IJRRECS/October 2013/Volume-1/Issue-6/1601-1610 **ISSN 2321-5461**

INTERNATIONAL JOURNAL OF REVIEWS ON RECENT ELECTRONICS AND COMPUTER SCIENCE

DETECTION OF HUMAN MOTION BASED ON BACKGROUND SUBTRACTION METHOD USING MORPHOLOGICAL FILTERING ON FPGA

Ch.Venkateswarlu[1], G.Lokesh Kumar[2]

[1]Asst.Prof, Dept of ECE, Pragati Engineering College, Surampalem, A.P, India
Email: bk.venkateswarlu@gmail.com
[2]M.Tech, Dept of ECE, Pragati Engineering College, Surampalem, A.P, India
Email: lokesh442.g@gmail.com

ABSTRACT:

The human motion detection is the most important part of the human body motion analysis, the purpose is to detect the moving human from the background image in video sequences, and for the follow-up treatment such as the target classification, the human motion tracking and behavior understanding, its effective detection plays a very important role. This paper presents a new technique morphological filtering is used to eliminate the noise and solve the background disturbance problem. This system works on a period pipelined flow on the Micro Blaze architecture of Spartan3 EDK. Field Programmable Gate Array (FPGA) has become a new device for implementation of algorithms of video image process applications. This paper explains the Background subtraction method using morphological filtering by using Micro blaze Processor presents in a FPGA using System C coding.

***Keywords**: Background Subtraction, FPGA, Morphological filtering, moving object detection.*

1. INTRODUCTION:

Field Programmable Gate Arrays (FPGA) is majorly used as a reconfigurable device, which can be used in the field of Image Processing. FPGA generally consists of large no of digital components such as look up tables, logic gates, flip-flops and many more, and its consists of memory ,and all there are interconnected through many inter connecting wires. All of the logic in an FPGA can be rewired, or reconfigured, with many different designs and according to the own needs. Normally Image Processing application can be implemented by using MATLAB software but in this paper our Background subtraction algorithm was implemented by using Spartan 3 FPGA Which consists Micro blaze processor which increases the speed of operation and it consists of high no of MAC units compare to the DSP processors so that we can achieve the speed of operation in the FPGA. The main process was because the software results are not accurate than the Hardware results to implement a hardware to existing Image Processing applications we are coming for FPGA implementation. In this project a high configurable Micro blaze processor was used our algorithm was written in the system C coding an Desynthesized using the XILLINX Platform Studio 10.1 and our output are seen through VB application which reads the pixels values of the image that comes from the FPGA to computer through UART communication. Any motion detection system based on background subtraction needs to handle a number of critical situations such as:

1. Noise image, due to a poor quality image source;

2. Gradual variations of the lighting conditions in the scene;

3. Small movements of non-static objects such as tree branches and bushes blowing in the wind;

4. Undeviating variations of the objects in the scene, such as cars that park (or depart after a long period);

5. Sudden changes in the light conditions, (e.g. sudden raining), or the presence of a light switch (the change from daylight to non-natural lights in the evening);

6. Movements of objects in the background that leave parts of it different from the background model;

7. Shadow regions that are projected by foreground objects and are detected as moving objects.

8. Multiple objects moving in the scene both for long and short periods;

The main objective of this paper is to develop an algorithm that can detect human motion at certain distance for object tracking applications. The approaches reviewed in the paper are

- B ackground subtraction
- A ppling morphological filter to the above step to remove noise
- A gain Appling a linear filer technique

II BACKGROUND SUBTRACTION METHOD

Detection of moving human in videos from static camera is widely performed by background subtraction method. The origin in the approach is that of detecting the moving objects from the difference between the existing frame and a reference frame, frequently called the “background copy”, or “background replica”[1]. **As** a baric, the background image must be a representation of the scene with no moving objects and must be kept regularly updated so as to adapt to the varying luminance conditions and geometry settings. More difficult models have extended the concept of “background subtraction” beyond its literal meaning. The background subtraction method is the common method of motion detection. It is a technology that uses the difference of the current image and the background image to detect the motion region, and it is generally able to provide data included object information. The key of this method lies in the initialization and update of the background image[2]. The effectiveness of both will affect the accuracy of test results. Therefore, this paper uses an effective method to initialize the background, and update the background in real time.

