

Video Redundancy Detection In Rushes Collection

Reede Ren
University of Glasgow
17 Lilybank Gardens
Glasgow, UK
reede@dcs.gla.ac.uk

P. Punitha
University of Glasgow
17 Lilybank Gardens
Glasgow, UK
punitha@dcs.gla.ac.uk

Joemon Jose
University of Glasgow
17 Lilybank Gardens
Glasgow, UK
jj@dcs.gla.ac.uk

ABSTRACT

The rushes is a collection of raw material videos. There are various redundancies, such as rainbow screen, clipboard shot, white/black view, and unnecessary re-take. This paper develops a set of solutions to remove these video redundancies as well as an effective system for video summarisation. We regard manual editing effects, e.g. clipboard shots, as differentiators in the visual language. A rushes video is therefore divided into a group of subsequences, each of which stands for a re-take instance. A graph matching algorithm is proposed to estimate the similarity between re-takes and suggests the best instance for content presentation. The experiments on the Rushes 2008 collection show that a video can be shortened to 4%-16% of the original size by redundancy detection. This significantly reduces the complexity in content selection and leads to an effective and efficient video summarisation system.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Video Summarisation

General Terms

video summarisation

Keywords

edit effect detection, re-take detection, attention-based content selection, rushes collection

1. INTRODUCTION

As a raw material collection, the rushes is made up by many re-takes of similar scenes. This indicates that only a very small ratio of a rushes video will appear in a final edited version [1]. Rushes summarisation is therefore a removal of redundant or unnecessary contents in many aspects. In addition, we find that it can cut a video to 4%-16% of the

original size by removing redundant contents in the Rushes 2008 collection. This significantly decreases the complexity in content topic selection and eases the later video abridgement.

The redundancy in the rushes could be roughly categorised into two classes, manual editing effects and unnecessary re-takes. Manual editing effects are short and usually meaningless video segments, such as clap shots, rainbow and black/white screens. These effects are artificially created to facilitate later processing, i.e. marking boundaries of a re-take. To some extents, they are *commas* in the visual language, which separate different re-take instances. Unnecessary re-takes are multiple records of a scene to: (1) provide different camera viewpoints; (2) correct actor's mistakes; or (3) remove technical failures. This indicates that re-take instances are similar but not identical. There are occasional omissions and extra insertions of visual shots. Moreover, according to the requirement of video summarisation, only one instance of a re-take could be accepted in the final summary. This leads to a research problem, how to identify the best presentation of video contents among multiple re-take instances. In short, the challenges of redundancy removal in the rushes collection are to: (1) detect manual editing effects; (2) identify instances of a re-take; and (3) decide the *best* re-take instance for video content representation.

The remainder of this paper is organised as follows. Section 2 describes the framework of our video summarisation system. Section 3 states a clustering algorithm to remove direct repetitions. Techniques to detect editing effects are presented in Section 4, involving rainbow screens, clipboard shots and meaningless views. Section 5 proposes a solution to discriminate re-take instances. A measurement of visual similarity is defined to compare visual key frames (Section 5.1). A graph matching algorithm is developed to match two similar but not identical re-take instances and therefore decides an optimised content representation (Section 5.2). Based on the work [6], an attention-based content topic selection is introduced by Section 6. Discussion and future work are found in Section 7.

2. SYSTEM FRAMEWORK

Figure 1 shows the framework of our video summarisation system. Comparing with the system of attention-based video abstraction [6], two new modules are improved, replay speed adjustment and content redundancy detection. The module of replay speed adjustment works in the stage

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

TVS'08, October 31, 2008, Vancouver, British Columbia, Canada.

Copyright 2008 ACM 978-1-60558-309-9/08/10 ...\$5.00.

of summary composition, which increases frame rate to reduce summary size. We regard this module as a safe guard to ensure the final summary video will not exceed the 2% limitation. However, this function is rarely activated due to the effectiveness of redundancy identification. The module of content redundancy detection identifies and removes editing effects, repetitive shots and unnecessary re-takes. Three sub-components are therefore involved, direct repetitive removal(Section 3), manual editing detection (Section 4), and re-take discrimination (Section 5).

