International Journal of Robotics and Automation (IJRA)
Vol. 2, No. 1, March 2013, pp. 17~25
ISSN: 2089-4856

Design and Implementation of Video Shot Detection on Field Programmable Gate Arrays

Jharna Majumdar[1], Darshan K M[2], Abhijith Vijayendra[2]
[1]Dept of CSE (PG), Nitte Meenakshi Institute of Technology, Yelahanka, Bangalore, Karnataka, India
[2]Image and Video Processing Lab, R&D NMIT, Nitte Meenakshi Institute of Technology, Yelahanka, Bangalore, Karnataka, India

Article Info

Article history:

Received Aug 12, 2012
Revised Nov 21, 2012
Accepted Dec 16, 2012

Keyword:

Display Controller
FPGA
Micro Blaze
MPMC
Processor Local Bus (PLB)
Video Shot Detection
Video to VFBC

ABSTRACT

Video has become an interactive medium of communication in everyday life. The sheer volume of video makes it extremely difficult to browse through and find the required data. Hence extraction of key frames from the video which represents the abstract of the entire video becomes necessary. The aim of the video shot detection is to find the position of the shot boundaries, so that key frames can be selected from each shot for subsequent processing such as video summarization, indexing etc. For most of the surveillance applications like video summery, face recognition etc., the hardware (real time) implementation of these algorithms becomes necessary. Here in this paper we present the architecture for simultaneous accessing of consecutive frames, which are then used for the implementation of various Video Shot Detection algorithms. We also present the real time implementation of three video shot detection algorithms using the above mentioned architecture on FPGA (Field Programmable Gate Arrays).

Corresponding Author:

Dr Jharna Majumdar,
Departement of Computer Science and Engineering,
Nitte Meenakshi Institute of Technology,
Govindapura, Gollahalli, Yelahanka, Bangalore – 560 064
Email: jharna.majumdar@gmail.com

1. INTRODUCTION

With the advancement of video technology, multimedia techniques have found applications in nearly every aspect of our life, such as national security, broadcasting, communication, entertainment, library science etc [2]. However, the increasing availability of digital data has not been accompanied by an increase in its accessibility hence an efficient methods for video summarization, indexing and retrieval are required to confront this situation [3]. Video summary is a process of presenting an abstract of entire video within a shorter period of time. It aims to provide a condensed video representation, which enable the user to quickly figure out general contents of a video. It also plays prime role where the resources such as storage, communication bandwidth and power are limited [8].

Video shot boundary detection is a fundamental step in video indexing and summarization. A shot designates a contiguous sequence of video frames recorded by an uninterrupted camera operation. A scene is a collection of shots which present different views of the same event and contain the same object of interest. The shot boundaries may be abrupt or gradual. Abrupt transitions (Hard cut) indicates the shot change that occurs between two consecutive frames whereas, gradual transitions such as Fade, Wipe, Dissolve are spread over several frames. It will be harder to detect gradual transitions, as the difference between consecutive frames is smaller.

Several shot change detection approaches have been proposed in past and most of them are based on comparison between consecutive frames. Some of the key features considered for the purpose of comparison include luminance, image edges, color ratio histograms, motion, transform coefficients etc. Zhang et al [2]

use a pixel based difference method and running histogram method to detect gradual as well as abrupt shot boundaries. Nagasaka and Tanaka et al [3] compared several simple methods based grey level and color histograms. R. Lienart et al [4] has reviewed several methods statistics based on color histograms and change in edges of frames. Boreczky and Rowe [5] has given comparison and classification of various shot boundaries techniques and their variations including histograms, discrete cosine transform, motion vector and block matching techniques. Lotfi Boussaid et al [6] discussed the real time implementation of shot detection algorithms for low resolution videos. The paper also discusses about the ease of development and testing of the hardware architecture in FPGA over ASIC.

