Video Summarization Preserving Dynamic Content

Francine Chen
FX Palo Alto Laboratory
3400 Hillview Ave, Bldg 4
Palo Alto, CA USA
chen@fxpal.com

Matthew Cooper
FX Palo Alto Laboratory
3400 Hillview Ave, Bldg 4
Palo Alto, CA USA
cooper@fxpal.com

John Adcock
FX Palo Alto Laboratory
3400 Hillview Ave, Bldg 4
Palo Alto, CA USA
adcock@fxpal.com

ABSTRACT

This paper describes a system for selecting excerpts from unedited video and presenting the excerpts in a short summary video for efficiently understanding the video contents. Color and motion features are used to divide the video into segments where the color distribution and camera motion are similar. Segments with and without camera motion are clustered separately to identify redundant video. Audio features are used to identify clapboard appearances for exclusion. Representative segments from each cluster are selected for presentation. To increase the original material contained within the summary and reduce the time required to view the summary, selected segments are played back at a higher rate based on the amount of detected camera motion in the segment. Pitch-preserving audio processing is used to better capture the sense of the original audio. Metadata about each segment is overlayed on the summary to help the viewer understand the context of the summary segments in the original video.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Abstracting methods*

General Terms

Algorithms, Experimentation, Performance

Keywords

video summarization, clustering, segmentation, presentation

1. INTRODUCTION

Video cameras are becoming ubiquitous as they are increasingly embedded in common devices such as cell phones and digital cameras. As evidenced by the explosion of user-generated video on the web, it has become easy for people to create video media. This increasing body of publicly shared video, which is largely unedited and unstructured, can be tedious and time consuming to search.

NIST has organized a track in TRECVID where research groups develop systems for summarizing rushes, specifically, unstructured BBC shows. The TRECVID Rushes task and data are described in Over et al. [9]. In this paper, we describe the system we developed for the Rushes task.

Our system selects short excerpts of video, trying to identify non-redundant segments containing action. The action may be due to objects moving in the video or the camera panning or zooming across a scene. The system also attempts to eliminate uninteresting segments, including color-bars, clapboards, and objects such as hands, arms, or backs that accidentally cover up the camera lens.

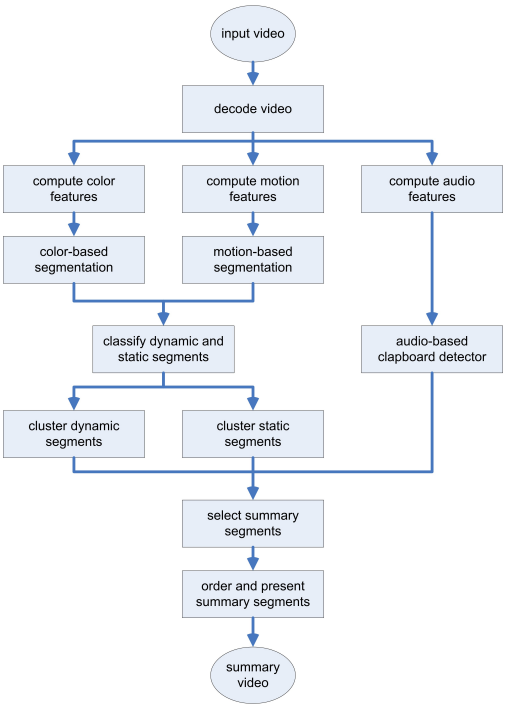

Figure 1: Block diagram of system

2. SYSTEM OVERVIEW

Figure 1 is an overview of the video summary system. The input video was decoded using FFmpeg. Three types

MM'07, Sept 23–29, 2007, Augsberg, Germany.

Figure 2: A frame with overlayed points and trails indicating the motion. The points on the bicyclist show little motion and the background shows the image shift due to a right to left pan.

of features are computed from the decoded frames: color, image motion, and audio. The color and motion features are used to segment the input video, and the segments are classified into dynamic camera motion, static camera, or ignore. The differentiation into dynamic and static camera was motivated in part by the Rushes task considering camera motion to be an event. The dynamic and static segments are clustered separately to identify redundancies. Audio is extracted and used to identify clapboard appearances. Segments in which an audio clap is identified are removed during summary segment selection. Finally, the selected segments are ordered and concatenated to create the summary video with overlayed metadata to help the viewer better understand the summary.

3. FEATURES

Color and motion features are computed from the video to identify segments which are then clustered and classified.

