AT&T RESEARCH AT TRECVID 2007

Zhu Liu, Eric Zavesky, David Gibbon, Behzad Shahraray, Patrick Haffner

AT&T Labs – Research
200 Laurel Avenue South
Middletown, NJ 07748
{zliu, ezavesky, dcg, behzad, haffner}@research.att.com

ABSTRACT

AT&T participated in two tasks at TRECVID 2007: shot boundary detection (SBD) and rushes summarization. The SBD system developed for TRECVID 2006 was enhanced for robustness and efficiency. New visual features are extracted for cut, dissolve, and fast dissolve detectors, and SVM based verification method is used to boost the accuracy. The speed is improved by a more streamlined processing with on-the-fly result fusion. We submitted 10 runs for SBD evaluation task. The best result (TT05) was achieved with the following configuration: SVM based verification method; more training data that includes 2004, 2005, and 2006 SBD data; no SVM boundary adjustment; training SVM with high generalization capability (e.g., a smaller value of C).

As a pilot task, rushes summarization aims to show the main objects and events in the raw material with least redundancy while maximizing the usability. We proposed a multimodal rushes summarization method that relies on both face and speech information.

Evaluation results show that the new SBD system is highly effective and the human centric rushes summarization approach is concise and easy to understand.

I. INTRODUCTION

TRECVID started as a video track of TREC (Text Retrieval Conference) in 2001 to encourage research in automatic segmentation, indexing, and content-based retrieval of digital video and in 2003 it became an independent evaluation. TRECVID 2007 contains three fundamental tasks: shot boundary detection (SBD), high-level feature extraction, and search (interactive, manually-assisted, and/or fully automatic), and one new pilot task: rushes summarization. AT&T submitted results for two tasks: shot boundary detection and rushes summarization.

Shot boundary detection has been widely studied for the last decade; early work can be found in [1-4]. TRECVID further stimulates the interest and effort in a much broader research community. AT&T's SBD system achieved good results in TRECVID 2006 [5], and this year, we further enhance the existing system. Three major improvements are: 1) New visual features are extracted for cut, dissolve, and fast dissolve detectors, 2) Support vector machine (SVM) based verification method is used to boost the accuracy and robustness of cut and fast dissolve detectors, 3) SBD processing is more streamlined with on-the-fly result fusion with low latency and an implementation of the algorithm in a Microsoft DirectShow filter to take advantage of a highly efficient MPEG codec. Evaluation results show that our SBD system is very time effective and accurate.

Rushes summarization is new pilot task in TRECVID 2007. Rushes are the raw material (extra video, B-rolls footage) used to produce a broadcast program. Rush material may consist of as much as 20 to 40 times the amount of material actually used in the finished product. Video summarization is a very interesting yet challenging task with recent work found in [6, 7]. We adopt a multimodal approach for rushes summarization. The system relies on both speech and face information to create a human centric video summary. A shot clustering algorithm is applied to remove the content redundancy. Evaluation results show that the summaries we created are concise and easy to understand.

This paper is organized as follows. Section II gives a detailed description of the shot boundary detection system. Section III addresses our work on the rushes summarization. Evaluation results are also presented and discussed in these sections respectively. Finally, we draw our conclusions in Section IV.

II. SHOT BOUNDARY DETECTION

2.1 Overview

Fig. 1 shows the high level diagram of the SBD system. There are three main components in our SBD system: visual feature extraction, shot boundary detectors, and result fusion. The top level of the algorithm runs in a loop, and every loop processes one video frame. Each new frame and the associated visual features are saved in circular buffers. The loop continues until all frames in the MPEG file are processed.

Given the wide varieties of shot transitions, it is difficult to handle all of them using a single detector. Our system

adopts a "divide and conquer" strategy. We devised six independent detectors, targeting for six dominant types of shot boundaries in the SBD task. They are cut, fade in, fade out, fast dissolve (less than 5 frames), dissolve, and wipe. Essentially, each detector is a finite state machine (FSM), which may have different numbers of states. Finally, the results of all detectors are fused and the overall SBD result is generated in the required format.

The SBD system in 2007 brought a few enhancements to the 2006 system. We employed more visual features and adopted data driven approaches in cut, fast dissolve, and dissolve detectors. In this section, we will focus on the new components. Interested readers may find more details of the 2006 system in [5].

