Indian Statistical Institute Kolkata

M.Tech (Computer Science) Dissertation

Outlier Detection and Motion Estimation Approach for Shot Boundary Detection

---

A dissertation submitted in partial fulfillment of the requirements
for the award of Master of Technology
in
Computer Science

Author:
Saurabh SHIGWAN
Roll No: CS1205

Supervisor:
Prof. Ashish GHOSH
Machine Intelligence Unit

Certificate of Approval

This is to certify that the thesis entitled **Outlier Detection and Motion Estimation Approach for Shot Boundary Detection** by **Saurabh Shigwan** towards partial fulfillment for the degree of **M.Tech. in Computer Science** at **India Statistical Institute, Kolkata**, embodies the work done under my supervision.

Prof. Ashish Ghosh,Head,
Machine Intelligence Unit,
Indian Statistical Institute.

Date : 16th $July$**,** 2014

Acknowledgements

At the end of my dissertation and my M.Tech. training at the Indian Statistical Institute Kolkata, I want to thank and give credit to all individuals who have provided me with invaluable assistance. Whether be it gentle guidance or access to materials or services that helped me a lot in my research work, it is greatly appreciated.

First and foremost I offer my sincerest gratitude to my supervisor, Prof. Ashish Ghosh, who has supported me throughout my thesis with his patience and knowledge. It was a memorable learning experience. For his patience, for all his advice and encouragement and for the way he helped me to think about problems with a broader perspective, I will always be grateful. One simply could not wish for a better or friendlier supervisor.

I would like to thank all the professors at the Indian Statistical Institute Kolkata who have made my educational life exciting and helped me to gain a better outlook on computer science. I would like to express my gratitude to Ramchandra Murthy,Rahul Roy, Badri Subudhi for interesting discussions. Without the precious suggestions from Ramchandra Murthy, it would have been considerably difficult to finish this report.

I would like to thank everybody at Indian Statistical Institute, Kolkata for providing a wonderful atmosphere for pursuing my studies. I thank all my classmates, seniors and juniors who have made the academic and non-academic experiences very delightful.

My most important acknowledgement goes to my family and friends who have filled my life with happiness.

Abstract

Shot Boundary Detection problem is most fundamental problem in content-based video retrieval system. The main idea of this work is to consider shot boundaries as outliers of a given feature space. Here we have considered shot boundaries as outlier, because within-shot boundaries are very large in number compared to shot boundaries. This is a novel concept, which helps to remove the complexity of gathering training set for supervised classification. To tackle next problem of getting some within-shot boundaries as outlier, we have used multi-resolution based motion estimation algorithm to check, whether outlier boundary is part of high motion of camera or object. So, this post-processing step of removal of false-positive points improves the final result of shot boundary detection.

Contents

List of Figures

Chapter 1

Introduction

Recent advances in multimedia technology, coupled with the significant increase in computer performance and the growth of the Internet, have led to the widespread use and availability of digital videos. The rapidly expanding applications of videos have spurred the growing demand of new technologies and tools for efficient indexing, browsing and retrieval of video data. The area of content based video retrieval, aiming to automate the indexing, retrieval and management of video, has attracted extensive research during the last decade. Shot Boundary Detection is step towards that goal.

1.1 Definitions

1.1.1 Video

3D signal, in which, first two dimensions reveal the visual content in horizontal and vertical direction, and third one reveals the variation of visual content over time.

1.1.2 Shot

A shot is defined as unbroken sequence of frames captured by a camera.

1.1.3 Shot Boundary

shot boundaries are characterized by a significant difference between successive frames in a video sequence. Two basic types of shot boundaries are usual:

- Sharp Changes.

- Gradual Changes (such as fade-out/fade-in, dissolve, wipe, and similar effects).

Focus of this work is on sharp changes (or hard cuts).

1.1.4   Shot Boundary Detection (SBD)

It aims to temporally segment the video into consecutive shots.

1.1.5   Visual Discontinuities

Significant change in their visual content shown by frames surrounding a shot boundary, is called as Visual Discontinuities.

1.2   Difference between Shot and Scene

A shot is an image sequence captured by a single camera in continuous time duration. While a scene is a collection of consecutive shots, which possess a semantic meaning. For example, Video of eating a breakfast is scene, but picking up bread, applying butter, taking plate, etc. could be shots.