3.1 Color Features

For each frame we extract three channel color histograms in the YUV colorspace. This is a simple and common feature parametrization [3]. We extract both global image histograms and block histograms using a uniform 4×4 spatial grid.

3.2 Motion Features

Motion-based features are computed for each frame of the videos using the Lucas-Kanade [5, 1] point-based tracking functions included in the OpenCV [8, 1] image processing system. For each point that is successfully tracked, the vector corresponding to its displacement between frames is computed. Points that cannot be matched between frames are ignored. The number of successfully tracked features (which is highly dependent on the degree of contrast and texture in the video) is reported in the feature vector.

For each motion vector the magnitude is computed. The magnitudes are summarized in a non-linearly spaced histogram with 14 bins. To capture directional information, the average angle of the vectors falling within each bin is also computed. This yields a pair of vector quantities: the magnitude histogram, and the vector containing the average direction of the motion vectors in each histogram bin. The boundaries of the magnitude histogram were manually chosen as follows (in units of pixels): 0.0, 0.5, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 6.0, 8.0, 10.0, 12.0, 15.0, 18.0, inf.

To capture camera motion (pans and zooms), the motion vectors are ordered by magnitude, and the 75% with the highest magnitudes are thrown out. The mean and variance of the x and y components of the remaining low-magnitude vectors are reported. In addition, the radial component (the vector projected onto the line joining the first point and the center of the image) of each motion vector is computed. The mean and variance of this radial projection is also reported. The final motion based feature vector contains 1+6+14+14=35 values.

4. SEGMENTATION

Segmentations based on color and motion are computed separately. The color-based segmentation identifies shot boundaries while the motion-based segmentation identifies pans, zooms, and colorbars.

4.1 Color-based Segmentation

Visual segmentation is based on the use of pairwise inter-frame similarity features and kernel correlation. The basic system is documented in [2] and has three main components: The first is the low-level color feature extraction described in Section 3.1. The second component is the generation of an incomplete inter-frame distance matrix. We build a matrix with elements $\mathbf{S}(i,j)$ equal to the chi-squared distance between the histograms from frames i and j. Because the distance measure is symmetric and $\mathbf{S}(i,i) = 0$, we compute only a small portion of the full distance matrix.

The final piece of the segmentation system is kernel correlation. In this step, a small square kernel matrix, $\mathbf{K}$, representing the ideal appearance of an abrupt shot boundary is correlated along the main diagonal of $\mathbf{S}$. The frame-indexed novelty score is a simple linear correlation:

$$\nu(n) = \sum_{l=-L}^{L-1} \sum_{m=-L}^{L-1} \mathbf{K}(l,m)\mathbf{S}(n+l, n+m) \quad . \qquad (1)$$

The kernel and distance matrix are transformed to a lag domain representation to expedite processing [2]. Throughout, $L = 36$. Once the novelty score is computed for each frame, simple peak detection is applied using a threshold and analysis of the first difference of ν.

4.2 Motion-based Segmentation

Horizontal pans, vertical pans, and zooms are identified separately. The horizontal and vertical pans are then combined into a single set of combined pan segments. All the motion features are smoothed before analysis begins.

The rate of camera motion often varies during a pan or zoom, starting and ending more slowly. A threshold representing the minimum amount of motion required for a pan or zoom to occur is used to identify candidate pans and zooms from the computed motion features. The endpoints of the candidates are identified as the first locations forward and backward from the high motion region that are less than a threshold, computed as the running average within a window of 2000 frames. The use of a running average helps in cases where the camera is relatively shaky.

Colorbars are identified by finding regions where there are very few motion points and very little global motion. Specifically, the magnitude of global motion in the x and y direction is computed and a threshold on the number of motion points is used to identify candidate colorbar segments. Thresholds

Figure 3: The audio waveform from a clapboard appearance. First, middle, and final frames from the 1.3 second segment are shown over the audio. The impulsive nature of the clap sound is apparent.

on the peak motion values and the average motion value in each segment are used to remove segments with too much global motion, e.g., an arm passing in front of the lens or quickly moving the camera to focus on an object.

4.3 Audio Clapboard Detection

Clapboards appear frequently in the videos and are undesirable as summary material. Simple analysis of the short-time log energy for loud impulsive onsets detects many of the clapboard instances enabling their exclusion from summaries. To implement the detector, log RMS energy was measured in 5ms windows and estimates of the onset slope and height of peaks in the log energy were tested against thresholds determined through examination of the data.