The process algorithm is described as follow:

1. Sequences of Video Frames
2. Frame Separation
3. Image Sequence
4. Separation of Image Sequence in Current Frame Image and Background Frame Image
5. Perform Background subtraction
6. Detection Of Moving Object
7. Perform Background updating
8. Noise Removal
9. Shape Analysis

Most of background subtraction algorithms follow a simple flow diagram as that shown in below figure

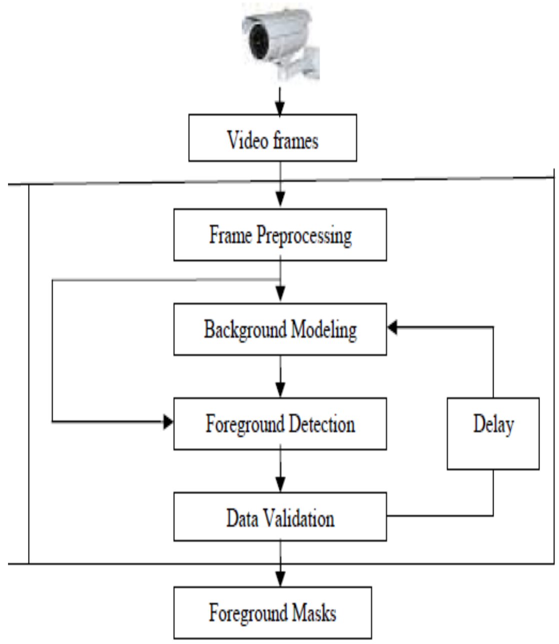

FIG: Flow Diagram Of a Generic Background Subtraction Algorithm

III MORPHOLOGY FILTER

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, you can construct a morphological operation that is sensitive to specific shapes in the input image.

Fundamental operations

The basic operations of mathematical morphology are the

1. Dilation
2. erosion

in these names capture the essence of operations; dilation increases the image, erosion makes it less

1. DILATION

Structural element S is applied to all pixels of binary image. Every time the origin of the structural element is combined with a single binary pixel, the entire structural element is wrapped and subsequent alteration of the corresponding pixels of binary image. The results of logical addition are written into the output binary image, which was originally initialized to zero.

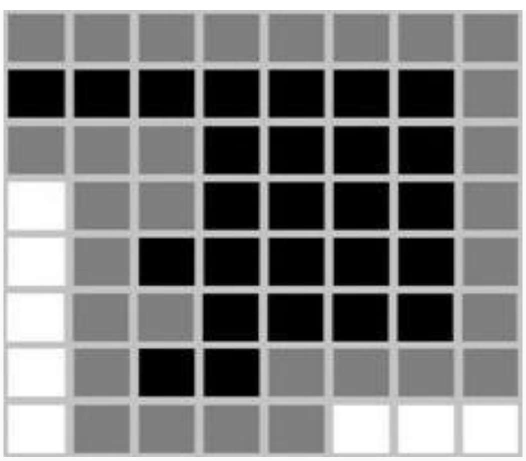

Image (B) dilation using structure element S

2. EROSION

When using erosion structural element also passes through all pixels of the image. If at a certain position every single pixel structuring element coincides with a single pixel binary image, then the logical disjunction of the central pixel structuring element with the corresponding pixel in the output image

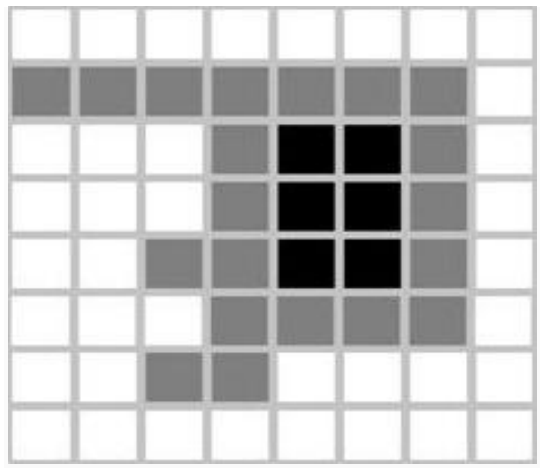

Image (B) erosion using structure element S

As a result of applying the erosion of all the objects smaller than the structural element will be erased, objects, connected by thin lines are disconnected and the sizes of all the objects are reduced.

