University of Central Florida at TRECVID 2004

Yun Zhai, Xiaochun Chao, Yunjun Zhang, Omar Javed, Alper Yilmaz
Fahd Rafi, Saad Ali, Orkun Alatas, Saad Khan, and Mubarak Shah

Computer Vision Laboratory
University of Central Florida
Orlando, Florida U.S.

1 Introduction

This year, the Computer Vision Group at University of Central Florida participated in two tasks in TRECVID 2004: High-Level Feature Extraction and Story Segmentation. For feature extraction task, we have developed the detection methods for “Madeleine Albright”, “Bill Clinton”, “Beach”, “Basketball Scored” and “People Walking/Running”. We used the adaboost technique, and has employed the speech recognition output in addition to visual cues. In story segmentation, we used a 2-phase approach. The video is initially segmented into coarse segmentation by finding the anchor persons. The coarse segmentation is then refined in the second phase by further detecting weather and sports stories and merging the semantically related stories, which is determined by the visual and text similarities.

2 Feature Extraction

We have submitted five features: Madeleine Albright, Bill Clinton, Beach, Basketball Scoring, and People Walking/Running. Section 2.1 describes the method for detecting person X (Albright and Clinton); section 2.2 describes the beach detector; section 2.3 presents the detection for the basketball scored; finally, section 2.4 presents the method for classifying the people walking/running shots.

2.1 Finding Person X in Broadcast News

Our approach to find a specific person X combines text cues from the given transcripts, face detection from key frames and face recognition. Figure 2.1 gives the overview of our algorithm.

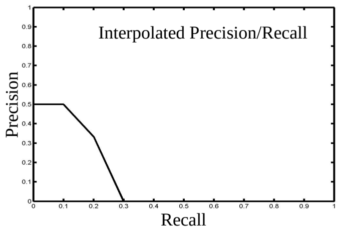
At the core of the text reinforcement algorithm is the “wordtime” data provided by the CMU Infomedia team. These wordtime files contain the time in milliseconds beginning from the start of the broadcast (videos) at which each particular word is spoken. The speech information in news broadcast videos is highly correlated with the visual data. This makes the wordtime data a powerful resource to help classifying video and detect features. In the case of detecting shots containing footage of Bill Clinton and/or Madeleine Albright, we observed from the training data that the occurrence of words “Bill Clinton” and “Madeleine Albright” is highly correlated with their appearance in the video. This is a natural consequence of either the news caster or the reporter introducing these two personalities. We therefore scanned the wordtime data for the relevant words and extracted the time, denoted as the hit time, at which they are spoken. The hit times are used to label the shots spanning the duration as candidate shots. These candidate shots then are further tested and refined by a Haar face detector, the details of which are described later.

The words that were scanned as hints for Bill Clinton were “Clinton” or “President”. The results showed that there is no need to scan for the word “Bill” since the former president is almost never introduced as simply “Bill”. In fact, scanning for “Bill” increases the number of false positives. At times Bill Clinton is referred only as the “President”, so we found it necessary to search for the word “President” also. In the case of Madeleine Albright, we scanned the wordtime data for the phrase “Madeleine Albright”.

After extracting all the text cues from the given transcripts, we filter the key frames based on the temporal location of the cues, only maintaining the frames which are within a preset threshold from the cues. According to our experimental results, the text cues give more reliable output than visual cues. Therefore by filtering the key frames based on text cues, we save large amount of computation time for the face detection on the key frames which actually contain no person X.

We assume that if a specific shot contains person X, there should exist a frontal view of the person X in the corresponding key frame. Based on this assumption, face regions were located in the shot’s key frames. The face detector we used is a modified version of the Haar-like feature face detector in OpenCV [1]. It only gives one face region output.

Total shots returned:	19
Total shots with features:	19
Total shots with feature returned:	4
Precision at shots with feature:	0.2105
Average Precision:	0.0899



Precision at n shots:	
5	0.4000
10	0.3000
15	0.2667
20	0.2000
30	0.1333
100	0.0400
200	0.0200
500	0.0090
1000	0.0040
2000	0.0020

Figure 1: Evaluation Results for Madeleine Albright. Values of interpolated precision and recall are plotted in the graph instead of listing them out.

Giving the results of the face detection, we need to recognize the specific person's face. Therefore we build two limited face recognizers for the two specific persons correspondingly based on extracted faces in training data set. We collect k sample faces $\{F_1, F_2, \cdots, F_k\}$ for person x and $n - k$ non-x faces $\{F_{k+1}, F_{k+2}, \cdots, F_n\}$ from the training data set. An Eigenspace is build for those faces $\{F_1, F_2, F_3, \cdots, F_n\}$ and and the Eigenfaces $\{eigF_1, eigF_2, \cdots, eigF_n\}$ are obtained. Next a Support Vector Machine (SVM) classifier is trained from the Eigenface data , and is used to recognize person x from the extracted faces in the testing data.