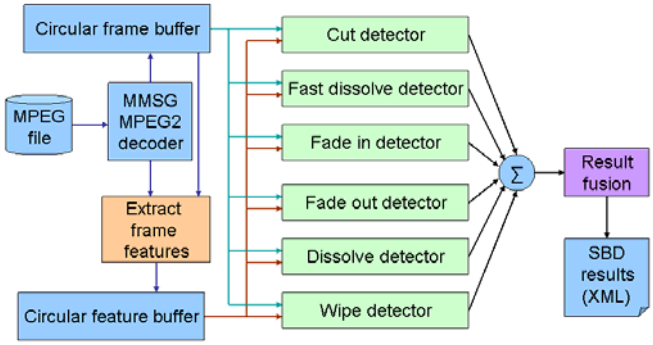


Fig. 1. Overview of the SBD system

2.2 Feature Extraction

For each frame, we extracted a set of visual features, which can be classified into two types: intra-frame and inter-frame visual features. The intra-frame features are extracted from a single, specific frame, and they include color histogram, edge, and related statistical features. The inter-frame features rely on the current frame and one previous frame, and they capture the motion compensated intensity matching errors and histogram changes.

Fig. 2 illustrates how these visual features are computed. The visual features are extracted from a central portion of the picture, which we called the region of interest (ROI). The ROI is marked by a dashed rectangle in Fig. 2, overlaid on the original image. The size of ROI is 288x192 pixels.

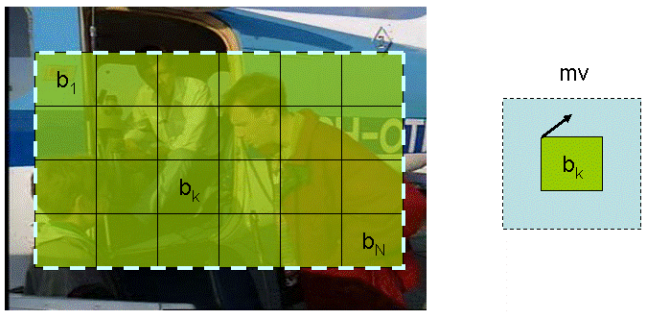


Fig. 2. Visual feature extraction

Within the ROI, we extract the histogram of red, green, blue, and intensity channels and compute a set of statistics, including the mean, the variance, the skewness (the 3rd order moment), and the flatness (the 4th order moment). For each pixel in the ROI, we compute its discontinuities in the horizontal (with respect to vertical) direction by Sobel operators [8]. If the value is higher than a preset threshold, the pixel is labeled as horizontal (respectively, vertical) edge pixel. Finally, we use the ratio of the total number of horizontal (respectively, vertical) edge pixels to the size of ROI as an edge based feature.

The temporal derivative (delta) of a feature (e.g., histogram mean) is fitted by a second-order polynomial to make it smooth. The delta values of histogram mean, variance, and dynamic range are computed as additional visual features.

Motion features are extracted based on smaller blocks within the ROI. Specifically, in Fig. 2, we split the ROI into 24 blocks (6 by 4), each with the size 48x48 pixels. The search range of motion vector for each block is set to 32x32. Either an exhaustive search for better accuracy or a hierarchical search for higher efficiency is used to estimate a block's motion vector. The motion features for each block, e.g., block k, include the motion vector (MV_k), the best matching error (ME_k), and the matching ratio (MR_k). The matching ratio is the ratio of the best matching error with the average matching error within the searching range, and it measures how good the matching is. The value is low when the best matching error is small and the block has significant texture. Based on the motion features of all blocks, we select the dominant motion vector and its percentage (the ratio of the number of blocks with this motion vector to the total number of blocks) as frame level features. We then rank all ME_k (resp. MR_k), and compute the order statistics, including the mean, ME_A; the median, ME_M; the average of the top 1/3, ME_H; and the average of the bottom 1/3, ME_L (resp. MR_A, MR_M, MR_H, MR_L). These features are effective in differentiating the localized visual changes (e.g., foreground changes only) from the frame wised visual changes. For example, high MR_H with low MR_A indicates a localized transition.

Based on the motion vectors of all blocks, we can determine the dominant motion vector and the percentage of blocks whose motion vectors are the same as the dominant one. If the dominant motion vector is non-trivial and the percentage is significant (e.g., more than 1/3), we set the global motion flag to be true for the frame, otherwise, false.

To cope with the false shot boundaries introduced by zooming effects, we developed a simple yet effective zooming detector. Fig. 3 illustrates the method we adopted to detect zooming. Frames i and i-1 are two adjacent frames. For each frame, we extracted the intensity values for the center row (horizontal bars h_i and h_{i-1}), and those for the center column (vertical bars v_i and v_{i-1}). Dynamic programming is used to search the optimal match between

the two horizontal bars, where the centers of the two bars are aligned. Fig. 3 shows an example of zooming out, and the best match path (MP_h) is marked in a solid line. The dotted solid line shows a possible match path for a case of zooming in. The tangent value of the angle of the match path (θ) is defined as the zooming factor. While zooming out, the factor is less than 1.0, and while zooming in, the factor is greater than 1.0.