How It Works

Useful background to the present description is given within the mathematical morphology section of the wordbook. The erosion operator takes 2itemsof knowledge as inputs. The primary is that the

image that is to be worn. The second may be a (usually small) set of coordinate points called a structuring component (also called a kernel). it's this structuring component that determines the precise result of the erosion on the input image.
The mathematical definition of abrasion for binary pictures is as follows:
Suppose that X is that the set of geometrician coordinates appreciate the input binary image, which K is that the set of coordinates for the structuring component.
Let Kx denote the interpretation of K so its origin is at x.
Then the erosion of X by K is just the set of all points x specified Kx may be a set of X.
The mathematical definition for grayscale erosion is identical except within the method during which the set of coordinates related to the input image comes. additionally, these coordinates area unit three-D instead of 2-D.
As associate degree example of binary erosion, suppose that the structuring component may be a 3×3 sq., with the origin at its center as shown in Figure one. Note that during this and a future diagram, foreground pixels area unitportrayed by 1's and background pixels by 0's.

1	1	1
1	1	1
1	1	1

Set of coordinate points =

{ (-1, -1), (0, -1), (1, -1),
(-1, 0), (0, 0), (1, 0),
(-1, 1), (0, 1), (1, 1) }

Figure 2.1 A 3×3 square structuring element

To work out the erosion of a binary input image by this structuring part, we have a tendency tocontemplateevery of the foreground pixels within the input image successively. for every foreground {pixel|pel|picturepart|component|constituent|element} (which we are going todecision the input component) we have a tendency toposition the structuring part on prime of the input image so the origin of the structuring element coincides with the input pixel coordinates. If for each {pixel|pel|picturepart|component|constituent|element} within the structuring element, the corresponding componentwithin the image beneathcould be a foreground component, then the input component is left because it is. If any of the corresponding pixels within the image ar background, however, the input componentis additionally set to background price.
For our example 3×3 structuring part, the result of this operation is to get rid of any foreground componentthat's not utterlyenclosed by alternative white pixels (assuming 8-connectedness). Such pixels should lie at the sides of white regions, sothe sensibleeffect is that foreground regions shrink (and holes withina part grow). Erosion is that thetwin of dilation, i.e. wearing foreground pixels is adore dilating the background pixels.
The background image initialization expression is as follows:

$$B_{\text{init}}(x,y) = \underset{k}{median}\, f_k(x,y) \qquad k = 1,2,\cdots n \qquad (1)$$

Where Binit is the intial background, n is the total number of frames selected.
For the background model can better adapt to light changes, the background needs to be updated in real time, so as to accurately

extract the moving object. In this paper, the update algorithm is as follows:

In detection of the moving object, the pixels judged as belonging to the moving object maintain the original background gray values, not be updated. For the pixels which are judged to be the background, we update the background model according to following rules:

$$B_{k+1}(x,y) = \beta B_k(x,y) + (1-\beta)F_k(x,y) \qquad (2)$$

Where *f3* E **(0,1)** is update coefficient, m this paper *f3* = 0.004. *Fk(x,y)* is the pixel gray value in the current frame. *Bk(x,y)* and *Bk+1(x,y)* are respectively the background value of the current frame and the next frame.

IV MOVING OBJECT DETECTION

After the background image *B(x, y)* is obtained, subtract the background image *B(x,y)* from the current frame *Fk (x, y).* If the pixel difference is greater than the set threshold *T* , then determines that the pixels appear in the moving object, otherwise, as the background pixels. The moving object can be detected after threshold operation. Its expression is as follows:

$$D_k(x,y) = \begin{cases} 1 & |F_k(x,y) - B_{k-1}(x,y)| > T \\ 0 & others \end{cases}$$

Where *Dk(x, y)* is the binary image of differential results. *T* is gray-scale threshold; its size determines the accuracy of object identification.

As in the algorithm *T* is a fixed value, only for an ideal situation, is not suitable for complex environment with lighting changes. Therefore, this paper proposes the dynamic threshold method, we dynamically changes the threshold value according to the lighting changes of the two images obtained. On this basis, add a dynamic threshold *!'1T* to the above algorithm. Its mathematical expression is as follows:

$$\Delta T = \lambda \bullet \frac{1}{M \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} |F(i,j) - B(i,j)| \qquad (4)$$

Then

$$D_k(x,y) = \begin{cases} 1 & |F_k(x,y) - B_{k-1}(x,y)| > T + \Delta T \\ 0 & others \end{cases} \qquad (5)$$

Where A is the inhibitory coefficient, set it to a value according to the requirements of practical applications, and the reference values is 2. *M* x *N* is the size of each image to deal with. *M* x *N* numerical results indicate the number of pixels in detection region. *!'1T* reflects the overall changes in the environment. If small changes in image illumination, dynamic threshold *!'1T* takes a very small value. Under the premise of enough pixels in the detection region, *!'1T* will tend to O. If the image illumination changes significantly, then the dynamic threshold *!'1T* will increase significantly. This method can effectively suppress the impact of light changes.