Shot is a fundamental element of video, which possess temporal information. A shot is easier to detect and process, compared to whole scene at a time.

1.3   Applications

Any video content based retrieval system requires key-frames inside that video. These key-frames can extracted efficiently, if individual shot can be extracted from that video. SBD proves it's importance here. There are many important field, where SBD can be used, like object detection, Activity recognition.

1.4   Problem Analysis

Our focus is to select such features which highlight this discontinuities more accurately. Let a m-dimensional feature vector, $W \subseteq \mathbb{R}^m$ is used to quantify variation in visual content from frame k to frame $k + T$, where, k is index of frames, while T is relative index. Let $P \subseteq W$ represents within-shot frames and $Q \subseteq W$ represents inter-shot frames. Then ideally $P \cap Q$ is must be equal to ϕ. But practically, it is not equal to ϕ, due to visual discontinuities presents within shots, like object or camera motion or illumination variation.

1.5   Existing Literature

Over the past decade, SBD has received a considerable amount of research attention, and a large variety of methods have been proposed to address the problem. Some recent reviews have reported that, using simple video intensity or color features, existing SBD methods can achieve performance up to 90% recall (the percentage of the number of correct detections from the actual shot boundaries) and 80% precision (the percentage of the number of correct detections in all the detected shot boundaries) for hard cuts for specific video data set [13, 9].

A number of existing methods manage to improve the performance of SBD by eliminating false positives (or false detections) and identifying false negatives (or miss detections) during the detection process [10].

Chapter 2

Block-wise mean of Image Difference

2.1 Definitions

2.1.1 Image Difference

It is a matrix of pixel-by-pixel intensity difference of two image matrices [9]. It mainly used to capture change in visual information from one image to next image.

2.1.2 Block-wise mean

It divides image matrix into $n \times n$ equally sized block. If we compute mean pixel intensity value for each block, then we get $n \times n$ block-wise mean matrix.

2.1.3 Block-wise mean of Image difference

It is defined as follow,

$$W_{ij}(t) = \frac{1}{S_z} \sum_{(x,y) \, \epsilon \, B_{ij}} |I(x,y,t) - I(x,y,t-1)| \tag{2.1}$$

Where, $i, j \epsilon \{1, 2, ..., n\}$, Sz– no. of pixel in each block. W matrix is an $n \times n$ sized matrix.

2.2 Representation

Consider W matrix as an n^2-dimensional feature vector. If we are computing this features for m length video sequence, and if we apply block-wise mean of image difference for each consecutive image frame, then we get such $m-1$ features. So we have convert visual content of frame to $m \times n^2$ dimensional data set.

2.3 Advantages and Disadvantages

Advantages

It possess both temporal and spatial information of two consecutive frames. Also it is comparatively easy to compute, compared to frequency domain features.

Disadvantages

It is sensitive to object motion and illumination variation.

Chapter 3

DBSCAN

3.1 Introduction

The DBSCAN (**Density-Based Spatial Clustering of Application with Noise**) algorithm can identify clusters in large spatial data sets by looking at the local density of database elements, using only one input parameter. Furthermore, the user gets a suggestion on which parameter value that would be suitable. Therefore, minimal knowledge of the domain is required. The DBSCAN can also determine what information should be classified as noise or outliers. In spite of this,its working process is quick and scales very well with the size of the database,almost linearly.

By using the density distribution of nodes in the database, DBSCAN can categorize these nodes into separate clusters that define the different classes. DBSCAN can find clusters of arbitrary shape. However, clusters that lie close to each other tend to belong to the same class [6].

3.2 Algorithm

Following section will describe working of DBSCAN algorithm. It's working is based on six definition and two lemma.

$\varepsilon\text{-}neighborhood$ **of a point:** $N_\varepsilon(p) = \{q \epsilon D \,|\, dist(p, q) < \varepsilon\}$

For a point to belong to a cluster it needs to have at least one other point that lies closer to it than the distance ε.
There are two kinds of points belonging to a cluster; there are border points and core points.

Directly density-reachable: Points q which satisfies following two condition w.r.t. core point p,

1. $p \,\epsilon\, N_\varepsilon(p)$

2. $|N_\varepsilon(p)| \geq Minpts$ (**Core point condition**)

The $\varepsilon\text{-}neighborhood$ of a border point tends to have significantly less points than the $\varepsilon\text{-}neighborhood$ of a core point. The border points will still be a part of the cluster and in order to include these points, they must belong to the $\varepsilon\text{-}neighborhood$ of a core point q.