Using the single pixel wide horizontal (vertical) bars, we can find possible horizontal (vertical) zooming factors efficiently. Based on the optimal horizontal and vertical matching paths, the entire frames are used to verify the zooming decision. For the case of zooming out, frame i-1 is shrunk and compared to corresponding portion in frame i. The verification for the case of zooming in is similar. If the overall matching error is small enough, we set the zooming flag of current frame to be true, otherwise, false.

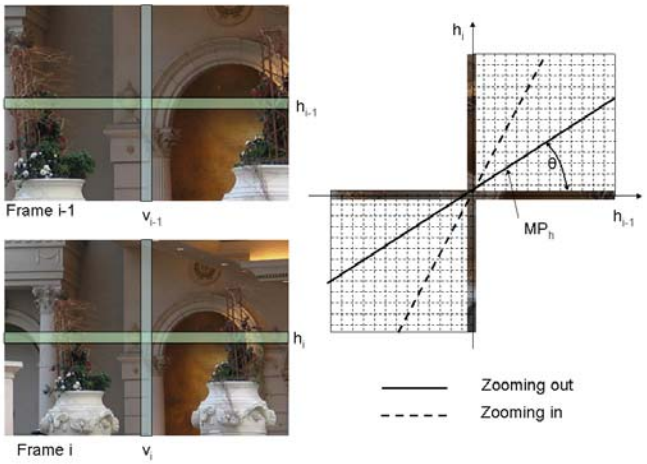


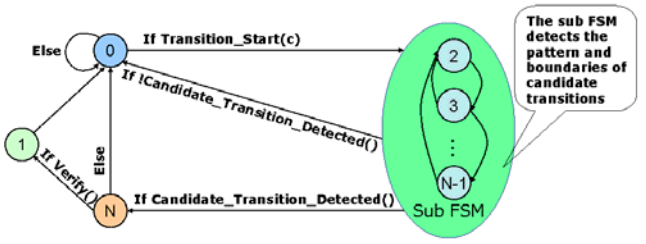
Fig. 3. Zooming detection

2.3 Shot Boundary Detectors

The AT&T SBD system contains 6 detectors, which detect 6 common shot boundaries: cut, fast dissolve (less than 5 frames), fade in, fade out, dissolve, and wipe. These 6 types of transitions cover most shot transitions in TRECVID sequences and they can be detected relatively reliably.

Fig. 4 illustrates the general FSM structure for all shot boundary detectors. State 0 is the initial state. When the transition start event is detected, the detector enters the sub FSM, which detects the target transition pattern, and locates the boundaries of the candidate transition. If the sub FSM fails to detect any candidate transition, it returns to state 0, otherwise, it enters state N. State N further verifies the candidate transition with more strict criteria, and if the verification succeeds, it transfers to state 1, which indicates that a transition is successfully detected, otherwise, it returns to the initial state. Although the six detectors share the same general FSM structure, their intrinsic logic and complexity is quite different.

The state of the FSM is determined by a set of state variables. There are three basic state variables that are common for all FSMs: state_id, which is the state of current FSM, start_frame, which is the last frame of previous shot, end_frame, which is the first frame of the new shot. Some detectors may have an additional state variable to track an adaptive threshold value used for determining the state transitions.



State	Description
0	Initial state
1	Transition detected state
2 ~ N-1	Candidate transition detection states
N	Verification state

State Variables
Current state
Transition start frame
Transition end frame
Detector specific variables

Fig. 4. General FSM for transition detectors

In the 2007 SBD system, we improved three detectors: cut, fast dissolve, and dissolve detectors. In the remainder of this section, we will focus on these three detectors.

2.3.1 Cut detector

To improve the effectiveness of the cut detector, we introduced more features for cut verification. For a candidate cut boundary at (pref, postf), these features are extracted from a verification window that starts at pref-2, and ends at postf+2. Table I lists all 22 features for cut verification. How these features are computed is also briefly described in Table I. The last feature, IsMono, is useful for the cases where two regular shots are connected by a sequence of black/white screens. The TRECVID reference labels consider this case a gradual transition, instead of two adjacent cuts.

