Sequence Alignment for Redundancy Removal in Video Rushes Summarization

Emilie Dumont and Bernard Mérialdo
Institut Eurécom
2229 route des Crêtes
06904 Sophia Antipolis, FRANCE
Emilie.Dumont,Bernard.Merialdo@eurecom.fr

ABSTRACT

In this paper, we describe our approach to the TRECVID 2008 BBC Rushes Summarization task. First, we remove junk frames and dynamically accelerate videos according to their motion activity to maximize the content per time unit. Then, we search identical sequences using a sequence alignment algorithm derived from bio-informatics and we identify and structure scenes in videos, then we select one take per scene. We select the most relevant sequences in order to maximize the content and finally, we compose our summary in an original presentation. The produced summaries have been evaluated in the TRECVID campaign.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Evaluation/methodology

General Terms

Algorithms, Performance

1. INTRODUCTION

Digital video documents are now widely available. Although powerful technologies now exist to create, play, store and transmit those documents, the analysis of the video content is still an open and active research challenge. In this paper, we focus on video summarization: the automatic creation of video summaries is a powerful tool which allows synthesizing the entire content of a video while preserving the most important or most representative sequences. A video summary will enable the viewer to quickly grab the essence of the document and decide if it is useful for its purpose or not.

Over the last number of years, various ideas and techniques have been proposed towards the effective summarization of video contents. Overviews of these techniques appear in [7], [4]. The TRECVID evaluation campaign focuses on the summarization of rushes video. Rushes are the raw material used to produce a video, and their summarization has particular characteristics. Recently, several techniques were developed to solve this task, the most often used techniques are based on a clustering of segments in order to eliminate visual redundancy, as in [3], or [2].

In this paper, we introduce an original approach that includes sequence alignment as a preprocessing step to structure video rushes. The rest of the paper is organized as follows: the next section explains our motivation and approach. In the following sections, we describe the details of our method. Finally, we will present the evaluation results provided by TRECVID.

TVS'08, October, 2008, Vancouver, BC, Canada.

2. GENERAL APPROACH

The system task in rushes summarization is, given a video from the rushes test collection, to automatically create an MPEG-1 summary clip less than or equal to a maximum duration (2 percent of the original duration) that shows the main objects (animate and inanimate) and events in the rushes video to be summarized. The summary should minimize the number of frames used and present the information in ways that maximize the usability of the summary and the speed of objects/event recognition.

Figure 1 provides a schematic view of our approach, the general approach that we propose contains a series of steps to produce the final summary:

- removal of junk frames,
- dynamic acceleration,
- parsing video into scenes and takes,
- selection of representative segments,
- creation of the final video.

In video rushes, a scene is represented by several takes of the same action. The various takes of a given scene are visually very similar, with differences dues to comments from the director, or unexpected events during the recording. During filmmaking, the film editor will first select which take of each scene will be used, to build a rough cut with the best shots. The next step is to create a fine cut by selecting the precise content of each shot and organize the sequence into a seamless story. Trimming, the process of shortening scenes by a few minutes, seconds, or even frames, is done during this phase. Our approach tries to

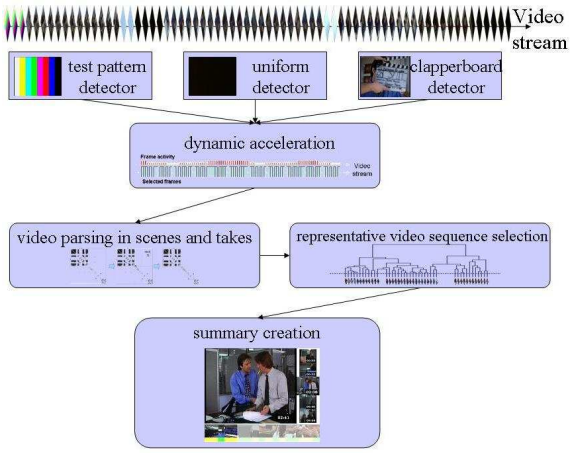

Figure 1: General scheme of our proposed approach

follow this process, with a first analysis of the scenes and takes, then a selection of the adequate video segments.

