Keyframe-based Video Summarization using Delaunay Clustering

PADMAVATHI MUNDUR, YONG RAO, YELENA YESHA

Department of Computer Science and Electrical Engineering

University of Maryland Baltimore County

1000 Hilltop Circle

Baltimore, MD 21250

{pmundur, yongrao1, yeyesha}@cs.umbc.edu

Abstract

Recent advances in technology have made tremendous amount of multimedia information available to the general population. An efficient way of dealing with this new development is to develop browsing tools that distill multimedia data as information oriented summaries. Such an approach will not only suit resource poor environments such as wireless and mobile, but also enhance browsing on the wired side for applications like digital libraries and repositories. Automatic summarization and indexing techniques will give users an opportunity to browse and select the multimedia document of their choice for complete viewing later. In this paper, we present a technique by which we can automatically gather the frames of interest in a video for purposes of summarization. Our proposed technique is based on using Delaunay Triangulation for clustering the frames in videos. We represent the frame contents as multi-dimensional point data and use Delaunay Triangulation for clustering them. We propose a novel video summarization technique by using Delaunay clusters that generates good quality summaries with fewer frames and less redundancy when compared to other schemes. In contrast to many of the other clustering techniques, the Delaunay clustering algorithm is fully automatic with no user specified parameters and is well suited for batch processing. We demonstrate these and other desirable properties of the proposed algorithm by testing it on a collection of videos from Open Video Project. We provide a meaningful comparison between results of the proposed summarization technique with Open Video storyboard and K-means clustering. We evaluate the results in terms of metrics that measure the content representational value of the proposed technique.

Keywords. Video Summarization, Delaunay Triangulation

1.0 Introduction

There has been a tremendous growth in the multimedia information available on the Web and elsewhere in the last five years. For instance, streaming video applications

have proliferated in the recent years spurred by the phenomenal increase, about 58% in the last year or two in home-based Internet users [1]. We have also seen a rise in the number of low bandwidth technologies such as wireless and mobile that are typically resource poor. Together these developments indicate the need for technologies that sift through vast amounts of multimedia information to facilitate full content selection based on previews. A browsing facility that provides an information oriented summary for selection of actual content is a necessity. This is perhaps more relevant in the context of digital libraries and repositories than many other applications. In this paper, we propose an *automatic* clustering method using *Delaunay Triangulation* for a video summarization application which can then be incorporated into a browsing tool for navigating a large scale video collection.

In [2], a panel of experts concluded that the core research in video content analysis must focus on developing techniques for automatic processing of videos to extract information. For an application like video summarization, we *gather* frames of interest based on one or more video features which are then used to extract and summarize information content in the videos. The result of generating such video summaries can range from just a collection of keyframes representing the essence of a video to generating a video clip summarizing the essential content of the video with temporal order in tact.

At one end of the spectrum, we have video summaries that can be *played* using browsing tools. The simplest of the techniques for this type of temporal video summary is variable speed fast forwarding using time compression; that is to play the full video at a higher speed but still be intelligible to the viewer about the video contents. A second popular technique for generating temporal video summary is the video skimming technique also called gisting popularized by the CMU's Informedia project [3] where video summaries are generated by incorporating both audio and video information from the source video and played using a browsing tool. At the other end of the spectrum are video summaries that are viewed statically rather than played. This type of summary generation is enabled by using keyframe-based approach. The summaries vary from a static pictorial overview of the video content to a collection that maintains temporal order and therefore, conveys time sequenced content. A shot detection based keyframe selection will yield a summary that

maintains temporal order but at the cost of increased redundancy. A clustering approach to video summarization results in a set of keyframes which may not preserve temporal order but eliminates redundancy. The significant challenge for keyframe-based video summarization is to devise methods for selecting keyframes that represent the essence of the video content and not just form a random collection of frames.

In this paper, we present a keyframe-based video summarization technique using clustering. The technique we propose for clustering video frames is based on Delaunay Triangulation (DT) which has been used in other domains such as data mining and is widely acknowledged to be fully *automatic*. We demonstrate that and other desirable properties of this clustering technique in our proposed video summarization algorithm. For instance, the proposed algorithm based on DT results in clusters of different sizes depending on the content distribution within a video. We define a metric called the *significance factor* for each keyframe based on the size of the cluster it comes from. We also define a metric called the *Compression factor* for each video which indicates the reduction in size with the summarized content compared to the number of original frames. A third metric called the *overlap factor* quantifies the content overlap between our summary and the Open Video storyboard to provide a meaningful comparison. Significance metric along with the overlap factor helps us evaluate the representational value of the summaries generated by the proposed method. Like most other keyframe-based approaches, the proposed algorithm does not preserve temporal order. On the other hand, the importance of content can be quickly conveyed to the user by displaying the keyframes in order of cluster significance. The proposed algorithm generates summaries that are devoid of redundancy.

The main contribution of this paper is twofold:

1. We develop an automatic video summarization technique based on Delaunay clustering that is well suited for batch processing of videos without user intervention. Each cluster shows very good clustering performance in terms of similarity of content. The clustering process results in clusters of different sizes based on the content represented in the original video. We are able to convey the content with fewer frames from the larger and more significant

clusters. This performance is consistently noted for the 50 videos in our test collection.

2. We develop an evaluation methodology that is objective in nature and provides a way of generating a meaningful comparison with the Open Video's storyboard. We evaluate the proposed algorithm using 50 videos from the Open Video Project and compare our results with the publicly available Open Video storyboard for each of those videos. The comparison with K-means clustering again demonstrates the automatic nature of the proposed technique.

The rest of the paper is organized as follows. In Section 2, we discuss related work. Delaunay clustering is presented in Section 3. In Section 4, we present our video summarization technique. We discuss summarization results in Section 5 and compare them to the results from Open Video Project's storyboard and K-means clustering. We conclude the paper in Section 6.

2.0 Related Work

Clustering analysis has been studied in other areas like data mining, machine learning, and statistics. Different clustering methods have been proposed with different capabilities and different computational requirements. Nevertheless, most clustering methods share their need for user-specified arguments and prior knowledge to produce their best results. For example, the density threshold parameter in [4] needs to be specified prior to the clustering process and the result depends on this predefined value. In [5], the total number of clusters N is required to be predefined, although they derived a formula for N. In [6], the authors use the content value computed from the most static cluster as the threshold to cluster the rest of the frames. This type of parameter tuning is expensive and inefficient for huge data sets because it demands pre-processing and/or several trial and error steps. More recently, spectral clustering [7], a clustering algorithm developed in the machine learning community has been applied to visual data clustering. As the authors in [8] show, spectral clustering can work on non-metric space and does not assume that the data in each cluster has convex distribution; it is free of singularity problem caused by high dimensionality of feature vectors. These properties make the spectral clustering algorithm a favorable choice for visual data clustering, since visual features are often high dimensional and

the distribution of each cluster is not necessarily convex or a Gaussian function. However, to find the *k* largest eigenvectors in the spectral clustering algorithm is an approximate solution for bi-partitioning the graph with the normalized cuts principle, whose exact solution is NP hard as shown in [8]. The time complexity of the implemented algorithm depends on the complexity of the eigenvalue decomposition algorithm which requires $O(n^3)$, where *n* is the number of items in the dataset.

In our system, the clustering process is based on Delaunay Triangulation [9-12]. It is acknowledged in spatial data mining and geographical database domains where DT has been used for clustering purposes that it is fully argument free and avoids the impact of noise. Similar to spectral clustering, it is realized by partitioning the graph into disjoint sub-graphs and can handle various data distributions including non-convex cluster shape. However, DT succinctly captures the spatial proximity relationships among data points and is computationally more efficient as it only requires $O(n \log n)$ time, where *n* is the size of the dataset. We also integrate Principal Component Analysis (PCA) [13] into the DT clustering algorithm to handle the high dimension problem. Because of its automatic nature, we are able to construct an automatic video processing architecture around it. In this paper, we demonstrate that aspect of DT by batch processing all our test video files without user participation at any point.