Table I. Features for cut verification

Feature	Computation
Corr_c	Correlation between frames pref and postf: Corr(pref, postf)
ME_H_c	ME_H between pref and postf: ME_H(pref, postf)
ME_L_c	ME_L between pref and postf: ME_L(pref, postf)
ME_M_c	ME_M between pref and postf: ME_M(pref, postf)
HistDist_c	Histogram distance between pref and postf: HistDist(pref, postf)
Corr_pre	Max{Corr(pref-2, postf), Corr(pref-1, postf)}
ME_H_pre	Min{ME_H(pref-2, postf), ME_H(pref-1, postf)}
ME_L_pre	Min{ME_L(pref-2, postf), ME_L(pref-1, postf)}
HistDist_pre	Min{HistDist(pref-2, postf), HistDist(pref-1, postf)}
ME_H_st	Max{ME_H(pref-2, pref-1), ME_H(pref-1, pref)}

HistDist_st	Max{HistDist(pref-2, pref-1), HistDist(pref-1, pref)}
Corr_st	Min{Corr(pref-2, pref-1), Corr(pref-1, pref)}
Corr_post	Max{Corr(pref, postf+1), Corr(pref, postf+2)}
ME_H_post	Min{ME_H(pref, postf+1), ME_H(pref, postf+2)}
ME_L_post	Min{ME_L(pref, postf+1), ME_L(pref, postf+2)}
HistDist_post	Min{HistDist(pref, postf+1), HistDist(pref, postf+2)}
ME_H_end	Max{ME_H(postf, postf+1), ME_H(postf+1, postf+2)}
HistDist_end	Max{HistDist(postf, postf+1), HistDist(postf+1, postf+2)}
Corr_end	Min{Corr(postf, postf+1), Corr(postf+1, postf+2)}
MaxVar	Maximum variance within the verification window {pref-2, pref-1, ..., postf+1, postf+2}
ME_H_ratio	Ratio of ME_H to the adaptive matching error threshold (a state variable of cut detector)
IsMono	True if either pref or postf is monochrome

In the 2006 SBD system, cut verification was mainly a threshold based method. In the 2007 system, we adopted a support vector machine (SVM) based verification engine using all the above mentioned 22 features as input. More details about SVM training can be found in Section 2.3.4.

2.3.2 Fast dissolve detector

In the 2007 system, there are two main improvements for the fast dissolve detector: more visual features and SVM based verification approach.

Let X, Y, and Z denote the start_frame, end_frame, and an intermediate frame of the fast dissolve transition. We require that the duration of the fast dissolve transition be less than 5 frames, so it is reasonable to assume that there is no motion involved in the transition. With this assumption, Z can be written as a linear combination of X and Y, $Z = \alpha X + (1 - \alpha)Y$, where $0 \le \alpha \le 1$. The value of α can be determined by a min square error criteria, and the minimum estimation error is denoted by EE_L. To measure how accurate the estimation is, we also compute the energies of the difference images X-Z and Y-Z, denoted by ED_X and ED_Y. The ratio of $\min(ED_X, ED_Y)$ to EE_L is used as the confidence (Conf) of the linear estimation.

Table II lists all features used for fast dissolve verification, where the candidate boundary is (pref, postf). How to compute these features is also briefly described in the table. These features are designed to capture the distinctive nature of a fast dissolve transition. The features Zoom_p and Motion_p aide in removal of the false fast dissolves that are induced by zooming or camera motion.

Table II. Features for fast dissolve verification

Feature	Computation
Corr_f	Correlation between frames pref and postf: Corr(pref, postf)
HistDist_f	Histogram distance between pref and postf: HistDist(pref, postf)
ME_H_f	ME_H between pref and postf: ME_H(pref, postf)
ME_L_f	ME_L between pref and postf: ME_L(pref, postf)
MaxError	Max{EE_L(pref+1), ..., EE_L(postf-1)}
MinConf	Min{Conf(pref+1), ..., Conf(postf-1)}
MaxAlpha	Max{|α (pref+1)-0.5|, ..., |α (postf-1)-0.5|}
MaxVar	Maximum intensity variance: Max{Var(pref), ..., Var(postf)}
ME_H_r	Min{ ME_H(pref, pref+1), ..., ME_H(postf-1, postf)}/ Max{ME_H(pref-2, pref-1), ME_H(pref-1, pref), ME_H(postf, postf+1), ME_H(postf+1, postf+2)}
HistDist_r	Min{ HistDist (pref, pref+1), ..., HistDist (postf-1, postf)}/ Max{ HistDist (pref-2, pref-1), HistDist (pref-1, pref), HistDist (postf, postf+1), HistDist (postf+1, postf+2)}
IsLowVar	True f perf or postf is a low variance frame
Zoom_p	Percentage of frames whose zooming flags are true
Motion_p	Percentage of frames whose global motion flags are true
Var_r	Min{Var(pref+1), ..., Var(postf-1)} / Min{Var(pref), Var(postf)}

In the 2006 system, fast dissolve verification was a heuristic rule based method. In 2007 system, we adopted a SVM based verification engine. All the above mentioned 14 features are used as SVM input. More details about SVM training can be found in Section 2.3.4.