To detect scenes and takes, we search for repetitive segments using a sequence alignment algorithm. Then, we remove visual redundancy by selecting one take in a scene. Finally, we cluster segments and select a set of relevant segments to be included in the summary. The selected segments are concatenated to compose the summary, which is also decorated with various extra information (timeline, keyframes and timestamps).

3. VIDEO PREPROCESSING

3.1 Junk frame removal

Rushes contain, in particular, a lot of uninteresting sequence of frames for example, test pattern frames, black frames, ... Figure 2 shows example of removed frames.

(a) Test patterns

(b) Uniform color frames

(c) Clapperboards

Figure 2: Example of junk frames

(a) Test patterns: Test patterns are used for calibrating or troubleshooting the downstream signal path. They are generated by test signal generators. They usually have a set of line-up patterns to enable television cameras and receivers to be adjusted to show the picture correctly. So, each company has a unique test pattern.
To detect them, we use a training set of 2452 BBC test pattern frames extracted from the training collection. We compute the mean hue histogram of frames in the training set to build a detector vector T. Frames which have an Euclidean distance with the detector vector T larger than a predefined threshold are removed.

(b) Uniform color frames: By their nature, rushes contain also empty sequences, e.g. black, white, gray, blue sequences ...
To detect them, we compute the entropy of the distribution of color pixels in HSV color space and we remove frames with an entropy lower than a predefined threshold.

(c) Clapperboards: In videotape production, a clapperboard is a device used to synchronize picture and sound; additionally the clapperboard is used to designate and mark particular scenes and takes recorded during a production. Clapperboards are useful for editing, but they should not be kept in the summary since they contain no video object or event.
To detect them, we use a training set of 9972 frames labeled as clapperboard, and 15501 frames labeled as no-clapperboard. For each frame, we compute a feature vector based on the HSV histogram of the central region of the frame, then we train a SVM classifier. This classifier is then used as a detector for new clapperboard frames.

3.2 Dynamic acceleration

Some video scenes are very long and contain very little visual movement, while others will be short and contain a lot of action. In order to maximize the visual content that is provided in the final summary, we use dynamic acceleration to show a video sequence during a duration related to its motion activity, show figure 3.

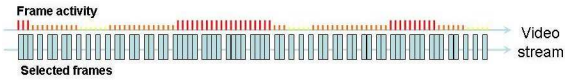

Figure 3: Acceleration according to video activity

First, we fix the mean acceleration of the entire video $ACCmean = 3$, and the maximum acceleration allowed for a sequence $ACCmax = 5$. For each frame f, we compute the motion activity $act(f)$. Then, for a video segment v with F frames, we sequentially select frames with an interval of:

$$jump(v) = min(ACCmax, \sum_{f \in v} act(f)/F * ACCmean)$$

Under all of the 39 summaries, we removed 0.81% of frames; e.g after junk frame removal and dynamic acceleration, a video has a average duration of 7111 frames.

4. VIDEO PARSING

Rushes video contains a lot of redundancy, we remove visual redundancy by detecting repetitive segments and by parsing video in scenes and takes. A complete study and explication of this method is presented in [1].

4.1 Sequence alignment

The Smith-Waterman algorithm [6] is used to compute the edit distance between two sequences (DNA, or protein) using a dynamic programming approach. This is done by creating a scoring matrix with cells which indicate the cost to change a sub-sequence of one to the sub-sequences of the other. By building on the edit distances of the sub-sequences, Smith-Waterman provides an efficient way to compare sequences by comparing segments of all possible lengths and optimizing the similarity measure. We propose to adapt this algorithm for our problem like in figure 4 with these constraints: two aligned sub-sequences can not contain the same segment, two segments can be aligned only once, and two aligned sub-sequences must have a minimal length fixed to 3 seconds.