The flow chart of moving human body extraction is shown inFig.l:

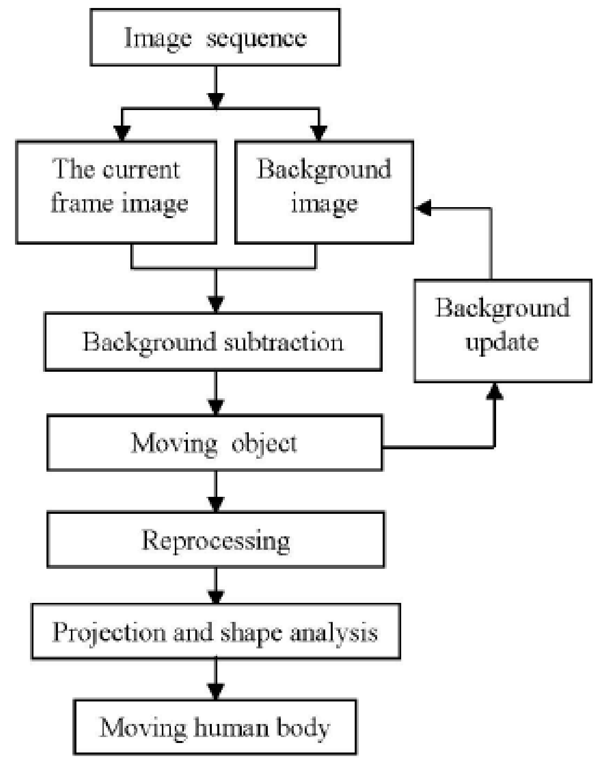

Figure 1. The flow chart of moving human body extraction

IV EXPERIMENTAL SETUP

Xilinx Platform Studio

The Xilinx Platform Studio (XPS) is the development environment or GUI used for designing the hardware portion of your embedded processor system. Xilinx Embedded Development Kit (EDK) is an integrated software tool suite for developing embedded systems with Xilinx Micro Blaze and PowerPC CPUs. EDK includes a variety of tools and applications to assist the designer to develop an embedded system right from the hardware creation to final implementation of the system on an FPGA. System design consists of the creation of the hardware and software components of the embedded processor system and the creation of a verification component is optional. A typical embedded system design project involves: hardware platform creation, hardware platform verification (simulation), software platform creation, software application creation, and software verification. Base System Builder is the wizard that is used to automatically generate a hardware platform according to the user specifications that is defined by the MHS (Microprocessor Hardware Specification) file. The MHS file defines the system architecture, peripherals and embedded processors]. The Platform Generation tool creates the hardware platform using the MHS file as input. The software plat defined by MSS (Microprocessor Software Specification) file which defines driver and library customization parameters for peripherals, processor customization parameters, standard 110 devices, interrupt handler routines, and other software related routines. The MSS file is an input to the Library Generator tool for customization of drivers, libraries and interrupts handlers.