Density reachable: A point p is $density-reachable$ from a point q with respect to ε and $MinPts$ if there is a chain of points $\{p_1 ..., p_n\}, (p_1 = q, p_n = p)$ such that p_{i+1} is directly density-reachable from p_i.

There are cases when two border points will belong to the same cluster but where the two border points don't share a specific core point. In these situations the points will not be density-reachable from each other. There must however be a core point q from which they are both density-reachable.

Density Connected: A point p is *density-connected* to a point q with respect to ε and $MinPts$ if there is a point o such that both, p and q are *density-reachable* from o with respect to ε and $MinPts$.

Clusters:

- If point p is a part of a cluster C and point q is *density-reachable* from point p with respect to a given distance and a minimum number of points within that distance, then q is also a part of cluster C.

- Two points belongs to the same cluster C, is the same as saying that p is *density-connected* to q with respect to the given distance and the number of points within that given distance.

Noise: A cluster can be formed from any of its core points and will always have the same shape.

Lemma1:

A cluster can be formed from any of its core points and will always have the same shape.

Lemma2:

Let p be a core point in cluster C with a given minimum distance (ε) and a minimum number of points within that distance ($MinPts$). If the set O is *density-reachable* from p with respect to the same ε and $MinPts$, then C is equal to the set O.

To find a cluster, DBSCAN starts with an arbitrary point p and retrieves all points *density-reachable* from p with respect to ε and $MinPts$. If p is a core point, this procedure yields a cluster with respect to ε and $MinPts$ (see *Lemma2*). If p is a border point then no points are *density-reachable* from p and DBSCAN visits the next point of the database.

3.3 Advantages and Disadvantages

3.3.1 Advantages

- DBSCAN does not require one to specify the number of clusters in the data a priori, as opposed to k-means.

- It has a notion of noise, and is robust to outliers.

- It requires just two parameters and is mostly insensitive to the ordering of the points in the database.

- It can find non-linearly separable clusters.

3.3.2 Disadvantages

- It cannot cluster data sets well with large differences in densities.

Chapter 4

Muti-Resolution Motion Estimation (MRME)

This algorithm is unique in itself. Instead of an inter-frame similarity measure which is directly intensity based, It exploits image motion information, which is generally more intrinsic to the video structure itself.

This shot change detection method is related to the computation, at each time instant, of the dominant image motion represented by a two-dimensional affine model [3].

4.1 Intuition and working

This algorithm involve estimation of motion parameters from two consecutive frames. It works by forming parameterized optical flow equation,which is to be minimized iteratively over time period. This minimization problem is solved by a robust estimation method [12, 1]. It involve approximating minimization problem to weight-least square estimation [5].

The output of this method is estimated values of motion parameters (like translation, rotation, scaling) and weight matrix containing weight values for each pixel. Pixel positions, which possess high weight value, follows motion of the estimated parameters. Pixel positions, which have low weight values, does not follow the estimated motion.

4.2 Dominant Motion Estimation

This algorithm make use of the by taking into account the usual first-order derivative of spatio-temporal derivatives of the intensity function. Since several motions may be present, we only seek for the estimation of the dominant one. The corresponding motion field between two successive images is represented by global 2-D parametric model.

Affine model W_Θ defined at $p = (x, y)$,

$$W_\Theta(t) = \begin{pmatrix} a1 + a2 * x + a3 * y \\ a4 + a5 * x + a6 * y \end{pmatrix} \tag{4.1}$$

where, $\{a1, a2, a3, a4, a5, a6\}$ are 2-D parameters of model.

This model is a good trade-off between complexity and representative. Jointly to the affine motion model, a seventh parameter is estimated, which is a global inter-frame intensity variation, denoted η.

The parameter vector $\Theta = (a1, a2, a3, a4, a5, a6, \eta)$ is estimated between images $I(t)$ and $I(t+1)$ as follows:

$$\Theta_{est} = \arg\min_{\Theta} \sum_{p \, \epsilon \, \xi} \rho(DFD_{\Theta}(p)) \tag{4.2}$$

where, $DFD_{\Theta} = I(p + W_{\Theta}(p), t+1) - I(p, t) + \eta$, (i.e. Displaced Frame Difference) and $\rho(x)$ is hard re-descending robust estimator and ξ is image grid.