The remainder of paper is organized as follows. Section 2 presents the objective of the work. Section 3 discusses the platform used and implementation setup. Section 4 describes the design and implementation four different methods for shot detection. In Section 5 performance of these methods is evaluated using recall and precision for various video segments and experimental results are presented in Section 6.

2. OBJECTIVE OF THE WORK

For most of the surveillance and border security applications real time implementation of shot detection algorithms is a key criteria. To operate in the real time condition, computational time must not exceed the blanking time between the consecutive frames. Since PC is not suitable for computationally intensive algorithms due to the serialized execution, it demands for the hardware solution where the process of parallelism and pipelined architecture accelerates the computation speed. The main idea underlying the shot detection is that, images in the vicinity of a transition are highly dissimilar. Hence, general approach to do this is to compare the adjacent frames in the video to extract the similarity measure between the observations. Since design development platform being used for this implementation lacks on chip (block ram) memory for storing the frames it is necessary to store them into the external memory (DDR2 RAM) and access consecutive frames at the same time, without compromising with the speed of access. In this paper we present the architecture for storing the frames into the external memory and accessing the consecutive frames simultaneously and also present the implementation of three video shot detection algorithms, namely, Pixel Difference, Histogram Difference, and Edge Change Ratio. Both the designs has been implemented and tested on Xilinx's Spartan3A DSP Video Starter Kit (VSK) for an incoming video of 800 x 600 resolution at 60fps. The paper also discusses the calculation of threshold for shot detection and performance evaluation of the above implemented algorithms.

3. IMPLEMENTATION SETUP

3.1 General Architecture

The setup for implementation consists of Video Starter Kit (VSK) housing a Spartan 3A DSP XCSD3400A FPGA connected to a Workstation acting as a DVI source of resolution 800 x 600 pixels delivering frames at 60 fps through a FPGA Mezzanine Card (FMC) Daughter card used for decoding the data arriving through the DVI interface [13][14][15][16]. The de-serialized input consists of V-Sync, H-Sync, Data Enable (DE) and 8 bit color red, green and blue data bus which serves as the input for the subsequent blocks.

The block diagram in the Figure 1 shows the 'Video Frame Buffer' setup of VSK.

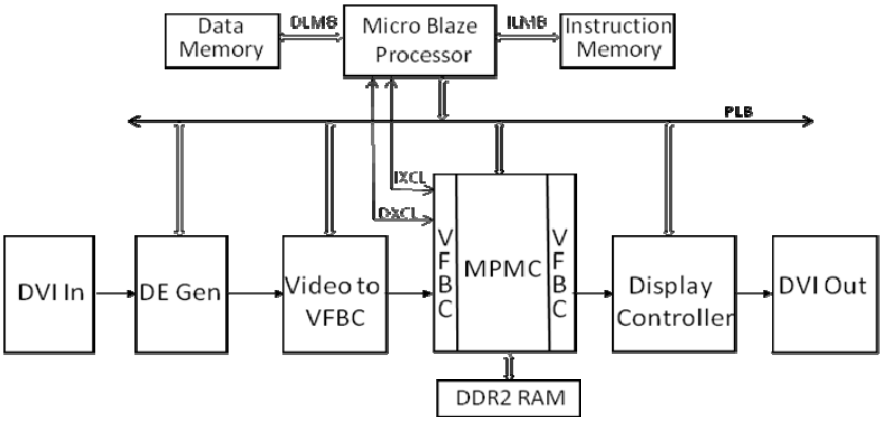


Figure 1. Video Frame Buffer Setup of VSK

The serialized data provided by the source is written into external memory (DDR2 RAM) by 'Video to VFBC' core of VSK through MPMC interface and again retrieved using 'Display Controller' core. The architecture for accessing two frames simultaneously from external memory is discussed in Section 3. The status reporting and controlling of different cores are carried out by Micro Blaze processor through Processor Local Bus [18].

