International Journal of Image Processing and Vision Science

Volume 2 | Issue 2 Article 7

October 2013

Shot boundary detection in videos using Graph Cut Sets

Shanmukhappa Angadi
Dept. of Computer science and Engg. Visvesvaraya Technological University Belgaum, INDIA,
vinay_angadi@yahoo.com

Vilas Naik
Dept. of Computer science and Engg. Basaveshwar Engineering College Bagalkot, INDIA,
vilasnaik_h@rediffmail.com

Shot boundary detection in videos using Graph Cut Sets

Shanmukhappa Angadi
Dept. of Computer science and Engg.
Visvesvaraya Technological University
Belgaum, INDIA
vinay_angadi@yahoo.com

Vilas Naik
Dept. of Computer science and Engg.
Basaveshwar Engineering College
Bagalkot, INDIA
vilasnaik_h@rediffmail.com

Abstract— **The *Shot Boundary Detection* (SBD) is an early step for most of the video applications involving understanding, indexing, characterization, or categorization of video. The SBD is temporal video segmentation and it has been an active topic of research in the area of content based video analysis. The research efforts have resulted in a variety of algorithms. The major methods that have been used for shot boundary detection include pixel intensity based, histogram-based, edge-based, and motion vectors based, technique. Recently researchers have attempted use of graph theory based methods for shot boundary detection. The proposed algorithm is one such graph based model and employs graph partition mechanism for detection of shot boundaries. Graph partition model is one of the graph theoretic segmentation algorithms, which offers data clustering by using a graph model. Pair-wise similarities between all data objects are used to construct a weighted graph represented as an *adjacency matrix* (*weighted similarity matrix*) that contains all necessary information for clustering. Representing the data set in the form of an edge-weighted graph converts the data clustering problem into a graph partitioning problem. The algorithm is experimented on sports and movie videos and the results indicate the promising performance.**

Keywords- *Shot Boundary Detection, graph theory, Graph partition model, Graph theoretic Segmentation.*

I. INTRODUCTION

Parsing a video into its basic temporal units -shots- is considered as the initial step in the process of video content analysis. A shot is a series of video frames taken by a single camera, such as, for instance, by zooming into a person or an object, or simply by panning along a landscape. The content is similar in shot regions. The regions where the significant content change occurs are, therefore, called shot boundaries. Since the SBD is a prerequisite step for most of the video applications involving the understanding, parsing, indexing, characterization, or categorization of video, temporal video segmentation has been an active topic of research in the area of content based video analysis. Most common approach to detecting shot boundaries is to search for large discontinuities in the visual content flow of a video. In order to achieve this aim, a continuity (similarity) signal needs to be calculated for the frame sequence to determine the temporal variations of the extracted features. The constructed signal provides us with an idea about how similar the images in the video sequence are. Obviously, the continuity signal constructed by such a way is expected to demonstrate high values within a shot, while it drops off significantly at the transition regions. Recent developments have seen the use of graphs for representing videos for various applications.

Graphs are flexible entities that have the potential for modeling video frames/ shots and need to be investigated for their applicability in Video summarization and selection. In graph theory, Graph clustering is clustering graph vertices on the basis of edge structure. A graph cluster is a connected component comprising of few vertices. It is a maximal clique of a graph with vertices within each cluster and very few edges between clusters. A natural notion of graph clustering is the separation of sparsely connected dense sub graphs from each other resulting in partitioning of graph in to subgraphs.. Firstly, each data element in the data set to be clustered is mapped to a node in the graph. The next, and maybe the most critical, step is to determine the similarity metric. All the data items in the set will be compared in a pair-wise manner with all the other data items according to this similarity metric. Whether two nodes are alike and therefore be in the same cluster is mostly determined by this similarity metrics. In other words, the criteria to determine “good pieces” are the similarity metrics [1].

Following the identification of the similarity metric, a weighted similarity matrix (or affinity matrix) is formed by using the pair-wise similarities between all data items. There is a row and column for each data item in the similarity matrix. The (i,j)th element of the matrix represents the similarity that is calculated for data item *i* and data item *j* based on the similarity metric [2]. All the elements of the similarity matrix is calculated in this manner. The motivation behind graph theoretic clustering algorithms is the idea that weighted similarity matrix contains all the information necessary for clustering [3]. Thus the video frames get represented as a graph in the form of relation(similarity) matrix. since the graph representation is made by using video frames as nodes of graph and similarity between frames as edges connecting nodes, the data clustering problem is transformed into a graph partitioning problem. Therefore, the final step is to obtain an appropriate algorithm to cut the graph into sub-graphs. Figure 1 summarizes the partitioning process. On the top left, graph representation of the problem in the form of an undirected weighted graph is shown with nine nodes and each edge of graph represents similarity value between pair of edges connected by the edge. On the top right is a common visualization of the cost adjacency (similarity) matrix of this graph. Larger similarity values are indicated with lighter color. The partitioning can be performed by means of similarity matrix and with any thresholding operation. If the similarity between two nodes is above a certain threshold it is assumed that two nodes are similar otherwise dissimilar and the edges between such pair of nodes is removed from the graph, as edges 1-5 and 3-7. Deleting of these edges segments graph in to two sub graphs as in lower part of the figure 1. The set of such edges which upon deletion partition the graph in to two subgraphs form a cut set.

The method proposed in this paper works on a weighted undirected graph, where the graphs are established by using the frames in the testing sequence. Each frame in the sequence corresponds to a node in the graph, whereas edge weights between the nodes are calculated by using pair wise similarities of frames.