The valid recognition rate is low because of the following reasons. First of all, the face detector misses a lot of faces, especially faces in small sizes, faces in side view, and faces in crowd. Secondly, the face detector sometimes detects only a part of the face which causes problems in recognition. It might be helpful if we use skin information to extract the complete face based on the face detection results. Finally, we only use key frames instead of multiple frames to extract faces, which is not robust since more frames provide more information. The evaluation results for detecting feature “Madeleine Albright” are shown in Figure 1, and the one for “Bill Clinton” is shown in Figure 3.

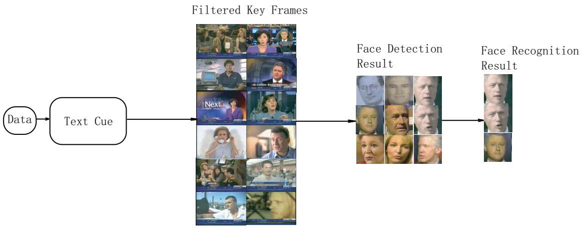
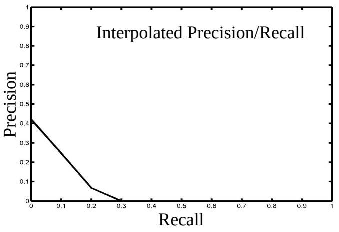


Figure 2: The three steps approach to recognize Person X

Total shots returned:	2000
Total shots with features:	409
Total shots with feature returned:	103
Precision at shots with feature:	0.1516
Average Precision:	0.0536



Precision at n shots:	
5	0.0000
10	0.0000
15	0.2000
20	0.2500
30	0.4000
100	0.3500
200	0.2400
500	0.1320
1000	0.0780
2000	0.0515

Figure 3: Evaluation Results for Bill Clinton.

2.2. Beach

The Beach feature was extracted from key frames using the color information and its spatial context. Low level features include color correlograms and color histograms in RGB and HSV color spaces. These were computed on entire images. Moreover, the images were divided into regions, and the regional color histograms were also computed. The regional histograms were used to capture the fact that the scenes with a beach has a very high probability of showing sky in the upper portion of the image and sand or water on the lower portion. The above described features were selected because they were the most discriminatory among all the features tested for the beach detection. Analysis through adaboost classifiers was carried out to make the selection.

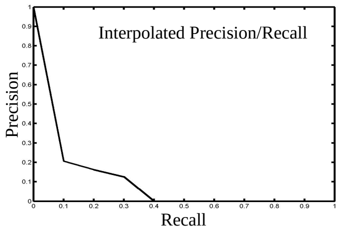
The Adaboost [2] frame work was used to train the classifier. The weak classifiers were simple thresholds on value of each dimension of the feature vector.

Beach in an image can appear in different settings which would affect the value of spatial color features. So we selected a subset of training examples from the TRECVID 2003 data set which conformed to a general beach setting (sand and water occupying substantial area). These hand picked examples were augmented by a data set collected from the internet. This was necessary due to the difference in the definition of the feature and the annotated data provided. The TRECVID 2003 training images included some obviously ”difficult” beach images which were excluded to avoid over fitting of our trained classifier. A very important factor in the selection of images for positive samples was the fact that the training and testing data had colors much different from what one would expect in images of beaches from high quality sources, e.g., the sky was more close to grey than blue in all videos.

The images labelled as beaches by the boosted classifier were then tested for motion. Images with high motion content were removed as one would not be expect a beach scene to have a high motion.

Finally, images were ordered using heuristics on relative

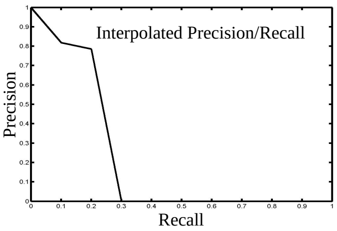
Total shots returned:	1183
Total shots with features:	374
Total shots with feature returned:	130
Precision at shots with feature:	0.1711
Average Precision:	0.0635



Precision at n shots:	
5	0.4000
10	0.2000
15	0.2000
20	0.2500
30	0.2667
100	0.2300
200	0.2000
500	0.1560
1000	0.1210
2000	0.0650

Figure 4: Evaluation Results for Beach.