It consider $\rho(x)$ function as Tukey's bi-weight function. The purpose of such a robust estimator [8] in contrast to a least mean square procedure, in which $\rho(x) = r^2$, is that the estimation is not affected by strong deviations from the model.

In particular, Tukey's function enforces a radical limitation of the influence of outlier. The function $\rho(x)$ and its derivative $\psi(x)$ on parameter C, and are respectively defined as:

$$\rho(x) = \begin{cases} \frac{x^6}{6} - \frac{C^2 x^4}{2} + \frac{C^4 x^2}{2} & |x| < C \\ \\ \frac{C^6}{6} & \text{otherwise} \end{cases} \tag{4.3}$$

$$\psi(x) = \begin{cases} x(x^2 - C^2)^2 & |x| < C \\ \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

Here parameter C is a small positive number. In our implementation, we have consider $C = 10$.

4.3 Multi-Resolution Algorithm

This algorithm helps to focus dominant motion occurring in frame. It works by creating multi-resolution pyramid of frames. Estimation procedure is done through iterative calculation of parameter at each step of resolution.

Since, visual information is less at lowest resolution, parameter is estimated by taking first order derivative of expression $DFD_{\Theta}(p)$ as follow:

$$\Theta^0_{est} = \arg\min_{\Theta^0} \sum_{p \, \epsilon \, \xi} \rho(r_{\Theta^0}(p)) \tag{4.5}$$

where, $r_{\Theta^0}(p) = I(p, t+1) - I(p, t) + \nabla I(p, t+1).W_{\Theta^0}(p) + \eta^0$

Then, increments are computed within resolution level and from a resolution level to next higher one. At step k, we aim at estimating the increment $\Delta\Theta^k$ given by $\Theta = \Theta^k + \Delta\Theta^k$, where Θ^k is current estimate of motion parameter vector. The estimated increment $\Delta\Theta^k$ is calculated as

$$\Delta\Theta^k_{est} = \arg\min_{\Delta\Theta^k} \sum_{p \, \epsilon \, \xi} \rho(r_{\Delta\Theta^k}(p)) \tag{4.6}$$

where, $r_{\Delta\Theta^k}(p) = I(p + W_{\Theta^k}, t+1) - I(p, t) + \nabla I(p + W_{\Theta^k}, t+1).W_{\Delta\Theta^k}(p) + \Delta\eta^k$
Then estimated increment is used to update Θ_{est} as $\Theta_{k+1} = \Theta_k + \Delta\Theta_k$.

Increments are computed and accumulated, until a convergence criterion is met or a given number of iterations is reached. The estimated motion parameter vector at a given level is projected onto the level of higher resolution and serves as an initial value to compute some more increments and thereby to refine Θ, this down to the finest resolution level in the image pyramid. The use of this robust estimator allows us to get an accurate computation of the dominant motion between two images, even if other motions are present.

4.4   Detection of shot change

The minimization problem, defined above, can be approximated to weighted least squared estimation [5] problem as follow:

$$\Theta_{est} \ = \ \arg\min_{\Theta} \sum_{p\,\epsilon\,\xi} \frac{1}{2}\,\omega(p).r_\Theta(p)^2 \tag{4.7}$$

where, $\omega(p) \ = \ \frac{\psi(r_\Theta(p))}{r_\Theta(p)}$

Once the dominant motion estimation step is completed, the final value of $\omega(p)$ indicates if a point p is likely or not to belong to the part of the image undergoing this dominant motion. Incidentally, points where motion equation is not valid are rejected by the robustness of estimator. In former case $\omega(p)$ is close or equal to one, in later case, $\omega(p)$ is equal or close to zero.

We define support of the dominant motion Sd as set of points p satisfying $\omega(p) \geq \nu$, where ν is a predefined threshold, The pertinent information here is in fact the size of this support.

Indeed, within a given shot, the size n_d of support S_d is supposed to remain nearly constant. On the other hand, if try to estimate a affine motion model between hard cut frames, $I(t)$ and $I(t+1)$, then no coherent estimation can be derived. So n_d is suddenly close to zero. So, by application of simple threshold, it can distinguish between shot boundaries and non-shot boundaries.

4.5   Advantages and Disadvantages

4.5.1   Advantages

Advantages of this method takes care of most of the object motion or camera motion scenario, which are main cause of visual discontinuities. Along with that, the seventh parameter (which is nothing but noise factor in optical constraint equation) takes care of background illumination variations. Due to this, it provides high n_d-values for frames within single shot and low n_d-values for shot boundaries.

