ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

Video Shot Detection on Embedded System

Priyanka S[1], Dr.Jharna Majumdar[2], Santhosh Kumar K L[3]

M.Tech, CSE, Nitte Meenakshi Institute of Technology, Bangalore, India [1]

Dean R&D, HOD CSE(PG), Nitte Meenakshi Institute of Technology, Bangalore, India[2]

Assistant Professor, CSE, Nitte Meenakshi Institute of Technology, Bangalore, India [3]

Abstract*:* Currently digital videos are used in different fields of technology and in daily activities of human. The processing of image or video provides an understanding of the particular scene that it describe and of course it is very essential task in technologies like Robotics, Video surveillance and Multimedia. So it has become an area of research in the recent years. Video shot boundary detection is one of the research works in the field of video processing. The Shot boundary detection plays very important role for a browsing, retrieving, classifying and detecting any event. Video shot transitions provide simple jump points for browsing the videos. There are so many methods to detect the shot change, which have been proposed, many researches are trying hard to put forward various algorithms of this. In this paper, we have tried to analyse the performance of some of the video shot detection methods on embedded system such as on Beagle Board.

Keywords: Video Shot Detection (VSD), BeagleBoard, Edge Change Ratio (ECR).

I. Introduction

Video is the one of the effective media in today's world. Many applications like multimedia information system, distance learning advertising, electronic publishing, broadcasting, video-on-demand entertainment, etc. A shot is a continuous-unbroken sequence of inter-related images or frames which captured from single camera or from more than one camera. A scene is a set of one or more adjoining shots that focus on object or object of interest. "a single picture is equal to thousand words", as an old saying: If every video document contain a collection of still images and the number of set might contain hundreds even thousands of images or frames, it is burdensome to imagine how hard it can be to detect certain information in a collection of video documents. A video shot detection is also called shot transition detection and even called as video shot boundary detection is defined as a video 'v' consisting of 'n' shots, finding the start (beginning) and end of each shot. There are different methods in video shot detection such as pixel difference, histogram difference methods such as chi-square, bin-wise & intersection and edge change ratio, mutual information, weighted variance, scaled metrics and etc.

II. Embedded System

An embedded system is a computer system which is present in larger system and designed for specific control functions i.e. to perform few dedicated functions, with constraints like real-time computing. It is embedded as portion of a complete device, which includes hardware and mechanical parts. An embedded system contains processing cores like microcontrollers or digital signal processor (DSP). A processor is the heart of the embedded system because it is an important unit in the embedded system hardware. Nowadays it is very common that many devices are controlled by the embedded system. Beagle boards are small computers with most of all the capability of today's desktop machines, without the huge, expense or noise.

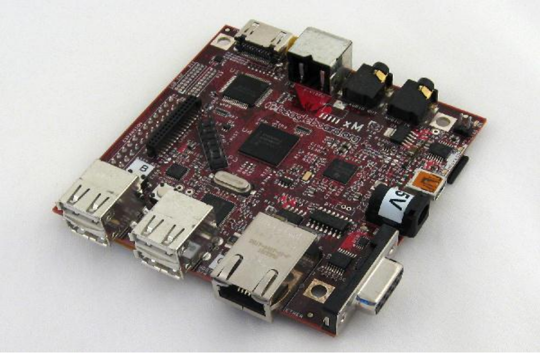


Figure.1: Beagle Board-xM

The Beagle board-xm is low power consumption, low cost, fan-less hardware single board computer, so called open source hardware which is easily available in market. Beagle board has all the functionality which is present in basic computer. The board utilize up to 2W of power because it consumes less power so additional freezing and heat sinks are not needed. The beagle board is designed in such a way that it addresses the open source community by removing on-board peripheral devices and providing important expansion buses like high-speed USB 2.0, Ethernet port, HDMI port etc.

Ubuntu comes under Linux operating system family and it is built on Debian's architecture and provide basic services to Linux server, desktop, phone, tablet and to operating system. Ubuntu is safest and fastest operating system. By default the programs of user, run with low fore deal and cannot corrupt the operating system or any other files of user's. So here Ubuntu is used with sdl and gstreamer library.

III.Proposed Methods

The input video is read frame by frame then every frame is converted into gray model. Then for that different video shot detections methods are applied.