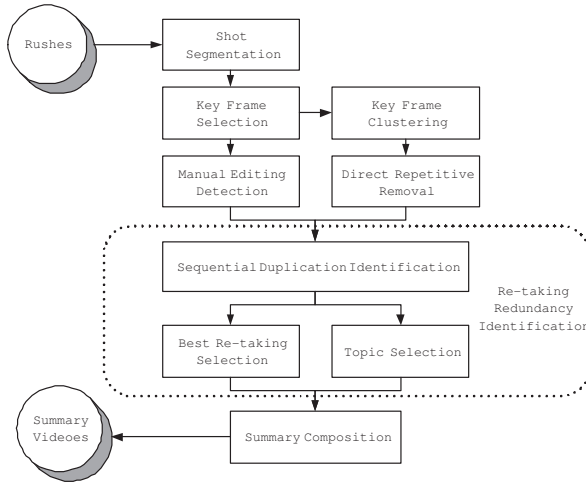


Figure 1: System Framework

The entire process of video summarisation is described as follows. A two-threshold algorithm [4] is employed to segment shots. We favour low thresholds to ensure the visual similarity inside shots. From each shot, a key frame is extracted and these key frames are clustered by visual similarity. Adjacent shots with key frames of the same cluster will be removed as direct repetitions. This step alleviates the over-cutting problem in shot segmentation and decreases computational complexity in the sequential similarity measurement. Meanwhile, a group of salient features, such as average local motion, shot boundary frequency and colour moment, are computed to estimate video attention [6]. Effects of manual editing are identified and used to divide a long video into a group of segments. We suppose each video segment denotes a possible re-take instance. The module of retaking discrimination computes the sequential similarity between video segments and therefore identifies different instances of a re-take. The module of topic selection chooses the most interesting moments. If the overall size of selected video clips exceeds the given 2% duration limitation, the module of replay speed adjustment will increase frame rate to shorten a video summary.

3. SHOT CLUSTERING

Principal component analysis (PCA) captures the most relevant features to use in classifying a group of objects to be recognised. PCA is a linear method for data feature extraction. It is a mathematical technique used to analyse correlated random variables to reduce the dimensionality of a data set. This reduction is achieved by selecting the first

few principal components. The mathematical background of PCA lies in eigen analysis. Thus to achieve good recognition rate, for all N , key frames representing the N shots of a rushes video, $N \times N$ eigen distance matrix, is computed, where each cell entry corresponds to the pairwise eigen distance of the respective key frames. The eigen distance matrix is then used for clustering shots. The clustering algorithm always tries to find the best fit for a fixed number of clusters. Fixing up a static value for the number of clusters, irrespective of the video and the number of shots detected in the video, is not meaningful. This is because, either the number of clusters might be wrong, or the clusters might not correspond to that of the data. There are two main approaches to determine the appropriate number of clusters, compatible cluster merging and validity measures. We use the validity measure to find the appropriate number of clusters for each video. Depending on the number of shots detected for each rushes video, minimum and maximum number of expected clusters is calculated. This minimum and maximum number of clusters reflects to the percentage of coverage of a rushes video. With this initial set up and a combination of the scalar validity measures, separation index, Alternative Dunn's index [2] and Xie and Beni's index [8] the most appropriate and optimal number of clusters for each video is found. K-means clustering is then used to categorise the shots into the most optimal number of clusters.

4. EDITING EFFECT DETECTION

Three types of editing effects are widely observed in the rushes collection: rainbow screen, clipboard shot and meaningless low quality view, i.e. black/white screen. These editing effects should be removed because they contribute few video contents. We suppose these effects play the role of differentiator in the video composition. Related algorithms and solutions for the identification of editing effects are discussed in the following sections.

4.1 Rainbow Screen

Rainbow screen is the colour bar which mostly appears at the beginning of a video, although some instances are occasionally found in the middle of video documents as well.

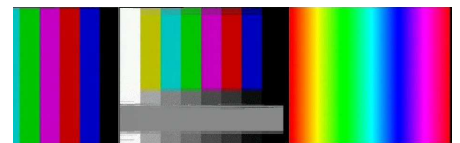


Figure 2: Rainbow Samples

Figure 2 shows some examples of rainbow screens in the Rushes 2007 collection. A significant character of rainbow screens is the distribution similarity among different colour components. Colour histograms of R, G, B colour channels are almost the same (Figure 3). We therefore compute insertion difference between $R - G$ and $R - B$ histograms for rainbow discrimination. 41 rainbow screens are collected from the Rushes 2007 collection to train a two-class Gaussian mixed model(GMM). The recall of rainbow detection in

the Rushes 2007 collection is 100% with the precision above 87%. In addition, most false detections are various meaningless views, such as blurred white/black screen. This is acceptable and even welcomed in editing effect detection.