A video summarization is a compact representation of a video sequence and is useful for various video applications such as video browsing and retrieval systems. A video summarization can be a preview sequence which is the concatenation of a limited number of selected video segments (video highlights) or can be a collection of keyframes which is a set of suitably chosen frames of a video. Although keyframe-based video summarization may lose the spatio-temporal properties and audio content in the original video sequence, it is the simplest and the most common method. When temporal order is maintained in selecting the keyframes, users can locate specific video segments of interest by choosing a particular keyframe using a browsing tool. Keyframes are also effective in representing visual content of a video sequence for retrieval purposes: video indexes may be constructed based on visual features of keyframes, and queries may be directed at keyframes using image retrieval techniques

[5,14]. In the next paragraph, we present details of research work in the area of keyframe-based video summarization.

Earlier approaches in keyframe-based video summarization mainly relied on shot boundary detection; after shots are detected, the keyframe is selected as the first frame appearing after each detected shot boundary [15], or the first and last frame in each shot [16]. But since shots may be of different types (cut, dissolve, wipe), the selected key frame may not adequately represent the video content. Recent approaches use graph theory [17], curve splitting [18] and clustering [4,5,6,8]. In [17], each frame is represented as a point in a high dimensional feature space; each shot is viewed as a proximity graph and each frame is a vertex in the graph. In this way, keyframe selection is equivalent to finding a cover of vertices that minimizes the total feature distance between the vertices and their neighboring points. This is an NP-Complete vertex cover problem. The authors use a greedy approach to get an approximate solution. In [18], a video sequence is represented as a curve in a high dimensional feature space. The authors use a multi-dimensional curve splitting algorithm to get a linearized curve characterized by "perceptually significant" points, which are connected by straight lines. A keyframe set is obtained by collecting frames found at those perceptually significant points. However, there is no comprehensive user study to prove that there is a clear connection between "perceptually significant" points and most memorable keyframes (highlights); the keyframes may not capture all important instances of a video. The authors in [4] propose an unsupervised clustering algorithm to group the frames into clusters based on color histogram feature. The frame which is closest to a cluster centroid is selected as keyframe. The number of clusters is controlled by a predefined density threshold value which can not guarantee optimal result. In [5], Hanjalic & Zhang use a partitioned clustering algorithm with cluster-validity analysis to select the optimal number of clusters for shots. The keyframe set represents frames that are closest to each cluster centroid. The resulting keyframe set is said to be optimal in terms of intra and inter cluster distance measures. In [6], the authors propose to use Singular Value Decomposition to deal with the high dimensionality of their feature vector reducing the dimension to 150 from 1125. They compute the value of the visual content metric of the most static frame cluster as the threshold value for the clustering process. In their scheme shots rather than keyframes are included in the summarization. In [8], the authors use a hierarchical clustering

algorithm based on spectral clustering that merges similar frames into clusters in a tree-structured representation with individual frames at the leaves. The authors use this type of clustering to derive a higher semantic level stories and shots and their work is not directly related to summarization.

2.1 Related Work Summary

We first introduced DT as a viable clustering technique for video summarization in [19]. In this paper, we expand on that initial effort with the batch processing of a larger video collection with metrics to evaluate the representational power of generating keyframes using DT clustering. We also include comparison with K-means clustering and Open Video Project's storybook generation. While many of the clustering based approaches discussed in the previous paragraphs are based on threshold specification and therefore not totally automatic, we demonstrate through experimentation and results that DT is fully automatic and well suited for batch processing without user intervention. The clustering overhead varies for different algorithms from $O(n)$ for K-means clustering to graph-based algorithms needing $O(n^2)$ -- DT as pointed out earlier requires $O(n \log n)$. The temporal order is not maintained in any of the keyframe-based summarization approaches discussed in [4-6, 8, 18] including our technique. While we use a data set of about 50 video segments from Open Video project, each 2 to 5 minutes long, others have used 15 to 30 second commercials [18] or 2-5 movies [4, 5]. In [6], the test data set includes a total of two hour content from news reports, documentaries, political debates, all used to summarize one single political event. Objective evaluation of the generated summaries is mostly absent in the cited references, while a few highlight quantifiable aspects of their respective algorithms [6]. In general, such objective analysis is a hard task made more challenging by the absence of benchmarked data sets and lack of agreed-upon metrics. Despite this shortcoming, the intellectual merit of the algorithms developed in previous work and our own, we argue is a step toward advancing the field. While acknowledging this state of research, our goal for this paper is to quantify as much as possible the representational power of our summarization technique.

3.0 Clustering using Delaunay Triangulation

Defining and detecting groups of frames of interest is the first key step in any type of content analysis application and clustering techniques have previously been used for that purpose. We propose a novel method based on DT to cluster multi-dimensional point data corresponding to the frame contents of the video. Clustering techniques used in previous research rely on user input and the quality of clustering depends on that input. For example, most of the clustering techniques currently used require a predefined number of clusters or threshold parameters. These parameters are mostly found by trial and error after many repetitions. This type of parameter tuning is expensive and inefficient for huge data sets. Several iterations are required before good quality clustering is arrived at. In contrast, the clustering method used in our technique generates clusters without any user specified arguments and is well suited for automatic batch processing of large data sets.

The basic idea in the proposed method is to represent individual frames as data points in generating a DT. Using DT, the inter-frame similarity relationship is mapped to the spatial proximity relationship among data points. The Delaunay Triangulation of a point set is the dual of the famous Voronoi Diagram, which is a partition of the space into cells, one for each data point, so that the cell for data point x consists of that region of the space that is closer to x than to any other data points. An edge in the Delaunay diagram connects points a and b if and only if the Voronoi cells containing a and b share a common boundary. Hence, edges of the DT capture spatial proximity [11]. The DT for a point set is unique and has the additional property that the circumcircle (or circumhypersphere for the general n dimensional space) of any triangle in the triangulation contains no other data point. In the Delaunay diagram, the edges can be grouped into two types: the *intra-cluster* edges and the *inter-cluster* edges. The former connect points within a cluster and the latter connect individual clusters. The inter-cluster edges reflect graph discontinuity and act as separating edges. The clustering process ends after the inter-cluster edges are identified and removed. The modeling and separating processes are shown in Figure 1. While all data points are connected in the Delaunay diagram, we remove the separating edges and the remaining connected components represent natural clusters.

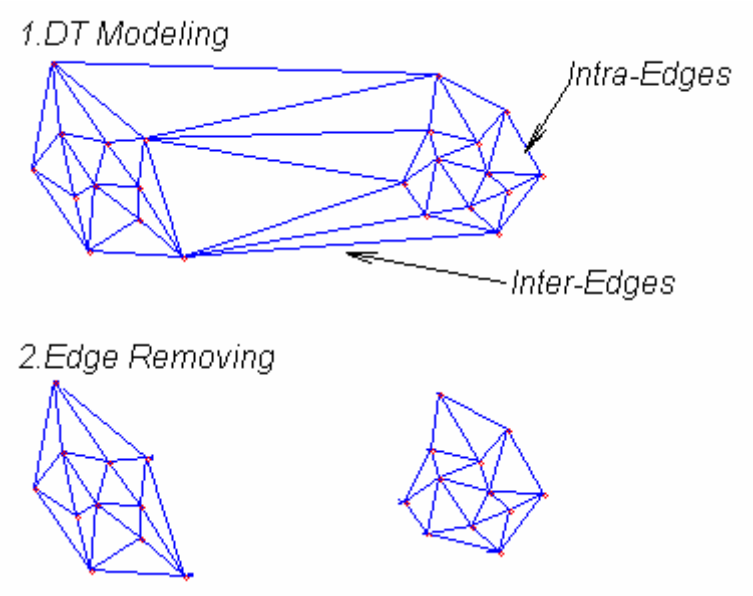

Figure 1. Delaunay Edges

To detect the separating (*inter-cluster*) edges, we notice that the end-points of these inter-cluster edges have greater variability in the edge length since these points connect both intra-cluster edges and inter-cluster edges. Intuitively, inter-cluster edges are longer than intra-edges. For each point in the DT, we calculate the length of each incident edge and determine the local mean length of all the incident edges for that point. The standard deviation of all those incident edges is also computed as the local standard deviation. The mean length and local standard deviation of the incident edges capture the local effect of a single point. The formal definitions for the mean edge length and local standard deviation for each data point follows from Definitions 1 and 2.