Given: A video sequence S is defined as a list of m one-second segments: $S = s_1 s_2 ... s_m$

- Hierarchical clustering: $S_l = c_1 c_2 ... c_m$ where c_i is the cluster of segment s_i of the clustering level l.
- $l = 0$.
- Compute $(m+1)*(m+1)$ normalized scoring matrix $\bar{M}_l$ where $\bar{M}_l[i][j]$ represents the cost of the sub-sequence alignment ending with segments s_i and s_j.
- Iteratively: find the best sub-sequence alignment, i.e. the maximal value $\bar{M}_l$.
  - If $\bar{M}_l > threshold$, we store this alignment and we update the scoring matrix.
  - Else $l = l + 1$ and we update the scoring matrix.

Output: A list of aligned sub-sequences.

Figure 4: Video Sequence Alignment algorithm

4.1.1 Hierarchical clustering

In order to detect the visual redundancy, we partition video into one-second segment, i.e. into 25 frames. Each one-second segment is represented by a HSV histogram of those frames. The algorithm starts with as many clusters as there are one-second segments, then at each step of the clustering, the number of clusters is reduced by one by merging the closest two clusters, until all segments are finally in the same cluster. The distance between two one-second segments is computed as the Euclidean distance, and the distance between two clusters is the average distance across all possible pairs of segments of each cluster.

4.1.2 Scoring matrix

The video sequence is defined as a list of one-second segments $S_l = c_1 c_2 ... c_m$ where c_i corresponds to the cluster of the segment $s8i$ at the clustering level l. The $(m+1)*(m+1)$ scoring matrix $M_l[i][j]$, for clarity we denote $M[i][j]$, is computed as:

$$M[i][0] = 0, \quad M[0][i] = 0 \text{ and } M[i][i] = 0 \ \ \forall i \in 0, ..., n$$

$$M[i][j] = max \begin{pmatrix} 0 \\ \begin{cases} M[i-1][j-1] + cos(\vec{i}, \vec{j}) + 1 & \text{if } c_i = c_j \\ M[i-1][j-1] + cos(\vec{i}, \vec{j}) - 2 & \text{if } c_i \neq c_j \end{cases} \\ M[i][j-1] + cos(\vec{i}, \vec{j}) - 3 \\ M[i-1][j] + cos(\vec{i}, \vec{j}) - 3 \end{pmatrix}$$

where $cos(\vec{i}, \vec{j})$ is the cosine between the HSV histogram of segment s_i, and the HSV histogram of segment s_j.
When $M[i][j] > 0$, the alignment of two sub-sequences ending in position i and j can be found by recursively adding the antecedent (u, v) which realizes the maximum of M starting from $M[i][j]$, until a zero value is found for $M[u][v]$. The number of antecedents is the length of the alignment $length(i, j)$. The matrix M is normalized by the $\bar{M}[i][j] = \frac{M[i][j]}{length(i,j)}$.

Figure 5 shows an example of a scoring matrix. The best value is equals to 1.98, but the length corresponding is smaller than 3. So, we select $\bar{M}[s6][s3] = 1.97$, and then we align $s1, s2, s3$ with $s4, s5, s6$ by a trace-backing.

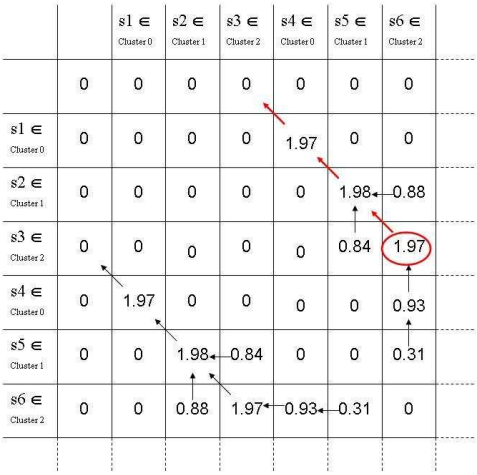

Figure 5: Example of scoring matrix

4.2 Scene detection and take selection

To parse video in scenes and takes, we use found alignment. A video sequence is defined as a list of frames $V = f1...f_n$. We construct a $n * n$ alignment matrix A where $A[f_i][f_j]$ is equals to the number of alignment found when the segment containing f_i and the segment containing f_j were aligned. If these two segments were not aligned, $A[f_i][f_j]$ is equal to the total number of found alignments plus one.
We remove false alignments by the following method: for each frame $f_i \in F$, we compute $rect(f_i)$ by:

$$rect(f_i) = \frac{\sum_{\forall f1 \in [first, f_i]} \sum_{\forall f2 \in [last, F]} A[f1][f2]}{\sum_{\forall f1 \in [first, f_i]} \sum_{\forall f2 \in [f_i, last]} 1}$$

