Automatic Summarization of Rushes Video using Bipartite Graphs

L. Bai
CLARITY: Centre for Sensor Web Technologies,
Dublin City University,
Glasnevin, Dublin 9, Ireland.
Present address: School of Information System & Management,
National Univ. of Defense Technology,
ChangSha, 410073, P.R. China.

Y. Hu and S. Lao
School of Information System & Management,
National Univ. of Defense Technology,
ChangSha, 410073, P.R. China.

A.F. Smeaton and N.E. O'Connor
CLARITY: Centre for Sensor Web Technologies,
Dublin City University,
Glasnevin, Dublin 9, Ireland.

October 3, 2009

Abstract

In this paper[1] we present a new approach for automatic summarization of rushes, or unstructured video. Our approach is composed of three major steps. First, based on shot and sub-shot segmentations, we filter sub-shots with low information content not likely to be useful in a summary. Second, a method using maximal matching in a bipartite graph is adapted to measure similarity between the remaining shots and to minimize inter-shot redundancy by removing repetitive retake shots common in rushes video. Finally, the presence of faces and motion intensity are characterised in each sub-shot. A measure of how representative the sub-shot is in the context of the overall video is then proposed. Video summaries composed of keyframe slideshows are then generated. In order to evaluate the effectiveness of this approach we re-run the evaluation carried out

[1]The original publication is available at www.springerlink.com. DOI: 10.1007/s11042-009-0398-1

by TRECVid, using the same dataset and evaluation metrics used in the TRECVid video summarization task in 2007 but with our own assessors. Results show that our approach leads to a significant improvement on our own work in terms of the fraction of the TRECVid summary ground truth included and is competitive with the best of other approaches in TRECVid 2007.

1 Introduction

Video summarization has recently become an active and popular research field, partly because of the growth in video sharing on the internet, and the fact that benchmark data and metrics for formal evaluation are now available through TRECVid [Over et al., 2007], [Over et al., 2008]. Video summaries provide a condensed version of a full-length video and should include the most important content from within the original video. Summaries can be used in a range of different media applications including browsing and search, TV program editing, and so on. A variety of approaches have been proposed for automatic summarisation based on redundancy detection [Byrne et al., 2007], frame clustering [Ferman and Tekalp, 2003], speech transcripts [Taskiran et al., 2006], and multiple information streams [Ma et al., 2002].

In 2007 and 2008, the National Institute of Standards and Technology (NIST) in Gaithersburg, Md. USA, coordinated an evaluation of automatic video summarization for rushes video, i.e. extra video, B-rolls footage, etc. This took place as part of a larger video benchmarking activity with worldwide participation which has been running since 2001, known as TRECVid [Smeaton et al., 2006]. The achievements of the dozens of participants in the TRECVid video summarisation task were presented at two workshops held in conjunction with the ACM Multimedia Conferences in Augsburg, Germany (2007) and in Vancouver, Canada (2008). The overall video summarization task, the data used, the evaluation metrics, etc., are described in two overview papers from those workshops [Over et al., 2007], [Over et al., 2008] and some of these details such as the data used, are described later in this paper. Importantly, in the TRECVid guidelines for rushes summarization, several criteria have been used for evaluating the automatically generated summaries, including the fraction of ground truth objects and ground truth events included by the summary (IN), the ease of understanding the summary (EA), the time needed for subjective judgment of the summary by an assessor (TT, VT), and the compactness of the summary (DU, XD).

For our participation in this task in 2007, we used a keyframe-based approach [Byrne et al., 2007] but it did not perform as well as expected, especially for the IN and EA criteria. The inclusion results (IN) placed our approach (mean: 0.38; median: 0.38; best: 0.70) among the 5 lowest scoring participants. Our low EA scores (mean: 2.53; median: 2.67) placed us second worst out of 25 participants. This poor performance encouraged us to undertake detailed failure analysis and motivated us to re-analyze the characteristics of rushes videos and

of how summaries for rushes should be generated.

There are two types of redundant information in rushes video. The first is content such as clapperboards, color bars, monochromatic shots and very short shots. This content is not related to the main content of the video and so is not of value in a summary. The second type of redundant content is repetition of some shots with near-identical material appearing in the second and subsequent shots. During program production, the same shot is often taken many times because an actor may fluff his/her lines or a director may want a second or subsequent "take" in case there are errors which would only become apparent in the post-production stage. A director may even slightly change the content of the original video by adding or deleting lines or may change the angle of the camera, for example. All these near-duplicates arise from the creative processes involved in filming but nonetheless they do represent repeated and thus redundant material. For summarization purposes, such re-taken shots should be detected and only one of them kept, removing others from the final summary.

