iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** VI **Month of publication:** June 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 3 Issue VI, June 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Combined Approach for Video Shot Boundary Detection with Monoslam and Image Quality Based Features

Amritanshu Upadhyay[1], Ravi Mishra[2]
[1]M.E. Scholar [VLSI Design], Dept. of ETC, [2] Sr. Assistant Professor Dept. of EEE
Shri Shankaracharya Technical Campus, Bhilai, Chhattisgarh

***Abstract*— The use of three dimensional information from video is rare in the video analysis literature due to the inherent difficulties of extracting accurate 3D measurements from a single view of a scene. Several methods have been published in recent years, however, that attempt to solve such a problem. They all use the same underlying meaning of exploiting camera motion in order to measure the parallax of visible objects in the scene. In this paper, we employ the use of such algorithms towards solving the problem of automatic shot boundary detection using MonoSLAM and Image Quality Features. The idea is to extract salient features from a video sequence and track them over time in order to estimate shot boundaries within the video. We detect shot boundaries in videos by observing the system's ability to successfully track features across frames.**
***Keywords*— Shot Detection, MonoSLAM, Image Quality Features.**

I. INTRODUCTION

Videos have become very popular in many areas such as communications, education and entertainment. A huge collection of video clips, live TV programs and movie pictures can be found on the Internet. Movies and edited videos consist of scenes, such as a dialog between two people. Scenes consist of one or more shots or consecutive frames as captured with a single camera. Locating transitions between shots, also called cuts or shot boundaries, is fundamental procedure for analyzing videos such as indexing videos, querying scenes, searching objects, or summarizing video contents. [11] Videos are organized in a hierarchical structure of stories, scenes, shots and frames as shown in Fig. 1. Such representations start with the actual video at the highest level. The video may be broken down into scenes, each containing some semantic meaning. Scenes can be broken down further into shots, which may or may not contain semantic meaning. These shots can then be broken down even further into the individual video frames that make up the video.

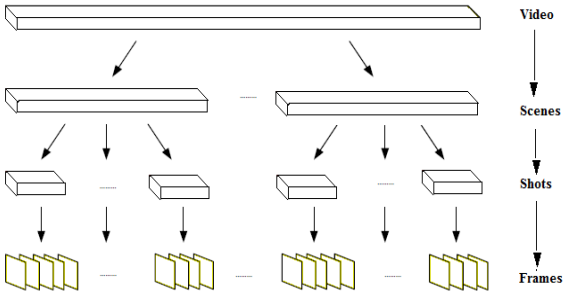


Fig. 1 Hierarchical structure of a video clip

The video shot is a basic structural building block of video sequences. Its boundaries need to be determined possibly automatically to allow for content-based video manipulation. A video shot can be defined as a sequence of frames captured by one camera in a single continuous action in time and space [13]. It should be a group of frames that have consistent visual (including color, texture, and motion) characteristics. A Shot boundary is the transition between two shots. It can be abrupt or gradual. According to whether the transition between shots is abrupt or gradual, the shot boundaries can be categorized into two types: cut (CUT) and gradual transition (GT). The GT can be further classified into dissolve, wipe, fade out/in (FOI), etc., according to the characteristics of the different editing effects. This paper presents a combined approach for video shot boundary detection using two dimensional and three dimensional features of an image. By using MonoSLAM and image quality based features we can detect shot boundaries within a video.

©IJRASET 2015: All Rights are Reserved

II. LITERATURE REVIEW

H. Y: Mark Liaoff et al [2002] proposed a novel dissolve detection algorithm which could avoid the mis-detection of motions by using binomial distribution model to systematically determine the threshold needed for discriminating a real dissolve from global or local motions. **Jesús Bescós [2004],** proposed a Real-Time Shot Change Detection over Online MPEG-2 Video where it describes a software module for video temporal segmentation which is able to detect both abrupt transitions and all types of gradual transitions in real time. **Guillermo Cisneros et al [2005]** proposed a paper on A Unified Model for Techniques on Video-Shot Transition Detection. The approach presented here is centred on mapping the space of inter-frame distances onto a new space of decision better suited to achieving a sequence independent thresholding. **Liuhong Liang et al [2005],** presented an Enhanced Shot Boundary Detection Using Video Text Information, in which a number of edge-based techniques have been proposed for detecting abrupt shot boundaries to avoid the influence of flashlights common in many video types, such as sports, news, entertainment and interviews videos. **Daniel DeMenthon et al [2006]** proposed a paper on Shot boundary detection based on Image correlation features in video. This paper is based on image correlation features in the videos. The cut detection is based on the so-called 2max ratio criterion in a sequential image buffer. The dissolve detection is based on the skipping image difference and linearity error in a sequential image buffer. **Kota Iwamoto et al [2007]** proposed Detection of wipes and digital video effects based on a pattern-independent model of image boundary line characteristics which is based on a new pattern independent model. These models rely on the characteristics of image boundary lines dividing the two image regions in the transitional frames.