3.2 Dual Display Controller Logic

The main idea underlying the shot detection is to compare the adjacent frames in the video. As there is a lack of on chip (block ram) memory for storing of the frames it is necessary to store them into the external memory (DDR2 RAM) and access consecutive frames at a time. In this section describes the general architecture of storing the frames into the external memory and accessing the consecutive frames.

As stated above 'Video to VFBC' core [13] manages the storing of video into frame buffers in external memory through VFBC interface on the MPMC [17]. There will be three frame buffers in the external memory, each of which stores one frame and frames are stored sequentially by 'Video to VFBC' core. The arrival of new frame makes the base address of 'Video to VFBC' to increment by a value equal to frame offset, thereby making the VFBC to start writing to the next buffer. The 'Display Controller' core reads video frames out of memory through VFBC interface on MPMC. The consecutive frames are accessed using the dual display controller logic as shown in Figure 2. By the time 'Video to VFBC' core is writing data to third buffer, two 'Display Controllers' are made to read the data from first and second buffers by assigning them the respective base addresses. Arrival of the next frame makes 'Video to VFBC' to point to the first buffer and 'Display Controllers' to point to the next two buffers and the cycle repeats. Here care should be taken to see that there will be no address collisions between the different cores accessing the external memory and both the 'display controllers' are switching on at the same time. In order for the proper synchronization among the two 'display controllers' chip enable is provided from the custom core that is used to enable both the display controllers simultaneously.

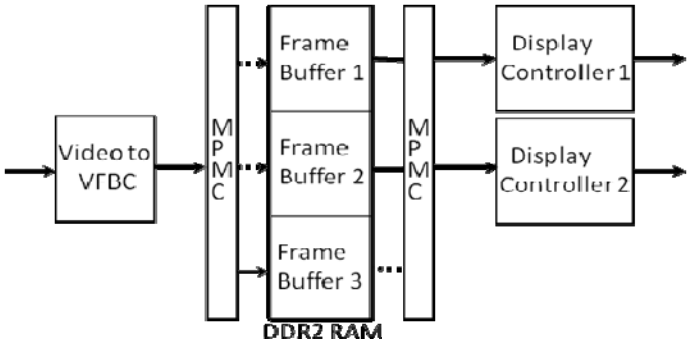


Figure 2. Dual Display Controller logic

The output of two display controllers, each of which containing RGB data along with Sync signals are connected to custom core where actual Video Shot Detection algorithm is implemented. Since both the display controllers starts simultaneously both the Sync signals will be same and either of them can be used. The flow diagram of modified 'Video Frame Buffer' setup of VSK housing dual display controller logic is shown in the Figure 3.

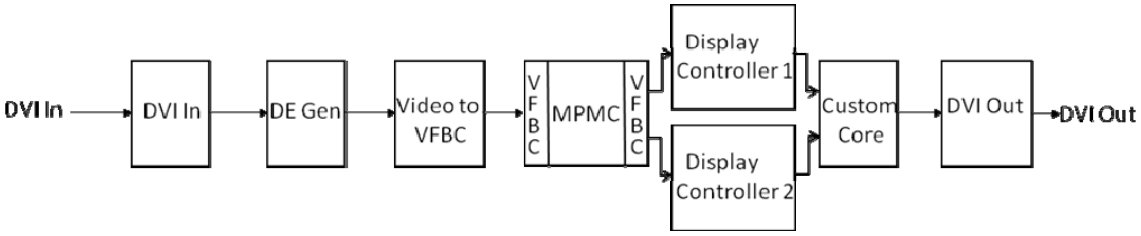


Figure 3. Flow diagram of Dual Display Controller logic

4. METHEDOLOGY

4.1 Pixel Difference

The idea behind the algorithm is that the intensities of the pixels changes rapidly at the shots. The pixel subtraction operator takes two consecutive images as input and produces the value which is the sum of the difference between corresponding pixel values of first image with that of the second image [4.5, 10]. As we are interested only in the change in intensity value from one frame to another the difference is taken to be an absolute value. Mathematically it is represented as