4.5.2   Disadvantages

Drawback of this method is its time complexity. It takes around 39 seconds to process two consecutive 320×240 dimensional frames. Also processing time directly proportional to frame dimension. As frame dimension increases processing time increases.

4.6 Results

Results of application of MRME on two of our video sets yields following results:
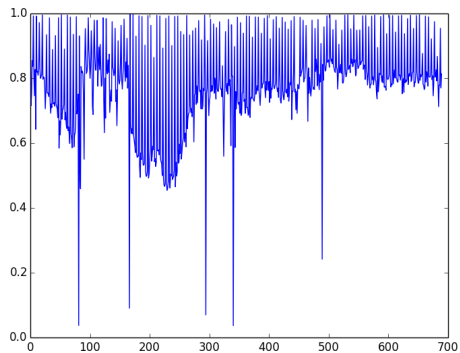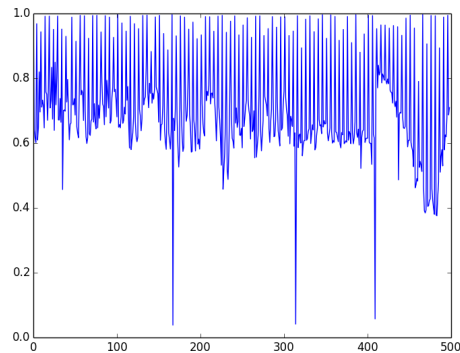
(a) video containing low activity (688 pts)   (b) video containing high activity (498 pts)

Figure 4.1: Normalized n_d value plot of **MRME** algorithm output for video frames

From above results, it is clear that, a threshold value between 0.4 or 0.5 is enough to distinguish between within-shot boundary points and shot-boundary points.

Chapter 5

Proposed Work

5.1 Motivation

We have observed that, if proper features of image frame is chosen then shot boundaries can be modeled as outliers. Here in this case, proper features are those features, which highlights the visual discontinuities between two consecutive frames. So **block-wise means of image difference** is one such features which is suitable in this situation. Since this feature is sensitive to visual discontinuities, generated feature vectors for shot boundary frames are far away from generated vector for within shot frames.

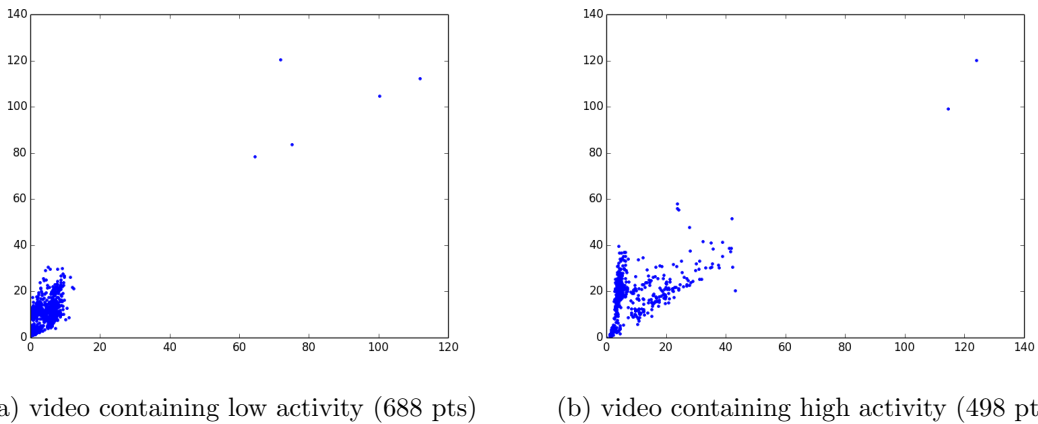Above observation can be visualize using following figure.

(a) video containing low activity (688 pts)   (b) video containing high activity (498 pts)

Figure 5.1: 2-D plot of **'block-wise means of image difference'** feature vector from video frames

From figure 5.1, we observed that, for video containing low activity, outliers in feature space are clearly separable. But in case of high activity video outliers are not clearly separable. So as we increase the number of blocks dividing image, dimensionality of feature space increases. According to, cover's theorem [4], feature are more separable in higher dimension than in a lower dimension. Hence, by increasing number of blocks, our feature space more separable. So outliers become outlier are relatively more separated.