The proposed algorithm represents the video in the form of an edge-weighted graph and employs a graph partitioning mechanism using cut set. The algorithm partitions the graph in to two subgraphs, each subgraph represents a segment of video belonging to separate shots. Thus methodology identifies boundary between two shots. The algorithm is experimented on sports and movie videos from YOUTUBE and YOUTUBE Action dataset and the results indicates the suitability of the algorithm for shot boundary detection.

The rest of the paper is organized in 4 sections. Section 2 provides an overview of literature. Section 3 presents the proposed algorithm and discussed it in detail. In Section 4 experimentation and results are discussed. Finally, in Section 5 conclusions are presented.

II. RELATED WORK

Video shots are video image sequences produced by a video camera. The transition process between one video sequence and another video sequence is named as video shot transition. Shot boundary detection is the most basic and one of the difficult problems in video summarization, content-based video retrieval. The problem has been studied in depth and numerous technologies have been proposed, for example, the template matching method, color histogram matching method[4], compression domain method[5], edge matching method[6] and color texture moment based method [7]. The researchers have presented diverse approaches for shot boundary detection and comparison of the performance of various approaches are found in[8,9,10].

Shot boundary detection (SBD) can be defined as parsing a video into its basic temporal units called shots [11]. Shot boundary detection (SBD) is the first and essential step for content-based video management and structural analysis. Many efforts to develop SBD algorithms is evident from the literature. Further, the SBD has become necessary step for applications such as video indexing, browsing, retrieval, and representation. All the algorithms developed are capable of parsing by temporal segmentation of video. Although graph theoretic segmentation algorithms are widely used in the fields of computer vision and pattern recognition [12], their applications in SBD area is not widespread. Segmentation via a graph partition model [13] is one of the promising graph theoretic segmentation algorithms, which offers data clustering by using a graph model. The authors in [14] have shown how graph theoretic approaches can yield significantly successful results in SBD area.

Some graph theoretical approaches for shot boundary detection have been presented in literature. A graph based method called *normalized cuts* [15] is presented as an efficient tool for image segmentation problem. The method proposed is a computational technique based on generalized eigenvalue problem in order to obtain optimized graph partitions. Another video shot boundary detection algorithm based on the novel graph theoretic concept, namely dominant sets is presented in [16]. Dominant sets are defined as a set of the nodes in a graph, mostly similar to each other and dissimilar to the others. In order to achieve this goal, candidate shot boundaries are determined by using simply pixel-wise differences between consequent frames. For each candidate position, a testing sequence is constructed by considering 4 frames before the candidate position and 2 frames after the candidate position.

Graph theory is a powerful tool for pattern representation and classification in fields, such as image processing and video analysis. The primary advantage of graph-based representation is that it can represent patterns and relationships among data easily. To take this advantage into video analysis, several studies have proposed the graph-based techniques. As video is a collection of large number of visual patterns(frames) and audio patterns, some of them are nearly similar. The video frames or segmented regions of video frames and spatial relationships among them are expressed as nodes and edges of graph. The graph clustering and graph partitioning algorithms can help in separating and grouping patterns with maximum connectivity and are similar in features.

The work proposed in this paper is motivated by the capacity of graph theoretical techniques to group similar

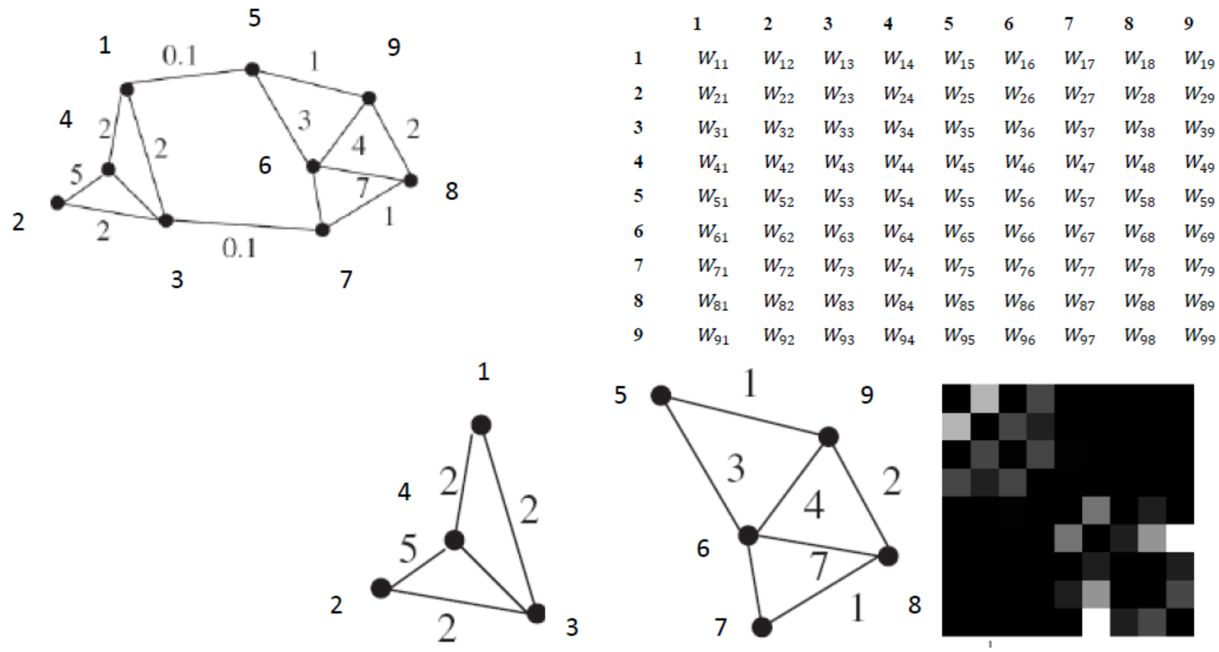


Figure 1 Graph Partitioning process by deleting two edges of lower similarity.