III. METHODOLOGY

The aim of the proposed methodology is to provide a GUI for simple and better video shot boundary detection using the 2D image features measurements and relative 3D structure of the scene by using MonoSLAM techniques.

A. Shot Boundary Detection Using MonoSLAM

MonoSLAM uses as input a video stream from a single monocular camera. This data is no different from the data available in a single video of an observed scene, and thus the MonoSLAM algorithm can technically be applied to videos as well. The main idea of our approach is to use this framework to track objects in the scene. If at any point the algorithm fails to track all of the previously-observed objects in the scene, we assume that a shot boundary has been detected. After initial startup, the map is updated using the Extended Kalman Filter (EKF). The map itself is composed of a state vector $\hat{x}$ and a covariance matrix P. The state vector $\hat{x}$ provides an estimate of the camera and visual features being tracked in the world. This vector is composed of the camera vector $\hat{x}_v$ and a feature vector $\hat{y}_i$ for each feature inserted into the map. The covariance matrix P is a square matrix that can be divided into individual sub-matrix elements, allowing the probability distribution to be approximated by a single multivariate Gaussian distribution. This state vector and covariance matrix can be written as

$$\hat{x} = \begin{pmatrix} \hat{x}_v \\ \hat{y}_1 \\ \hat{y}_2 \\ \vdots \end{pmatrix}, \quad P = \begin{bmatrix} P_{xx} & P_{xy_1} & P_{xy_2} & \dots \\ P_{y_1x} & P_{y_1y_1} & P_{y_1y_2} & \dots \\ P_{y_2x} & P_{y_2y_1} & P_{y_2y_2} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \quad (1)$$

The covariance matrix P is a full covariance matrix that models the relationship between features in the probabilistic map. This allows MonoSLAM to accurately localize the camera and compute the 3D locations of each feature relative to each other very accurately. As mentioned above, the state vector is composed of the camera and feature vectors, which keep track of the main elements in the scene. The camera vector $\hat{x}_v$ keeps track of the extrinsic camera parameters, while each feature vector $\hat{y}_i$ stores the 3D location and orientation of each feature being tracked. These vectors can be written as:

$$\hat{x}_v = \begin{pmatrix} r^{WC} \\ q^{WC} \\ v^W \\ \omega^W \end{pmatrix}, \quad y_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ \theta_i \\ \phi_i \\ \rho_i \end{pmatrix}. \quad (2)$$

Here, the rWC parameters stores the 3D location of the camera, qWC stores the relative orientation of the camera, while the v^W and ω^W store the camera's velocity and angular velocity respectively. For each feature vector $\hat{y}_i$, the first three terms (xi; yi; zi) define the 3D location of the camera's optical center at the time the feature was first observed, (θ_i, ϕ_i) is the azimuth and

©IJRASET 2015: All Rights are Reserved

elevation for the ray $m(\theta_i, \phi_i)$ from the camera to the observed feature, and ρ_i is the inverse depth $\left(\frac{1}{d_i}\right)$ of the feature along this ray. These parameters keep track of a 3D point located at:

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \frac{1}{\rho_i}\mathbf{m}(\theta_i, \phi_i). \qquad (3)$$

This basic scene geometry is illustrated in Fig 2.