Since our feature is based on image difference based, feature space have following special properties.

1. points representing within-shot frame are much larger in number compared to points representing shot-boundary.

2. points representing within-shot frame forms dense cloud.

3. points representing shot-boundary sparse compared to it's counterpart.

Because of it, we can treat shot boundaries as outliers in feature space and these outliers are not linearly separable. So threshold based outlier detection will not work. Clustering based method is potential option here.

Because of special properties of our feature space, We need a clustering algorithm, which is sensitive to density. DBSCAN is one such algorithm, which suits our requirements. DBSCAN clusters most of within-shot frame points, which are forming dense cloud, into single cluster and rest sparse points are treated as outlier.

As mentioned in Chapter 1, along with shot boundary, there are other two reasons for visual discontinuities. Especially, rapid camera or object motion create very high visual discontinuity in a particular shot. Due to which, feature points, representing such shots, is also away from dense-cloud, and get detected as outlier by DBSCAN. Hence, our detected outlier points contains extra points beside those representing shot boundaries. Such false-positive points can be removed by extra post-processing. As explained in Chapter 4, MRME method is best suitable for this situation.

5.2 Algorithm

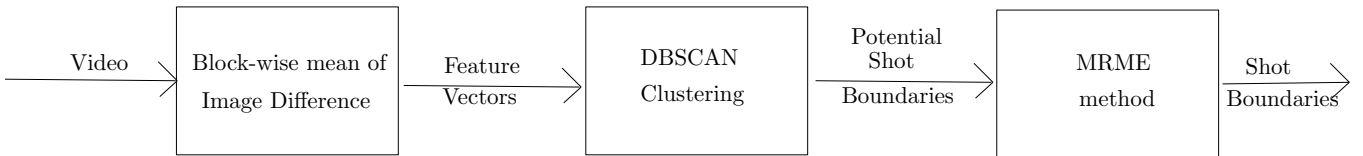Our proposed algorithm can be visualize as following modular diagram:

Figure 5.2: Block diagram of Proposed Algorithm

Here feature vector is n_2-dimensional vector as mentioned in Chapter 2. Potential shot boundaries are nothing but outliers generated by DBSCAN algorithm. Finally, false-positive outlier points, which are not shot boundaries, are detected and removed by MRME algorithm, as mentioned in Chapter 4.

5.2.1 Parameter Modification of MRME

In original MRME algorithm (Chapter 4), parameter C, in equation 4.3 and 4.4, is constant. This parameter C actually represents allowable error in particular iteration. If value of C is varied over iteration, then we get significant improvement in final output. one possible variation, we have implemented is, like for first two or three iteration, value of C is set to a high number, i.e. near 100, then from third iteration onwards value of C is gradually decreased to 10.

Following are improved results of output mentioned in Figure 4.1.
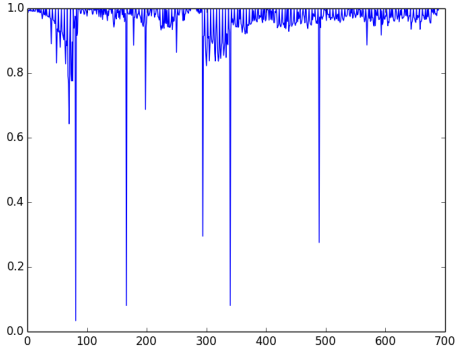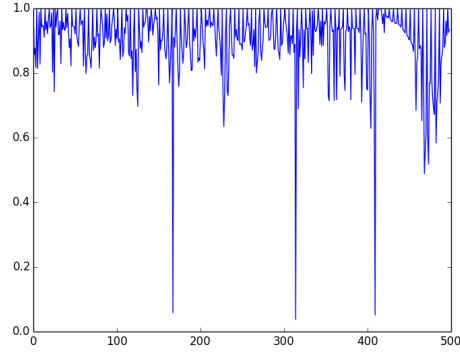
(a) video containing low activity (688 pts)   (b) video containing high activity (498 pts)

Figure 5.3: Improved normalized n_d value plot of **MRME** algorithm output for video frames

5.3   Results

Output of algorithm can be analyzed using following data:

	Data from Proposed Method				
			No. of Shot detected		
Video no.	Length(in frames)	No. of shots	clustering	proposed	MRME
1	498	3	16	3	3
2	689	5	9	5	5
3	348	1	1	1	1
4	382	2	3	2	2
5	1611	4	40	7	7