The simple nature of the detector is such that it cannot distinguish between clapboards and other sounds with impulsive onsets, though in practice the detector has a fairly low false-alarm rate and rarely exhibits systematic false alarms that completely eliminate key segments from the summary.

5. SEGMENT CLASSIFICATION

The color-based and motion-based segmentations are used to identify two types of segments: those containing camera motion, or *dynamic segments*, and those segments where the camera is relatively steady, or *static segments*. A third class of segments, those that will not be considered further, is also created.

We differentiate between dynamic segments and static segments so that in the clustering step, we can cluster the two types separately. Empirically, we observed that the static segments tend to be much more similar to each other, presumably because the background is relatively stable, in contrast to the dynamic segments.Thus, different features can be used to cluster dynamic segments. In addition the Rushes task considered camera motion an event, so we explicitly identified camera motion events.

Segment classification begins by filtering the motion-based segments to remove those that are not suitable for inclusion in a summary. These include segments that are very short (we used a 20 frame minimum to be kept), those for which the motion features indicate that the motion was very fast, and colorbars. The remaining motion segments are labeled dynamic segments. Static segments are identified as those segments which are not motion segments and have not been removed from consideration. Next, the color-based segmentation is combined with the motion-based segmentation to further divide the motion segments.

6. CLUSTERING SEGMENTS

We used clustering to remove redundant shots. Because of the different characteristics of the shots with camera movement and the shots where the camera was relatively steady, different features and clustering methods were employed.

6.1 Dynamic Segment Clustering

The dynamic clustering step takes as input the boundaries of the dynamic segments and the color and motion features extracted per frame. The dynamic clustering step includes two main components: dimension reduction and spectral clustering.

We use probabilistic latent semantic analysis (PLSA) [4] For dimension reduction and apply it separately to the color and motion features. We normalize each histogram per-channel and per-block for color data and per frame for motion data to sum to one. We then accumulate the normalized counts over the entire motion segment. PLSA learns a generative model for the data according to:

$$P(s_i, c_j) = P(s_i) \sum_{k=1}^{K} P(z_k|s_i) P(c_j|z_k) \quad . \tag{2}$$

denoting the i^{th} segment and j^{th} bin by s_i and c_j, respectively. We construct $K = 9$ dimensional latent variable spaces indexed by z_k.

For clustering, we use the latent variable distributions for both the motion features and the color features in each motion segment. Our approach is inspired by [6] and [7]. We use these features to build a $N_S \times N_S$ similarity matrix, where N_S is the number of motion segments. For this step, we treat each latent variable distribution as a histogram, and compare them using the Bhattachraya coefficient:

$$d_*(s_i, s_j) = \left(1 - \sum_k \sqrt{P_*(z_k|s_i) P_*(z_k|s_j)}\right)^{\frac{1}{2}} . \tag{3}$$

This distance is computed for both the motion ($* = m$) and color ($* = c$) features, where the z_k are indices into each respective latent subspace.

We add a temporal term to the similarity measure which penalizes clusters with segments far apart in time:

$$d_t(s_i, s_j) = \frac{| < s_i > - < s_j > |}{N} \quad , \tag{4}$$

where $< s_i >$ is the mean frame number for the i^{th} segment and N is the number of frames in the video. The three distance measures above are combined into a single exponential inter-segment similarity matrix:

$$\mathbf{S}_S(i,j) = \exp\left(- \sum_{*=m,c,t} \frac{(d_*(s_i, s_j))^2}{2\sigma_*^2}\right). \tag{5}$$

Here $\sigma_m = \sigma_c = 0.25$ and σ_v was set to the average distance in frames between segments that are 9 segments apart in the source video. These choices are modifications of those in [7].

We use the inter-segment similarity matrix as input to the spectral clustering algorithm described in [6] which was also used in [7]. $\mathbf{S}_S$ is processed and scaled, and then its eigenvectors are computed. We used the eigenvalues to do a coarse rank estimation, finding the number of eigenvalues that represent 45% of the total energy in the eigenspectrum, and only retaining those. This in turn determines the number of clusters. For segment clustering, the eigenvectors are

clustered using k-means with the number of clusters determined in the rank estimation above.

6.2 Static Segment Clustering

Segments that are labeled as static segments are clustered using single-link agglomerative clustering because it was found to produce more balanced clusters for this type of data. For features, we used the average of the block histogram values in each segment. In our experiments, the type of clustering had a larger effect on the static segment cluster results than the distance measures or the use of time information. So for simplicity we used the Euclidean distance without time information. The effects that we noticed may be due to the static color-based segment features having much smaller variance than those of the dynamic segments.