Our approach described in this paper is an enhancement on what we produced for TRECVid in 2007 and focuses on representative frames selection, useless content detection and removal, re-take detection and content filtering and ranking among the remaining selected shots. In order to select representative frames which represent video content with as much precision as possible, we calculate the difference between consecutive frames based on color features at the pixel level in each shot and we use a geometrical approach to select representative frames. Although we don't explicitly segment sub-shots, our method for keyframe selection guarantees that representative frames in each sub-shot are selected as both the sum of differences and length of the shot are considered. SVM classifiers are trained based on the TRECVid development data to detect color bars and monochromatic frames which are regarded as having no value in a video summary. Clapperboard clips are removed by an existing method for Near-Duplicate Keyframe (NDK) detection. After filtering this non content-bearing material, we reduce inter-shot redundancy by removing repeated retake-shots. Maximal matching based on the Hungarian algorithm is then adopted to measure the similarity between retake shots at the level of keyframes. Finally, we reduce the intra-shot redundancy of the remaining shots in two steps:

1. We remove similar sub-shots by calculating the color similarity between keyframes that represent sub-shots;

2. We detect the important content including the presence of a face and motion intensity to score remaining keyframes and keep the keyframes with higher score according to the time limitation requirements of the final summary.

Figure 1 describes our overall approach to rushes summarization. First, a given rushes video is structured into shots and sub-shots and useless sub-shots are filtered (see Section 2 and Section 3). Then, overall inter-shot redundancy is reduced by removing repetitive re-take shots (see Section 4). Finally, a measure

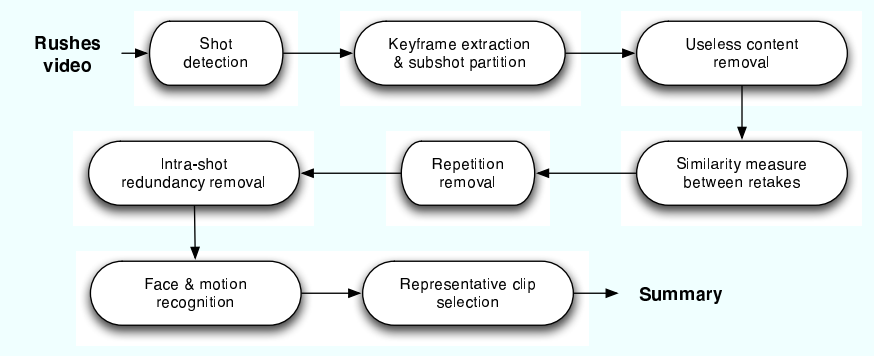


Figure 1: Our approach to rushes video summarization

is proposed to score the presence of faces and motion for intra-shot redundancy removal (see Section 5). We present a summary of our experimental results in Section 6 and some conclusions in Section 7.

2 Video Structuring

Given the raw nature of rushes video and the fact that it has little structure or organisation, the first thing we need to do is to structure it by detecting shots and sub-shots and extract keyframes as representatives from each sub-shot. We do this using shot detection, which we now describe.

2.1 Shot Detection

Since all rushes videos are unedited, hard cuts typically dominate the transitions used because the cameraperson will switch off the camera between "takes" while the next shot is being set up. For this reason, we focus only on detection of hard cuts. In our work we use a mutual information measure between two successive frames calculated separately for each RGB channel. The mutual information between two successive frames is calculated separately for each of the R, G and B channels. In the case of the R component, the element $C^R_{t,t+1}(i,j), 0 \leq i,j \leq (N-1)$, N being the number of gray levels in the image, corresponds to the probability that a pixel with gray level i in frame f_t has gray level j in frame f_{t+1}. The mutual information of frame f_k, f_l for the R component is expressed as:

$$I^R_{k,l} \; = \; - \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C^R_{k,l}(i,j) \; log \frac{C^R_{k,l}(i,j)}{C^R_k(i) C^R_L(j)} \tag{1}$$

The total mutual information between frames f_k and f_l is defined as:

$$I_{k,l} \; = \; I^R_{k,l} \; + \; I^G_{k,l} \; + \; I^B_{k,l} \tag{2}$$

A smaller value of the mutual information leads to a high probability of larger difference in the content between two frames. Local mutual information mean values on a temporal window W of size N_w for frame f_t are calculated as:

$$\overline{I_t} = \frac{\sum_{i=1}^{N_W+t} I_{i,i+1}}{N_W} \tag{3}$$

The standard deviation of mutual information on the window is calculated as:

$$\sigma_I = \sqrt{\frac{\sum_{i=t}^{N_W+t} (I_{i,i+1} - \overline{I}_t)^2}{N}} \tag{4}$$

The quantity $\frac{|\overline{I}_t - I_{t,t+1}|}{\sigma_I}$ is then compared to a threshold H, which represents the mutual information variation at frame f_t deviating from the mean value and determines a boundary frame. The threshold H is set empirically based on experimental results using a data set with annotated boundary frames. Essentially, the mutual information used in our approach measures the relative change value of color feature, which is not sensitive to the absolute difference value of color feature. H could in theory be adapted for various different video types. However, the rushes videos include many different video types and we found that values of H for different video types are very close to each other in a large number of experiments for boundary frame detection anyway. Assuming that the video sequence has a length of N frames, the shot boundary determination algorithm may be summarized as follows:

Step 1: calculate the mutual information time series $I_{t,t+1}$ with $0 \leq t \leq N - N_W$.

Step 2: calculate $\overline{I}_t$ and σ_I at each temporal window in which f_t is the first frame.