Total shots returned:	94
Total shots with features:	103
Total shots with feature returned:	37
Precision at shots with feature:	0.3592
Average Precision:	0.2632



Precision at n shots:	
5	1.0000
10	0.8000
15	0.7333
20	0.8000
30	0.7667
100	0.3700
200	0.1850
500	0.0740
1000	0.0270
2000	0.0185

Figure 5: Evaluation Results for Basketball Scored.

RGB channel strength in the image. Images with a high blue content were given a higher rank and images with a high green content were given a low rank. Images with high red content were in between. These crude heuristics were based on observations from TRECVID 2003 data. Within these ranks, the images were sorted according to the confidence provided by the boosted classifier for each image. Evaluation result for feature “Beach” is shown in Figure 4.

2.3. Basketball

We adopted a two step approach to detect basketball going through the hoop. The first step was aimed at detecting the basketball game. While the second step determines which of the basketball games detected in the first step contains the ball going through the hoop. The features used for detecting basket ball game were similar to beach detection. We used color correlograms and color histograms as the feature. These features were also image based. The intuition for these features follows from the rich color information present in any basketball game setting which can be exploited effectively for detection purposes.

The Adaboost training framework similar to the beach was employed. We used threshold based weak classifiers for each dimension of the feature vector. The classifier was trained on TRECVID 2003 development data set. A large number of positive examples were added from the internet resources. This enabled the final classifier to have a better performance on a generalized set of basketball games.

The shots returned by the first step were refined by our textual analyzer. At the core of our text reinforcement is the CMU wordtime data. To detect the basketball hoops we observed that when a hoop-passing is made it is very likely that the newscaster will report the match scores. These scores appear in the wordtime data along with the time at which they are spoken. We searched the wordtime files for basketball scores and extracted the time, called the hit time. The hit times are used to label the shots spanning the duration as candidate shots. If the shots returned by boosted classifier are within these candidate shots we label it as a shot with a basket in it. Evaluation result for feature “basketball scored” are shown in Figure 5.

2.4. People Walking/Running

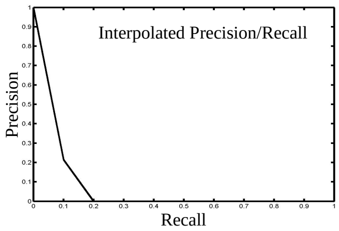
The TRECVID data set contained a large variety of scenes with multiple walking or running people. Persons were observed in a wide range of poses and scales in such scenes. In many cases there was significant person to person occlusion. There was also significant variation in the camera views. However, there were some common attributes of these scenes also. One commonality in many of the the walking/running scenes was the presence of camera motion, since the camera usually followed the walking people to keep them centered in the image. Furthermore, usually in such scenes the faces of the people were at least partially visible. In addition, many sports shots specially those of basketball and football contained multiple walking or running people.

In view of the above given observations, we used several cues to detect the walking/running features. These cues consisted of face detection, skin detection, ego-motion estimation and sports-shot detection.

- Face Detection: Face detection was performed using the Adaboost algorithm. The training set consisted of both front and side view poses as positive face examples. The negative face examples consisted of images of a variety of potential backgrounds. Haar like features obtained from the these examples were used to train the boosted classifier.

- Skin Detection: Skin was detected using the naive Bayes classifier. Normalized RGB color values and the variance of R,G,B in a 3×3 neighborhood were used as features for skin detection. The distribution of these features for both skin and non-skin examples were approximated using histograms. In the test phase a per-pixel (skin or non-skin) decision was made based on

Total shots returned:	2000
Total shots with features:	1695
Total shots with feature returned:	273
Precision at shots with feature:	0.1434
Average Precision:	0.0752



Precision at n shots:	
5	0.8000
10	0.9000
15	0.9333
20	0.9500
30	0.9667
100	0.8100
200	0.5100
500	0.2940
1000	0.1810
2000	0.1365

Figure 6: Evaluation Results for People Walking/Running.

the likelihood ratio of these distributions. Some skin detection results are shown in Figure 2.4.

- Global Motion Estimation: The global motion was estimated by fitting affine parameters to block motion vectors. Large translation parameters indicated a pan,tilt or a translational movement of the cameras.
- Sports Shot Detection: Shots containing basket ball and football play were detected using a combination of color and audio cues. For more detail of this method please see the ‘basket ball going through the hoop’ feature in Section.

A rule based system employing the above mentioned cues was used to make the final walking/running detection decision. A shot was labelled as contained multiple walking or running people if

- multiple faces or significant skin regions were detected in the key-frame along with camera translation, pan or tilt motion, or
- a sport (football, basketball) shot is detected.

System evaluation results are shown in Figure 6.