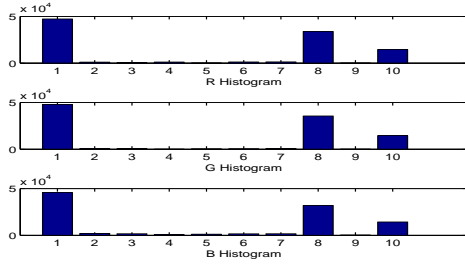


Figure 3: Colour Histograms of A Rainbow

4.2 Clipboard Shot

We categorise clipboard shots into two groups, small board and large board, according to the appearance of clipboards in visual frames. In the case of small board, a clipboard is displayed entirely and occurs less than one half area of a visual frame (Figure 4). We manually cut more than 300 video objects of small clipboards from the Rushes 2007/8 collection and normalise them to the size of 20×15 pixels. In addition, 3,000 visual regions of the same size are randomly cut as negative examples. A support vector machine detector (SVM) is therefore trained as clipboard detector. A four-layer pyramid is created for every key frame and scanned by the SVM detector from coarse to high resolutions. If a clipboard is found, this scan process will stop; the visual frame is labelled as with clipboard and the shot is recognised as clipboard shot.



Figure 4: Small Clipboard Examples

In the case of large board, only a part of a clipboard is shown in a key frame, which occupies most visual area (Fig 5). We develop a direct pathway to deal with this case. Three low-level features are computed in a 3×3 region map, including black ratio, white ratio, and block variance. In addition, we transform a key frame into 12 gray scales [0, 11]. A pixel with gray intensity zero will be regarded as black and that with 11 as white. Black and white ratio is the rate of black/white pixels in the given region. Block variance is the standard deviation of gray intensity. Therefore, a 27-dimension feature vector is created. We collect 216 large clipboard examples and more than 600 non-clipboard visual frames. A support vector machine is trained for large clipboard detection.

To improve the precision of clipboard detection, we develop a post-processing step of dynamic programming [7]. This step



Figure 5: Large Clipboard Examples

smooths the label sequence of clipboard vs non-clipboard. The transmission probability is set as 0.2 for different states, such as the change from *with* to *without* clipboard; and 0.8 for similar states, e.g. from *with* to *with* clipboard (Figure 6). In addition, we sample rushes videos at 1/5 for clipboard detection. Most clipboard shots includes more than 25 frames (longer than 1 second). The dynamic length is therefore set to six. This indicates that we only accept a clipboard shot longer than 30 frames.

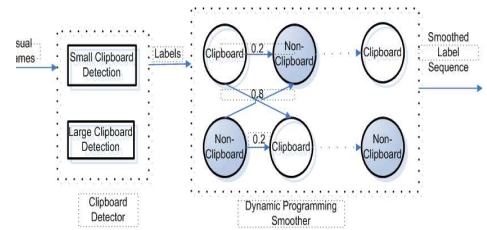


Figure 6: Clipboard Shot Detection

The clipboard detection is a complex task. The average precision is 77.2% for small clipboard and 69.5% for large clipboards in Rushes 2007 collection. In addition, the post-process of dynamic programming could improve precision about 7% for small clipboard and 12% for large clipboard.

4.3 Meaningless View

Meaningless views are low quality visual frames due to technical failures, e.g. over-exposure and black/white screen. Many variations are observed in the rushes collection 2007 and some examples are shown in Figure 7. A common character of meaningless views are the lack of image details. Therefore, we employ a measurement of image quality, local harmonic activity map [3], for the discrimination of meaningless views. A threshold is set to pick out visual frames with low local harmonic activity as meaningless views. In addition, such a threshold relies on the production quality of raw videos as well as the definition of meaningless views.

5. RE-TAKE DISCRIMINATION

By means of editing effect detection, we identify the *com-mas* in the visual language as well as divide a long video stream into segments. In addition, each video segment usually consists of several shots. We assume such a segment standing for a re-take instance. Therefore, the re-take discrimination is to cluster these segments into a few groups (re-takes), according to the similarity in content presenta-

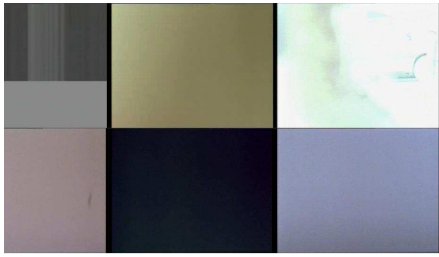


Figure 7: Black/White Samples

tion. Two aspects are involved: (1) the visual similarity between related key frames; and (2) the sequential similarity between video segments. We value the second aspect because such a sequential similarity is closely associated with the presentation structure of video contents.