frames in to cluster and mostly these similar frames represent a shot, which will further help in segregating frames belong to separate shot this can help for identifying boundary between shots. However, still some more capabilities of graph theory can be exploited to attain better results in various steps of video processing. In this direction the proposed methodology employs graph cut set determined by empirically defined threshold used to identify edges connecting nodes of less similarity. The graph partitioning is done with help of cut set for separating frames belonging to separate shot there by detecting shot boundary. The complete description of the proposed methodology for shot boundary detection is given in the subsequent section.

III. GRAPH PARTITION MODEL FOR SHOT BOUNDARY DETECTION

Graph theoretic segmentation algorithms are widely used in the fields of computer vision and pattern recognition. Segmentation with graph partition model is one of the graph theoretic algorithms, that facilitates data clustering/segmentation by using a graph model. Pair-wise similarities between all data objects are used to construct a weighted graph which is represented as an adjacency matrix (weight matrix or similarity matrix) as depicted in figure 2. Representing the data set in the form of an edge-weighted graph converts the data clustering problem into a graph partitioning problem.

Given a weighted graph *G* with node set *V,* edge set *E* and weight matrix *W*, the problem is to partition the graph into two sub-graphs *A* and *B* using an objective function. In graph theory, clustering algorithms mainly differ based

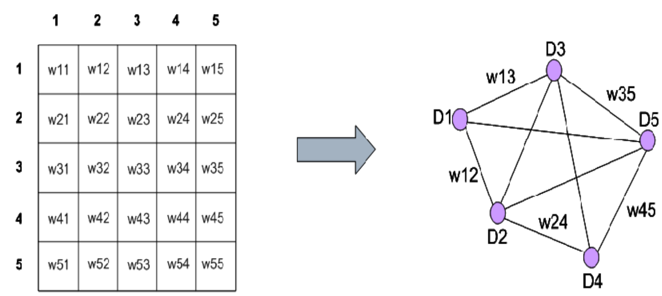


Figure2. Similarity Matrix representation of data and its graph model

on the selected objective function. There are several objective functions mostly used in graph theory such as min-max cut [17], normalized cut, ratio cut and minimum cut. These object functions cluster nodes of higher similarity and similarity among the nodes across the cluster will be low. This clustering will lead to partitioning of graph into sub graphs containing clustered nodes. Literature shows that min-max cut algorithm performs the best among others. Therefore, in this work, min-max cut algorithm is adopted as the objective function. Min-max cut principle aims to maximize the similarity within a cluster and minimize the similarity between clusters. The similarity between nodes *i* and *j* is denoted by w_{ij} [0,1]. The larger the w_{ij} the stronger the connectivity is between the nodes *i* and *j*. The cut divides the graph G into two sub-graphs A and B. The similarity or the association between nodes is their edge weight w_{ij}. Thus the similarity between sub graph A and sub graph B is the cut size as defined in equation

$$\text{cut (A, B)} = w(A, B) \tag{1}$$

$$w(A,B) = \sum_{i \in A, j \in B} w_{ij}$$

The association within sub-graph A which is similarity between nodes within sub graph A is defined as in equation (2).

$$assoc(A) = \sum_{i,j \in A} w_{ij} \qquad (2)$$

The partition principle [19][20] always requires minimized *cut(A, B)* and maximized *assoc (A)* and *assoc (B)*.

A graph can be constructed by treating each sample (i.e. each frame in this case) within a data set as a node and linking an edge between each pair of the nodes. By defining the weight of the edge as the similarity of the samples, clustering can lead to graph partition. From the shot boundary detection point of view, the objective function *Mcut* as in equation 3 should be minimized

$$M_{cut}(A,B) = \frac{cut(A,B)}{assoc(A)} + \frac{cut(A,B)}{assoc(B)} \qquad (3)$$

The minimized value of M_{cut} tries to minimize the association between the two sub-graphs while maximizing the association within each sub-graph [20]. The proposed algorithm for shot boundary detection employing graph partitioning approach is summarized in the algorithm described below.

A. The Proposed Shot Boundary Detection Algorithm

Algorithm

Input: *Frames in group of 10 frames*

$frame_i$ *= 1: 10*

Output*: The algorithm returns two sub graphs indicating presence of shot boundary or null.*

Step 1: *For every* $frame_i$ *ie* $frame_1$ *to* $frame_{10}$

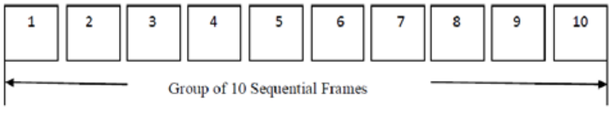


Figure 3. Group of 10 sequential frames

Calculate the Mutual Information between every $frame_i$ *and* $frame_1$ *to* $frame_{10}$ *and represent in matrix form which will be used as adjacency matrix (similarity matrix)*