At the beginning, we fix $first = 0 \wedge last = F$, we search the maximal value of $rect[f_i]$, and if this value is greater than a

threshold, we remove alignments of f_i. And recursively, we restart with $first = 0 \wedge last = f_i$, and $first = f_i \wedge last = F$. So, we obtain a new alignment matrix with scene boundaries, see figure 6.

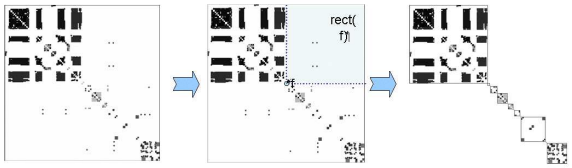

Figure 6: Example of alignment matrix

Now, the idea is to select only one take per scene in order to remove visual redundancy. We would like to select the most complete take, so the longest sequence without repetition. Iteratively, we compute, for all frames $f \in V$, the number of successive frames without redundancy and we select the longest sequence, and remove all frames aligned with them. Table 1 shows statistics of sequence alignments.

	Test videos	Training videos
Number of alignments per video	303.575	394
Number of alignments per frame	2.763	4.181
Percentage of aligned frames	0.713	0.821
Percentage of removed frames	0.372	0.232

Table 1: Statistics of alignment sequence

5. SUMMARY CREATION

In previous sections, we explained our method to remove junk frames, and visual redundancy. So, now we have only interesting frames, and we want to select a set of these frames to compose the final summary. We propose to make a selection of the most relevant sequences whose content overlaps as little as possible, a sequence is a set of successive frames.

Segment selection: Frames are removed, video was decomposed in sequences where a sequence is a set of successive frames. Long sequences are decomposed in several sequences. We chose to fix the maximal length of a sequence equals to 2 seconds, e.g 50 frames. Sequences are clustered by an agglomerative hierarchical clustering. Each iteration of the algorithm provides a different clustering of sequences. The idea is to choose the clustering level which best close to a duration of 2% of the original video. And finally, we select a set of sequences which covers all events by selecting the medoid of each cluster, see figure 7.

Summary presentation: For the final presentation, we decided to include not only the selected frames, but also additional information elements to provide the user with a more global view of the summary. The selected frames are reduced to 80% and placed in the upper left corner of the final frame. On the right side, we provided a list of icons for the keyframes of the selected segments, the icon for the

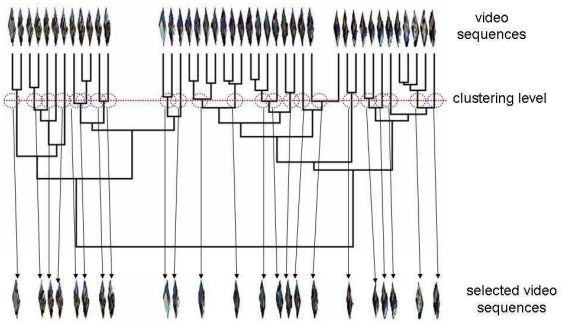

Figure 7: Scheme of segment selection

current segment being slightly enlarged. By discussing with professional video editors, we got the information that the time information for a frame was crucial if the raw material had to be accessed later, therefore we inserted a time information in each frame and icon. On the bottom part, we constructed a visual timeline by concatenating the center vertical line from all the selected frames. This timeline is initially dimmed, and progressively discovered as the summary is being played. Figure 8 is an illustration of the resulting summary frame.

At the end of the summary, within the 2% limit, we insert

Figure 8: Example of summary presentation

frames showing an overview of all keyframes, as illustrated in figure 9.