5.1 Key Frame Similarity Estimation

We extract one key frame from every shot in a video segment. These key frames are divided into 3×3 region maps. For each region, a colour histogram is computed and a weight is assigned (Figure 8). The similarity between key frames is estimated as a weighted sum of intersection distances between region based colour histograms. Additionally, the intersection distance will be one if two histograms are the same and zero for totally different. Such a measurement will be normalised throughout a rushes video.

2	1	2
1	2	1
2	1	2

Figure 8: Weights of Region Map

A two-class Gaussian mixed model is learnt for each video, which simulates the distribution of frame similarity. The threshold for similar vs non-similar decision is therefore computed as the average of Gaussian model means.

An optional step is carried out to remove short video segments. For a segment with less than five shots, we test whether all key frames could be found in any other long video segments. If a high key frame coverage is achieved, i.e. 80%, such a short video segment will be ignored because it can be covered by another instance. This step reduces the number of video segments and alleviates the problem of multiple overlaying in the following computation of sequential similarity.

5.2 Sequential Matching

The module of sequential matching checks whether a video segment could be replaced or not. Such a process is similar to the comparison between two strings, if we regard a video segment as an ordered sequence of key frames. However, there are unpredictable omissions and insertions of video

shots. This indicates: (1) some key frames may be lost; (2) a random number of characters may be inserted. Therefore, the search of a possible match is equivalent to finding a bi-directional pathway in a $N \times M$ directional graph, where N and M are the length of two re-take instances, respectively. The computational complexity will be $O(n^{NM})$ for a whole search. In addition, a sliding window of a given length may be ineffective for sequential matching. This is because the duration of video segments is contingent with the precision of editing effect detection. Some video segments may be twice longer than others. Moreover, there is absent a common video structure in the rushes collection. It is difficult to train an efficient Markov model to adopt these omissions and insertions. Nevertheless, we also think about the employment of key video objects. For example, an actor can effectively *nail* all instances of a re-take. But this will introduce extra complexity, such as face discrimination and object segmentation, as well as the semantic uncertainty: "what is the key object in the video segment?".

A bi-directional searching algorithm (Algorithm 1) is developed to find possible matches between two video segments. This algorithm decides (1) whether all key frames of a video segment could find similar frames in another video segment; and (2) whether the appearance order of key frames between two video segments are similar. Three steps are developed. Firstly, we compute the frame similarity of all key frames between video segments. A candidate collection is created for each key frame, which lists frame numbers of the top N most similar key frames in the other video segment. Secondly, we search these candidate collections in the order of key frame appearance. In each collection, we suppose a possible match is the candidate with the smallest frame number but larger than any others already in the matching sequence. We will label a key frame unique, if such a match could not be found. When this searching experiences all collections, a possible pathway is found from one video segment to the other. A similar searching process is carried out for the other video segment as well. Thirdly, we compare the number of unique key frames and the length of possible matching sequence. If the unique key frame number is larger than one third of matching sequence length, these video segments will be supposed to be non-relevant, otherwise relevant. The relevant video segment with more unique key frames will be held as a better re-take instance.

In addition, this algorithm compares two video segments once, which may be inefficient when there are too many video segments. A possible solution is to limit the comparison among adjacent video segments with similar length. This could significantly reduce the times of comparison.

6. TOPIC SELECTION

The module of topic selection is to abridge selected video segments. We estimate the distribution of attention intensity among segments [6]. In this system, we increase the weight of two aspects in attention estimation: (1) the duration of a re-take instance; and (2) audio-visual variations, e.g. strong motion. On one hand, a long re-take instance indicates that ad-hoc directors are satisfied with this sequence when production. On the other hand, a complex visual or audio situation usually provides rich information about video contents. Clips with the highest attention inten-

Data: A pair of video segments $\{S_1, S_2\}$, each of which consists of a key frame group

$K_n = \{k_{in} | k_{in} \in S_n, i = 1 \dots N_n, n \in \{1, 2\}\}$, Q the candidate collection size