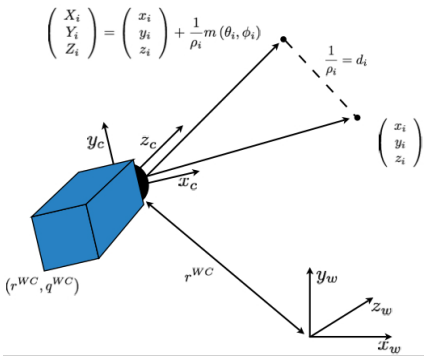


Fig 2. Basic geometry of the scene

1) Features Extraction: Image features are initially found in the image by searching for salient image regions using RANSAC algorithm. Random Sample Consensus (RANSAC) has become one of the most successful techniques for robust estimation from a data set that may contain outliers. It works by constructing model hypotheses from random minimal data subsets and evaluating their validity from the support of the whole data. Combination of RANSAC plus Extended Kalman Filter (EKF) that uses the available prior probabilistic information from the EKF in the RANSAC model hypothesizes stage.

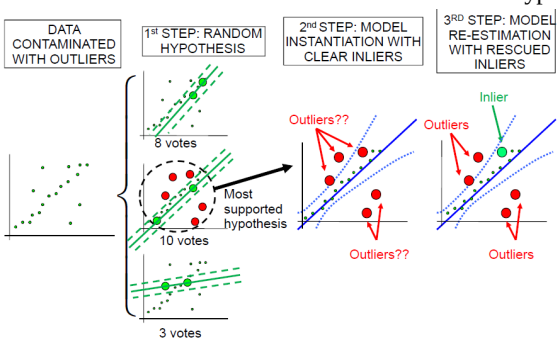


Fig 3 RANSAC steps for the simple 2D line estimation example

First, random hypotheses are generated from data samples of size two, the minimum to define a line. The most supported one is selected, and data voting for this hypothesis is considered inliers. Model parameters are estimated from those clear inliers in a second step. Finally, the remaining data points consistent with this latest model are rescued and the model is re-estimated again.

2) Shot Detection : When a feature is first initialized, it is immediately inserted into the map with a large possible depth range of [1; ∞]. As the feature is re observed over time, the EKF re-estimates the depth of each feature until it converges to a more accurate depth value. For this to occur there must be enough parallax observed by the camera; otherwise, the system assumes the features are at infinity. The camera vector is updated at each iteration via the EKF according to this model. At each iteration, the system updates the camera parameters as well as the feature vectors for all successfully tracked features. If the system fails to successfully track a given feature at any point in time, that feature is marked as one to be possibly deleted. If the same feature is unsuccessfully tracked for many consecutive frames, it is removed from the map and deleted from the system. If at any point the algorithm marks all of the visible features to be possibly deleted, then we assume a shot boundary has occurred.

©IJRASET 2015: All Rights are Reserved

Volume 3 Issue VI, June 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

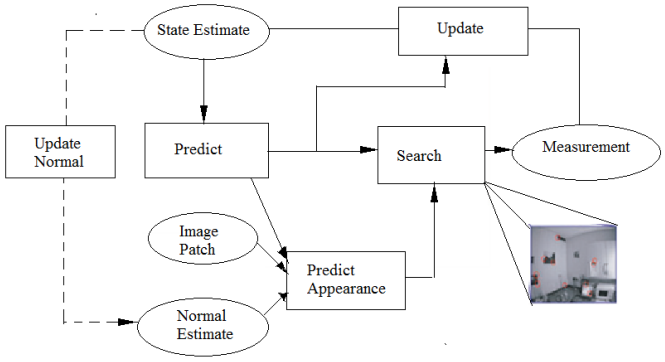


Fig 4. Processing cycle for estimating the 3D orientation of planar feature surfaces

Fig. 3 displays the processing steps in feature orientation estimation. When a new feature is added to the map, we initialize an estimate of its surface normal which is parallel to the current viewing direction, but with large uncertainty. We currently make the simplifying approximation that estimates of feature normals are only weakly correlated to those of camera and feature positions. Normal estimates are therefore not stored in the main SLAM state vector, but maintained in a separate two-parameter EKF for each feature.

B. Shot Boundary Detection Using 2D Image Quality Features

We use Edge detection, Histogram difference, Thinning, SSIM (Structured Similarity Index Metrics), Mean Squared Error, Peak Signal to Noise Ratio, Mean Deviation and Standard Deviation techniques to detect video shots.

1) Edge Detection: In this method the edges of successive aligned frames are detected first and then the edge pixels are paired with nearby edge pixels in the other image to find out if any new edges have entered the image or if some old edges have disappeared. We use Canny edge Detector.