DOI 10.17148/IJARCCE.2015.4811

Those methods are explained below.

A. Histogram Difference Methods

1) Histogram Difference using Chi-Square:

This approach is the best test for evaluation and effective way to detect the shot changes by comparing the histogram difference method. It is defined by following formula:

$$D_{chi} = \frac{1}{n^2} \sum_i^{256} \frac{(h_1[i]-h_2[i])^2}{\max(h_1[i],h_2[i])}, h_1[i] \neq 0, h_2[i] \neq 0$$

In the above formula as mentioned i is not equal to zero and it will keep on increment till the last frame of the input video. The h_1 and h_2 are the histogram of current frame and next frames. The difference between h_1 and h_2 is calculated, when there is maximum difference between h_1 and h_2 then it is compared with threshold value and if the h_1 and h_2 difference greater than threshold value shot will be assumed.

2) Histogram Difference using Intersection:

This approach is also one of the histogram based method which similar to the method which explained above i.e. Chi-square. The Histogram difference between two frames can be obtained using intersection formula. The Intersection formula is as follows:

$$\text{Intersection } (h_1, h_2) = \frac{\sum_i \min(h_1[i],h_2[i])}{N} \ (i \neq 0)$$

$$D_{INT} = 1\text{-Intersection } (h_1, h_2)$$

In the above formula as mentioned i is not equal to zero and it will keep on increment till the last frame of the input video. The h_1 and h_2 are the histogram of current frame and next frames; N is the total number of frames. The difference between h_1 and h_2 is calculated using the above intersection formula, when there is maximum difference between h_1 and h_2 then it is compared with threshold value and if the h_1 and h_2 difference greater than threshold value shot will be assumed.

B. Edge Change Ratio

The ECR is one of the methods to detect the shots in the video. In this approach instead of colour histogram difference the difference between edges are detected in the adjacent frames. Each frame in the input digital video is converted in to grey scale image then Sobel filtering applied to detect the edges. This method looks for similar edges in the corresponding frames to detect the shot boundary[3]. In this method, the entering and exiting edge of frame is calculated using Sobel edge detection method. The ECR value varies between 0 and 1, maximum ECR value which is greater than threshold represent the shot change.

The ratio of edge change between current and previous frame is calculated using following formula:

$$\text{ECR} = \max\left(\frac{EC_{n-1}^{in}}{S_{n-1}}, \frac{EC_n^{out}}{S_n}\right)$$

Where,

EC_{n-1}^{in} Represents number of entering edges

EC_n^{out} Represents number of exiting edges

S_{n-1} *and* S_n Represents number of edge pixels in *frame n* and *frame n-1* respectively.

C. Scaled Metrics

All the Video Shot Detection methods have advantages as well as disadvantages but one of the common problems in other Video Shot Detection is flashlights. The flashlights introduce false detection of shot change in the given video. So in order to overcome this Scaled Metrics method is used. In this approach frame difference is computed by using local histogram method for comparison which are more robust to object and camera motion and important information is obtained to produce more accurate results. Then dynamically scaling the frame difference by Log-formula to compress and enhance the frame difference.

The proposed Log-Formula in this scaled metrics is defined as follows:

$$d_{log} = c \times \log (1+ d^2)$$

$$c = \frac{\max(d_{log})}{\max(\log(1+d^2))}$$

Where,

d : is the frame difference extracted from using any histogram difference method.

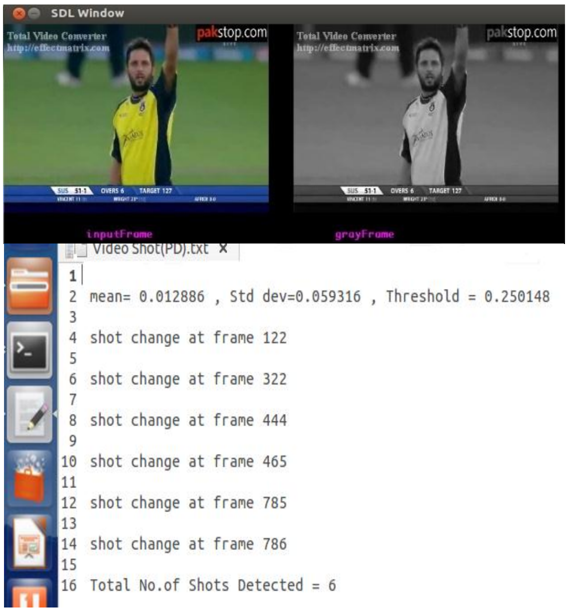
c : is the constant value calculated from d.