2.3.3 Dissolve detector

The majority of the gradual transitions are dissolves. Therefore, the performance of the dissolve detector largely determines the performance of detecting gradual transition. In the 2006 system, we developed a set of 66 features for dissolve verification, which were proven to be very effective. For the 2007 system, we added two new features based on the candidate dissolve boundary: the percentage of frames with zooming flag set and the percentage of frames with the motion flag set (similar to Zoom_p and Motion_p described in Table. II). These two features help to reduce the false positives introduced by motion and zooming.

Similar to the 2006 system, we used an SVM based verification method for a dissolve detector. More details about SVM training are given in the next section.

2.3.4 SVM Models

Support vector machines are now standard for fast and robust classification. While this discriminative classifier greatly reduces training time by analyzing only marginal samples, care must be given to the training parameters and underlying kernel used in an SVM. For our experiments, we evaluated radial basis functions in a 3-fold validation process. We searched 7 linear settings and 70 RBF settings with random subsets of our training set split into 80/20

training/testing partitions. All features are globally normalized to one before they are analyzed by the SVM.

2.4 Fusion of Detector Results

In the 2006 system, fusion of detector results occurs when all frames are processed. We first sorted the list of raw results by their starting frames and merge all overlapped fade out and fade in transitions into a single FOI transition. Then the overlapped transitions are removed based on their priorities. The adopted priority order is (from highest to lowest) FOI, dissolve, fast dissolve, cut, and wipe. The final step is to map the system types into two categories: cut and gradual. All shot boundaries except cuts are mapped into gradual. The 2007 SBD system kept the same logic for result fusion, but re-implemented it such that the fusion is conducted on-the-fly with low latency. Now, it is a one-pass process instead of the two-pass process in the 2006 system.

2.5 Evaluation Results

The TRECVID 2007 SBD evaluation data contains 17 sequences, totally about 7 hours. There are both color and black/white videos in these sequences. Compared to the evaluation data of TRECVID 2006, the 2007 data has more cuts (about 90%, compared to 48.7% in 2006) and longer shots (275 frames/shot, compared to 158 frames/shot in 2006). Our evaluation was conducted on a Windows 2003 server with dual Intel Xeon 5110 1.6GHz CPUs, and it took less than 2 hours 5 minutes to finish each run.

Table III shows the 10 runs we submitted for the shot boundary detection task. For runs 1, 2, 9, and 10, the cut verification is threshold based rules, and for the other runs, cut verification is SVM based. For all runs, the dissolve verification and fast dissolve verification are SVM based. In terms of the training data for the cut, dissolve, and fast dissolve SVM models, runs 1 to 4 rely on the SBD evaluation data in TRECVID 2005 and 2006; runs 5-10 use the SBD data in TRECVID 2004 also.

As a tradeoff between the precision and the recall, we introduced a bias for the boundary used in SVM classification. Adjusting the SVM boundary to include more negative samples (e.g., a value of -0.1) increases the recall rate and decreases the precision. Table 1 lists the SVM bias value used in different runs. A value of zero means no bias is enforced. The C value in SVM training controls the tradeoff between the training error and the SVM margin, which affects the generalization capability of SVM. Higher C reduces training error, but decreases the margin at the same time. For different runs, we use different values of C. The last column of table III basically gives an idea of the C values we used.

The best results of AT&T's submissions in different categories are shown in Table IV. For example, run 5 achieves the best overall result and the best gradual results among the 10 runs. Run 3 achieved the best cut and frame based gradual detection result.

Gradual transition detectors provide good performance, which enabled the AT&T system to be one of the top contenders. The frame based gradual transition performance of the AT&T system also achieved good performance, which means that the proposed gradual transition (mainly the dissolve) boundary location approaches are very effective.

Table III. AT&T's 10 submissions for SBD

Run	SVM based cut verification	Training dataset	SVM boundary adjustment	SVM generalization
1	no	2005 & 2006 SBD data	-0.1	High
2			0.0	
3	yes		0.0	
4			-0.03	
5		2004, 2005 & 2006 SBD data	0.0	
6			-0.03	
7			-0.10	
8			0.0	Low
9	no		-0.10	
10			-0.03	High

Table IV. The best runs of AT&T's SBD submissions

Run	Category	Performance (%)		
		Recall	Precision	F-Measure
5	**Overall**	**95.6**	**95.4**	**95.5**
	Cut	97.9	96.6	97.2
	Gradual	**70.9**	**80.2**	**75.3**
	Frame based	71.8	93.3	81.2
3	Overall	95.5	95.3	95.4
	Cut	**97.7**	**96.8**	**97.2**
	Gradual	70.4	78	74
	Frame based	**74.2**	**93.3**	**82.7**

III. RUSHES SUMMARIZATION

3.1 Overview

Rushes are the raw material (extra video, B-rolls footage) used to produce a video broadcast. Rushes material may consist of as much as 20 to 40 times the amount of material actually used in the finished product. The rushes summarization task attempts to construct a short video clip that includes the major objects and events of the video to be summarized.