$$D\,(i) = \sum |f_i - f_{i+1}|$$

The top module of Pixel Difference method is shown in Figure 4. The incoming frames are converted into gray scale in Color Space Conversion (CSC) subsystem. Their sum of the difference is calculated and its absolute value D (i) is stored. Since the streaming data is directly subtracted without storing anywhere there is no data loss due to computational delay. If the calculated D (i) >T, where T is some threshold then shot change is assumed. Thus hard cuts and shot transitions can be detected as single peaks in the time series of the sum of pixel differences of contiguous frames from a given source.

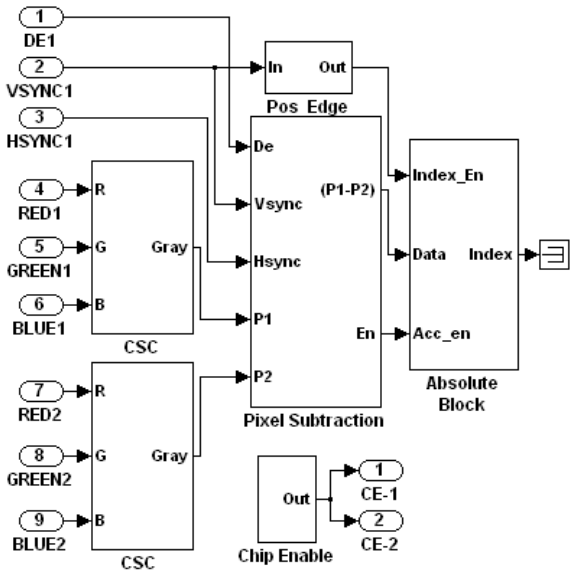


Figure 4. Top Level Module of Pixel Difference Method

4.2 Histogram Difference

Histogram subtraction is another variant of reliable histogram based detection algorithm [6, 9]. The histogram subtraction operator takes two images as input and produces as output value equal to the difference of the histogram of first image with that of the second image. The histogram subtraction is performed by the determination of the histogram of the images followed by their subtraction

$$\mathrm{D}(i) = \sum |H_i - H_{i+1}|$$

The top module of Histogram Difference method is shown in Fig 5. The histogram of the incoming consecutive frames is calculated separately and is stored into the Block Ram. At the *posedge* of *Vsync* (end of the frame) the histograms of two frames are accessed simultaneously and their sum of the difference is calculated and its absolute value *D (i)* is stored. Since accessing of histogram hardly takes 256 cycles *D (i)* can be calculated much before the arrival of the next frame hence no data will be lost. Further the shot is detected by the *D (i)* value exceeding certain threshold.

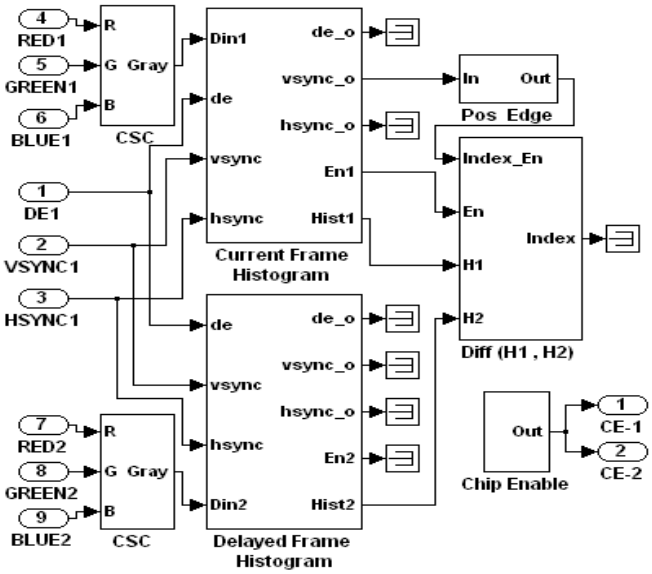


Figure 5.Top Level Module of Histogram Difference Method

4.3 Edge Change Ratio

This approach looks not at the color differences, but at the differences between the edges detected in adjacent frames [11]. Each frame in the digital video is turned into a grey-scale image and Sobel filtering applied to detect edges. The method looks for similar edges in adjacent frames to detect a shot boundary. In this approach, an entering and exiting edge of a frame is considered using Sobel edge detection method. The ratio of edge change between frame *n* and frame *n-1* is calculated using,