With agglomerative clustering, a method is needed for defining clusters. We used a semi-adaptive threshold, where the threshold is a function of the knee of the sorted heights from the cluster tree. The idea here is that similar segments have smaller distances and so their heights are generally lower than the height at the knee. When there are just a few segments (we defined this as 20 or less), a fixed threshold is used to define the clusters, since there are too few segments to reliably estimate the knee.

7. SEGMENT SELECTION

We attempted to automatically detect videos with content that is not amenable to cluster-based summarization. This applied to all videos for which no dynamic segments were detected. If in addition, fewer than 9 static segments are detected, we assume that the video contains mostly long static shots. In this case, the summary then consists of three second excerpts taken from the center of each segment. These segments played back at 1.5 times normal speed comprise the summaries in this case. In retrospect, this approach generated short summaries that excluded various object-based events that didn't produce appreciable motion.

In the normal case, in which both static and dynamic segments are detected, we first select the dynamic segments under the assumption that this was more appropriate to the Rushes task. We select representative segments per cluster using the inter-segment similarity matrix produced by the measure of (5). We can then compute the similarity between each segment $\{s_c : c \in C\}$ and its segment cluster C:

$$\mathrm{Sim}(s_c, C) = \frac{1}{|C|} \sum_{\hat{c} \in C} S_S(c, \hat{c}) \quad . \tag{6}$$

We select a dynamic representative segment with index $c^* \in C$ as

$$c^* = \underset{c \in C}{\mathrm{argmax}} \left(\mathrm{Sim}(s_c, C) - \mathrm{Sim}(s_c, P) \right) \quad . \tag{7}$$

where P denotes the index set for previously selected segments in the summary. This measure tries to select segments that are both representative of the cluster and different from previously selected segments. This process is repeated to select one representative segment from each cluster of dynamic segments, or until the duration time limit is exhausted. The selected segments are included in their entirety up to a maximum duration of six seconds. If the dynamic segment is longer than six seconds, the segment is truncated to remove the beginning of the segment such that six seconds remain.

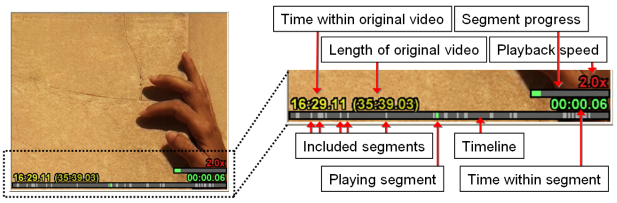

Figure 4: A frame from a video summary showing overlayed timeline and information.

After selecting dynamic excerpts, static segments are included. In the first step, static segments are removed from consideration if they overlap selected dynamic segments, contain detected clapboards, or belong to unique (singleton) clusters. Starting from the static segment cluster with the longest total duration and proceeding in descending order, a segment is selected from each cluster. The selection proceeds according to (7) where the similarity matrix used is the same distance matrix (Euclidean distance between average segment block histogram) used for the agglomerative clustering.

For each selected static segment, we determine a segment excerpt based on a frame-indexed activity score computed from the motion features of Section 3.2. Given the 14 bin motion magnitude histogram of frame f as $m_f(b), b = 1, \cdots, 14$, we compute

$$\hat{m}_f = \sum_{b=1}^{14} (b \cdot m_f(b)) \quad .$$

The activity score is the output of a 73 point median filter applied to the above score. Local maxima in the activity score correspond to frame ranges with high levels of generic activity or object motion. For summarization, we excerpt the three second portion of a selected static segment with the highest average activity score. The process is repeated to include additional static segments, possibly revisiting various clusters, until either all static segments are included or the maximum summary duration is reached.

8. SUMMARY PRESENTATION

The selected segments are ordered by the beginning time of the earliest segment in the cluster to which each selected segment belongs, which we hypothesized would make it easier for the evaluators to match the shot against a list of shots.

When the summary video is rendered, visual cues are overlayed to provide information to the viewer about the context of the summary segments. This is shown in Figure 4 The time of the currently playing segment within the original video is shown alongside the total length of the original video. A timeline representing the original video is also shown with shading marking the portions of the original video which are included in the summary. The currently playing segment is highlighted on the same timeline. Also shown are the current playback rate (which may change from segment to segment), the play time of the current segment, and a progress bar which indicates what proportion of the current segment has been played. We also modified the audio of the sped up segments to restore pitch.