	f_1	f_2	f_3	f_4	f_5	f_7	f_7	f_8	f_9	f_{10}
f_1	$I(f_1,f_1)$	$I(f_1,f_2)$	$I(f_1,f_3)$	$I(f_1,f_4)$	$I(f_1,f_5)$	$I(f_1,f_6)$	$I(f_1,f_7)$	$I(f_1,f_8)$	$I(f_1,f_9)$	$I(f_1,f_{10})$
f_2	$I(f_2,f_1)$	$I(f_2,f_2)$	$I(f_2,f_3)$	$I(f_2,f_4)$	$I(f_2,f_5)$	$I(f_2,f_6)$	$I(f_2,f_7)$	$I(f_2,f_8)$	$I(f_2,f_9)$	$I(f_2,f_{10})$
f_3	$I(f_3,f_1)$	$I(f_3,f_2)$	$I(f_3,f_3)$	$I(f_3,f_4)$	$I(f_3,f_5)$	$I(f_3,f_6)$	$I(f_3,f_7)$	$I(f_3,f_8)$	$I(f_3,f_9)$	$I(f_3,f_{10})$
f_4	$I(f_4,f_1)$	$I(f_4,f_2)$	$I(f_4,f_3)$	$I(f_4,f_4)$	$I(f_4,f_5)$	$I(f_4,f_6)$	$I(f_4,f_7)$	$I(f_4,f_8)$	$I(f_4,f_9)$	$I(f_4,f_{10})$
f_5	$I(f_5,f_1)$	$I(f_5,f_2)$	$I(f_5,f_3)$	$I(f_5,f_4)$	$I(f_5,f_5)$	$I(f_5,f_6)$	$I(f_5,f_7)$	$I(f_5,f_8)$	$I(f_5,f_9)$	$I(f_5,f_{10})$
f_6	$I(f_6,f_1)$	$I(f_6,f_2)$	$I(f_6,f_3)$	$I(f_6,f_4)$	$I(f_6,f_5)$	$I(f_6,f_6)$	$I(f_6,f_7)$	$I(f_6,f_8)$	$I(f_6,f_9)$	$I(f_6,f_{10})$
f_7	$I(f_7,f_1)$	$I(f_7,f_2)$	$I(f_7,f_3)$	$I(f_7,f_4)$	$I(f_7,f_5)$	$I(f_7,f_6)$	$I(f_7,f_7)$	$I(f_7,f_8)$	$I(f_7,f_9)$	$I(f_7,f_{10})$
f_8	$I(f_8,f_1)$	$I(f_8,f_2)$	$I(f_8,f_3)$	$I(f_8,f_4)$	$I(f_8,f_5)$	$I(f_8,f_6)$	$I(f_8,f_7)$	$I(f_8,f_8)$	$I(f_8,f_9)$	$I(f_8,f_{10})$
f_9	$I(f_9,f_1)$	$I(f_9,f_2)$	$I(f_9,f_3)$	$I(f_9,f_4)$	$I(f_9,f_5)$	$I(f_9,f_6)$	$I(f_9,f_7)$	$I(f_9,f_8)$	$I(f_9,f_9)$	$I(f_9,f_{10})$
f_{10}	$I(f_{10},f_1)$	$I(f_{10},f_2)$	$I(f_{10},f_3)$	$I(f_{10},f_4)$	$I(f_{10},f_5)$	$I(f_{10},f_6)$	$I(f_{10},f_7)$	$I(f_{10},f_8)$	$I(f_{10},f_9)$	$I(f_{10},f_{10})$

The mutual information between every pair of frames is calculated as shown

$$I_{k,l}^{R} = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} C_{k,l}^{R}(i,j) \log \frac{C_{k,l}^{R}(i,j)}{C_{k}^{R}(i) C_{l}^{R}(j)}.$$

$$I_{k,l=}^{G} \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} C_{k,l}^{G}\ (i,j)\ log\ \frac{C_{k,l}^{G}\ (i,j)}{C_{k,}^{G}\ (i)\ C_{l}^{G}\ (j)}$$

$$I_{k,l=}^{B} \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} C_{k,l(i,j)}^{B}\ log\ \frac{C_{k,l(i,j)}^{B}}{C_{k,(i)}^{B}\ C_{l(j)}^{B}}$$

The total mutual information (MI) calculated between frames f_k *and* f_l *is defined as*

$$I(f_k.\ f_l) = I_{k,l}^{R} + I_{k,l}^{G} + I_{k,l}^{B}.$$

the mutual information between every pair of frames is equivalent to similarity between those frames

Step 3: *After weight matrix /adjacency matrix/ similarity matrix is computed give a graph model Observe every row for mutual information values between pair of frames.*

If for $frame_k$*: k : 1 to 10*

I($\boldsymbol{frame_k}$ ***,*** $\boldsymbol{frame_{l+1}}$***) ≤ 0.25 I(***$\boldsymbol{frame_k}$ ***,*** $\boldsymbol{frame_l}$***)***

0.25 is empirically selected threshold.

Then *There exists a cut /shot boundary.*

Else *There will not be a shot boundary*

Step 4: *Delete the edges of graph connecting pair of nodes satisfying the condition given in step3. As in the figure 4 if all the edges that connect frames i-3, i-2,i-1 and i to frames i+1, i+2 till the last frame in the group are of mutual information value below threshold then shot boundary/shot cut exists between frame i and i+1 . Set of these edges is "cut set"upon deletion they partion graph in to two graphs with nodes as frames from opposite side of boundary of shot.*

The graph representation of example shown in figure 4(a) is as in the figure 4(b), in which all the frames of video under consideration are represented as nodes of graph. The edges connecting each pair of nodes have weight equivalent to mutual information between pair of nodes. A set of edges connecting $frame_{i+1}$ and $frame_{i+2}$ to remaining frames (as shown in figure 4(a)) exhibit low mutual information (low similarity value). If these edges are deleted the graph gets partitioned in to two sub graphs as shown in figure 4(c). Each sub graph now represents frames from a separate shot. The set of edges deleted forms a *cut set*.