Table 5.1: Results

	Data from Proposed Method				
			Running Time (in Seconds)		
Video no.	Length(in frames)	No. of shots	clustering	proposed	MRME
1	498	3	17.26	615.66	18625.2
2	689	5	23.84	351.44	25079.6
3	348	1	12.18	46.3	11873.76
4	382	2	13.28	154.54	14783.4
5	1611	4	55.52	1482.72	57480.48

Table 5.2: Run Time Required

5.4   Advantages

Proposed method has one major advantage over direct MRME algorithm, that is **time complexity**. A it can be seen from results that, proposed method require very less time compared to direct MRME method. It provides **accuracy of MRME** method with very less time. Proposed method is applicable to wide variety of videos.

5.5   Conclusion and Future Scope

In this work, we have focused on providing such algorithm to SBD problem, which can provide accuracy as par any motion estimate technique with less time complexity. Proposed method is a combination of two method, one is clustering based (Block-wise mean of image difference + DBSCAN) and other one is motion estimation based (MRME).

Clustering based method have less time complexity and less accuracy. Motion estimation based method have very high time complexity and high accuracy. So proposed method get combined advantages of both methods, i.e. less time complexity and high accuracy.

Limitation of proposed method is that, it is best suitable for videos containing single dominant motion. For more that one dominant motion this method is not providing good results. There are methods available for local motion estimation [11, 7, 2]. So there is scope for improvement for videos containing multiple dominant motions.

Implementation of proposed method is currently not using any parallel computation. Hence, there is scope for improvement in time complexity, if algorithm implemented to make use of parallel computation.

Bibliography

[1] James R. Bergen, P. Anandan, Keith J. Hanna, and Rajesh Hingorani, *Hierarchical model-based motion estimation*, Proceedings of the Second European Conference on Computer Vision (London, UK, UK), ECCV '92, Springer-Verlag, 1992, pp. 237–252.

[2] Michael J. Black and P. Anandan, *The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields*, Comput. Vis. Image Underst. **63** (1996), no. 1, 75–104.

[3] P. Bouthemy, M. Gelgon, and F. Ganansia, *A unified approach to shot change detection and camera motion characterization*, IEEE Trans. Cir. and Sys. for Video Technol. **9** (1999), no. 7, 1030–1044.

[4] T.M. Cover, *Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition*, Electronic Computers, IEEE Transactions on, no. 3, IEEE, June 1965, pp. 326–334.

[5] Norman Richard Draper and Harry Smith, *Applied regression analysis*, Wiley series in probability and mathematical statistics, Wiley, New York [u.a.], 1966.

[6] Martin Ester, Hans peter Kriegel, Jörg S, and Xiaowei Xu, *A density-based algorithm for discovering clusters in large spatial databases with noise*, Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining. Portland, AAAI Press, 1996, pp. 226–231.

[7] Gunnar Farneback, *Two-frame motion estimation based on polynomial expansion*, Proceedings of the 13th Scandinavian Conference on Image Analysis (Berlin, Heidelberg), SCIA'03, Springer-Verlag, 2003, pp. 363–370.

[8] P.J. Huber, J. Wiley, and W. InterScience, *Robust statistics*, Wiley New York, 1981.

[9] Irena Koprinska and Sergio Carrato, *Temporal video segmentation: A survey*, Signal Processing: Image Communication **16** (2001), no. 5, 477 – 500.

[10] Hong Lu and Yap-Peng Tan, *An effective post-refinement method for shot boundary detection*, IEEE Trans. Cir. and Sys. for Video Technol. **15** (2005), no. 11, 1407–1421.

[11] Bruce D. Lucas and Takeo Kanade, *An iterative image registration technique with an application to stereo vision*, Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2 (San Francisco, CA, USA), IJCAI'81, Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[12] J. M. Odobez and P. Bouthemy, *Robust multiresolution estimation of parametric motion models*, Jal of Vis. Comm. and Image Representation (1995).

[13] Jinhui Yuan, Huiyi Wang, Lan Xiao, Wujie Zheng, Jianmin Li, Fuzong Lin, and Bo Zhang, *A formal study of shot boundary detection*, IEEE Trans. Cir. and Sys. for Video Technol. **17** (2007), no. 2, 168–186.