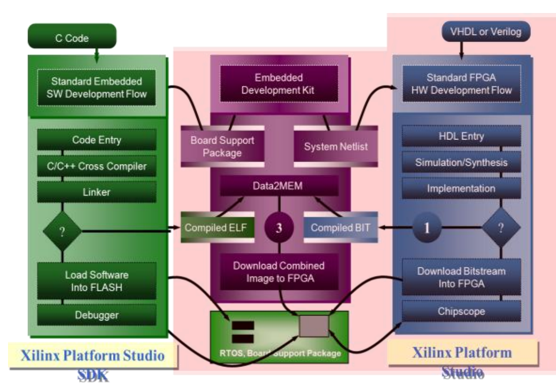

FIG: Embedded Development Kit Design Flow

The creation of the verification platform is optional and is based on the hard platform. The MHS file is taken as an input by the Simgen tool to create simulation files for a specific simulator. Three types of simulation models can be generated by the Simgen tool: behavioral, structural and timing models. Some other useful tools available in EDK are Platform Studio which provides the GUI for creating the MHS and MSS files. Create / Import IP Wizard which allows the creation of the designer's own peripheral and import them into EDK projects. Platform Generator customizes and generates the processor system in the form of hardware net lists. Library Generator tool configures libraries, device drivers, file systems and interrupt handlers for embedded processor system. Bit stream initializes the instruction memory of processors on the FPGA shown in figure2. GNU Compiler tools are used for compiling and linking application executables for each processor in the system [6]. There are two options available for debugging the application created using EDK namely: Xilinx Microprocessor Debug (XMD) for debugging the application software using a Microprocessor Debug Module (MDM) in the embedded processor system, and Software Debugger that invokes the software debugger corresponding to the compiler being used for the processor. C. Software Development Kit Xilinx Platform Studio Software Development Kit (SDK) is an integrated development environment, complimentary to XPS, that is used for C/C++ embedded software application creation and verification. SDK is built on the Eclipse open source framework. Soft Development Kit (SDK) is a suite of tools that enables you to design a software application for selected Soft IP Cores in the Xilinx Embedded Development Kit (EDK).The software application can be written in a "C or C++" then the complete embedded processor system for user application will be completed, else debug & download the bit file into FPGA. Then FPGA behaves like processor implemented on Xilinx Field Programmable Gate Array (FPGA) device.

V RESULTS

The Algorithm is implemented in Micro blaze Processor and the results are furnished in the tabulation below