The construction of weight matrix using mutual information for graph representation of video is described in next section.

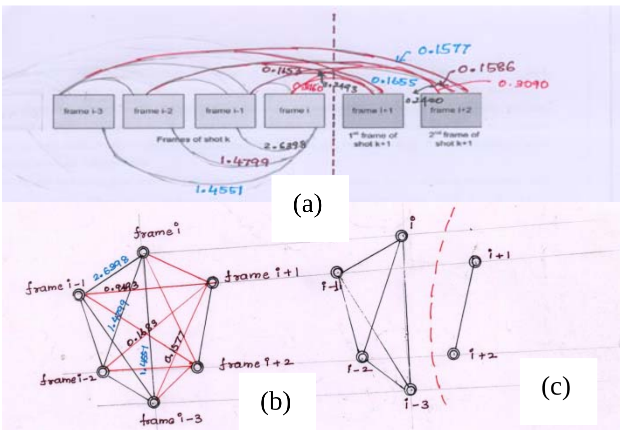


Figure 4, Graph representation for the position of partition

B. Mutual Information for construction of weight matrix of Graph representation

One can see the problem of shot boundary detection as a problem of graph partitioning. The video frames can be represented as nodes in a graph whose edge weights correspond to the pairwise similarities of the frames. In order to detect the video shots, one has to discover and disconnect the weak connections between the nodes, thus partitioning the graph to subgraphs ideally corresponding to the shots. As a measure of similarity between two frames mutual information (MI) is employed because it exploits the inter-frame information flow in a more compact way than frame subtraction. Difference in content between two frames, leads to low values of mutual information.

In proposed solution, the mutual information between two frames is calculated separately for each of the RGB color components. In the case of the *R* component, the element $C^R_{t,t+1}(i,j)$, $0 \le i,\ j \le N-1$ (*N* being the number of intensity levels in the image), corresponds to the probability that a pixel with color intensity *i* in frame f_t has color intensity *j* in frame f_{t+1}. In other words, $C^R_{t,t+1}(i,j)$equals to the number of pixels which change from color intensity *i* in frame f_t to color intensity *j* in frame f_{t+1}, divided by the total number of pixels in the video frame. The mutual information $I^R_{k,l}$ of frames f_k, f_l for the *R* component is expressed as equation 4

$$I^R_{k,l} = -\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \mathbf{C}^R_{k,l}(i,j)\log\frac{\mathbf{C}^R_{k,l}(i,j)}{\mathbf{C}^R_k(i)\mathbf{C}^R_l(j)}. \qquad (4)$$

Where

$C^R_{k,l}$ *(i. j): Joint Probability that a pixel with intensity level i in frame* f_k *has intensity level j in frame* f_l *for R Component*

The mutual information $I^R_{k,l}$ is the relative entropy between the joint distribution $C^R_{k,l}$ *(i. j):* and the product distribution $\mathrm{C}^R_{k,}\ (i)\ \mathrm{C}^R_l\ (j)$.

Similarly the mutual information for Blue (I^B) Green (I^G) components can be found as

$$I^G_{k,l}=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \mathrm{C}^G_{k,l}\ (i,j)\ log\ \frac{\mathrm{C}^G_{k,l}\ (i,j)}{\mathrm{C}^G_{k,}\ (i)\ \mathrm{C}^G_l\ (j)}$$

$$I^B_{k,l}=\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \mathrm{C}^B_{k,l(i,j)}\ log\ \frac{\mathrm{C}^B_{k,l(i,j)}}{\mathrm{C}^B_{k,(i)}\ \mathrm{C}^B_{l(j)}}$$

The total mutual information (MI) calculated between frames f_k and f_l is defined as in (5)

$$I(f_k, f_l) = I^R_{k,l} \mid I^G_{k,l} \mid I^B_{k,l}. \qquad (5)$$

Since the aim is to cluster the sequence into shots, It is not necessary to calculate the mutual information between all pairs of frames in a video sequence, because relations between frames, which are far apart are not important for the shot cut detection task. Thus, the method uses only mutual information calculated between frames in a group of 10 temporally consecutive frames.

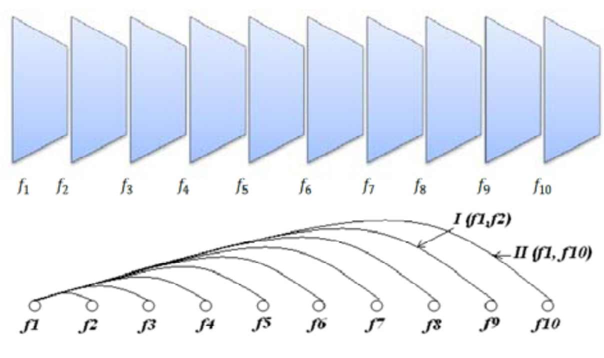


Figure 5 Calculating Mutual information between a frame and all other frames from group of 10 frames

The figure 5 above depicts the computation of mutual information values between pair of frames and using the equations described above. The weight matrix W for group of 10 frames for the graph representation can be given as in figure 6.