IV. SYSTEM SPECIFICATION

Software Requirements on CPU and on Beagle-Board are as follows: Operating System used Linux; Ubuntu 12.10 with specific libraries is Gstreamer and SDL.

CPU must contain RAM 1GB or higher and Processor must be 1.84 GHZ dual core or Higher. Beagle-Board must contain 1GB memory in it.

V. RESULTS AND ANALYSIS



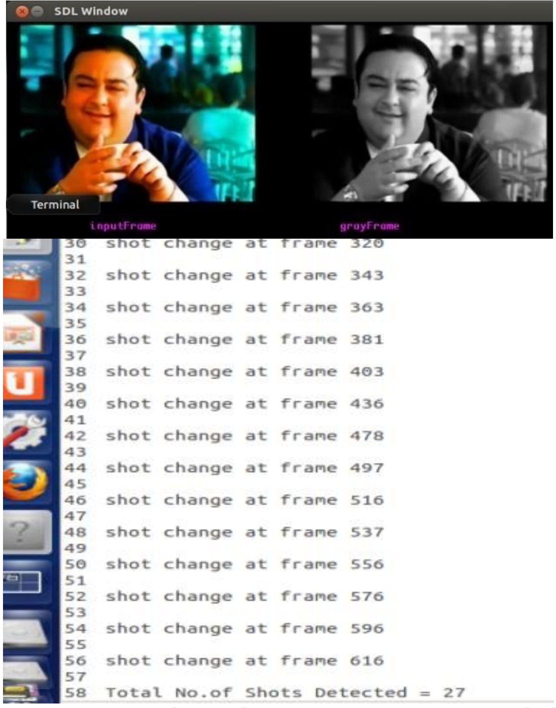
```

mean= 0.012886 , Std dev=0.059316 , Threshold = 0.250148

shot change at frame 122

shot change at frame 322

shot change at frame 444

shot change at frame 465

shot change at frame 785

shot change at frame 786

Total No.of Shots Detected = 6
```

Figure.2: Input video and output on SDL screen and result

The Fig.2 represents the input video given to the histogram-difference method using chi-square and also

DOI 10.17148/IJARCCE.2015.4811

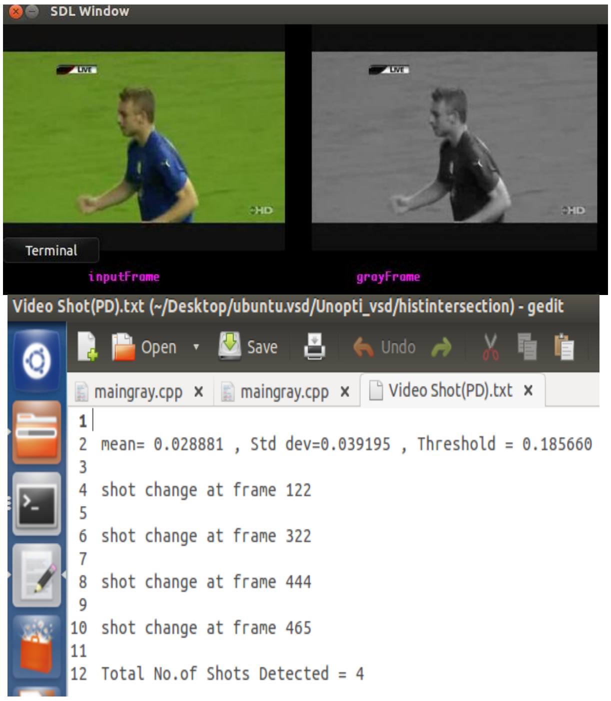
shows the output of the video on SDL screen with the result i.e. shots detected in the input video.



```
shot change at frame 320

shot change at frame 343

shot change at frame 363

shot change at frame 381

shot change at frame 403

shot change at frame 436

shot change at frame 478

shot change at frame 497

shot change at frame 516

shot change at frame 537

shot change at frame 556

shot change at frame 576

shot change at frame 596

shot change at frame 616

Total No.of Shots Detected = 27
```