Definition 1. The mean length of edges incident to each point p_i is denoted by Local_Mean_Length(p_i) and is defined as

$$\text{Local\_Mean\_Length}(p_i) = \frac{1}{d(p_i)} \sum_{j=1}^{d(p_i)} |e_j|$$

Where $d(p_i)$ denotes to the number of Delaunay edges incident to p_i and $|e_j|$ denotes to the length of Delaunay edges incident to p_i.

Definition 2. The local standard deviation of the length of the edges incident to p_i is denoted by Local_Dev(p_i) and is defined as

$$\text{Local\_Dev}(p_i) = \sqrt{\frac{1}{d(p_i)} \sum_{j=1}^{d(p_i)} (Local\_Mean\_Length(p_i) - |e_j|)^2}$$

To incorporate both global and local effects, we take the average of local standard deviation of the edges at all points in the Delaunay diagram as a global length standard deviation as defined in Definition 3.

Definition 3. The mean of the local standard deviation of all edges is denoted by Global_Dev(P) and is defined as

$$\text{Global\_Dev}(P) = \frac{1}{N} \sum_{i=1}^{N} Local\_Dev(p_i)$$

Where N is the number of total points and P is the set of the points.

All edges that are longer than the local mean length plus global standard deviation are classified as inter-edges (Definition 5) and form the separating edge between clusters. The formal definition for short and separating edges in terms of mean edge length of a point and mean standard deviation are captured in Definitions 4 and 5 below.

Definition 4. A short edge (*intra-cluster edge*) is denoted by Short_Edge(p_i) and is defined as

$$\text{Short\_Edge}(p_i) = \left\{ e_j \mid |e_j| < Local\_Mean\_Length(p_i) - Global\_Dev(P) \right\}$$

Definition 5. A Separating edge (*inter-cluster edge*) is denoted by Separating_Edge(p_i) and is defined as

$$\text{Separating\_Edge}(p_i) = \left\{ e_j \mid |e_j| > Local\_Mean\_Length(p_i) + Global\_Dev(P) \right\}$$

The algorithmic steps involved in generating Delaunay clusters are summarized in Figure 2.

1. Generate DT for the multi-dimensional point data;
2. Calculate the length of each incident edge for each point;
3. Calculate mean length of incident edges for each point;
4. Calculate local standard deviation of length of incident edges for each point;
5. Calculate global standard deviation as the average of local standard deviation;
6. Identify all edges longer than local mean length plus global standard deviation as inter-cluster edges;
7. Remove inter-cluster edges to obtain Delaunay clusters.

Figure 2. Delaunay Clustering Algorithm

More on the definitions of Delaunay edges and related concepts can be found in [9, 12]. Efficient construction of DT is discussed in [10-12]. The computation of Delaunay Triangulation can be done effectively in *O(nlogn)* time and the identification of inter and intra edges can be done in *O(n)* time where *n* is the number of frames processed.

4.0 Video Summarization Technique

Our proposed summarization technique depends on removing the visual-content redundancy among video frames. Like many other approaches, the entire video material is first grouped into clusters, each containing frames of similar visual content. By representing each cluster with its most representative frame, a set of keyframes is obtained which then summarizes the given sequence.

As shown in Figure 3, the general keyframe based video summarization system contains three major steps: pre-sampling, clustering, and keyframe selection. The first step is to obtain the video frames from the original video. Earlier approaches based on shot detections return a fixed or variable number of frames per shot. This shot based approach may still contain redundancies because similar content may exist in several

shots. For example, in news videos, the anchor person will appear many times in

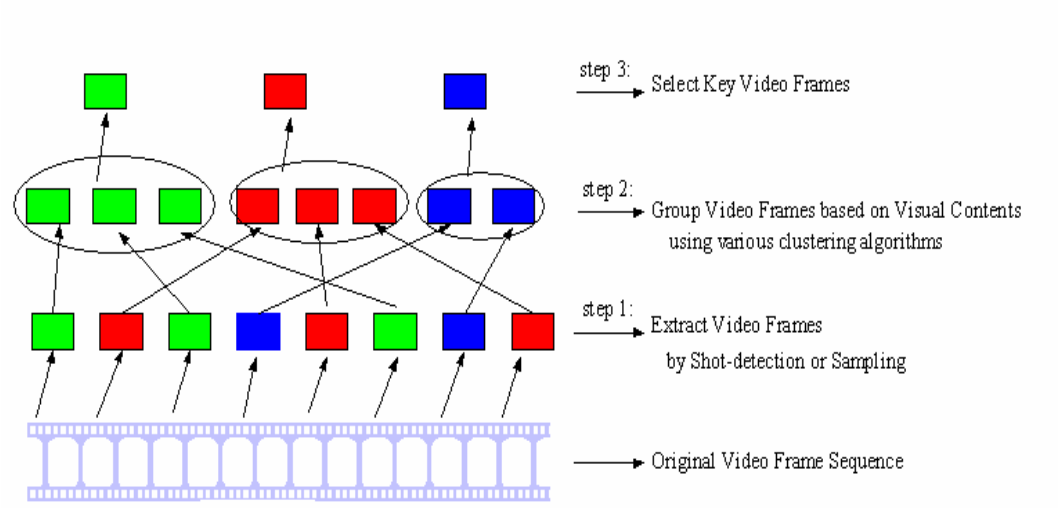

Figure 3. Key Frame-based Video Summarization

several video shots and those same frames may appear repeatedly in the summary. In contrast, we work on the video frames directly and cluster the frames that have similar content obviating the need for shot detection. All video frames extracted from the original video are processed. However, we can pre-sample the video frames to reduce the number of frames that will be used in the clustering algorithm. As we show later in our experiments, the quality of the summaries is not affected by pre-sampling. The rationale for pre-sampling is that any video with a digitization rate of 30 frames per second, when fully decompressed, will generate too many frames with redundancies among consecutive frames. It is possible to capture the video content with pre-sampled video frames using a judicious sampling rate. In steps 2 and 3, we perform Delaunay clustering and select keyframes. We provide a summary of the keyframe selection algorithm in Figure 4 and describe it in the following paragraph.

4.1 Algorithm Description

The first step in the proposed summarization algorithm is feature extraction. We use color histograms to represent the visual content of video frames in our experiments for testing the clustering algorithm. Each frame is represented by a 256-bin color histogram in the HSV color space where there are 16 levels in H, 4 levels in S and 4 levels in V according to the MPEG-7 generic color histogram descriptor [20]. Once each frame is represented by this 256-bin color histogram, we order it as a 256-D row

vector and stack such vectors for each frame into a matrix which represents the video content of the original video. Since color histograms tend to generate sparse matrices, we use Principal Component Analysis (PCA) [13] to reduce the dimensions of the matrix but still capture the video content. The benefit of using the PCA algorithm is the reduction in processing time. After applying PCA, each frame with the *m*-dimensional (*m* = 256 in our case) raw feature space is projected on to a *d*-dimensional refined feature space where *d* is the number of the selected Principal Components (PCs). The value of *d* is a design parameter and in our experiments, we choose the first *d* largest PCs that contribute to more than 90% of the total variation. Therefore, *d* is not a fixed value as in [6]. Later in our experimental results, we show that a small number of *d* of about 7 is sufficient to capture 90% or more of the total variation for most of the videos in our test collection. We apply the DT algorithm to data points in reduced dimension and generate Delaunay diagram. Individual clusters are identified using the process described in Section 3. Once the clustering process is complete we find the keyframe for each cluster using the centroid of that cluster.