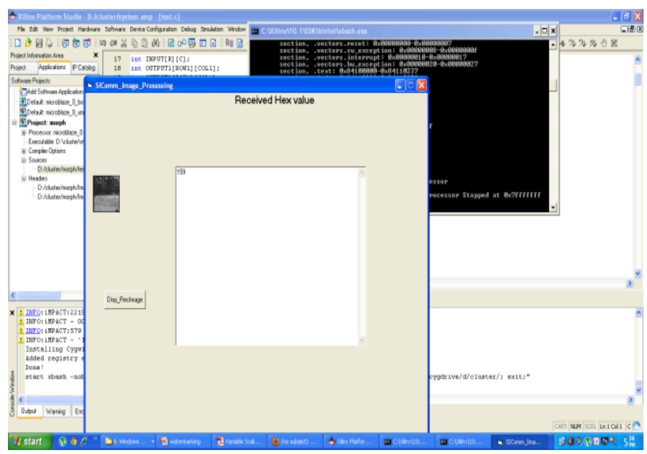

Fig5.1: Background Image reading in VB window

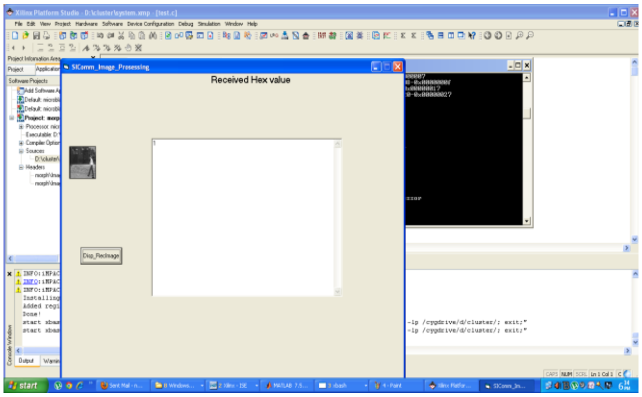

Fig5.2: Foreground Image

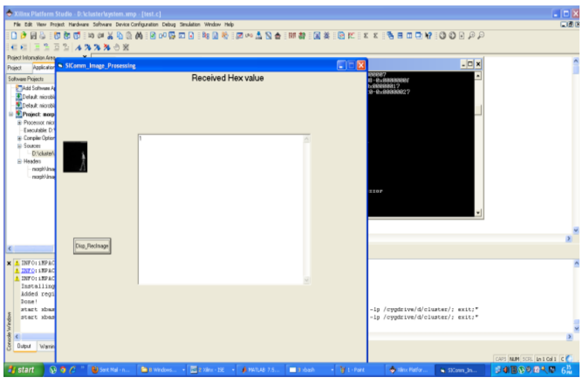

Fig5.3: Background subtracted image

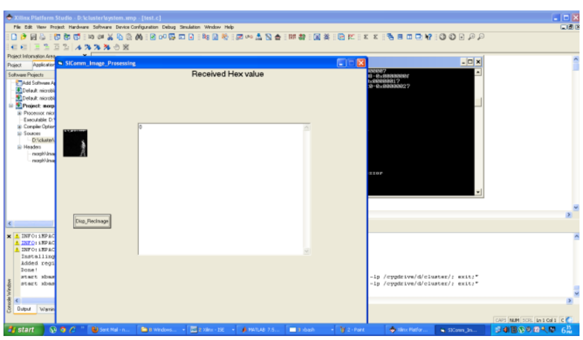

Fig5.4: Morphological Output

```
Device utilization summary:
---------------------------

Selected Device : 3s200tq144-4

 Number of Slices:                    1880  out of   1920    97%
 Number of Slice Flip Flops:          2118  out of   3840    55%
 Number of 4 input LUTs:              2971  out of   3840    77%
    Number used as logic:             2418
    Number used as Shift registers:    297
    Number used as RAMs:               256
 Number of IOs:                         62
 Number of bonded IOBs:                 62  out of     97    63%
    IOB Flip Flops:                     64
 Number of BRAMs:                        4  out of     12    33%
 Number of MULT18X18s:                   3  out of     12    25%
 Number of GCLKs:                        4  out of      8    50%
 Number of DCMs:                         1  out of      4    25%
```

Fig5.5: Synthesis report

Conclusion:

In this work moving object motion detection on background subtraction algorithmic rule was developed. In this work a morphological filtering is used to eliminate the noise and background disturbance problem.

This system works on a period pipelined flow on the Micro Blaze architecture of Spartan3 EDK. On the opposite hand, synthesis results show that space consumption is low, using simply 100 percent of logic components of FPGA for moving object detection system, permitting the implementation of this method over inexpensive FPGAs.

REFERENCES:

[1] T. Wigand, G. J. Sullivan, G. Bjontegaad, and A. Luthra, “Overview of the H.264/AVC video coding standard,”

IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560–575, Jul. 2003.

[2] Z. He, Y. Liang, L. Chen, I. Ahmad, and D.Wu, "Power-rate-distortion Analysis for wireless video communication under energy constraints," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 5, pp. 645–658, May 2005.

[3] H. F. Ates and Y. Altunbasak, "Rate-distortion and complexity optimized Motion estimation for H.264 video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 2, pp. 159–171, Feb. 2008.

[4] A. Bahari, T. Arslan, and A. T. Erdogan, "Low-power H.264 video compression architectures for mobile communication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 9, pp. 1251–1261, Sep. 2009

[5] C.T. Johnston, K.T.Gribbon, D.G.Bailey, "Implementing Image Processing Algorithms on FPGAs", Eleventh Electronics New Zealand Conference, Palmerstone North, New Zealand, 2004.

[6] Bruce A. Draper, J. Ross Beveridge, A.P. Willem Böhm, Charles Ross, Monica Chawathe, "Accelerated Image Processing on FPGAs", IEEE Transactions on Image Processing, Vol. 12, No. 12. Pp. 1543-1551, 2003.

[7] A. Castillo, J. Vázquez, J. Ortegón y C. Rodríguez, "Prácticas de laboratorio para estudiantes de ingeniería con FPGA", IEEE Latin America Transactions, Vol. 6, No.2, pp. 130-136, 2008.

[8] K. T. Gribbon, D. G. Bailey and C. T. Johnston, "Design Patterns for Image Processing Algorithm Development on FPGAs", TENCON 2005, pp. 1-6, November 21-24, 2005.

[9]"Digital Image Processing"3rd Edition by Rafael C. Gonzalez and Richard E. Woods, Addison Wesley, 2009

[10] P.Maragos "Morphological filtering for image enhancement and feature detection " in A.C. Bavik(eds.). The image & video processing handbook. Elsevier Academic Press 2005

[11] Jie Yang, Ran Yang," A Novel Edge detection based segmentation algorithm for polar metric SAR images", The international Archives of photogrammetric, Remote Sensing & Spatial information Sciences Vol.XXXVII .Part B7 Beijing 2008

[12] Shankar Krishna murthy", Histogram based Morphological Edge Detector", IEEE Transaction on Geosciences and Remote Sensing Vol.32 no.4, July 1994.