Step 3: if $\frac{|\overline{I}_t - I_{t,t+1}|}{\sigma_I} \geq H$, frame f_t is determined as a shot boundary.

We evaluated the effectiveness of this approach on the TRECVid development data for shot boundary detection and it achieved an overall performance of 93.4% recall and 91.5% precision, which is acceptably close to the state of the art [Smeaton et al., 2009].

2.2 Sub-shot Partition

In rushes video, each shot usually contains not only the scripted action, but also other material that is not related to the story of whatever is being filmed, such as camera adjustments, discussions between the director and actors, background noise from the film crew as a shot is being set up, environmental noise, and unintentional camera motion. Furthermore, the scripted action usually contains

varied content because of camera and/or object movements. In video summarization, we aim to remove video segments not related to the storyline and to include only selections from the remaining video segments. One keyframe for each shot, however, is not sufficient for this purpose and so we partition each shot into sub-shots corresponding to different content.

We split each frame into 8×8 pixel grids and calculate the mean and variance of RGB color in each grid. f_{ij} is the feature vector of the jth grid in the ith frame. Euclidean distance is used to measure the difference between neighboring frames F_i and F_j as follows:

$$Diff(F_i, F_{i+1}) \ = \ \sum_j ||f_{ij} - f_{(i+1)j}|| \tag{5}$$

Usually, in one sub-shot the cumulative frame difference $\sum_i Diff(F_i, F_{i+1})$ shows gradual change. High curvature points within the curve of cumulative frame differences are likely to indicate the sub-shot boundaries and we exploit this in our work. We denote the straight line passing through p_i, p_j as $p_i p_j$, where p_i, p_j are the points on the curve of cumulative frame difference and i, j are frame indexes. We define the distance between the point p_x on the curve $p_i p_j$ and the line $\overline{p_i p_j}$ as $Dist(p_x, \overline{p_i p_j})$. Let P_x denote the projection of p_x on the line $\overline{p_i p_j}$, so:

$$Px \ = \ p_i + \mu(p_j \ - \ p_i) \tag{6}$$

where

$$\mu = \frac{(p_x - p_i) \bullet (p_j - p_i)}{(p_j - p_i) \bullet (p_j - p_i)} \tag{7}$$

so that

$$Dist(p_x, \overline{p_i p_j}) = ||p_x - P_x|| \tag{8}$$

According to the definitions above, we propose a simple but efficient sub-shot segmentation method as follows:

- Set the number of frames in shot S, NF= frames_in_shot_S;
- For each point on the curve of cumulative frame difference $p_1 p_{NF}$, calculate the distance $Dist(p_k, \overline{p_1 p_{NF}})$. Seek the point$p_k$, which has the maximum $Dist(p_k, \overline{p_1 p_{NF}})$. If $Dist(p_k, p_1 p_{NF}) > C_{dist}$, mark p_k as a high curvature point.
- For each point on the curve $p_i p_k$ and $p_k p_{NF}$, calculate the distance and find the point with maximum distance $p_{k2} p_{k3}$, if $Dist(p_{k2}, \overline{p_1 p_k}) > C_{dist}$, mark p_{k2} as a high curvature point; similarly if $Dist(p_{k3}, \overline{p_k p_{NF}}) > C_{dist}$, mark p_{k3} as a high curvature point.
- Update the curves to be processed as $p_{k1}, p_{k2}, p_{k2} p_k \ldots$; Repeat the calculations above. If all distances calculated are smaller than C_{dist}, then exit.

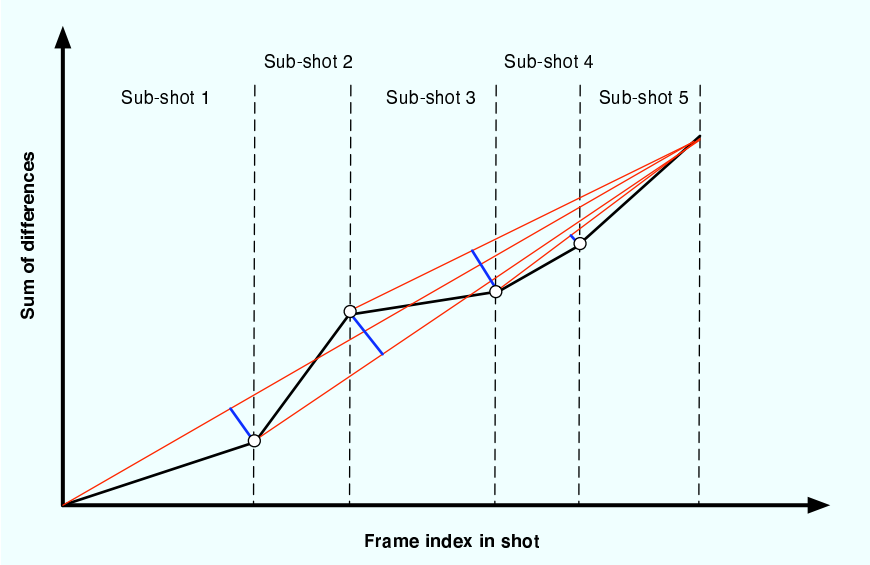


Figure 2: Illustration of sub-shot partitioning algorithm

- All high curvature points selected are sorted in order for sub-shot segmentation boundaries.