Figure.3: Input video and output on SDL screen with the result

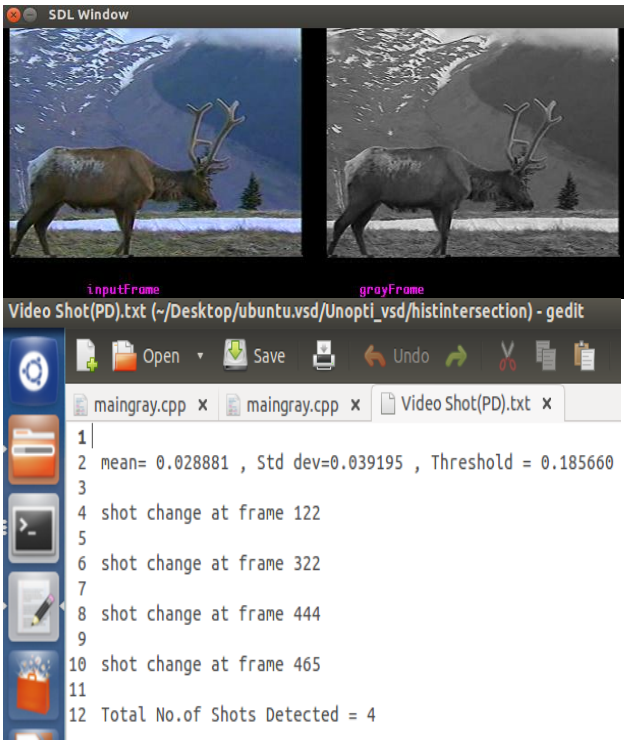
The Fig.3 represents the input video given to the histogram-difference method using intersection and also shows the output of the video on SDL screen with the result i.e. shots detected in the input video.



```

mean= 0.028881 , Std dev=0.039195 , Threshold = 0.185660

shot change at frame 122

shot change at frame 322

shot change at frame 444

shot change at frame 465

Total No.of Shots Detected = 4
```

Figure.4: Input video and output in SDL screen and result

The Fig.4 represents the input video given to the Edge Change Ratio (ECR) and also shows the output of the video on SDL screen with the result i.e. shots detected in the input video.



```

mean= 0.028881 , Std dev=0.039195 , Threshold = 0.185660

shot change at frame 122

shot change at frame 322

shot change at frame 444

shot change at frame 465

Total No.of Shots Detected = 4
```

Figure.5: Input video and output in SDL screen and result

The Fig.5 represents the input video given to the Scaled Metrics and also shows the output of the video on SDL screen with the result i.e. shots detected in the input video.

TABLE I: ANALYSIS OF VIDEO SHOT DETECTION

Methods	Shots detected in the following videos											
	Entertainment			Cricket			Football			Exotic		
	TS	FP	MS	TS	FP	MS	TS	FP	MS	TS	FP	MS
Chi-Square	32	0	36	6	0	0	6	0	0	11	0	4
Intersection	27	0	41	4	0	2	4	2	0	13	0	2
ECR	35	0	33	4	0	2	2	0	4	8	0	7
ScaledMetrics	30	0	38	18	12	0	6	0	0	4	0	11

TS: Total number of shots detected.
FP: False Positive.
MS: Missed shots.

The above table.1 contains the number of shots detected in the four different videos. The Entertainment video contains 68shots, Cricket, Football video contains 6shots each and Exotic video contain 15shots. So according to the analysis Histogram Difference using Chi-square is the best Video shot Detection method, next Histogram Difference using Intersection, Edge Change Ratio is good then Scaled Metrics.