1. Sampling the frames from the input video sequence, sampling rate is a design parameter.
   This step is optional. The sampled n selected frames are further processed.
2. Extract the color histogram feature from each of the n selected frames and create the feature-frame matrix $\vec{A}$.
   After this step, each frame is represented by a 256-dimensional feature vector and the video sequence is represented by this matrix.
3. Perform the PCA on $\vec{A}$ to get the refined feature matrix $\vec{B}$, which is a $n \times d$ matrix, where n is the total selected frames and d is the selected number of Principal Components.
   After this step, each frame is represented by a d-dimensional feature vector, or a point in d-dimensional space. Note that the number d is dependent on the content of each video. We select the largest d principal components that represent at least 90% total variance.
4. Build the Delaunay Diagram for the n points in d-dimensions.
5. Find the Separating Edges in the Delaunay Diagram.
6. Remove these Separating Edges to get the clusters.
   The above three steps are the core of our automatic clustering process.
7. Select the frame that is nearest to the center of each cluster as the keyframe.

Figure 4. Keyframe Selection Algorithm

5.0 Summarization Results

In this section, we present details of implementation, test video set, and an evaluation of the results by comparing them to both OpenVideo (OV) Storyboard and results from K-Means clustering.

5.1 Experimental Setup

Implementation The code for constructing the DT is based on QHULL [21]. Our system is implemented using MATLAB. QHULL implementation using MATLAB works well for datasets in low dimensions but is problematic for handling large datasets in more than 16-dimensions as documented in [22]; however, as we discovered with our implementation, 7 is the maximum number of dimensions that can be used without runtime problems. Due to such constraints, we test video segments of 2 to 5 minutes and limit the number of dimensions to 7 using PCA. The number of frames in the videos is further reduced by pre-sampling. The sampling rate we use in our experiment is one out of every 10 frames which gives us a sample of 3 fps for a video with 30 fps. The video lengths and the sampling rate are comparable to datasets used by other researchers (See Related Work Section and [6]). Before we build the DT, the pre-sampled video is reduced in dimension using PCA. In the experimental setup, the largest number of PCs that represents at least 90% total variance is selected. However, this number can not be larger than 7 for reasons mentioned above. Later in our experiments, we show that we can capture most of the content for a majority of the test collection with PCs less than 7. Given that the complexity of DT is $O(nlogn)$, testing full videos in higher dimensions is possible with a better DT implementation. The theoretical runtime efficiency of DT is borne out by our experiments where we notice that the processing times are roughly between 9 and 10 times the video length. We run our algorithm using MATLAB R12 on a Dell PC with 2.4GHz CPU and 1GB RAM running Windows XP. All videos in the test collection were setup for batch processing in the experiments.

Test Video Set For the experiments, the test video segments are MPEG compressed and downloaded from the Open Video Project's shared digital video repository [23]. They are first decompressed using official MPEG codec from MPEG Software Simulation Group [24]. We test the summarization algorithm on 50

randomly chosen video segments, each of length between 2 to 5 minutes, pertaining to news and documentaries in color. Using global color histogram feature provides a computationally effective way of detecting the overall differences in keyframes. There is no single method that works for all genres of videos. For sports video, summarization method using motion feature may work better than other features [2]. As we show later in our experiments, using just the color feature still generates good quality summaries and is in tune with our goal of developing an automatic technique with less overhead.

Evaluation Procedure Developing an objective evaluation procedure for a video summarization method is difficult as has been acknowledged by other researchers in this field. Much of the problem comes from the absence of standardized metrics, the creation of which is challenging due to the disparities between feature-based low level analysis and higher level semantics of the video content. Having made these observations, we attempt to make use of those characteristics of our algorithm that can be quantified and measured. Towards that end, we define three metrics: 1) *Significance Factor* is used to denote the significance of the content represented by each cluster using the number of frames in each cluster; 2) *Overlap Factor* is used to make a meaningful comparison between our summaries and OV storyboard; 3) *Compression Factor* is used to quantify the reduction in frame count from the original video source.

We define the significance of each cluster as

$$Significance\_Factor(l) = \frac{C_l}{\sum_{j=1}^{K} C_j} \quad where \ C_l$$ is the number of frames in cluster

l, and K is the number of total clusters.

The overlap factor for video *i* is defined as

$$Overlap\_Factor(i) = \frac{\sum C_k}{\sum_{j=1}^{K} C_j}$$

where *k* represents a subset of clusters in our summary. The keyframes of these clusters are found in OV storyboard as indicated by the content similarities. The overlap factor determines the cumulative significance of the overlapped content between the two techniques using the cluster significance factor defined above. The rationale behind this idea of overlap factor is to penalize if our summary is not fully represented in OV. The shortfall in the overlap also tells us the significance of the missed content. Where OV contains non-redundant keyframes that are not represented in our summary, we have no way of assessing the significance or the cluster origin of those missed frames without a detailed examination of all DT clusters. In the interest of a simplified and automatic process, we ignore such post-processing steps. If our summary contains more frames than OV, the overlap factor will be less than 100% but we are able to assess the significance of missed content in OV storyboard using the cluster significance factor.

The Compression factor video *i* is defined as

$$Compression\_Factor(i) = \frac{K}{N}$$

where *K* is the number of keyframes and *N* is the total number of processed frames in the original video. This metric gives an indication of the size of the summary with respect to the original content. For a temporal video summary, this quantity is generally measured as speedup. However, for a static keyframe-based summary like ours, the reduction in the number of frames provides a good metric. In the next section, we use these metrics to provide an objective evaluation of the comparison between OV storyboard and our summary.

5.2 Comparison with OV Storyboard

The Open Video Project is a valuable resource for comparing the performance of the proposed summarization technique with a third party algorithm. The storyboard for each video presented at the OV Project's website [23] is generated using the algorithm from [18] and some manual intervention to refine results so generated; the scheme for generating them is published in [25]. In this section, we provide pairwise comparison of our results with OV storyboard for each video segment. While this comparison is meaningful, we can not completely depend on the OV storyboard as ground truth as it contains temporal order and for that reason there may be redundant

frames. In our result, we present keyframes in the order of the significance factor; the first keyframe is always from the largest cluster. Our display of the keyframes does not preserve temporal order. However, we can compare content coverage by visually examining individual frame content and determining the overlap between the two summaries. This type of comparison is made for all 50 videos in our collection. Table 1 below provides a summary of the performance for all 50 videos. Based on the experimental results, we find that 32 of the 50 videos have fewer keyframes than OV storyboard; 2 videos have identical summaries; 10 videos show more number of keyframes than the OV storyboard; 6 of the 50 videos show mismatched keyframes with not much overlap. With 3 of those 6 videos, there were decoding problems unrelated to our algorithm performance. The other 3 videos show very low variance after PCA, below 70% for the number of PCs equal to 7, the maximum for the MATLAB implementation. As a result of this, a large portion of the content is lost and the representational value of the summary is also lost. This problem can be easily fixed with a better implementation of DT. In addition, only a small percentage of videos in the collection are affected by this problem.

Results for all 50 videos in our collection are made available at http://www.csee.umbc.edu/~pmundur/m2net.html.

Table 1. Experimental Results Summary

Total Clips	Category 1 Same Keyframes	Category 2 Fewer Keyframes	Category 3 More Keyframes	Category 4 Mismatched Keyframes
50	2	32	10	6

In the following paragraphs, we present results for 10 videos from the test collection that fall into one of the categories in Table 1. They are chosen with no particular disposition but represent the range of performance for the other videos in the test collection. The results for the 10 videos are summarized in Table 2 and discussed in detail using Figures 5-14 (referred to as DT Summary in all the figures).