$$ECR_n = max\ (\frac{I_n^{in}}{S_n}, \frac{I_{n-1}^{out}}{S_{n-1}})$$

where, I_n^{in} represents number of entering edges I_{n-1}^{out} represents number of exiting edges, S_n and S_{n-1} represents number of edge pixels in *frame n* and *frame n-1* respectively.

The top module of Edge Change Ratio is shown in Figure 6.

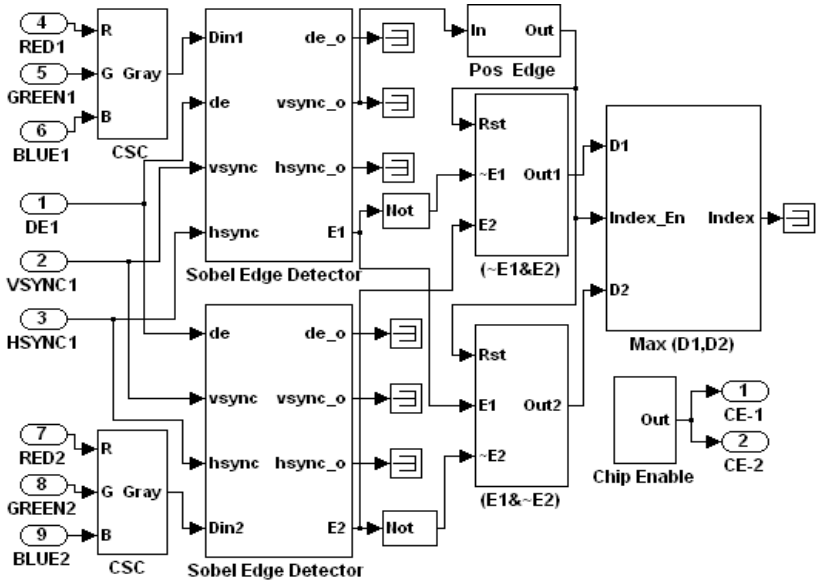


Figure 6.Top Level Module of Edge Change Ratio

Sobel Edge Detector [22] is applied to the incoming frames to implement the above mentioned formula. Obtained ECR value varies between 0 and 1. Maximum ECR value above some threshold θ represents shot change between the frames.

The principle behind the edge detection approach is that it can counter problems caused by fades and dissolves and other transitions which are invariant to gradual color changes. With edge detection, even when there are gradual transitions between shots, there should always be a pair of adjacent frames where an edge is detected in one, and not in the other and identifying an occurrence of this on a large scale locates a shot transition.

5. PERFORMANCE EVALUATION

The performance of shot detection methods in video sequence can be improved by the use of threshold that adapts itself to the sequence statistics. The mean μ and standard deviation σ which are used towards the computation of this threshold are as shown in the equations 1 and 2.

$$\mu = \frac{1}{N-1}\sum_{i=0}^{N-1} D(i,i+1) \quad -(1)$$

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=0}^{N-1}[D(i,i+1)-\mu]^2} \quad -(2)$$

where, N is the number of frames in the video sequence and $D(i,i+1)$ is the discontinuity value and the threshold T is obtained by $\mu + k\sigma$ where k is a pre specified constant.