Fig. 5 shows the diagram of the proposed rushes summarization method. Our system adopts a multimodal approach for rushes summarization. The system relies on speech and face information to create a human centric video summary. The video is first segmented into shots and three keyframes are selected for each shot. Based on the dissimilarities of corresponding keyframes, we compute the

shot distance matrix, and apply a hierarchical agglomerate clustering (HAC) algorithm to remove redundancy. For each cluster, the longest shot is kept, and the total budget (less than 4% of the original duration) is assigned to all chosen shots based on their durations. Within each shot, we pick one continuous segment that contains most speech and face occurrences. The final video summary is simply the concatenation of all selected segments in their original time order.

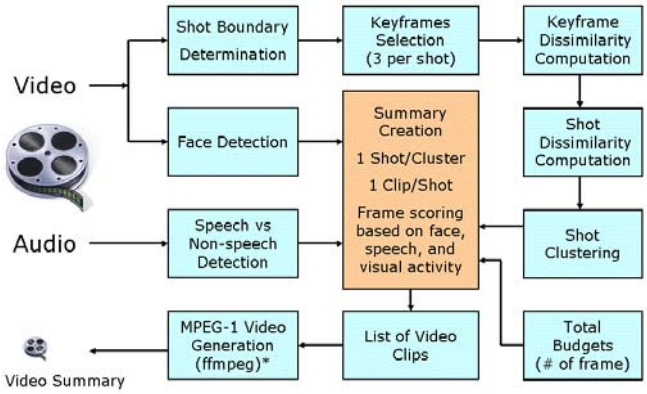


Fig. 5. Diagram of rushes summarization algorithm

3.2 Video Summarization Algorithm

In this section, we give more details on the components of shot clustering, face detection, speech detection, and the summary creation.

3.2.1 Shot clustering

We use our SBD system developed for TRECVID 2006 to detect the shot boundaries in the video. Within a shot, since there may be zooming, panning, tilting, and other camera motions, the visual content can change significantly. To deal with the content dynamics within a shot, we pick 3 keyframes that are uniformly sampled from the shot to represent the content of the shot. For example, in a six frame shot, we pick the first frame, the third frame, and the fifth frame as the keyframes.

Once we have all keyframes for each shot, we compute the distance between two shots based on the distances between all pairs of keyframes, one from each shot. Suppose that shot X has three keyframes $\{x_1, x_2, x_3\}$, and shot Y has three keyframes $\{y_1, y_2, y_3\}$. We use the matching error ME_A (see section 2.2) between keyframes x_i and y_j as their distance d_{ij}. Then the shot distance D(X, Y) can be written as a weighted summation of all keyframe distances.

$$D(X,Y) = \sum_{i=1}^{3}\sum_{j=1}^{3} w_{ij} d_{ij} ,$$

where w_{ij} satisfies the following criteria.

$$w_{ij} \geq 0$$

$$\sum_{i=1}^{3} w_{ij} = 1$$

$$\sum_{j=1}^{3} w_{ij} = 1$$

There exists a set of weight $\{w_{ij}\}$ that minimizes D(X, Y). Interested readers can find more details in [10]. We use the minimum value as the distance between shots X and Y.

$$D(X,Y) = \min_{\{w_{ij}\}} \sum_{i=1}^{3}\sum_{j=1}^{3} w_{ij} d_{ij} .$$

Having determined the shot distances, we apply hierarchical agglomerative clustering (HAC) algorithm to cluster all shots. A threshold of 0.5 is used to terminate the clustering algorithm. To remove the redundant content, we only pick the longest shot from each cluster to be part of the video summary.

3.2.2 Face detection

Face detection and tracking has drawn a lot of attention in the last two decades. Some recent work can be found in two excellent overview papers [11, 12]. In this work, we adopt the face detection module in OpenCV [13], which implements the AdaBoost based face detection algorithm proposed by Viola and Jones in [14].

To track faces in MPEG-1 sequence, we package the OpenCV face detection module into a Windows DirectShow filter. We then construct a software network of filters to detect faces in each frame that are passed to the video summary creation module.