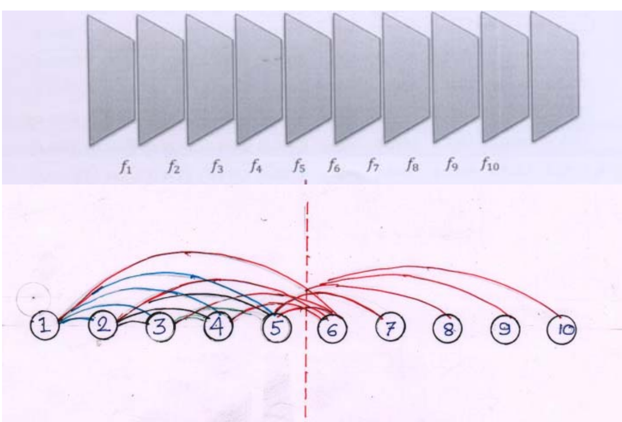
	f_1	f_2	f_3	f_4	f_5	f_7	f_7	f_8	f_9	f_{10}
f_1	$I(f_1,f_1)$	$I(f_1,f_2)$	$I(f_1,f_3)$	$I(f_1,f_4)$	$I(f_1,f_5)$	$I(f_1,f_6)$	$I(f_1,f_7)$	$I(f_1,f_8)$	$I(f_1,f_9)$	$I(f_1,f_{10})$
f_2	$I(f_2,f_1)$	$I(f_2,f_2)$	$I(f_2,f_3)$	$I(f_2,f_4)$	$I(f_2,f_5)$	$I(f_2,f_6)$	$I(f_2,f_7)$	$I(f_2,f_8)$	$I(f_2,f_9)$	$I(f_2,f_{10})$
f_3	$I(f_3,f_1)$	$I(f_3,f_2)$	$I(f_3,f_3)$	$I(f_3,f_4)$	$I(f_3,f_5)$	$I(f_3,f_6)$	$I(f_3,f_7)$	$I(f_3,f_8)$	$I(f_3,f_9)$	$I(f_3,f_{10})$
f_4	$I(f_4,f_1)$	$I(f_4,f_2)$	$I(f_4,f_3)$	$I(f_4,f_4)$	$I(f_4,f_5)$	$I(f_4,f_6)$	$I(f_4,f_7)$	$I(f_4,f_8)$	$I(f_4,f_9)$	$I(f_4,f_{10})$
f_5	$I(f_5,f_1)$	$I(f_5,f_2)$	$I(f_5,f_3)$	$I(f_5,f_4)$	$I(f_5,f_5)$	$I(f_5,f_6)$	$I(f_5,f_7)$	$I(f_5,f_8)$	$I(f_5,f_9)$	$I(f_5,f_{10})$
f_6	$I(f_6,f_1)$	$I(f_6,f_2)$	$I(f_6,f_3)$	$I(f_6,f_4)$	$I(f_6,f_5)$	$I(f_6,f_6)$	$I(f_6,f_7)$	$I(f_6,f_8)$	$I(f_6,f_9)$	$I(f_6,f_{10})$
f_7	$I(f_7,f_1)$	$I(f_7,f_2)$	$I(f_7,f_3)$	$I(f_7,f_4)$	$I(f_7,f_5)$	$I(f_7,f_6)$	$I(f_7,f_7)$	$I(f_7,f_8)$	$I(f_7,f_9)$	$I(f_7,f_{10})$
f_8	$I(f_8,f_1)$	$I(f_8,f_2)$	$I(f_8,f_3)$	$I(f_8,f_4)$	$I(f_8,f_5)$	$I(f_8,f_6)$	$I(f_8,f_7)$	$I(f_8,f_8)$	$I(f_8,f_9)$	$I(f_8,f_{10})$
f_9	$I(f_9,f_1)$	$I(f_9,f_2)$	$I(f_9,f_3)$	$I(f_9,f_4)$	$I(f_9,f_5)$	$I(f_9,f_6)$	$I(f_9,f_7)$	$I(f_9,f_8)$	$I(f_9,f_9)$	$I(f_9,f_{10})$
f_{10}	$I(f_{10},f_1)$	$I(f_{10},f_2)$	$I(f_{10},f_3)$	$I(f_{10},f_4)$	$I(f_{10},f_5)$	$I(f_{10},f_6)$	$I(f_{10},f_7)$	$I(f_{10},f_8)$	$I(f_{10},f_9)$	$I(f_{10},f_{10})$

Figure 6. Weight matrix for the graph representation of group of 10 frames

The mechanism using weight matrix for shot boundary detection is presented in ensuing section C.

C. Shot boundary detection

Given a weighted graph G with node set V, edge set E and weight matrix W representing 10 frames of video , the problem of shot boundary detection is to partition the graph into two sub-graphs A and B .The graph cut principle aims to minimize the similarity between clusters and maximize the similarity within a cluster. The similarity between nodes i and j is denoted by w_{ij} $I(f_i, f_j)$. The larger the w_{ij}, the stronger the connectivity is between the nodes i and j and smaller the w_{ij} value weaker is the connectivity between the nodes. The proposed work will detect the shot boundary based on connectivity between frames on both sides of boundary by calculating edge weight values by representing frames as nodes of graph and mutual information between them as edge weight. The method identifies shot boundary by looking at variation in connectivity between frames. If the weight of edge connecting node i and node j falls below the 25% of the edge weight connecting node i and node j-1 then a boundary exists between node j-1 and node j. This is illustrated in figure 7 below.



Node 1	*Node 2*	*Node 3*	*Node 4*
$I_{16} = 0.0870$	$I_{26} = 0.1655$	$I_{36} = 0.1653$	$I_{46} = 0.2493$
$I_{15} = 0.9690$	$I_{25} = 1.4551$	$I_{35} = 1.4779$	$I_{45} = 2.6398$
$I_{16} < 0.25 I_{15}$	$I_{26} < 0.25 I_{25}$	$I_{36} < 0.25 I_{35}$	$I_{46} < 0.25 I_{45}$

Figure 7. Shot boundary detection between frame 5 and frame 6. Shot boundary exists between f_5 and f_6 if I(1,6) <= 0.25 I(1,5).