Given the total number of shots and their locations the performance of the different algorithms are measured using number of shots correctly detected, number of false positive and number of missed shots. In [5, 10, 12], commonly used performance measures are recall and precision. Recall quantifies what proportions of the correct entities (shot change in our case) are detected, while precision quantifies what proportion of the detected entities are correct. It is given by the equations 3 and 4

$$Recall = \frac{correct}{correct + missed} \quad -(3)$$

$$Precision = \frac{correct}{correct + false\ positive} \quad -(4)$$

6. RESULTS AND DISCUSSION

The shot boundary detection algorithms were tested on three videos with diverse features. Table 1 below gives the actual shot details of different videos used. Figure 8 shows the graphs obtained from different methods, applied on the video sequence presented in Figure 7 (Park Avenue). First the algorithms were being tested in Visual Studio C++ and then designed and implemented on FPGA. The device on the board Spartan-3A DSP FPGA i.e., XC3SD3400A has 3400K gates, 23872 slices, 47744 Flip Flops, 47744 LUTs, 469 IOBs and 126 configurable internal Block RAM of 18 kb each. Over all Resource utilization (including VSK) for each method is as depicted in Table 2. Performance evaluation using recall and precision is presented in Table 3.

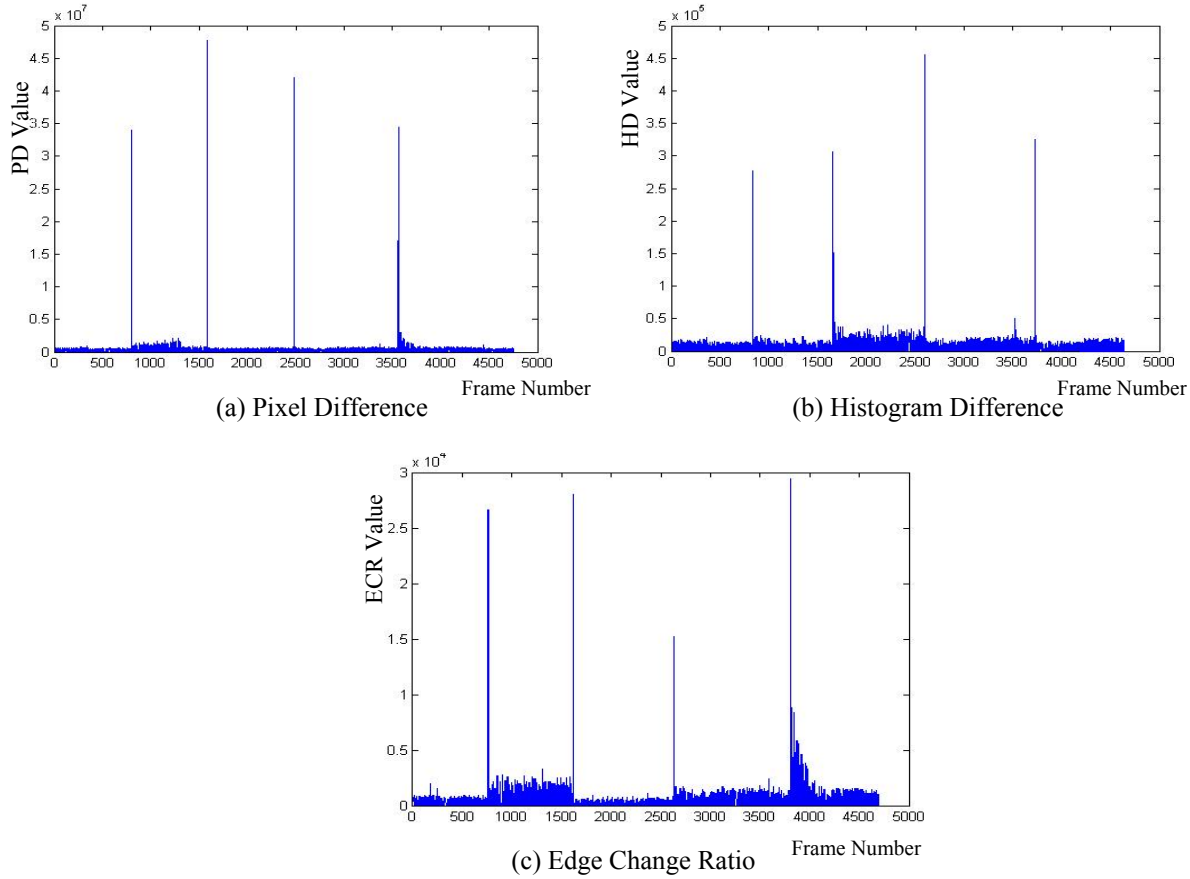
Table 1. Input video sequence details

Video	No. of actual shots in a video
Park Avenue	4
Wild Life	16
News	20

Table 2. Resource Estimation for different methods

Method	Slices %	Flip Flops %	LUTs %	BRAMs %	DSP 48 Core %
Pixel Difference	56	27	30	54	2
Histogram Difference	59	28	33	65	2
Edge Change Ratio	61	35	34	62	2

Figure 7. Input Video Sequence-1 (Park Avenue)



(a) Pixel Difference (b) Histogram Difference

(c) Edge Change Ratio

Figure 8. Graphical results of the shot boundaries for the input video shown in the Figure 7 using different methods

Table 3. Performance Evaluation Results for different video sequence

Video	Pixel Difference		Histogram Difference		Edge Change Ratio	