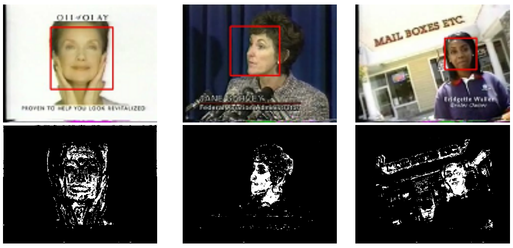


Figure 7: First Row: Some face detection results. Second Row: Skin detection results on the images shown in the first row

3 Story Segmentation

In the news videos, we often observe the following pattern: first, the anchor person appears to introduce some news story. Then, the camera switch to the outside of the studio, for example, the scene of the airplane crash site, with or without a reporter. After traversing around the key sites, the camera switches back to the studio, and the anchor person starts another news story. It can be summarized in this form: [anchor]→[shots of story1]→[anchor]→[shots of story2]→[...]. This pattern can be represented by the Shot Connectivity Graph (SCG) (Figure 3). In this graph, the nodes represent the shots in the video. The similar shots are represented by a single node. The edges connecting the nodes are the transitions between the shots, as shown in Figure 3. The stories in the video correspond to the large cycles in the SCG that are connected at the node representing the anchor. Our goal for segmenting the stories in the video is then same as finding these cycles in the SCG. We have developed an efficient and robust framework to segment the news programs into story topics. The framework contains two phases: (1). the initial segmentation based on the detections of the anchor person, including both the main anchor and the sub-anchor(s), and (2). the refinement based further detections of the weather and sports stories and the merging of the semantically related adjacent stories.

In the first phase, we detect the cycles that are connected at the “anchor” node, such that the story segmentation is transformed to detecting the “anchor” shots in the video. The properties of the extended facial regions in the key-frames of the shots are analyzed for clustering the similar shots into corresponding groups. Note that besides the large cycles, there are also smaller cycles that are embedded in the bigger ones. This can be explained as the appearance of the reporters, interviewers, or the sub-anchors for a specific news story, e.g., finance news. We also consider the last case as a portion of the news story segments. The detection method for the sub-anchor(s) is same as the detection for the main anchor.

In the second phase, the initial segmentation by the detection of the anchor(s) is refined by further detecting news stories with special format and merging the semantically related stories. For some news stories with special formats, there is no anchor involved. These stories are “hidden” in the large cycles in the SCG. Other techniques are used to “discover” them from the initial story segments. There are two kinds of special stories we incorporated into our system: weather news and sports news. The color pattern of the shots is examined to filter out the candidate weather shots. Then, these candidate weather shots are verified by their motion content. The largest continuous segment of the remaining weather shots form the weather story. For the detection of the sports story, we used the text correlation of

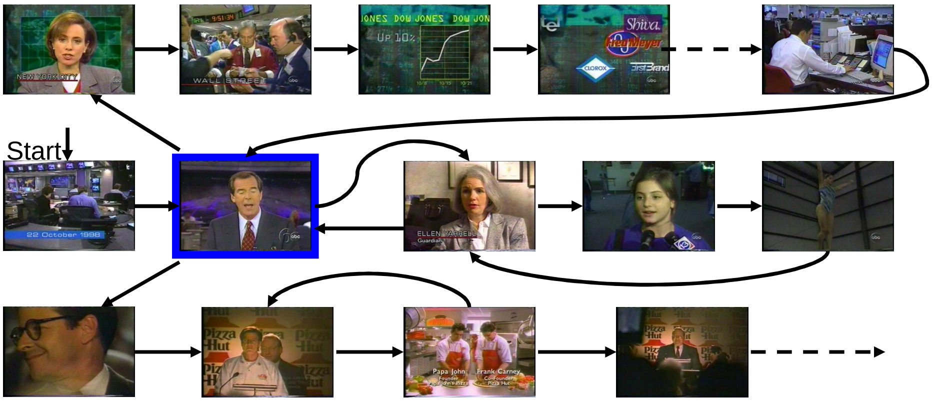
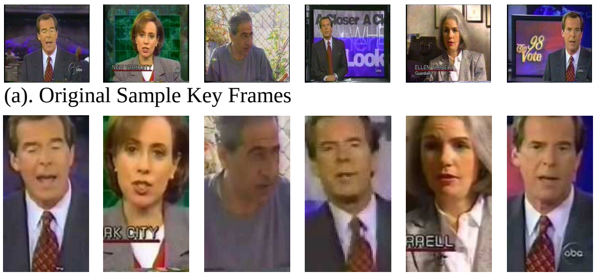