Table 2. Experimental Results

Video Segment Title	**#Frame**	**#Cluster**	**Compression (%)**	**Overlap (%)**	**Significance (%)**
Drift Ice as a Geologic Agent, segment 7	1940	4	0.21	100	(32, 28,27,13)
Drift Ice as a Geologic Agent, segment 10	1400	5	0.36	100	(36, 25, 22,14, 3)
A New Horizon, segment 5	2900	7	0.23	96	(37,25,17,9,8,3,1)
A New Horizon, segment 8	1810	7	0.38	80	(21,20,16,13,13, 10,7)
The Voyage of the Lee, Segment 15	2270	5	0.22	97	(49,22,18,8,3)
The Future of Energy Gases, Segment 3	2930	8	0.27	97	(30,21,17,10,8,7,4,3)
America's New Frontier, segment 3	2160	5	0.23	100	(27,23,18,17,15)
America's New Frontier, segment 4	3700	7	0.19	100	(18,18,17,15,15, 10,7)
America's New Frontier, segment 10	4820	6	0.12	100	(21,20,19,15, 14,11)
Drift Ice as a Geologic Agent, segment 5	2180	8	0.37	79	(18,17,16,14,13,12,5,5)

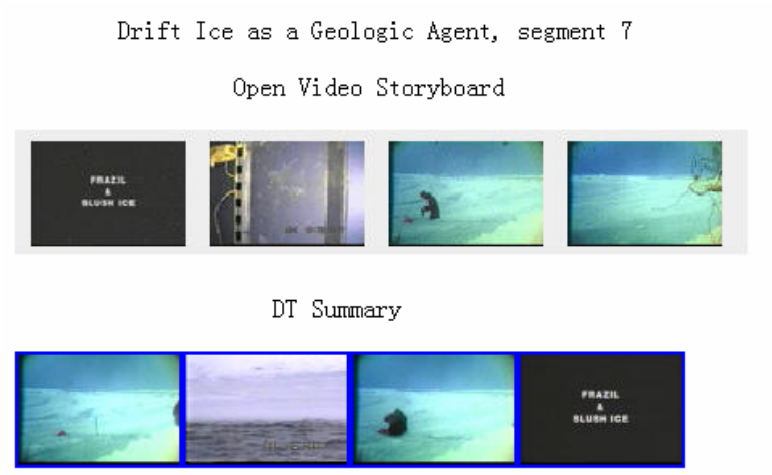

Figure 5. OV Storyboard Versus DT Summary for Drift Ice, Segment 7

Figures 5 and 6 show results for two videos where the DT summary is identical to OV storyboard (category 1 from Table 1). Both contain the same number of keyframes and result in 100% overlap factor. The first three clusters in DT summary have a combined significance factor of about 85% indicating that the content represented by those keyframes and their respective clusters accounts for 85% of the frames from the original video. The summarized content is less than 0.5% of the original number of frames.

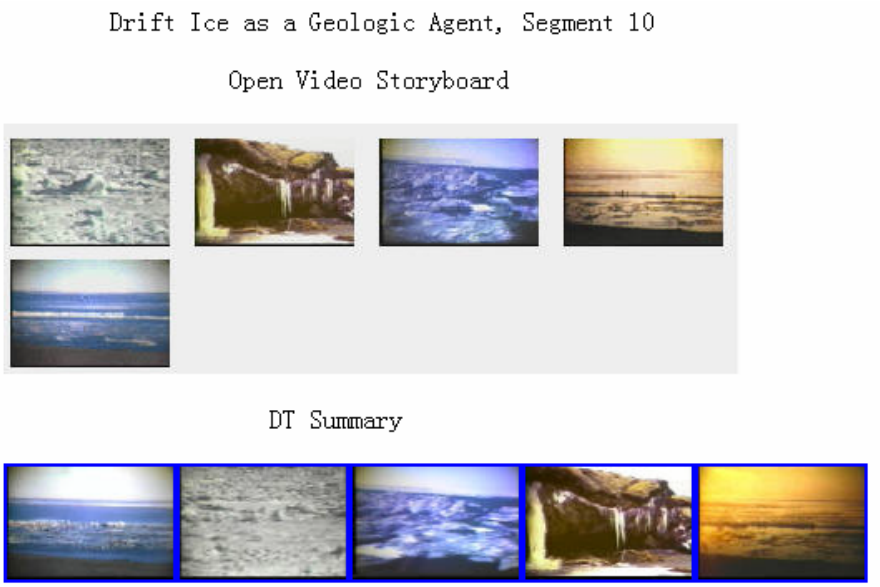

Figure 6. OV Storyboard versus DT Summary for Drift Ice, Segment 10

Figures 7 through 13 represent the videos which have fewer keyframes than OV storyboard (category 2 from Table 1) but represent an overlapped content of about 80% to 100%. The number of extra frames in OV is the result of redundancy or the presence of keyframes not in DT summary. For instance, in Figure 7, there are 5 keyframes in common between OV and DT accounting for 96% overlap factor. Similarly, in Figure 8, we only have 4 keyframes in common resulting in an overlap of 80%. The next two videos in Figures 9 and 10 while still having fewer keyframes than OV storyboard, show an overlap factor of 97%. About 3% of the content as represented in the last keyframe in the DT summary is not found in OV storyboard. The video summaries shown in Figures 11 to 13 show 100% overlap with all of the

keyframes in DT summary being represented in the OV storyboard while still containing fewer keyframes in the DT summary. All of the keyframes in DT summary are displayed in the order of significance of the cluster they come from. By displaying the key frames in the order of their significance factor, users will get a quick view of the original content since the size of the cluster corresponds to the length/content of the video. There are 32 out of 50 videos that generate fewer keyframes than the OV storyboard in our collection. More than half of them exhibit an overlap factor of 80% or more with about a very small number (4 out of 50) of them showing an overlap of 50%-60%. The reason for the low overlap is that the keyframes from the larger clusters in DT summary are not represented in the OV storyboard.