	Recall	Precision	Recall	Precision	Recall	Precision
Park Avenue	1	1	1	1	0.8	0.86
Wild Life	0.857	1	0.857	1	0.857	1
News	1	0.714	1	1	1	1

7. CONCLUSION

The general architecture proposed for simultaneous access of consecutive frames can also be used to implement various other shot detection algorithms that involve comparison of similarity or dissimilarity between the adjacent frames. The performance result shows that the proposed methods work in promising way for the abrupt cuts and it can be further improved to detect gradual transitions like fade, wipe and dissolve.

ACKNOWLEDGEMENT

The authors express their sincere gratitude to Dr. N.R. Shetty, Director and Dr. H.C. Nagaraj, Principal Nitte Meenakshi Institute of Technology for their constant support and encouragement during the course of the work.

REFERENCES

[1] Leonardo Chang, Ivis Rodes, Heydi Mendez, and Ernesto del Toro. "Best-Shot Selection for Video Face Recognition Using FPGA." *Springer-Verlag Berlin Heidelberg.* 2006.

[2] H.J.Zhang, A. Kankanhalli, and S.W.Smoliar. "Automatic Partitioning of Full Motion Video in Multimedia Systems." *Volume 1,pp 10-28.* 1993.

[3] A. Nagasaka and Y.Tanaka. "Automatic Video indexing and Full VideoSearch for Object Appearances." *pp1 13-117. Elsevier Science Publisher, 1992.*

[4] Rainer Lienhart. "Comparision of Automatic Shot Boundary Detection Algorithms." *CA 95052-8819. Microcomputer Research Labs, Intel Corporation, Santa Clara.*

[5] J.S. Boreczky and L.A. Rowe. "Comparison of Video Shot Boundary Detection Techniques." *Journal of Electronic Imaging (in Storage and Retrieval for Image and Video Databases IV, proc. SPIE vol. 2670,), 1996.*

[6] Lotfi Boussaid, Abdellatif Mtibaa, Mohamed Abid and Michel Paindavoine. "A real-time shot cut detector: Hardware implementation." *Elsevier*, *2007*.

[7] H.H.YU and W.Wolf. "A Hierarchical Multiresolution Video Shot Transition Detection Scheme." *Journal of Computer Vision and Image Understanding, vol.75, no. 1/2, pp.196-213*, 1999.

[8] C. Sujatha. "Video Summary With Shot Detection". Hubli: BVBCET, 2009.

