ABSTRACT
In this paper, we present a method for BBC rushes summarization. In the proposed method, first the input video is decomposed into fragments by comparing consecutive frames. Next, these fragments are grouped by a clustering method. Using the clustering result, consecutive fragments are grouped into segments. Then the adjacent segments are merged if the distance between them falls below a threshold. Finally, to generate the summaries, we select a subset of the frames of the longest segment in each cluster. The performance of the proposed method on TRECVID 2007 test set is reported.

Categories and Subject Descriptors
H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing—Indexing methods; I.5.3 [Pattern Recognition]: Clustering—algorithms

General Terms
Algorithms, Experimentation, Performance

Keywords
video summarization, video skimming, rushes content

1. INTRODUCTION
With the rapid growth of many multimedia databases, there needs to be effective and scalable tools for indexing and retrieval based on video contents. One of the those important tools which can support this is video summarization. Summarization is useful to users as they can look at a short summary clip of a long video and decide easily whether it is relevant to them. Furthermore, summary videos can help users to browse and navigate large video archives efficiently and effectively.

In this paper, we focus on generating video summary for BBC rushes. Definition of rushes are quoted from [5]: "Rushes are the raw material (extra video, B-rolls footage) used to produce a video. Twenty to forty times times as much material may be shot as actually becomes part of the finished product. Rushes usually have only natural sound. Actors are only sometimes present. So very little if any information is encoded in speech. Rushes contain many frames or sequences of frames that are highly repetitive, e.g., many takes of the same scene redone due to errors (e.g. an actor gets his lines wrong, a plane flies over, etc.), long segments in which the camera is fixed on a given scene or barely moving. A significant part of the material might qualify as stock footage - reusable shots of people, objects, events, locations.". From the definition, there are two main challenges in solving this problem:

- Redundancy at frame level: there are many frames recorded for events or objects. Especially, in the case of objects, only several tens of frames can be used to recognize them compared to several hundreds or thousands of frames recorded in the original video.

- Redundancy at segment level: In general, the summary consists of segments in which each segment describes some object or event [6]. In datasets such as BBC rushes, many similar segments exist but to detect them are very difficult due to large variations in length, acquired conditions (e.g. wind changes between two takes), motions, scenes and so on.

In order to handle these challenges, we propose a method using clustering techniques to group similar frames and segments into clusters and only the representative segment of each cluster is used for summary generation. Specifically, first we group similar consecutive frames into fragments by using a naive fragment detection technique which is a simple version of shot boundary detection to eliminate the redundancy at frame level. Next we do clustering on these fragments and use the clustering result to combine consecutive fragments belonging to the same cluster into segments. The adjacent segments are then merged into larger segments to reduce the redundancy at segment level. Finally, to construct the summaries, the largest segments of the clusters are selected and only a subset of frames of each selected segment is used for making the summary clip. Experimental results on TRECVID 2007 test set show that our proposed method is promising.
2. SYSTEM OVERVIEW

2.1 Video Decomposition

First of all, the input video is decomposed into fragments. Each fragment consists of consecutive frames which look visually similarly and have no significant change in motion. To do so, each 352x288 frame image is divided into regions using a 7x6 grid (each region in this grid is approximately 50x48 pixels). Then the features extracted from these regions are concatenated to form a feature vector for the input frame image.

Since color moments have been successfully used in retrieval systems [1] and proved to be efficient and effective in representing color distributions of images [7], we use these values to represent the image content. The first order (mean), the second order (variance) and the third order (skewness) color moments are defined as:

\[ \mu_i = \frac{1}{N} \sum_{j=1}^{N} f_{ij} \]

\[ \sigma_i = \left( \frac{1}{N} \sum_{j=1}^{N} (f_{ij} - \mu_i)^2 \right)^{\frac{1}{2}} \]

\[ s_i = \left( \frac{1}{N} \sum_{j=1}^{N} (f_{ij} - \mu_i)^3 \right)^{\frac{1}{2}} \]

where \( f_{ij} \) is the value of the \( i \)-th color component of the image pixel \( j \), and \( N \) is the number of pixels in the image.

To detect fragment boundaries, the similarity between two consecutive frames is computed. If the similarity is below a threshold \( T \), a new fragment will be formed. The similarity used in this experiment is the Euclidean distance between two feature vectors extracted from the frames. The fragments shorter than \( k \)-seconds will be merged with adjacent fragments to ensure no output fragments shorter than \( \ell \)-seconds. \( k \) can be 1 or 2 seconds.

By using a strict threshold \( T \) and the color moment feature, fragments with visual coherence can be found as shown in Figure 1.

![Figure 1: The middle frames of three detected fragments. Head rotation of the woman makes the difference between two fragments 2 and 3.](image)

2.2 Fragment Clustering

Normally, adjacent fragments might be similar than far apart fragments. Furthermore, a group of continuous fragments (called a segment) often share some visual cues. This motivates us to do clustering on fragments.