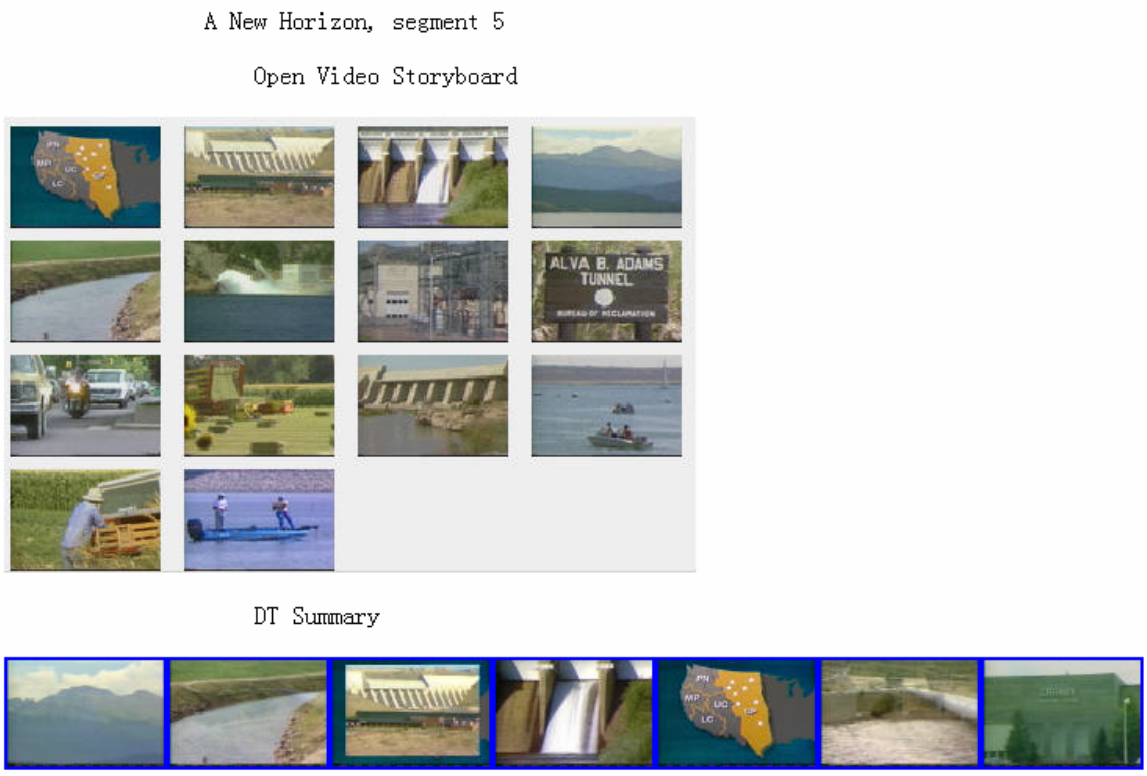

Figure 7. OV Storyboard versus DT Summary for A New Horizon, Segment 5

Figure 8. OV Storyboard versus DT Summary for A New Horizon, Segment 8

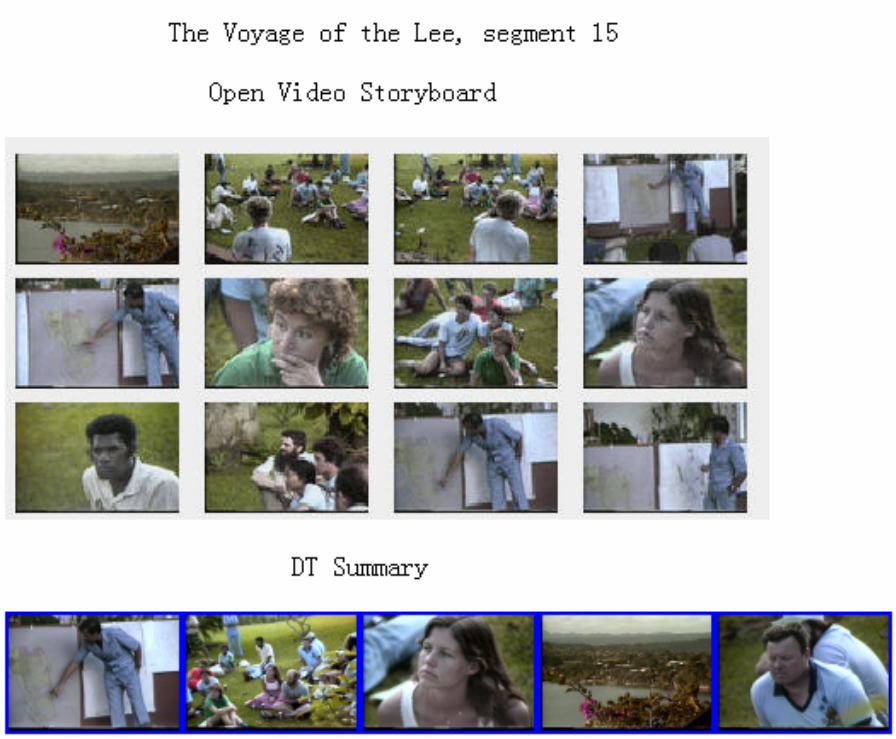

Figure 9. OV Storyboard versus DT Summary for Voyage of the Lee, Segment 15

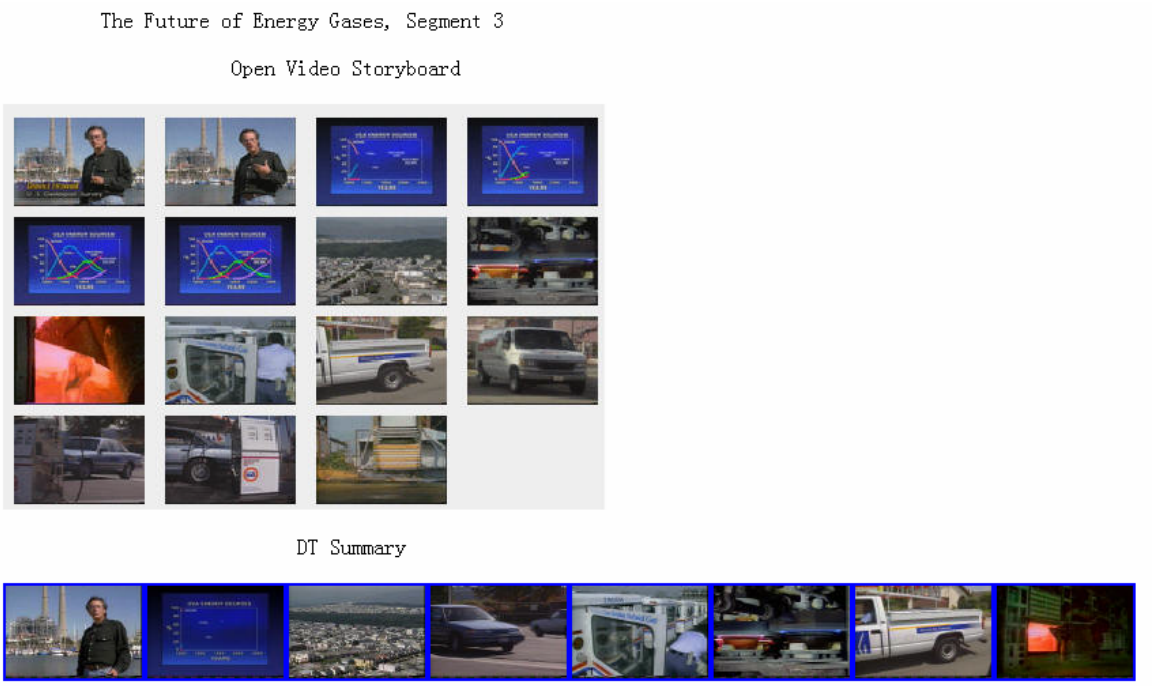

Figure 10. OV Storyboard versus DT Summary for Future of Energy Gases,Segment3

Figure 11. OV Storyboard versus DT Summary for New Frontier, Segment 3

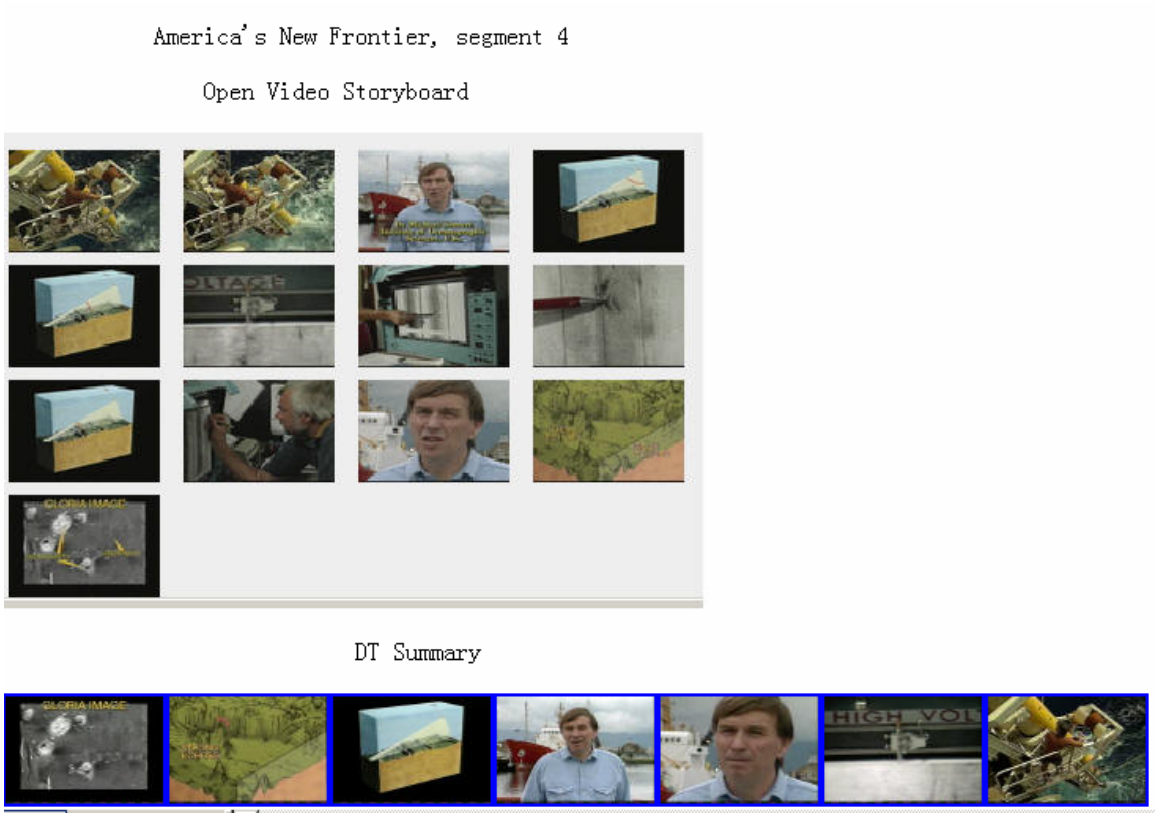

Figure 12. OV Storyboard versus DT Summary for New Frontier, Segment 4

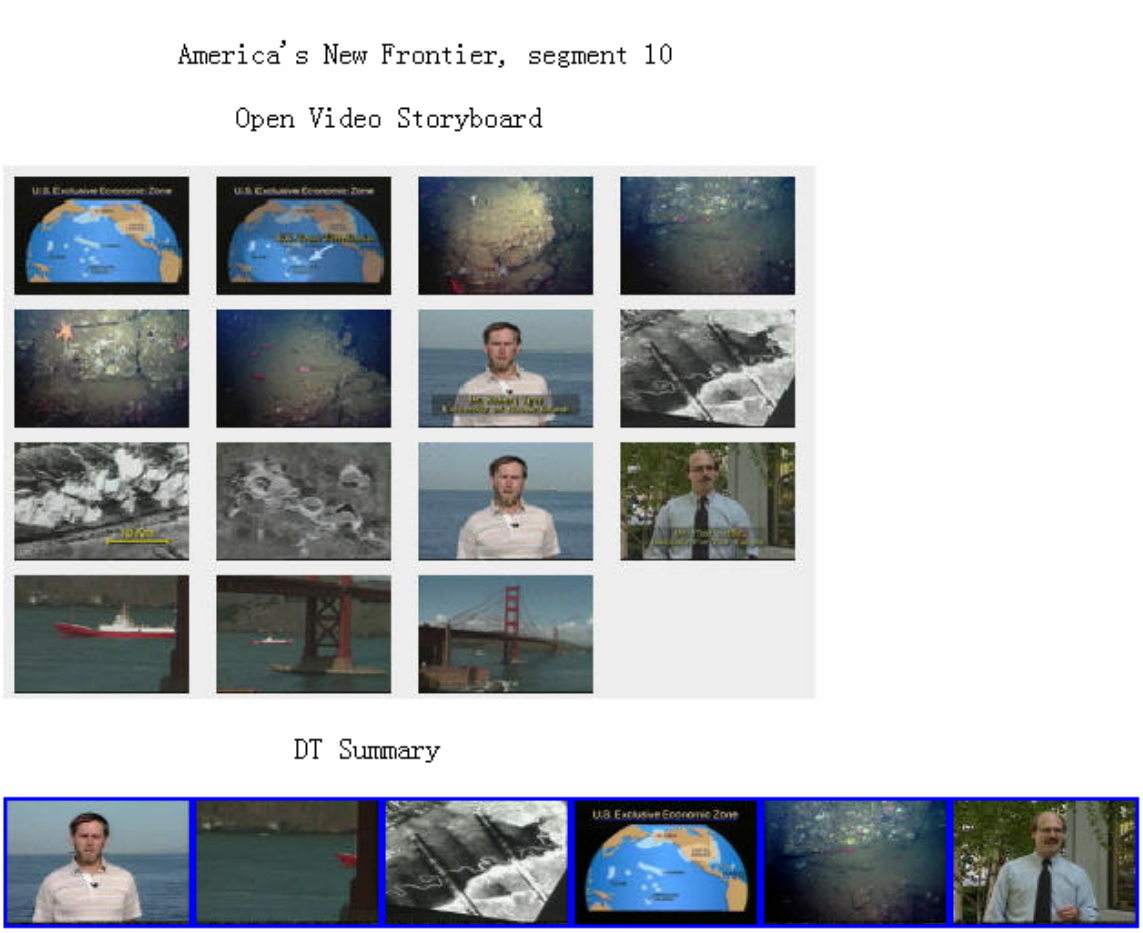

Figure 13. OV Storyboard versus DT Summary for New Frontier, Segment 10

Finally in category 3 from Table 1, DT summary contains more key frames than the OV storyboard for some videos -- 10 of the 50 videos fall in this category and exhibit a high overlap indicating common keyframes between the two summaries. An example of such a result is shown in Figure 14. The overlap factor is 79% for the video shown. We notice that a critical keyframe from the DT summary accounting for about 16% cluster significance is missing in OV. For this category of videos, since some of the keyframes from DT are always missing in OV, the overlap factor will be less than 100%. Many of the extra keyframes in the DT summary however, provide more content coverage as indicated by the cluster significance factor and as shown in the result in Figure 14. This performance is consistent with the other 9 videos in the test collection (See the URL mentioned earlier in this section for complete results).

Figure 14. OV Storyboard versus DT Summary for Drift Ice, Segment 5

In summary, our proposed summarization technique has the following advantages:

1. The proposed algorithm does not require user specified thresholds or other tunable parameters and is well suited for batch processing as we have demonstrated with our experiments;
2. Each cluster shows very good clustering performance in terms of similarity of content; the clustering process results in clusters of different sizes based on the content represented in the original video. We are able to convey the content with fewer frames from the larger and more significant clusters.