Figure 2 explains this idea more clearly. After sub-shot partitioning, the keyframes are selected as the midpoints between two consecutive high curvature points.

3 Removal of Non Content-Bearing or Useless Video

In rushes video, some of the useless content captured consists of actions recorded by the camera, and some consists of content inserted during video recording. Examples of this useless content contained in rushes video are illustrated in Figure 3. These include color bars inserted for colour calibration, monochromatic shots also inserted for calibration of the camera and to assist with metering light levels, clapperboards used to create a visual record of the program, scene, shot and other metadata, and shots which are very short, of the order of 1 second or less. All of these should be removed from the video summary.

Figure 3: Examples of video content to be removed from the video summary

Shots less than 1 second in duration are removed automatically. For shots consisting of color bars and monochromatic shots, four MPEG-7 features including color layout, scalable color, edge histogram and homogenous texture are extracted from all keyframes in the corresponding shots for all of the video, using a commonly available platform known as the AceToolbox [O'Connor et al., 2005], as follows:

- The *scalable color* descriptor from the MPEG-7 XM is extracted for each keyframe. This is a Haar transform-based encoding scheme applied across values of a uniform quantization of the HSV space to 256 bins, after a non-linear mapping into a four-bit representation, giving higher significance to small values. The Haar transform consists of a sum operation (a low-pass filter) and a difference operation (a high-pass filter). Summing pairs of adjacent lines results in a histogram with half the number of bins. Performing this process iteratively, we obtain histograms of 128, 64, 32 and 16 bins respectively.

- The *color layout* descriptor is designed to capture the spatial distribution of color in an image. By default, the input image is divided into 64 (8×8) blocks and their average colors are derived (YCrCb color space). These are then transformed into a series of coefficients by an 8×8 discrete cosine transformation (DCT). A few low-frequency coefficients are selected using zigzag scanning and quantized to form the description.

- The Canny algorithm [Canny, 1986] is used for edge detection in a multi-stage process. First, the frame image is smoothed by Gaussian convolution and a 2-D first derivative operator then highlights ridges. The algorithm then tracks along the top of these ridges and sets to zero all pixels not actually on the ridge top so as to give a thin line in the output. Finally, we compute an edge direction histogram from the edge image.

- Homogeneous texture is based on the use of Gabor functions which are sinusoidal modulated Gaussian. In a set, all filters are similar in the sense that they can be generated from one filter (called the mother wavelet or the basis wavelet) simply by translation, scaling and rotation. For this reason the set of filters can be seen as a set of wavelets. Nevertheless, it does not satisfy orthogonality and it is efficient for analysis but not for reconstruction. On the other hand, it provides very good properties of scale and rotation invariance. The frequency space is partitioned into 30 channels with 6 equal divisions in the angular direction ($30\,^\circ$ intervals) and 5 octave divisions in the radial direction. The values of the standard deviation were chosen such that the contour sections of the Gaussian envelopes coincide at their half magnitude.

Following low-level feature extraction, we use support vector machine (SVM) classifiers, trained to recognize color bars and monochromatic shots. We employ the algorithm for Near-Duplicate Keyframe (NDK) detection described in

[Ngo et al., 2006] to detect clapperboards. A set of 50 example keyframes of clapperboards were extracted from the TRECVid development set. The regions where clapperboards are present were manually annotated. Among the keyframes of each shot in the given rushes video, we detect the key points and match them with the example clapperboards. If enough matches are found that lie in the annotated regions, the keyframe is detected as a clapperboard and removed.

4 Re-take Shot Detection and Removal

As mentioned earlier, in rushes the same scene can be re-shot many times in order to eliminate actor or filming mistakes. In such cases, the re-taken shots should be detected automatically and the most satisfactory one kept, removing the others from the final summarization. Rows 1, 2 and 3 in Figure 4 show the keyframes extracted from three re-taken shots in rushes test video ID: MRS044500.