[9] Jordi Mas, Gabriel Fernandez. "Video Shot Boundary Detection based on Color Histogram." *Notebook Papers TRECVID2003. Gaithersburg, Maryland, October 2003.*

[10] M.H. Kolekar, S. Sengupta. "Video Shot Boundary Detection: A Comparative Study of Three Popular Approaches." *National Conference on Communication (NCC). I.I.Sc,Bangalore, India: pp.419-423, Jan 2004.*

[11] Ramin Zabih, Justin Miller, Kevin Mai. "A Feature Based Algorithm for Detecting and Classifying Scene Breaks." *Internatoinal Multimedia Conference and Exhibition, Multimedia Systems. San Francisco , California, 1995. 189-200.*

[12] Ullas Gargi, Rangachar Kasturi, and Susan H. Strayer. "Performance Characterization of Video - Shot - Change Detection Methods." *IEEE Transactons On Circuits and Systems for Video Technology. February 2000.*

[13] "Spartan – 3A DSP FPGA Video Starter Kit, Software User Guide, ug 514."

[14] "Spartan – 3A DSP FPGA Family: Data Sheet ds610."

[15] "Spartan – 3A DSP FPGA Family: FMC – Video Daughter Card quick start guide ug456."

[16] "Spartan - 3A DSP FPGA Video Starter Kit, Getting Started Guide, ug455.

[17] Spartan – 3A DSP FPGA Family:Multi Port Memory Controller, Data Sheet ds 643."

[18] Xilinx Chipscope Pro 10.1 Software and Cores User Guide, UG029.

[19] Xilinx MicroBlaze Processor Reference Guide, EDK 10.1, UG081.

[20] Xilinx System Generator for DSP, Version 10.1.01, User's Guide.

[21] Xilinx Embedded Dvelopment Kit, Version 10.1, Reference Guide.

[22] Sudeep K C, Dr Jharna Majumdar, "A Novel Architecture for Real Time Implementation of Edge Detectors on FPGA", *International Journal of Computer Science (IJCSI), Vol. 8, Issue 1, January 2011*

BIOGRAPHIES OF AUTHORS

Dr. Jharna Majumdar is currently working as Dean R & D and Professor and Head of Computer Science and Engineering (PG) at Nitte Meenakshi Institute of Technology, Bangalore. Prior to this Dr. Majumdar served Aeronautical Development Establishment, Defence Research and Development Organization (DRDO), Ministry of Defence, Govt. of India from 1990 to 2007 as Research Scientist and Head of Aerial Image Exploitation Division, Bangalore. Dr. Majumdar has 37 years. of experience in R & D and Academics in the country and abroad. She has published large number of papers in National, International Conferences and Journals.Her research areas include Image and Video Processing for defense and non defense application, Robot Vision, Vision based autonomous guided systems, development of Computer Vision algorithms in FPGA etc.. Recently as the Project Coordinator, she has provided leadership to a team of undergraduate students from seven engineering for the development of STUDSAT, the smallest STUDent SATellite of PICO Category (Less than 1 Kg), first time developed in the country.

Darshan K.M graduated from Nitte Meenakshi Institute of Technology in 2011 from Dept. of Electronics and Communication. His research areas include implementation of image and video processing algorithms on reconfigurable hardware. He has been working as a Research Associate in R & D of Nitte Meenakshi Institute of Technology since his graduation. Persently, he will be persuing his masters degree in Microelectronics and Microsystems engineering from the University of Freiburg, Germany.

Abhijith Vijayendra graduated from Nitte Meenakshi Institute of Technology in 2011 from Dept. of Electronics and Communication. His research areas include the design and implementation of computationally intensive algorithms on reconfigurable hardware for the real time performance. He has been working as a Research Associate in R & D of Nitte Meenakshi Institute of Technology since his graduation. Persently, he will be persuing his masters degree in Digital VLSI and Computer Architecture from the University of Southern California, Los Angeles, USA.