For performance measure following formulas is used:

$$\text{Recall} = \frac{No.of\ shots\ detected}{No.of\ shots\ detected+missed}$$

$$\text{Precision} = \frac{No.of\ shots\ detected}{No.of\ shots\ detected+false\ positive}$$

TABLE III: ANALYSIS OF VSD METHODS PERFORMANCE USING RECALL AND PRECISION

Methods	Ent		Cricket		Football		Exotic	
	R%	P%	R%	P%	R%	P%	R%	P%
Chi-Square	47	100	100	100	100	100	73.3	100
Intersection	39.7	100	66.6	100	66.6	100	86.6	100
ECR	51.4	100	66.6	100	33.3	100	53.3	100
Scaled Metric	44.1	100	100	60	100	100	40	100

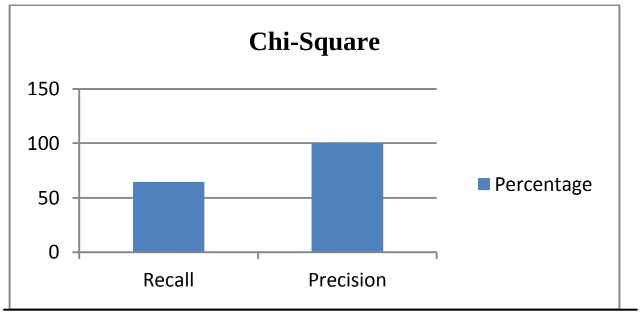


Figure.6: Graph for performance measure of Chi-square

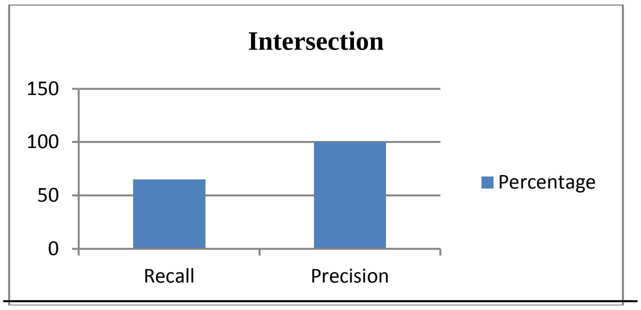


Figure.7: Graph for performance measure of Intersection

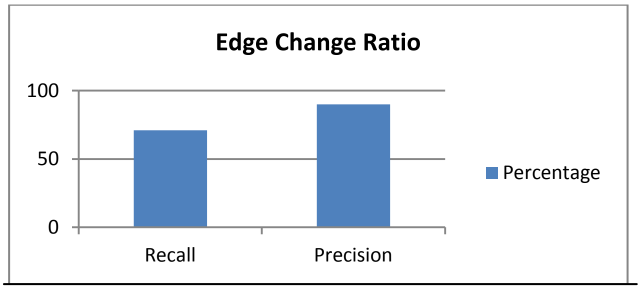


Figure.8: Graph for performance measure of Edge Change ratio

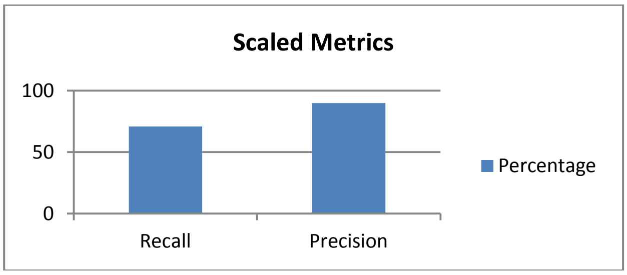


Figure.9: Graph for performance measure of Scaled Metrics

The above figure 6, 7, 8 and 9 are the graph obtained from table II. From the graph we can conclude that Histogram Difference method Chi-Square is the best method. Second best method is Histogram Difference method Intersection and next is ECR and Scaled Metrics.

TABLE IIIII: TIMING COMPARISON BETWEEN CPU AND BEAGLE-BOARD

Methods	Software Timing		Hardware Timing	
	Per total frame	Per frame	Per total frame	Per frame
Chi-Square	1856ms	3.008ms	921ms	9.49ms
Intersection	3708ms	6.009ms	30746ms	49.83ms
ECR	35976ms	5.829ms	40956ms	66.37ms
ScaledMetrics	11321ms	18.34ms	43953ms	71.23ms

The above Table.III contains the timing analysis between software and hardware i.e. the time takes for the single frame computation as well as for total frame computation both in software and in hardware. In Hardware obviously it take little bit more time because the configuration of the Beagle Board will be less compared to CPU.

VI. CONCLUSION

The work proposed in this paper is very useful in the field of robotics video surveillance and multimedia. Shot boundary detection plays very important role for a browsing, retrieving, classifying and detecting any event. Some simple methods are presented here, but used on hardware i.e. on Beagle Board. The evaluation of VSD methods proved that Histogram Difference using Chi-square and Intersection outperforms well.