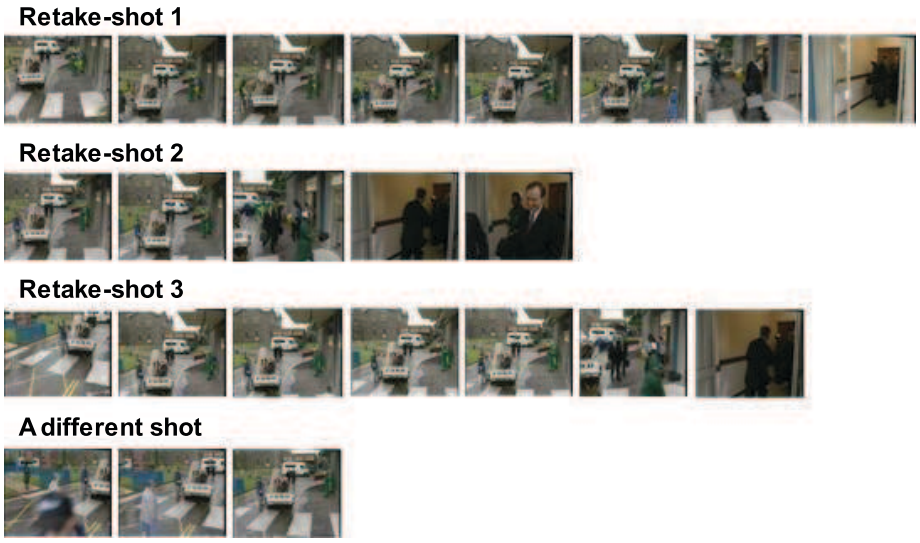


Figure 4: Examples of re-taken shots from rushes video

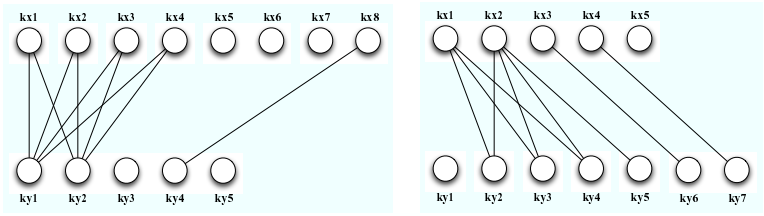
We assume that the similarity between shots can be measured according to the similarity of keyframes extracted from corresponding shots. Thus, the re-taken shots are detected by modeling the continuity of similar keyframes. Motivated by maximal matching in bipartite graphs, we propose an approach for similarity detection between video shots based on this matching technique.

The key for re-take shot detection is how to measure the similarity between two shots. The detailed motivation of our approach is, firstly, that shot similarity should be measured by sub-shot similarity relationships because the re-take

shots in rushes video are frequent, the content is complex, and using sub-shots can represent video content more precisely compared with using one keyframe for one shot. The bipartite graph is used to model the sub-shot similarity relationships and maximal matching in bipartite graphs is used to measure the similarity between the shots. Our second motivation is based on the fact that the content in one sub-shot is consistent, and so it iss appropriate to extract one keyframe to represent a corresponding sub-shot. We calculate the similarity between two keyframes to measure the similarity between the corresponding sub-shots. Keyframe similarity is calculated according to the difference among the spatial color histogram and texture features between two keyframes.

A bipartite graph is a connected undirected graph such that the vertices of G are partitioned into two sets X and Y and every edge of G has one end point in X and the other in Y. Matching M in G is a set of edges that have no end points in common. The maximum bipartite matching problem is how to find a matching with the greatest number of edges over all matchings.

According to the definitions of bipartite graphs and maximum matching, a shot can be expressed as: $S = \{k_1, k_2, \ldots, k_n\}$, where k_i represents the i^{th} keyframe. So, for two shots, $Sx = \{kx_1, kx_2, \ldots, kx_n\}$ and $Sy = \{ky_1, ky_2, \ldots, ky_n\}$, the similar keyframes between Sx and Sy can be expressed by a bipartite graph $G = \{Sx, Sy, E\}$, where $V = Sx \cup Sy$, $E = \{e_{ij}\}$, indicates kx_i is similar to ky_j. Figure 5 illustrates two examples of bipartite graphs for retake-shot 1, retake-shot 2 and retake-shot 3, those shots introduced in Figure 4.



(a) (b)

Figure 5: Two examples of bipartite graphs for re-taken shots from Figure 4: (a) shot 1 and shot 2; (b) shot 2 and shot 3

Clearly, there exist many similar pairs of keyframes between pairs of retaken shots. In our experiments however, we also find there often exist similar keyframes within the one retaken-shot. This results in one-to-many, many-to-one and many-to-many relations in a bipartite graph. In this case, there will be many similar keyframe pairs found between two dissimilar shots. The bipartite graph between retake-shot 3 and a different shot shown in Figure 5 illustrates such a case in Figure .

If we use the number of similar keyframe pairs to determine which are the retake-shots, 4 similar keyframe pairs are found in the Sx shot shown in Figure 6 and exceed half of the keyframes in Sx. In this case, Sx is likely determined to be similar to Sy, whilst this is not the case in practice.

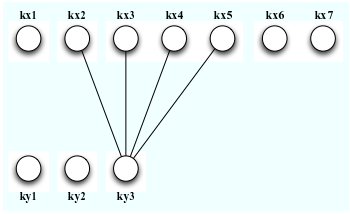


Figure 6: A bipartite graph between two dissimilar shots

In our approach, the similarity between two shots is measured by the maximal matching of similar keyframes in the bipartite graph model. The Hungarian algorithm [Dai et al., 1995] is used to calculate maximal matching $M, M \subseteq E$. If $M \geq min\{\lceil \frac{2}{3}n \rceil, \lceil \frac{2}{3}m \rceil\}$ where n, m are the number of keyframes in the two shots, it is determined that one shot is similar with respect to the other. Figure 7 shows the maximal matching results of the examples shown in Figure 5 and Figure 6.