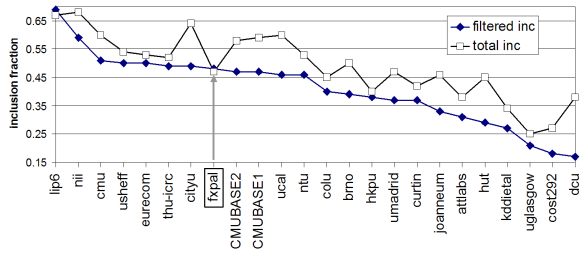

Figure 5: Mean inclusion by group. □ denotes baseline inclusion results, ♦ denotes results of only camera-motion based inclusions.

MRS155534		MS216210		MRS042543	
judge	score	judge	score	judge	score
2	.58	1	.67	1	.67
2	.25	1	.75	1	.67
3	.58	3	.67	4	.50
3	.42	3	.58	4	.42
6	.50	4	.50	7	.83
6	.50	4	.33	7	.67

Table 1: Inclusion scores for three test files

9. RESULTS AND CONCLUSIONS

The system was developed on the 47 development videos and evaluated on the 42 unedited test videos from the BBC as part of the TRECVID 2007 Rushes task [9]. Two of the primary evaluation measures were the fraction of the labeled inclusions found in the summary and whether the summary was easy to understand. In terms of both easiness and inclusions, our performance was in the middle of the 24 systems evaluated [9]. To discern whether we did better at detecting inclusions related to camera motion than other sorts of inclusions, we computed the mean fraction of inclusion of only those that contained the camera-motion related words: pan, zoom, and tilt. 62 of the 484 evaluated inclusions, from 25 of the 42 videos, pass this filter. Evaluated on this subset our mean inclusion ratio (we count only the final evaluation from each assessor to reproduce the NIST summary results) is nearly unchanged, but our relative performance goes from 14^{th} to 8^{th} overall, edging out both of the baselines in the process. See Figure 5. It is interesting to note that only 2 groups perform better on this subset and the great majority perform notably worse.

Each of the summaries were judged by three different judges, and a subset were judged twice by each judge. Some of the videos appeared to be more difficult to judge, and we noted that there can be quite a range in judgements, even by the same judge. For example, the inclusion scores by the judges for videos MRS155534, MS216210, and MRS042543 are shown in Table 1. Note the range of scores for a video, even by the same judge.

Two related measures that we thought would be useful are the amount of time spent judging the inclusions and the amount of time paused during judging, both relative to the summary duration. These would be quantitative measures of whether the judges could rapidly understand what was in an inclusion. Our performance was top-6 based on mean of the judgement-time based measure and top-8 based on mean of the pause-time based measure.

We think handling dynamic segments separately from static segments was a good approach. However, in retrospect, a more detailed analysis of local motion within a segment might help to differentiate when a person is standing and talking versus, say, walking into a room, which would improve our inclusion performance. We used three second static segments in our summaries, thinking that anything shorter would be hard to understand. However, the results indicate that the judges for the task, who were allowed to pause the video, had no problem with one second segments, and so we would use shorter segments, allowing for more segments, in the future.

10. REFERENCES

[1] J.-Y. Bouget. Pyramidal implementation of the lucas kanade feature tracker. description of the algorithm. Technical report, Intel Corporation Microprocessor Research Lab, 2000.

[2] M. Cooper and J. Foote. Scene boundary detection via video self-similarity analysis. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 378–81, 2001.

[3] B. Gunsel, M. Ferman, and A. M. Tekalp. Temporal video segmentation using unsupervised clustering and semantic object tracking. *Journal of Electronic Imaging*, 7:592–604, July 1998.

[4] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.*, 42(1-2):177–196, 2001.

[5] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981.

[6] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14: Proceedings of the 2001.*, 2001.

[7] J.-M. Odobez, D. Gatica-Perez, and M. Guillemot. Video shot clustering using spectral methods. In *Int. Workshop on Content-based Multimedia Indexing (CBMI)*, 2003.

[8] Open source computer vision library. http://www.intel.com/technology/computing/opencv/.

[9] P. Over, A. F. Smeaton, and P. Kelly. The TRECVID 2007 BBC rushes summarization evaluation pilot. In *Proceedings of the TRECVID Workshop on Video Summarization (TVS'07)*, pages 1–15, New York, NY, September 2007. ACM Press.