ACKNOWLEDGMENT

"A good guidance takes us a long way in achieving our goals", I am very grateful to **Dr. Jharna Majumdar**, Dean R&D, HOD of CSE Dept.(PG), for guiding and kind cooperation, advise & support. And I am thankful to **Mr. Santosh Kumar K.L.** Asst. Professor of M.Tech CSE Dept (PG), for his valuable guidance.

REFERENCES

[1] "An Algorithm for Shot Boundary Detection and Key Frame Extraction Using Histogram Difference" by Ganesh.I.Rathod, Dipali.A.Nikam, International Journal of Emerging Technology and Advanced Engineering. Website: www-ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certifeid Journal, Volume 3, Issue 8, August 2013).

[2] "Performance Evalution of Interframe Histogram Diffrence Technique for Advertisment Detection in Digital Videos" by Manish Chhabra, Student Member, IEEE and Parminder Singh Reel, Member, IEEE. 2011.

[3] "Automatic Shot Boundary Detection Combining Color, Edge, And Motion Features Of Adjacent Frames" by A. Jacobs, A. Miene, G. T. Ioannidis, and O. Herzog.

[4] "Key Frame Extraction using Edge Change Ratio for Shot Segmentation" by Azra Nasreen , Dr Shobha G, International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 11, November 2013.

[5] "Shot boundary detection via similarity analysis" by Matthew Cooper, Jonathan Foote, John Adcock, and Sandeep Cas.

[6] "Design of User Interface for Shot Boundary Detection using Classical Methods" by Prof. Anand P. Mankodia , Prof. Satish K. Shah, International Journal Of Innovative Research In Electrical,

DOI 10.17148/IJARCCE.2015.4811

Electronics, Instrumentation And Control Engineering Vol. 1, Issue 7, October 2013.

[7] "Model-Based Programming of Intelligent Embedded Systems and Robotic Space Explorers" by Brian C. Williams, Michel D. Ingham, Seung H. Chung, And Paul H. Elliott, proceedings of the ieee, vol. 91, no. 1, january 2003.

[8] "Stream Image Processing on a Dual-Core Embedded System" by Michael G. Benjamin and David Kaeli, Northeastern University, Computer Architecture Research Laboratory, Springer-Verlag Berlin Heidelberg 2007.

[9] "A Remote Laboratory for Real-Time Digital Image Processing on Embedded Systems" by A. Kalantzopoulos, D. Markonis and E. Zigouris, University of Patras, Greece, Conference ICL2009 September 23-25, 2009 Villach, Austria.

[10] "A Real-Time Photogrammetry System Based On Embedded Architecture" by S. Y. Zheng, L. Gui, X. N. Wang, D. Ma , The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-5, 2014 ISPRS Technical Commission V Symposium, 23 – 25 June 2014, Riva del Garda, Italy.

BIOGRAPHIES

Priyanka S received the B.E degree in Information Science and Engineering from Jawaharlal Nehru National College of Engineering affiliated to Visvesvaraya Technological University, Belgaum in 2013 and is currently pursuing M.Tech degree in Computer Science and Engineering from Nitte Meenakshi Institute of Technology affiliated to Visvesvaraya Technological University, Belgaum.

Dr.Jharna Majumdar, presently the Dean of R&D and the HOD of CSE (PG). She served DRDO from 1990 to 2007 and retired as Scientist G and Head of Aerial Image Exploitation Division, Aeronautical Development Establishment (DRDO), Bangalore, India. Dr. Majumdar received B.Tech (Hons.) in Electronics and Electrical Engineering and Post Graduate in Computer Technology from Indian Institute of Technology Kharagpur in 1969 and 1970 respectively. She received her PhD (Electrical Engineering) in 1980.

Santhosh Kumar K L received B.E. and M.Tech in Computer Science and Engineering from Visvesvaraya Technological University. He is currently working as a Lecturer in the Department of Computer Science & Engg. Nitte Meenakshi Institute of Technology, Bangalore. His areas of interest are image and video processing, computer vision and pattern recognition.

DOI 10.17148/IJARCCE.2015.4811