3.2.3 Speech detection

For speech and non-speech classification, we segment an audio signal into audio clips, which are 3 seconds long on average and each clip consists of overlapping frames. The features of each audio clip are determined from the sub-features of the associated frames. Each frame is 32 millisecond (ms) long, overlapping with the previous one by 22 ms. Eight features are computed for each frame. They are root mean square volume, zero crossing rate, pitch, frequency centroid, frequency bandwidth, and 3 energy ratios in subbands. We extract 14 features for each audio clip based on frame-level features. The 14 clip-level features are 1) volume standard deviation (VSD), 2) volume dynamic range (VDR), 3) volume undulation (VU), 4) non-silence ratio (NSR), 5) standard deviation of zero crossing rate (ZSTD), 6) 4-Hz modulation energy (4ME), 7) standard

deviation of pitch (PSTD), 8) smooth pitch ratio (SPR), 9) non-pitch ratio (NPR), 10) frequency centroid (FC), 11) frequency bandwidth (BW), 12-14) energy ratio in subbands 1 - 3 (ERSB1, ERSB2, and ERSB3). For a detailed description of these features, please refer to [4]. Based on our prior experiments, Gaussian mixture models (GMMs) with 4 mixtures provide good performance of speech and non-speech classification. In this paper, we assume that the covariance matrix of each Gaussian mixture is diagonal.

3.2.4 Video summary creation

The video summary creation module is the core component that combines results from other modules and creates the recipe for a video summary.

One requirement of rushes summarization is that the duration of a video summary is at most 4% of the original video to be summarized. This gives us an overall budget, B (number of frames), to be assigned to different shots that are picked by shot clustering. Assume that we have N shots, $\{S_1, S_2, \ldots, S_N\}$, and shot S_i contains frames $\{f^i_1, f^i_2, \ldots f^i_{Di}\}$, where D_i is the number of frames in S_i. For each frame f^i_j, we assign an importance value v^i_j for it as follows,

$$v^i{}_j = \begin{cases} ME_A & \text{if no face or speech} \\ ME_A + 1 & \text{if speech \& no face} \\ ME_A + 2 & \text{if speech \& one face} \\ ME_A + 3 & \text{if speech \& more than one face} \end{cases}$$

The importance value of shot S_i, denoted by $V_{i,}$, is the maximum frame important value within the shot. The budget for shot b_i is given by the following formula,

$$b_i = \frac{V_i \cdot D_i}{\sum_{j=1}^{N} V_j \cdot D_j} B$$

The last step is to choose one continuous segment for each shot. Our method is straightforward – simply picking the segment with the maximum accumulated frame scores. To remove the beginning scenes with the movie clap board in most of the shots, we ignore the beginning 4 seconds within each shot when we determine continuous segments.

The final video summary is created by concatenating all segments in their original temporal order. The accompanying audio streams are also kept such that the generated video summary provides a richer viewing experience.

3.3 Results

The TRECVID 2007 rushes summarization task provides a development dataset and a test dataset. The development dataset contains 47 MPEG-1 videos, totally 19.5 hours, and the test dataset contains 42 videos, totally about 18 hours. We generate the visual summarization results using the same server that runs the SBD task. Specifically, it is a Windows 2003 server with dual Intel Xeon 5110 1.6GHz CPUs. Overall, it took 37.5 hours to process all the data. Table V shows the detailed processing time used by different components.

Table V. Processing time for rushes summarization (hours)

SBD	Face tracking	Speech detection	Shot clustering	Summary rendition	Other
6	24	1	5.5	0.5	0.5

There are 7 performance measurements for the summarization: 1) Duration of the summary in seconds (DU), 2) Difference between target and actual summary size in seconds (XD), 3) Total time spent judging the inclusions in seconds (TT), 4) Total video play time judging the inclusions (VT), 5) Fraction of inclusions found in summary (IN), 6) Was the summary easy to understand (EA) (1=strongly disagree, 5=strongly agree), 7) Was there a lot of duplicate video (RE) (1=strongly agree, 5 = strongly disagree).

Table VI shows the evaluation results of our submission. One possible reason that our system performs relatively poor for IN is that our approach is more human centric and it does not put enough emphasis on other objects or events. Nevertheless, the TT, EA and RE scores of our system are decent, which means the human centric approach is actually effective in removing redundancy and easy to understand. Figures 6 and 7 draw the EA vs. TT, and EA vs. RE plots, which clearly show the effectiveness of our rushes summary.