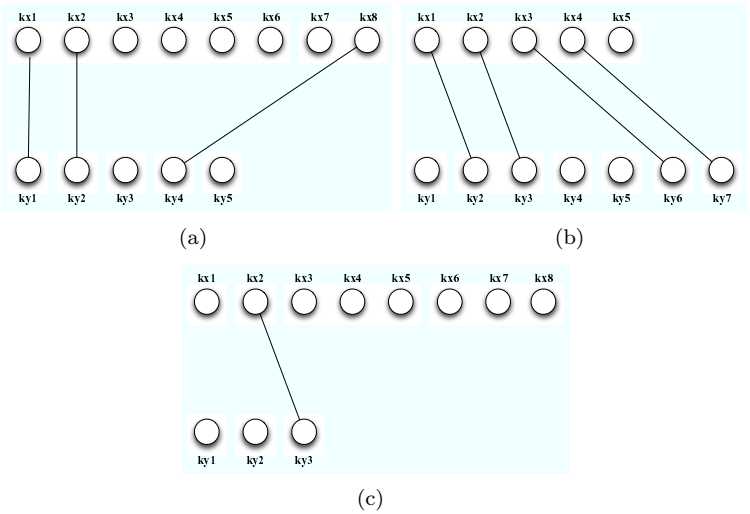


Figure 7: Examples of maximal matching results for shots in Figures 5 and 6

From Figure 6, we can see that the maximal matching of dissimilar shots is 1. From this, it is relatively straightforward to determine true retake-shots according to maximal matching.

The matching steps using the Hungarian algorithm are now described. We assume that a given bipartite graph is $G_k = \{Sx, Sy_k, E_k\}$; Mark "0" expresses the vertex that is not searched, mark "1" expresses the saturation vertex and mark "2" expresses the vertex that cannot increase the matching score.

Step1: Given an initial matching M, mark all vertices as "1";
Step2: Check if every vertex in Sx has a non-"0" mark.

If yes, M is the maximal matching. End.

If no, find a vertex marked "0" $x_0 \in Sx$, let $A \leftarrow \{x_0\}, B \leftarrow \emptyset$.
Step3: Check if $N(A) = B(N(A) \subseteq SY_k)$. $N(A)$ expresses the vertices belonging to Sy_k that neighbor with the vertices in A. $B(N(A) \subseteq SY_k)$ expresses the vertices belonging to Sy_k that neighbor with the vertices in $N(A)$.

If yes, x_0 cannot increase matching, mark x_0 as "2", go to Step 2;

If no, find a vertex Sy_i in $N(A) - B$, check if Sy_i is marked with"1".

If yes, there exists an edge $(Sy_i, z) \in M$, let $A \leftarrow A \cup \{z\}, B \leftarrow B \cup \{Sy_i\}$, and go to Step 3.

If no, there exists an augmenting path from $x + 0$ to Sy_i, let $M \leftarrow M \oplus P$ mark X_0 and Sy_i as "1", go to Step 2.

The complexity of this algorithm is $O(ne)$, where n is the number of vertices of Sx in the bipartite graph $G = \{Sx, Sy, E\}$ and e is the number of edges. After measuring the similarity of shots, re-take shots are detected, the last shot is kept and others are removed because the last retake shot is usually the most satisfactory one from the point of view of appearing in a summary.

5 Selecting Representative Shots and Summary Generation

After we perform the detection and removal of low-value content and repetitive re-take shots, useful content is kept as candidate material for inclusion in the generated summary. However the volume of the remaining video content typically exceeds an amount that would be deemed useful to make up a summary. Indeed the useful duration limit in the TRECVid summarisation guidelines was set at 4% of the original video in 2007, and 2% of the original video in 2008. This means that the most representative video clips need to be selected from the remaining content in order to generate the final summary. In our work, we extract motion and face factors to rank how representative each remaining sub-shot is in the context of the overall video.

A three-stage process, achieved using the aceToolbox [O'Connor et al., 2005], is used to describe the level of motion activity in each sub-shot. First, MPEG-1 motion vector data is extracted from the video. Next, the percentage of non-zero blocks in the frame (where a high percentage indicates higher motion activity) is calculated for each frame in the video. Finally, this per-frame data is used along with the shot boundary data calculated previously, to compute an average motion measure for the entire sub-shot. As a result, each keyframe in a given sub-shot is assigned the same measure of motion activity.

Our face detection processing extends the Bayesian Discriminating Feature (BDF) originally proposed by Liu [Liu, 2003] for detecting frontal faces in grayscale images. Using a statistical skin color model [Cooray and O'Connor, 2005], we can detect multiple faces at various sizes and orientations within color images. Ideally this processing would be carried out for each frame of the original footage; however, for efficiency we only perform this operation on the detected keyframes. While this reduction in processing time potentially results in a loss of information, such as the prevalence of faces across shots, it ensures efficient processing while still providing enough information to reliably enhance summary construction.

Sub-shot duration is important for sub-shot selection so we use the following simple weighting to combine the factors.

$$\begin{aligned} Score &= (Number-of-faces/Max-faces-in-footage \times 0.3) + \\ &\quad (Amount-of-motion \times 0.3) + \\ &\quad (Duration-of-subshot/Total-duration-all \times 0.4) \end{aligned}$$

Once the representative scores for sub-shots are calculated, those sub-shots with highest scores are selected according to the summary duration limitation. Finally, 1-second clips centred around the keyframe in each selected sub-shot are extracted for generating our final summary.