2) Histogram Difference: This method computes gray or color histograms of the two consecutive frames of video. If the difference between the two histograms is above a threshold, a shot boundary is assumed.

$$D(i,i+1) = \sum_{j=1}^{n} |H_i(j) - H_{i+1}(j)| \qquad (4)$$

Equation (4) is used for histogram based shot boundary detection. Where j indicates the gray level value. Hi(j) is the histogram for the gray level j in the frame i and n is the total number of gray levels. This method is less sensitive to object a camera motion. This method detects hard-cut, fade and dissolves and fails when there is large amount of motion. Histograms are the most common method used to detect shot boundaries. The simplest histogram method computes gray level or color histograms of the two images. If the bin-wise difference between the two histograms is above threshold, a shot boundary is assumed. Histogram differences is very similar to Sum of absolute differences. The difference is that HD computes the difference between the histograms of two consecutive frames; a histogram is a table that contains for each color within a frame the number of pixels that are shaded in that color.

3) SSIM: The structural similarity (SSIM) index is a method for measuring the similarity between two images. The SSIM index is a full reference metric; in other words, the measuring of image quality based on an initial uncompressed or distortion-free image as reference. The SSIM metric is calculated on various windows of an image. The measure between two windows x and y of common size N×N is:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \qquad (5)$$

with

μ_x the average of x;

μ_y the average of y;

σ_x^2 the variance of x;

©IJRASET 2015: All Rights are Reserved

σ_y^2 the variance of y;

σ_{xy} the covariance of x and y;

$c_1=(k_1 L)^2$, $c_2=(k_2 L)^2$ two variables to stabilize the division with weak denominator;

L the dynamic range of the pixel-values (typically this is $2^{\#bits\ per\ pixel}-1$);

$k_1=0.01$ and $k_2=0.03$ by default.

The resultant SSIM index is a decimal value between -1 and 1, and value 1 is only reachable in the case of two identical sets of data.

4) MSE: The MSE is the cumulative squared error between the compressed and the original image. A lower value for MSE means lesser error.

For m×n monochrome image I and its noisy approximation K, MSE is defined as:

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2 \quad (6)$$

5) PSNR: Peak signal-to-noise ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale.

The PSNR (in dB) is defined as:

$$\begin{aligned} PSNR &= 10\cdot\log_{10}\left(\frac{MAX_I^2}{MSE}\right) \\ &= 20\cdot\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \\ &= 20\cdot\log_{10}(MAX_I) - 10\cdot\log_{10}(MSE) \end{aligned} \quad (7)$$

As seen from the inverse relation between the MSE and PSNR, this translates to a high value of PSNR. Logically, a higher value of PSNR is good because it means that the ratio of Signal to Noise is higher. Here, the 'signal' is the original image, and the 'noise' is the error in reconstruction.

IV. RESULT & DISCUSSION

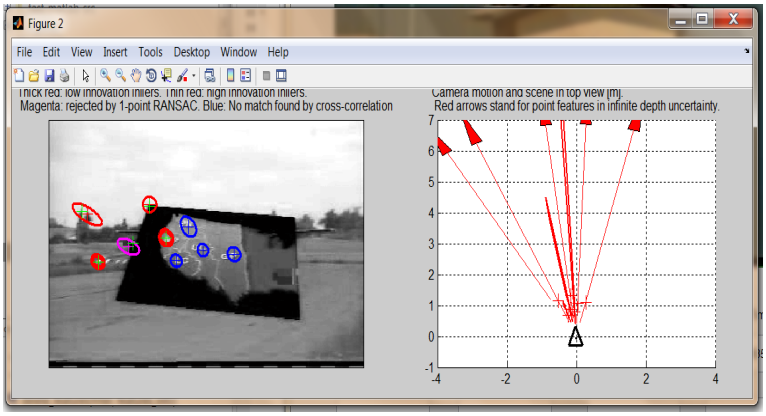


Fig 5. Tracking features in a video.
Thick red: low innovation inliers. Thin red: high innovation inliers. Magenta: rejected by RANSAC. Blue: No match found by cross correlation