Table VI. Evaluation results of rushes summarization

Measure	Mean score	Median score
DU	54.76	59.15
XD	5.11	5.49
TT	95.66	86.33
VT	54.95	59.5
IN	0.38	0.35
EA	3.37	3.33
RE	3.89	4

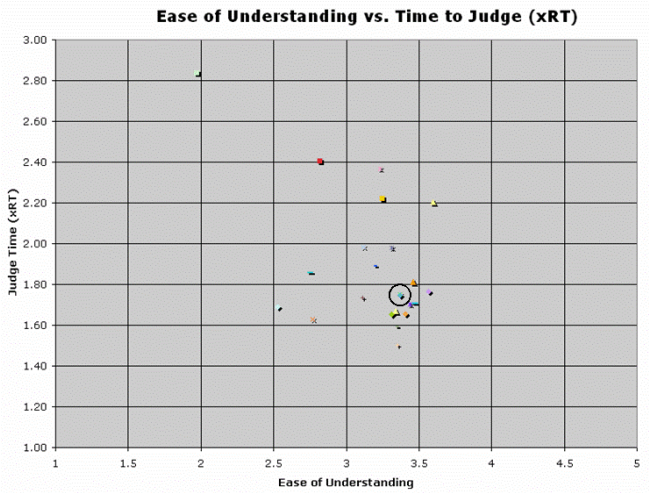


Fig. 6. Ease of understanding vs. Judge time

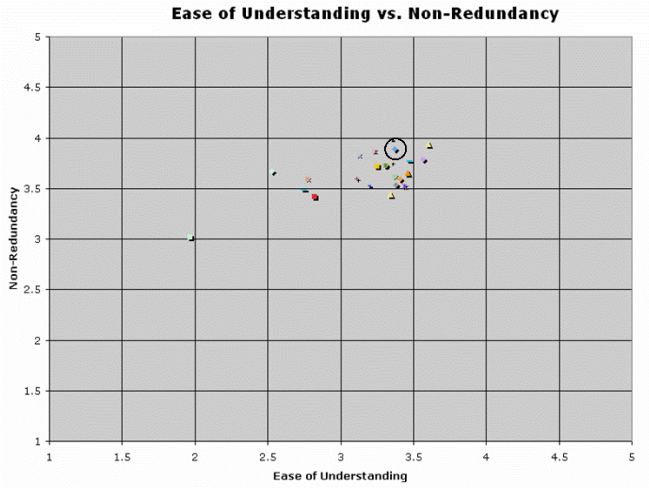


Fig. 7. Ease of understanding vs. Non-redundancy

IV. CONCLUSIONS

In this paper, we reported the AT&T system for TRECVID 2007 evaluation. AT&T participated in two tasks: shot boundary detection and rushes summarization. We enhanced the 2006 SBD system by utilizing more visual features and employing SVM based verification for cut, dissolve, and fast dissolve detectors. The proposed rushes summarization method is a human centric approach, where both speech and face information is exploited. The evaluation results show that the new SBD algorithm is effective and promising, and the proposed rushes summarization method is effective and easy to understand.

V. REFERENCES

[1] H. J. Zhang, A. Kankanhalli, s. W. Smoliar, "Automatic Partitioning of Full-motion Video," ACM Multimedia System, Vol. 1, No. 1, pp. 10-28, 1993.

[2] B. L. Yeo and B. Liu, "Rapid Scene Analysis on Compressed Video," IEEE Transactions on Circuits and Systems for Video Technologies, 5(6), pp. 533-544, 1995.

[3] B. Shahraray, "Scene Change Detection and Content-based Sampling of Video Sequences," in Digital Video Compression: Algorithms and Technologies 1995, *Proc. SPIE 2419*, February 1995.

[4] Y. Wang, Z. Liu, and J. Huang, "Multimedia Content Analysis Using Audio and Visual Information," *IEEE Signal Processing Magazine*, pp.12-36, Nov. 2000.

[5] Z. Liu, D. Gibbon, E. Zavesky, B. Shahraray, P. Haffner, "AT&T Research at TRECVID 2006," TRECVID 2006 Workshop, Gaithersburg, MD, Nov. 13-14, 2006.

[6] Z. Xiong, A. Divakaran, Y. Rui, and T. Huang, *A Unified Framework for Video Summarization, Browsing & Retrieval: with Applications to Consumer and Surveillance Video*, Academic Press, 2005.

[7] J. Pan, H. Yang, C. Faloutsos, "MMSS: Multi-modal Story-Oriented Video Summarization," *ICDM 2004*, Brighton, UK, Nov. 1-4, 2004.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing,* Addison Wesley, 1993.

[9] D. Gibbon, Z. Liu, and B. Shahraray, "The MIRACLE video search engine," *IEEE CCNC*, Jan. 2006.

[10] Z. Liu, Q. Huang, "A New Distance Measure for Probability Distribution Function of Mixture Type," *ICASSP-2000*, Istanbul, Turkey, June 5-9, 2000.

[11] M. Yang, D. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 24, No. 1, Jan. 2002.

[12] W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips, "Face Recognition: A Literature Survey," *ACM Computing Surveys*, 2003, pp. 399-458.

[13] Open Source Computer Vision Library, http://www.intel.com/technology/computing/opencv/

[14] Paul Viola and Michael Jones, "Robust Real-time Object Detection," IJCV 2001.