6 Dataset and Experiments Results

Using our approach described in this paper, we generated the summaries for all test rushes videos in the TRECVid 2007 rushes summarization evaluation. The data used in this evaluation consisted of MPEG-1 files corresponding to rushes video recorded for TV programs, specifically the BBC dramatic series Casualty, House of Elliot, Jonathan Creek, Ancient Greece, Between the Lines and others. The files were 25 minutes in duration on average, an artifact of the fact they were recorded onto tapes initially, and digitized subsequently. The task set to the TRECVid participants was to generate a video summary with no interaction except one single play through with unlimited optional pauses, such that it maximises viewers efficiency at recognising objects and events as quickly as possible, a true definition of what a summary should do.

In evaluating the effectiveness of a video summary, it would be very difficult to formally identify all content in a source video, do likewise for a summary and then compare them in a way that was repeatable and scalable to large numbers of generated summaries. In TRECVid, the organisers created partial ground truths for 42 original videos and human judges or assessors then viewed each summary and judged it against the list of important segments making up the ground truth. While this is an approximation of the effectiveness of a summary it is a scalable approach which is repeatable, which is what we do in this paper.

Twenty-two research groups completed submissions to the TRECVid BBC Rushes summarisation evaluation in 2007 and the overview paper from the summarizaion workshop points to the different approaches taken by the groups, as

well as their relative performances [Over et al., 2007]. Possibly the most surprising result was the good performance of the baseline systems, which were based on crude frame sampling approaches[2].

The seven criteria set by the TRECVid guidelines for summarization evaluation in 2007 and used again in 2008 are:

- EA: Easy to understand: (1 strongly disagree — 5 strongly agree);
- RE: Little duplicate video: (1 strongly disagree — 5 strong agree);
- IN: Fraction of inclusions found in the summary (0 — 1);
- DU: Duration of the summary (sec);
- XD: Difference between target and actual summary size (sec);
- TT: Total time spent judging the inclusions (sec);
- VT: Video play time (vs. pause) to judge the inclusions (sec).

IN, DU and XD are objective criteria which we can calculate directly in order to evaluate our summaries and allow direct comparison with the published results from TRECVid. However, EA, RE, TT and VT are criteria that depend on subjective judgments by assessors. Thus for a complete evaluation of our proposed approach it was necessary to re-run the evaluation performed by NIST with our own test subjects. Ten participants (all students in the School of Information System & Management, National University of Defense Technology, China) were selected to review the generated summaries under the exact same guidelines and setup as carried out by NIST in the TRECVid evaluation and they gave their score for the four subjective criteria.

Of course, by running our own evaluation of summary content outside the TRECVid process we could potentially introduce new subjective variations into the evaluation process. To investigate this, we first evaluated three sets of summaries using our own participant assessors: the two baseline summary sets used in TRECVid and our own original submission to TRECVid in 2007. The experimental results we obtained with our own assessors compared to the official results reported from TRECVid assessors are shown in Table 1.

The results in Table 1 show there exists only a small difference in the subjective judgments between our participant assessors and assessors used by NIST. This is understandable given that different people have different skills, intellects, powers of discernment, etc. However, from Table 1 we can see that the difference of judgments between our assessors and NIST assessors is small. From this we conclude our participants' evaluations on the subjective criteria are reasonable and credible. Given this, we proceeded to re-run the complete evaluation of summaries we have generated.

[2]Because of the relatively excellent performance of baseline runs in TRECVid Summarisation 2007, we use these as a basis for comparison against our own work since they were almost the best.

Table 1: Experimental results for the comparison between our assessors and NIST assessors

Criterion		EA	RE	TT	VT
TRECVid Baseline 1	Our Assessors	3.12	3.26	115.45	73.20
	NIST Assessors	3.33	3.33	110.67	66.67
TRECVid Baseline2	Our Assessors	3.35	3.30	118.10	70.38
	NIST Assessors	3.67	3.67	109.17	63.83
Our original TRECVid 2007	Our Assessors	2.29	3.33	76.78	48.49
	NIST Assessors	2.67	3.67	70.83	42.67

Table 2: Experimental results for IN (inclusion), DU (duration) and XU (target vs. summary size)

Criterion	IN	DU	XD
TRECVid Baseline1	0.60	66.40	-2.28
TRECVid Baseline2	0.62	64.60	-0.89
Mean of all 22 teams	0.48	49.54	10.33
Our original TRECVid	0.38	40.90	8.65
Our enhanced	**0.78**	**41.61**	**18.83**

Table 3: Experimental results for EA (ease), RE (duplication), TT (time judging) and VT (video playback)

Criterion	EA	RE	TT	VT
TRECVid Baseline1	3.12	3.26	115.45	73.20
TRECVid Baseline2	3.35	3.30	118.10	70.38
Our original TRECVid	2.29	3.33	76.78	48.49
Our enhanced	**3.74**	**3.88**	**89.21**	**44.50**

The experimental results averaged over all of our summaries for all of the test videos in TRECVid summarisation 2007 are shown in Table 2 and Table 3. The results in Table 2 show our enhanced approach results in a big improvement in IN score (+0.40, more than double) with a slightly longer duration of summaries (+0.71 sec, 1.7%) compared with our original approach. Of particular note is the fact that our enhanced approachs XD is 18.83, which is 8.5 sec longer than the mean of the other 22 teams. This is because in our approach we tend to retain the valuable content from original source rushes as much as possible within the summary duration requirement. Table 3 shows the evaluation results for the four subjective criteria. Clearly we obtain very encouraging results for the EA and RE. These experimental results clearly show that our enhanced approach performs competitively compared with the other TRECVid teams and the baselines.