IV. Experimentation and Results

The algorithm is implemented in Matlab and experiments are performed on a Core2 Duo 2.40 GHz Windows machine with 2GB of RAM. The experiments are conducted in order to validate the effectiveness of the proposed shot boundary detection algorithm and results are presented in the following. The performance of the proposed algorithm is evaluated on the YOU TUBE sports videos, Soccer clips and Open Videos data set. The performance of the proposed algorithm is evaluated using standard evaluation metrics.

In the proposed work first Mutual Information (MI) values between every frame and rest of the frames in group of 10 consecutive frames are computed as explained in the section B. A weight matrix is constructed for graph representation of video. The frames are represented as nodes of graph and edge weight is represented by Mutual information between two frames connected by that edge. A weight matrix is presented in figure 8.

8.4734	2.1200	2.1184	1.2109	0.9690	0.0870	0.0867	0.0854	0.0857	0.0911
2.1200	8.6683	7.5163	1.8915	1.4551	0.1655	0.1577	0.1534	0.1508	0.1473
2.1184	7.5163	8.6909	1.9228	1.4799	0.1653	0.1586	0.1539	0.1541	0.1474
1.2109	1.8915	1.9228	8.8815	2.6398	0.2493	0.2420	0.2375	0.2402	0.2292
0.9690	1.4551	1.4799	2.6398	9.1199	0.3160	0.3090	0.3094	0.3087	0.3023
0.0870	0.1655	0.1653	0.2493	0.3160	7.9925	4.9240	3.9573	3.6393	3.3291
0.0867	0.1577	0.1586	0.2420	0.3090	4.9240	8.0209	4.3507	3.9991	3.5407
0.0854	0.1534	0.1539	0.2375	0.3094	3.9573	4.3507	8.0669	5.3453	4.1995
0.0857	0.1508	0.1541	0.2402	0.3087	3.6393	3.9991	5.3453	8.0680	4.7657
0.0911	0.1473	0.1474	0.2292	0.3023	3.3291	3.5407	4.1995	4.7657	8.0971

Figure 8. Cost Adjacency (A) matrix representation of graph

The graph representation and partition process is shown in the figure 9. The values encircled in the cost adjacency matrix stand for lower edge weight values as per step 3 of proposed algorithm in section 3.The edges which are of lower weight value indicate lower similarity between nodes connected by them, In figure 9 the edges with low weight value are shown in red line, these edges upon deletion partition the graph into two subgraphs as in figure 9(b). Set of these edges become *Cut set.* The nodes in each sub graph represent frames belonging to two separate shots as shown in figure 4(a). From the graph model representation the shot boundaries are detected as explained in the section 3.

The performance of the proposed model is evaluated using precision and recall as evaluation metrics. The precision measure is defined as the ratio of number of correctly detected cuts to the sum of correctly detected and falsely detected cuts of a video data and recall is defined as the ratio of number of detected cuts to the sum of detected and undetected cuts as indicated in equation(5). These parameters were obtained for the proposed model on eight different video samples.

$$\left.\begin{aligned} Recall &= \frac{Number\ of\ correctly\ detected\ boundaries}{Number\ of\ True\ boundaries} \\ Precision &= \frac{Number\ of\ correctly\ detected\ boundaries}{Number\ of\ totally\ detected\ boundaries} \end{aligned}\right\} \quad (5)$$

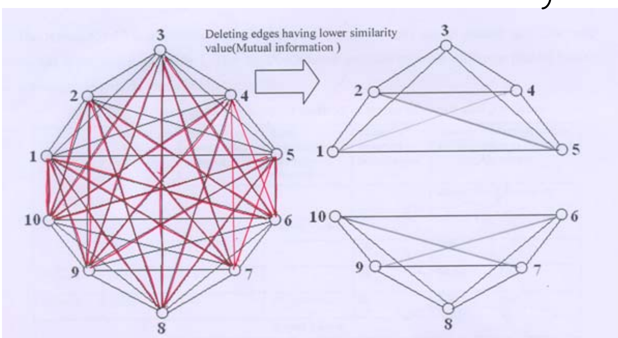


Figure 9. (a) Graph representation (b) Graph partitioned in to two sub graphs after deleting edges of lower similarity values shown in similarity matrix in Figure 8.

The results for 10 test videos randomly selected from You tube action dataset and You tube movies is presented in Table 1. The shot boundaries detected between different pair of frames in example videos is presented in figure 10.

The results obtained over videos of two genres namely movie and sports videos reveal the suitability of proposed algorithm for shot boundary detection in two types of videos. The shot boundary detection(SBD) in sports video is challenging [18] as compared to SBD in movie videos but still the proposed algorithm is capable of attaining higher performance in both the genres. The performance of proposed algorithm is found to be stable even with large size videos.

V. Conclusion

In this work, a graph modeling based shot boundary detection algorithm is described. The algorithm is tested with YOUTUBE Action data set and selected movie videos and performance is evaluated with recall and precision metrics. The algorithm proposed, models the group of every 10 frames as a undirected graph with help of weighted adjacency matrix. The similarity between two nodes of graph represents the edge connecting nodes where nodes are frames of the video. The similarity measure between pair of nodes is represented by mutual information. The shot boundary is detected by identifying set of edges with low mutual information value that upon deletion partition the graph in two sub graphs. The two subgraphs symbolize frames belong to two separate shots. The average values of performance parameters namely recall up to 95.86% and precision up to 96% for the proposed SBD algorithm indicates that the proposed mechanism is suitable for shot boundary detection from movie and sports videos and is a promising tool for consideration and experimentation with other types of videos.