Figure 9: Example of montage of keyframes

6. EVALUATION

A complete evaluation has been done in the TRECVID campaign [5]. This evaluation is based on several measures, in particular:

- IN- Fraction of inclusions found in the summary
- JU- Summary contained lots of junk
- RE- Summary contained lots of duplicate video
- TE- Summary had a pleasant tempo/rhythm

JU, RE and TE measures vary from 1 to 5 where 1 seems a bad quality and 5 a good quality. We propose to compute a normalized average of these criteria by:

$$Mean = \frac{IN + (JU-1)/4 + (RE-1)/4 + (TE-1)/4}{4}$$

Table 2 shows the evaluation results.

Criteria	IN	JU	RE	TE	Mean
Baseline	0.83	2.66	2.02	1.44	0.36
Min	0.07	2.52	2.02	1.44	0.35
Max	0.83	3.64	3.99	3.38	0.49
Mean	0.44	3.15	3.27	2.72	0.42
Eurecom	0.39	2.62	3.50	2.75	0.39

Table 2: Results on TRECVID 2008 summarization task

The baseline algorithm simply presents the entire video at 50x normal speed. So, the fraction of inclusions found in the summary is very large, it is not equal to 1 because some very short topics appear in only a few frames and may be lost during acceleration. Still, the baseline presents the best results for IN, and quasi worst results for other criteria. In conclusion, the only interest of this baseline is to provide an upper bound for IN, and a lower bound for other criteria.
We can see that our JU indicator shows a bad value, worse than the baseline, although the baseline does not remove any junk sequence. The explanation is that, for clapperboards, we chose to have a minimal number of false positive, so that, due to the visual variability of the clapperboard sequences, some of them were kept in the final summaries. In contrast, clapperboard sequences are not very long, so the baseline which selects only one frame out of 50 represented those sequences by only a few frames, making them virtually invisible. For the test pattern frames, we checked that our model removed all test pattern frames except only 2, this representing a total of 2 seconds over all summaries. Also, our uniform frame detector left 5 seconds of junk frames over all summaries. This bad result about JU can also be due to the presence of sequences like in figure 10 where a hand covers part of the screen, providing visual movement without interesting content. Keeping too many junk sequences prevents from showing enough interesting content in the summaries, and reduces the IN indicator. Other results on RE and TE show that our approach is valid and that the sequence alignment provides useful structuring.

7. CONCLUSION

This paper presented the solution for the rushes summarization task of TRECVID 2008 developed by Eurécom.

Figure 10: Junk frames

Our system is composed on several steps: first, we removed junk frames like test pattern, clapperboard and uniform color frames. Then we dynamically accelerated the video. Repetitive sequences are removed and we selected the best sequences through clustering. Finally, we organized the summary with an original presentation of relevant information.
We compared our results with the baseline and others. Results show good and bad aspects of our system: duplication are removed successfully, the summary is pleasant to watch but during the sequence selection step, the system keeps too many junk sequences.

8. ACKNOWLEDGMENTS

The research leading to this paper was supported by the Institut Eurécom and by the European Commission under contract FP6-027026, Knowledge Space of semantic inference for automatic annotation and retrieval of multimedia content - K-Space.
BBC 2008 Rushes video is copyrighted. The BBC 2008 Rushes video used in this work is provided for research purposes by the BBC through the TREC Information Retrieval Research Collection.

9. REFERENCES

[1] E. Dumont and B. Mérialdo. Rushes video parsing using video sequence alignment. In *Submitted to MMM 2009, the 15th International MultiMedia Modeling Conference, 7-9 January 2009, Sophia-Antipolis, France*.

[2] E. Dumont and B. Mérialdo. Split-screen dynamically accelerated video summaries. In *MM 2007, 15th international ACM conference on multimedia, September 24-29, 2007, Augsburg, Germany*, Sep 2007.

[3] D.-D. Le and S. Satoh. National institute of informatics, japan at trecvid 2007: Bbc rushes summarization. In *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, 2007.

[4] A. Money and H. Agius. Video summarisation: A conceptual framework and survey of the state of the art. In *Journal of Visual Communication and Image Representation*, 2007.

[5] P. Over, A. F. Smeaton, and G. Awad. The TRECVid 2008 BBC rushes summarization evaluation. In *TVS '08: Proceedings of the International Workshop on TRECVID Video Summarization*, 2008.

[6] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 1981.

[7] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2007.