7 Conclusion and Discussion

Rushes videos are captured by professional cameramen as the early stage of the video production lifecycle. As an unedited version of the final video, they include many useless and redundant or repeated shots. Although the structure of the video and the threading of the storyline are not directly available, rushes are organized based on shot structure.

In this work, we employ shot and sub-shot detection for video structuring, we train SVMs for removing useless content, and we model the similarity of keyframes between two shots by bipartite graphs. We then measure shot similarity by maximal matching for re-take shot detection. Based on consideration of motion, face and duration, sub-shots are ranked and the most representative clips are selected for final summary generation. This corresponds to a significantly extended approach compared to our original TRECVid submission. To evaluate this new approach, we re-ran the evaluation procedure ourselves with our own assessors. Experimental results indicate that our subjective evaluation is in line with that originally carried out by NIST. Our improved approach clearly demonstrates improvements compared to our original approach, but more importantly compared to the TRECVid baselines and the other teams who participated in the evaluation.

Not withstanding this, the summarization problem clearly still remains challenging. Indeed, most submissions cannot significantly outperform the two baselines, which are simply based on fixed-length shot selection and visual clustering. This poses the key question as to whether a deeper semantic understanding of the content can help in this regard and this is something for future work.

Acknowledgements:

This work was funded by the National High Technology Development 863 Program of China (2006AA01Z316), the National Natural Science Foundation of China (60572137 and 60875048) and by Science Foundation Ireland as part of the CLARITY CSET (07/CE/I1147). The authors thank the reviewers for their helpful and insightful feedback.

References

[Byrne et al., 2007] Byrne, D., Kehoe, P., Lee, H., Ó'Conaire, C., Smeaton, A. F., O'Connor, N. E., and Jones, G. J. (2007). A user-centered approach to rushes summarisation via highlight-detected keyframes. In *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pages 35–39, New York, NY, USA. ACM.

[Canny, 1986] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698.

[Cooray and O'Connor, 2005] Cooray, S. and O'Connor, N. (2005). Hybrid technique for face detection in color images. In *IEEE Conference on advanced video and signal based Surveillance, AVSS, Italy*, pages 253–258.

[Dai et al., 1995] Dai, Y., Hu, G., and Chen, W. (1995). Graph theory and algebra structure. *Beijing: Tsinghua University Press, (in Chinese)*, pages 89–91.

[Ferman and Tekalp, 2003] Ferman, A. and Tekalp, A. (2003). Two-stage hierarchical video summary extraction to match low-level user browsing preferences. *IEEE Transactions on Multimedia*, 5(2):244–256.

[Liu, 2003] Liu, C. (2003). A Bayesian Discriminating Features Method for Face Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 741–754.

[Ma et al., 2002] Ma, Y., Lu, L., Zhang, H., and Li, M. (2002). A user attention model for video summarization. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 533–542. ACM New York, NY, USA.

[Ngo et al., 2006] Ngo, C., Zhao, W., and Jiang, Y. (2006). Fast tracking of near-duplicate keyframes in broadcast domain with transitivity propagation. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 845–854. ACM New York, NY, USA.

[O'Connor et al., 2005] O'Connor, N., Cooke, E., le Borgne, H., Blighe, M., and Adamek, T. (2005). The AceToolbox: Low-Level Audiovisual Feature Extraction for Retrieval and Classification. In *2nd IEE European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies*.

[Over et al., 2008] Over, P., Smeaton, A. F., and Awad, G. (2008). The TRECVid 2008 BBC rushes summarization evaluation. In *TVS '08: Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, pages 1–20, New York, NY, USA. ACM.

[Over et al., 2007] Over, P., Smeaton, A. F., and Kelly, P. (2007). The TRECVid 2007 BBC rushes summarization evaluation pilot. In *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pages 1–15, New York, NY, USA. ACM.

[Smeaton et al., 2009] Smeaton, A. F., Over, P., and Doherty, A. R. (2009). Video shot boundary detection: Seven years of trecvid activity. *Computer Vision and Image Understanding*, In Press, Corrected Proof:–.

[Smeaton et al., 2006] Smeaton, A. F., Over, P., and Kraaij, W. (2006). Evaluation campaigns and TRECVid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA. ACM Press.

[Taskiran et al., 2006] Taskiran, C., Pizlo, Z., Amir, A., Ponceleon, D., and Delp, E. (2006). Automated video program summarization using speech transcripts. *IEEE Transactions on Multimedia*, 8(4):775–791.