3. We have shown a meaningful comparison of our results with OV storyboard for majority of the videos in the test collection where we demonstrated that the summaries generated by the proposed algorithm are as good as OV storyboard and in some instances, better. Where we generate fewer keyframes than OV, we have shown that the content coverage from DT summary in most videos is as good as OV storyboard. For videos where we generate more keyframes than OV, we have shown a way of assessing significance of missed content in OV.

5.3 Comparison with K-Means Clustering

In this section, we provide a comparison between DT summary and summaries generated using K-means clustering. We choose K-means because of its low computational overhead and reasonably good performance. We still have to decide on

an optimal number of clusters to obtain the required content coverage if we use K-means clustering.

The results presented are for the test video segment 'America's New Frontier', segment 4 (file name UGS02_004.mpeg). The K-means clustering algorithm is from [26]. We set the value of *K*, the number of clusters to be 3, 5, 11 and 20 respectively.

In the proposed DT scheme, the clusters of different sizes based on the duration of the video content result in better quality summaries. As we increase the number of clusters in K-means clustering, we notice that there are a lot more redundant frames in the summary. This type of redundancy is eliminated in our clustering scheme because there is no set number of clusters that the content needs to be distributed to as in K-means clustering. Figure 15 represents the keyframes using our algorithm. Keyframes from K-means Clustering for different values of *K* are shown in Figure 16. Notice the absence of any redundancy and also the content coverage in the DT summary to be about the same as in the summary for *K= 11* from K-means clustering shown in Figure 16. Comparing the two results, we note that the DT advantage over K-means is its suitability to automatic batch processing with no user specified parameters such as the number of clusters and similar content coverage with fewer frames.