For each fragment, we select the middle frame as the representative frame for that fragment and use the feature vector extracted from this frames for clustering fragments. We use GreedyRSC [3, 2] to find clusters with high precision (the number of clusters is determined automatically) [4]. Figure 2 shows members of a cluster.

Cluster 07 - 14 members

![Figure 2: An example of the clustering result in which fragments in this cluster show a shot of trees and then a motion of a women toward the camera.](image)

After this step, we have fragments with labels as shown in Figure 3:

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>...</th>
<th>Fn</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>C1</td>
<td>C2</td>
<td>C3</td>
<td>...</td>
<td>Ck</td>
</tr>
</tbody>
</table>

Figure 3: Fragments and their labels after the clustering process. \( F_i \) is \( i \)-th fragment and \( C_k \) is the cluster that \( F_i \) belongs to.

We merge consecutive fragments that belong to the same cluster into segments. For example, fragments \( F_1 \) and \( F_2 \) are merged into a segment, \( F_3 \) and \( F_4 \) form two different segments.

2.3 Segment Merging

There are two critical drawbacks in any clustering method. The first one is fragmentation clustering in which samples of one clusters are put into several different clusters and the second one is poor cluster boundary in which irrelevant and noisy samples might belong to. Some examples as shown in Figure 4 and Figure 5 clearly indicate that these drawbacks are inevitable. The main reason is it is difficult to find the appropriate representation and similarity measure.

To handle this problem, we use temporal information as a clue. Specifically, we compute the similarity between adjacent segments and check whether this similarity is lower than a threshold \( T \) or not. If it is the case, these two segments can be merged into a new segment and the same process is repeated with the new segment and its adjacent segment. Finally, two clusters \( C_l \) and \( C_j \) are merged if \( S_{ik} \) and \( S_{jl} \) have been merged where \( S_{ik} \in C_l \) and \( S_{jl} \in C_j \).

In order to compute the similarity between two segments, we compute the similarity between representative frames of these segments. A representative frame of a segment is defined as the frame closest to the mean frame of that segment.
The mean frame of a segment is simply computed as mean of frames in that segment.

2.4 Summary Generation

For each cluster, we select the longest segment to generate the summaries since it likely contains much information for the cluster that it represents. Due to the constraint on the length of the summary, only a subset of frames in the selected segment are used. The selection process of these frames is described as follows:

- Compute the total number of frames of the selected segments: \( N_s = \sum_{i=1}^{k} |S_i| \), where \( k \) is the number of final clusters, \( S_i \) is the longest segment of cluster \( C_i \).

- Compute new skim ratio (in frames)

\[
\text{SkimRatio}_{\text{new}} = \frac{N_s \times \text{SkimRatio}_{\text{max}}}{N}
\]

where \( N \) is the total number of frames in the input video and \( \text{SkimRatio}_{\text{max}} \) is the maximum skim ratio required. For example, if \( N_s = 30K \), \( N = 50K \) and the maximum of the length of the summary is set to 4%, so the \( \text{SkimRatio}_{\text{max}} = 25 \); and then

\[
\text{SkimRatio}_{\text{new}} = 25 \times \frac{30K}{50K} = 15.
\]

- Select frames in the selected segments at sampling rate \( \text{SkimRatio}_{\text{new}} \), i.e. frames 0, 15, 30, ....

3. EXPERIMENTAL RESULTS

We tested our proposed method for TRECVID 2007 test set [6]. As for generating time, since we processed all frames in a video for decomposition, the time for computing fragments is quite long. It took around one hour for a 30-minute video. The clustering process and the other post processing are much faster. On average, time for generating summary of our system is 4,406 seconds.

Compared to other approaches [6], our system obtained high recall as shown in Figure 7 but at the same time was judged as difficult to understand (Figure 8) and many duplications Figure 9.

Our summary generation strategy, in which the frames included in the summary clip are selected at a certain sampling rate, has advantages and disadvantages. The advantage is it can cover most of the contents of the original video and therefore it explains why the recall of our system is high. The disadvantage is it is difficult to understand since it normally has big jump between two consecutive frames in the summary clip if the new skim ratio is larger than 15 frames. Therefore, one simple possible way to make the summary not only easy to understand but also reduce redundancy is to select three portion of the selected segment (first, middle and last) and to set the skim ratio lower than 15 frames.

4. CONCLUSION

We have introduced a method for generating video summary for BBC rushes. The proposed method uses color moments as image representation and clustering techniques for eliminate redundancy. We also identify challenges in the clustering process that are fragmentation clustering and poor cluster boundary. In the future, we will focus on handling these challenges. One idea is to use agglomerative
clustering for post-processing GreedyRSC clustering results. This combination is promising since GreedyRSC clustering tries to group samples based on shared-neighbor set rather than real distance while agglomerative clustering uses real distance for merging similar groups.

5. ACKNOWLEDGMENTS

We would like to thank anonymous reviewers for their helpful and supportive comments.

6. REFERENCES