©IJRASET 2015: All Rights are Reserved

TABLE 1 RESULT OF IMAGE QUALITY FEATURES BASED SHOT DETECTION

Image Quality Features	No. of Consecutive Frames														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Edge Difference	0	0	8	6	11	11	30	28	8	15	9	9	1	8	13
SSIM	1	1	0.9315	0.9227	0.9297	0.9335	0.9321	0.9472	0.9584	0.9609	0.9732	0.9777	0.9778	0.9873	0.9909
MSE	0	0	0.0215	0.0197	0.0192	0.0215	0.0191	0.0146	0.0129	0.0127	0.0127	0.0101	0.0107	0.0105	0.0063
PSNR	Inf	Inf	64.839	65.199	65.327	64.826	65.348	66.519	67.035	67.108	67.120	68.084	67.865	67.919	70.141
Mean Deviation	0	0	0.8777	1.0793	1.9911	0.9963	1.3991	2.0990	0.7788	1.2903	1.3584	0.4310	0.7252	1.0122	0.2599
Standard Deviation	0	0	0.4580	0.6843	0.5629	0.0703	0.0898	0.1045	0.4027	0.3846	0.2150	0.3135	0.2163	0.0898	0.1925

The table shows the values of image quality features for 15 consecutive frames. When there is large difference in the values, a new shot has been detected.