Figure 15. DT Summary for New Frontier, Segment 4

6.0 Conclusion and Future Work

In this paper, we proposed an automatic video summarization technique based on Delaunay Triangulation. The summarization technique is free of user-specified modeling parameters and generates video summaries by capturing the visual content of the original videos in fewer frames than other summarization techniques. We presented meaningful comparisons of the results from the proposed DT algorithm to OV storyboard by defining metrics such as significance factor, overlap factor, and the compression factor all of which evaluate the representational power of the proposed

DT summarization technique. We also demonstrated the DT advantage for batch processing over K-means clustering where the optimal number of clusters needs to be predefined.

In our future work, the Delaunay clustering will be applied to cluster video frames on several different features such as text, audio, and motion features. As can be seen from the results, the clustering performance is especially good in separating frames with text content or close-up of the speakers into separate clusters. Keyframes with text or face and their associated clusters can then be processed in a text detection or face recognition module for information extraction purposes. It appears and remains part of future work that many other content analysis-based applications can be designed using the proposed Delaunay clustering as the core layer in an integrated automatic video processing architecture.

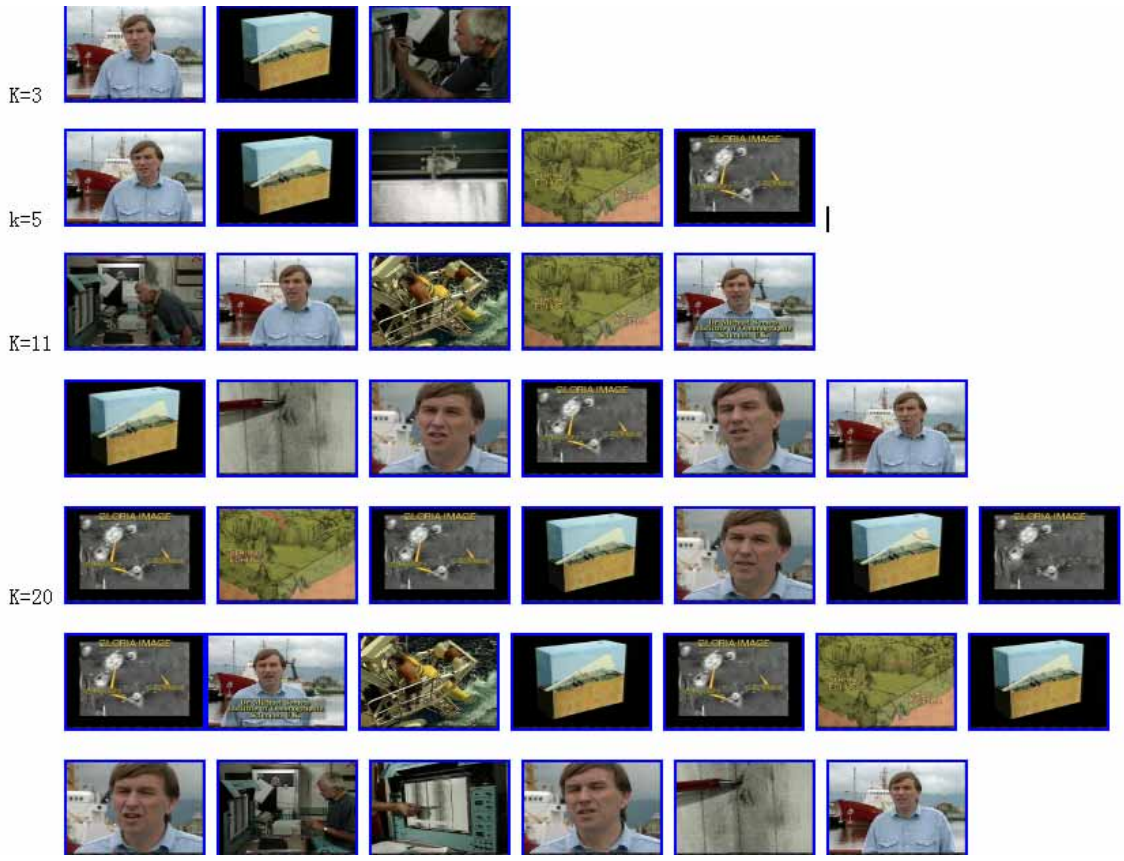

Figure 16. K-Means Summary for New Frontier, Segment 4

References

1. Akamai Technologies, Akamai Streaming – When Performance Matters, *White Paper*, 2004.
2. N. Dimitrova, H. J. Zhang, B. Shahray, I. Sezan, T. Huang, and A. Zakhor. Applications of video content analysis and retrieval. *IEEE Multimedia*, pp 42-55, July-September 2002.
3. M. G. Christel, M. A. Smith, C. R. Taylor, D. B. Winkler. Evolving video skims into useful multimedia abstractions. In Proceedings of *ACM Conference on Human Factors in Computing Systems*, April 1998.
4. Y. Zhuang, Y. Rui, T. S. Huang and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. In Proceedings of *IEEE International Conference on Image Processing*, 1998.
5. A. Hanjalic and H. J. Zhang. An integrated scheme for automated video abstraction based on unsupervised cluster-validity analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8), December 1999.
6. Y. Gong and X. Liu. Video summarization and Retrieval using Singular Value Decomposition. *ACM Multimedia Systems Journal*, 9(2), pp 157-168, August 2003.
7. A. Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In Proceedings of *Neural Information Processing Systems*, 2002.
8. D. Q. Zhang, C. Y. Lin, S. F. Chang, and J. R. Smith. Semantic video clustering across sources using bipartite spectral clustering. In Proceedings of *IEEE Conference on Multimedia and Expo (ICME)*, 2004.
9. V. Estivill-Castro and I. Lee. Autoclust: Automatic clustering via boundary extraction for massive point-data sets. In Proceedings of the fifth *International Conference on Geocomputation*, 2000.
10. S. Fortune. A sweepline algorithm for Voronoi diagrams. Algorithmic, 2(2), 1987.
11. S. Fortune. Voronoi diagrams and Delaunay Triangulation. In *Computing in Euclidean Geometry*, D. Z. Du and F. Hwang (eds.), World Scientific Publ. 1992.
12. R. A. Dwyer. A faster divide and conquer algorithm for constructing Delaunay triangulations. *Algorithmic*, 2(2), 1987.
13. E. Sahouria and A. Zakhor. Content analysis of video using principal components. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8), December 1999.
14. H. Zhang, J. Y. Wang, Y. Altunbasak. Content-based retrieval and compression: A unified solution. In Proceedings of *International Conference on Image Processing (ICIP)*, 1997.

15. B. Shahray. Gibbon: Automatic generation of pictorial transcripts of video programs. In Proceedings of *IS&T/SPIE Digital Video Compression: Algorithms and Technologies*. 1995.
16. H. Ueda, T. Miyatake, and S. Yoshizawa. Impact: An interactive natural picture dedicated multimedia authoring systems. In Proceedings of *ACM SIGCHI*, April 1991.
17. H. S. Chang, S. S. Sull, and S. U. Lee. Efficient video indexing scheme for content based retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(8), December 1999.
18. D. DeMenthon, V. Kobla, D. Doermann. Video Summarization by curve simplification. In Proceedings of *Computer Visualization and Pattern Recognition (CVPR)*, 1998.
19. Y. Rao, P. Mundur, and Y. Yesha. Automatic video summarization for wireless and mobile environments. In Proceedings of *IEEE Computer Communication (ICC)*, June 2004.
20. B. S. Manjunath, J. R. Ohm, V. V. Vasudevan, and A. Yamada. MPEG-7 color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(11), June 2001.
21. Homepage for QHULL. http://www.qhull.org/
22. QHULL FAQ at http://www.qhull.org/html/qh-faq.htm
23. The Open Video Project at http://www.open-video.org/
24. The MPEG Software Simulation Group at http://www.mpeg.org/MPEG/MSSG/
25. G. Marchionini and G. Geisler. The open video digital library. *D-Lib Magazine*, 8(12), December 2002.
26. Sing-Tze Bow. *Pattern recognition and image processing*, Publisher Marcel Dekker, Inc., New York, 2002.