Table 1.0 Performance of proposed method in terms of recall and precision

	Number of frames tested	Number of Boundaries present	Total boundaries correctly detected	False detected boundaries	Shot boundary detection performance of Proposed Algorithm	
					Recall %	Precision
Movie Videos						
Video 1	2000	12	12	00	100	100
Video 2	2000	18	17	00	94.44	100
Video 3	2000	10	10	00	100	100
Sports Videos						
Video 4	2000	15	14	00	93.33	100
Video 5	2000	16	15	01	93.75	93.75
Video 6	2000	14	13	01	92.85	92.85
Video 7	2000	08	08	00	100	100
Video 8	2000	17	16	01	94.11	94.11
Large Size Videos						
Video 9	4500	30	28	02	93.33	93.33
Videos 10	4500	32	31	03	96.85	91.11

VI. References

[1] D. A. F. J. Ponce, **Computer Vision: A Modern Approach:** Prentice Hall, 2002.

[2] J. Yuan, H. Wang, L. Xiao *et al.*, **"A Formal Study of Shot Boundary Detection,"** *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 168-186, 2007.

[3] C. Ding, X. He, H. Zha *et al.*, "**A min-max cut algorithm for graph partitioning and data clustering,"** *Proc. IEEE International Conference on Data Mining*, 2001.

[4] Hua Zhang Ruimin Hu Lin Song,2011 "**A shot boundary detection method based on color feature"** in International Conference on Computer Science and Network Technology (ICCSNT)

[5] Chattopadhyay, T. 2011 "**Video shot boundary detection using compressed domain features of H.264"** in 11th International Conference on Intelligent Systems Design and Applications (ISDA).

[6] Don Adjeroh, M. C. Lee, N. Banda, Uma Kandaswamy 2009 " **Adaptive Edge-Oriented Shot Boundary Detection" in** EURASIP Journal on Image and Video Processing Volume 2009.

[7] B. H. Shekar, M. Sharmila Kumari, Raghuram Holla, 2011**"Shot Boundary Detection Algorithm Based on Color Texture Moments" chapter in** Communications in Computer and Information Science Volume 142, 2011, pp 591-594

[8] Yuan J., Wang, H., and Xiao, L. (2007). **"A formal study of shot boundary detection".** IEEE Trans. Circuits Syst. VideoTechnol, Vol. 17, No. 2, pp.168 186.

[9] Li, S., Lee, M. (2007). **"Effective detection of various wipe transitions".** IEEE Trans. Circuits Syst. Video Technol, Vol.17, No. 6, pp.663 673.

[10] Grana, C., Cucchiara, R.(2007). **"Linear transition detection as a unified shot detection approach".** IEEE

Trans. Circuits Syst. Video Techno, Vol.17, No. 4, pp.483 489.

[11] A. Hanjalic, **“Shot Boundary Detection: Unraveled and Resolved?,”** *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 12, no. 2, pp. 90-105, February 2002.

[12] U. Sakarya, and Z. Telatar, **“Graph partition based scene boundary detection,”** in *Proc. The 5th International Symposium on Image and Signal Processing and Analysis (ISPA 2007)*, Istanbul, Turkey, 2007, pp. 544-549.

[13] J. Yuan, B. Zhang, and F. Lin, “**Graph partition model for robust temporal data segmentation,**” *Proc. of PAKDD*, pp. 539-542, 2005.

[14] Ufuk Sakarya · Ziya Telatar, 2008 **“Graph-based multilevel temporal video segmentation” in Springer Journal on** Multimedia Systems 14:277–290, DOI 10.1007/s00530-008-0145-x.

[15] Shi J, Malik, J. 2000**, “Normalized cuts and image segmentation”**. in IEEE Transaction on . Pattern Analysis. Machine . Intelligence. **22**, 888–905 (2000).

[16] Asan, E. “**Video shot boundary detection by graph-theoretic dominant sets approach” in** 24th International Symposium on Computer and Information Sciences, 2009. ISCIS 2009.

[17] C. Ding, X. He, H. Zha *et al.*, “**A min-max cut algorithm for graph partitioning and data clustering,”** *Proc. IEEE International Conference on Data Mining*, 2001.

[18] *Matko Šarić, Hrvoje Dujmić, Domagoj Baričević 2008*,**“Shot Boundary Detection in Soccer Video using Twin-comparison Algorithm and Dominant Color Region”** in Journal of Information and Organisational sciencesVol 32, No 1 (2008)

[19] Vincent D. Blondel,Anah Gajardo,Maureen Heymans,Pierre Senellart,Paul Van Dooren 2004 **“A Measure of Similarity between Graph Vertices: Applications to Synonym Extraction and Web Searching”** SIAM REVIEW 2004 Society for Industrial and Applied Mathematics Vol. 46, No. 4, pp. 647–666.

[20] Chris H.Q. Din, Xiaofeng He, Hongyuan Zhab, Ming Gu, Horst D. Simon 2001 **“A Min-max Cut Algorithm for Graph Partitioning and Data Clustering” in proceedings of International** on Data Mining. ICDM 2001.

Figure 10. Shot boundaries detected between frames 125 and 126, 457and 458, 242 and 243, 551 and 552