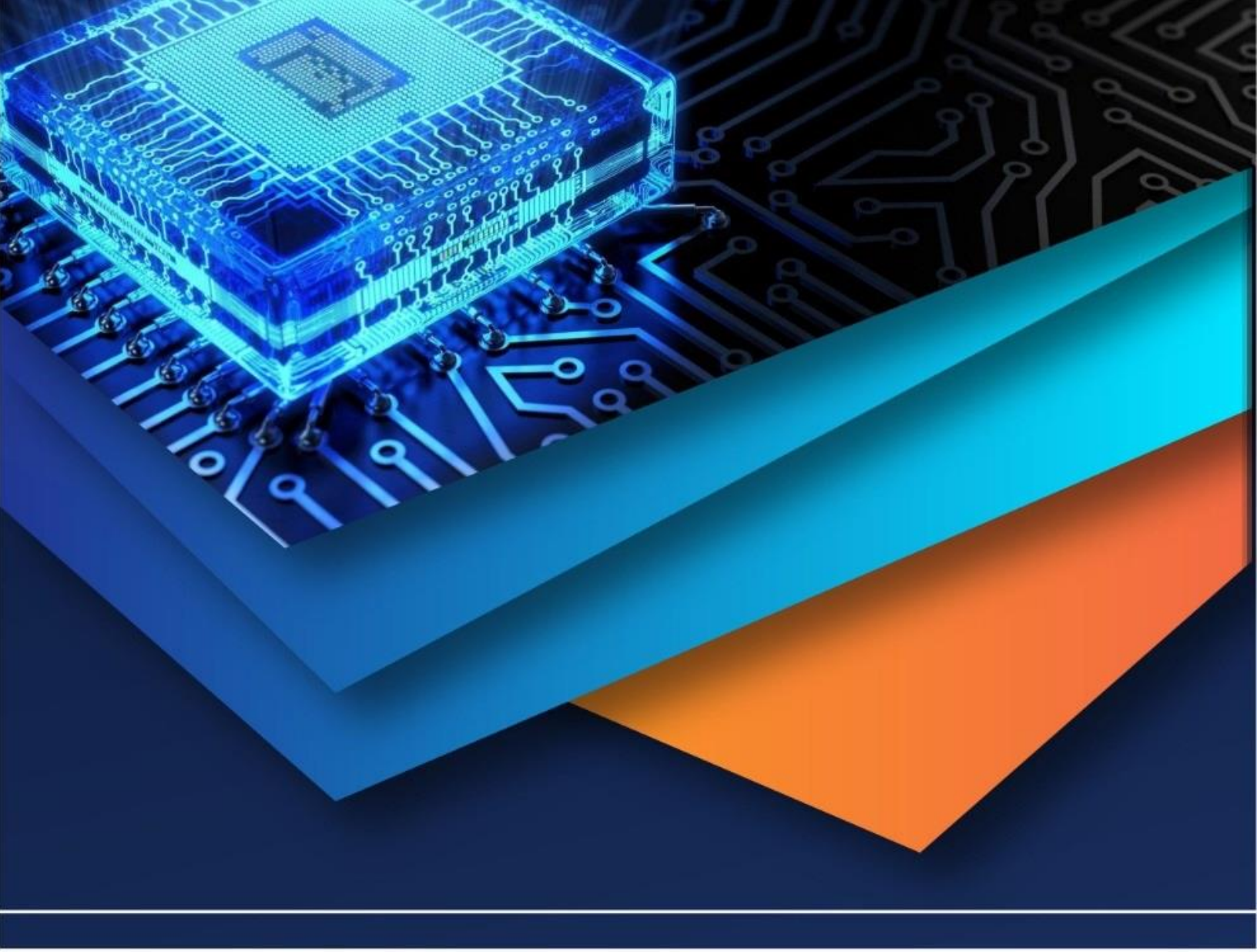
V. CONCLUSION

Shot boundary detection is temporal video segmentation, and is the process of identifying the transitions between the adjacent shots. The work in video processing and analysis is a very important part of searching and browsing of digital video. We can use the shot boundaries to analyze video data in shot in greater depth such as video indexing, shot similarity etc. Various methods to perform object and feature tracking, we show that SLAM algorithms provide a powerful solution to object tracking. More specifically, we employ the use of the MonoSLAM algorithm. By modelling the scene with a probabilistic 3D map, we are able to track objects in the scene while estimating their relative 3D position. By using the inherent three dimensional structure of the observed scene, we can track different objects in the scene accurately. Combined approach for video shot boundary detection using two dimensional and three dimensional features of an image has been developed. By using MonoSLAM and image quality based features we can detect shot boundaries within a video very effectively.

REFERENCES

[1] H. Y: Mark Liaoff C. W. Su*, H. R. Tyanf and L. H. Chen "A motion-tolerant dissolve detection algorithm" IEEE 2nd Pacific-Rim Conference on Multimedia, Beijing, China, ~V0l.2195, oct 2002
[2] Jesús Bescós, "Real-Time Shot Change Detection over Online MPEG-2 Video". IEEE transactions on circuits and systems for video technology, vol. 14, no. 4, april 2004.
[3] Jesús Bescós, Guillermo Cisneros, José M. Martínez, José M. Menéndez, and Julián Cabrera "A Unified Model for Techniques on Video-Shot Transition Detection" IEEE transactions on multimedia, vol. 7, no. 2, april 2005
[4] Zhe Ming Lu and Yong Shi "Fast Video Shot Boundary Detection Based on SVD and Pattern Matching-"Image processing IEEE Transactions (Volume: 22, Issue: 12), Dec. 2013
[5] Arturo Donate and Xiuwen Liu, "Shot Boundary Detection in Videos Using Robust Three-Dimensional Tracking" 978-1-4244-7030-3/10, IEEE 2010
[6] Kyoungmin Lee, and Mathias Kolsch "Shot Boundary Detection with Graph Theory using Keypoint Features and Color Histograms" Winter Conference on Applications of Computer Vision, IEEE 2015.
[7] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(6):1052–1067, 2007.
[8] Lenka Krulikovská, Jaroslav Polec, "An Efficient Method of Shot Cut Detection" World Academy of Science, Engineering and Technology Vol:6 March, 2012.
[9] Mr. Sandip T. Dhagdi, Dr. P.R. Deshmukh "Key frame Based Video Summarization Using Automatic Threshold & Edge Matching Rate" International Journal of Scientific and Research Publications, Volume 2, Issue 7, July 2012.
[10] Xin-Wen Xu, Guo-Hui Li, Jian Yuan, A Shot Boundary Detection Method For News Video Based On Object Segmentation And Tracking, Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 12-15 July 2008.
[11] Partha Pratim Mohanta, Sanjoy Kumar Saha, Member, IEEE, and Bhabatosh Chanda[2012] Member IEEE, "A Model-Based Shot Boundary Detection Technique Using Frame Transition Parameters" IEEE transactions on multimedia, vol. 14, NO. 1, february 2012.
[12] Javier Civera , Oscar G. Grasa , Andrew J. Davison, J. M. M. Montiel "1-Point RANSAC for EKF Filtering. Application to Real-Time Structure from Motion and Visual Odometry"
[13] Zhang Xi1, Xu Fang, Song Zhen, Mei Zhibin, Video Flame Detection Algorithm Based On Multi-Feature Fusion Technique, IEEE 2012.

©IJRASET 2015: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)