Result: The number of unique key frames U_1 and the matching sequence M_{12} ;

```

foreach  $k_i1 \in K_1$  do
  create a candidate collection  $QK_{i1}$ ;
  calculate visual similarity from  $k_i1$  to elements in
   $K_2$ ;
  insert the appearance order of the top  $Q^{th}$  element
  in  $K_2$  into  $QK_{i1}$ ;
end
 $U_1=0$ ;
for  $i = 1$  to  $N_1$  do
  if  $M_{12}$  is empty then
    insert  $M_{12}$  the minimum element of  $QK_{i1}$ ;
    continue;
  end
  while  $QK_{i1}$  is not empty do
    find the minimum element  $q$  of  $QK_{i1}$ ;
    if  $q$  larger than all elements in  $M_{12}$  then
      add  $q$  to  $M_{12}$ ;
      break;
    end
    remove  $q$  from  $QK_{i1}$ ;
  end
  if  $QK_{i1}$  is empty then
     $U_1++$ ;
  end
end

```

Algorithm 1: Searching Possible Match Sequence

sity are extracted from video segments for summarisation.

7. DISCUSSION AND FUTURE WORK

We present our video summarisation system for the Rushes 2008 collection. The major contribution is to develop an solution to detect various video redundancies, including rainbow screen, clipboard shot, meaningless low quality view and unnecessary re-take. These instances cover most content redundancies in the rushes collection.

We propose that editing effects are differentials, i.e. *commas* in the visual language. This leads to an efficient graph matching algorithm to estimate the sequential similarity between two re-take instances. A solution is therefore developed to find all instances of a re-take as well as decide the best instance for video content representation. The evaluation report [5] shows that our system provides a good balance between the summary duration and content topic coverage. Moreover, a short judgement period indicates these summaries are easy to be understood. This shows the effectiveness of our attention-based topic selection strategy.

Re-take detection is a complex task. A few challenges can seriously decrease the algorithmic performance of matching sequence search (Algorithm 1): (1) too many short video segments; (2) incomplete video segments; (3) key frame extraction. A short video segment is mostly caused by some technique failure in the production. This indicates some unique frames are introduced and results in a possible low

key frame coverage, given the short segment duration. In the case of incomplete video segment, a re-take instance actually consists of several video segments. The sequential comparison between adjacent video segments is therefore ineffective. To deal with incomplete video segments, a cross-segment search is necessary, which merges these segments to find a better content presentation. However, we leave this for future work. The selection of key frames plays a prominent role in this video summarisation system. A good collection of key frames can significantly improve the performance of re-take detection, vice versa. However, it is hard to justify what kind of key frame is ideal for video summarisation.

We discard audio information in this work, because audio clips can hardly be incorporated into a summary video. Additionally, an audio stream in the rushes collection is full of background noise and meaningless speeches, such as "ready...camera". It may incur extra complexity in content matching. However, the detection of manual editing effects can partially remove these helpless audio clips. This will lead to a possible pathway for the employment of audio information.

8. ACKNOWLEDGMENTS

The research leading to this paper was supported by the European Commission under contracts FP6-045032 (Semedia).

9. REFERENCES

- [1] W. Bailer, F. Lee, and G. Thallinger. Detecting and clustering multiple takes of one scene. In *MMM*, pages 80–89, 2008.
- [2] A. Bensaid, L. Hall, J. Bezdek, L. Clarke, M. Silbiger, J. Arrington, and R. Murtagh. Validity-guided (re)clustering with applications to image segmentation. *IEEE Transactions on Fuzzy Systems*, 4(3):112–123, 1996.
- [3] I. P. Gunawan and M. Ghanbari. Reduced-reference picture quality estimation by using local harmonic amplitude information. In *in Proc. London Communications Symposium*, pages 137–140, University College London, UK, September 2003.
- [4] R. Lienhart. Comparisons of automatic shot boundary detection algorithms. In *Proc of SPIE Storage and Retrieval for Image and Video Database*, volume 3656, pages 290–301, 1999.
- [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*, pages 1–20, New York, NY, USA, 2008. ACM.
- [6] R. Ren, P. Punitha, J. M. Jose, and J. Urban. Attention-based video summarisation in rushes collection. In *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pages 89–93, New York, NY, USA, 2007. ACM.
- [7] L. Xie, L. Kennedy, S.-F. Chang, A. Divakaran, H. Sun, and C.-Y. Lin. Discovering meaningful multimedia patterns with audio-visual concepts and associated text. In *ICIP*, Singapore, Oct 2004.
- [8] X. L. Xie and G. A. Beni. Validity measure for fuzzy clustering. *IEEE Trans. PAMI*, 3(8):841–846, 1991.