Figure 8: Shot Connectivity Graph. The node with blue bounding box represents the anchor person shots in the news video. In this simple example, the video consists of two news stories and one commercial.

the shots to the sporting words. Similar to the weather story segmentation, the near by sports shots are grouped into the sports story. It maybe possible that the initial segmentations from the first phase are not semantically independent. For example, for a particular story, the anchor may appear more than once, and this will cause multiple cycles in the SCG. In the situation, merging of the semantically related stories is needed. Two adjacent stories are merged together if they present similar pattern in either visual appearance or word narration, or both. The visual similarity is computed as the color similarity between the non-anchor shots in the adjacent stories. The narrative similarity is defined as the Normalized Text Similarity (NTS) based on the automatic speech recognition (ASR) output of the videos. The visual and text similarities are later combined to represent the overall similarity between the stories.

3.1. Phase I - Anchor Detection

We construct the SCG by representing the shots with same person by a single node. There are two common approaches for clustering the similar shots: (1) similarity measures based on the global features, for example, the color histograms of the key frames; (2) similarities based on the correlation of the human faces. The problem for the first approach is that if the settings of the studio have changed, the global features for the anchor shots may has less similarity. In the latter situation, the face correlation is sensitive to the face pose, lighting condition, etc. Therefore, it tends to create several clusters for one same person. To overcome these problems, we used the “body”, an extended region of the face. In one news video, the anchor has the same dress all the time. We take this fact as the cue for this problem. For each shot in the video, we take the middle frame as the key frame for that shot, detect the face by [10], and find the body region by extending the face regions to cover the upper body of the person. The similarity of two shots s_i and s_j is defined as the histogram intersection of the body patches f_i and f_j:


(a). Original Sample Key Frames

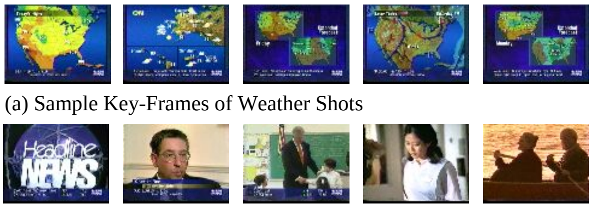
(b). Corresponding Body Regions Based on the Face Detection

Figure 9: The top row (a) are the sample key-frames. Row (b) are the body regions based on face detection. The global feature comparison fails to cluster the anchor together if applied on the top row.

$$HI(f_i, f_j) = \sum_{b \in allbins} min(H_i(b), H_j(b)) \qquad (1)$$

where $H_i(b)$ and $H_j(b)$ are the b-th bin in the histogram of the “body” patches f_i and f_j, respectively. Some example “body” patches are shown in Figure 9. Non-facial shots are considered having zero similarity to others. The shots are then clustered into groups using iso-data, each of those groups corresponds to a particular person. If a key-frame of a shot contains multiple “bodies”, the shot is clustered into the existing largest group with high similarity. Eventually, the shots that contain the main anchor form the largest cluster in the video. Once the anchor shots are detected, the video is segmented into initial stories by taking every anchor shot as the start points of the news stories.

Usually, in the news section of special interests, the main anchor is switched to an expert or sub-anchor. For example, such phenomenon can be found in finance news. The



(a) Sample Key-Frames of Weather Shots

(b) Sample Key-Frames of Non-Weather Shots

Figure 10: Row (a) shows the example key-frames of weather shots; Row (b) shows the key-frames of non-weather shots.

sub-anchor also appears multiple times with different stories focuses. Reappearing sub-anchors result in small cycles in the SCG. Note that many of the stories do not have sub-anchor. In addition, some of the stories also present the small cycles due to other reasons: reporters or interviewers. However, sub-anchor usually appears more times than other miscellaneous persons. Therefore, the true sub-anchor can be classified by examining the size of the largest group. Only the groups with sufficient facial shots are declared as the sub-anchor shots. The detections of the main anchor and the sub-anchors provide the initial result of the story segmentation.

3.2. Phase II - Refinement

3.2.1 Weather Detection

In the news story segmentation, segments related to weather forecast are considered as separate stories from the general stories. To classify a weather shot, we use both the color and motion information of the video. For the weather shots, there are certain color patterns in the visual signal, such as greenish and bluish. Some example key-frames can be found in Figure 10. Furthermore, to ensure that the audience can capture the important weather information, the motion content of the shot should be low.

From the training data set, we have the key-frames of the weather forecast shots. For a key-frame k_m, a color histogram $H(k_m)$ in RGB channels is computed. The histograms for all the key-frames then are clustered into distinctive groups based on the mean and the variance of the RGB channels. These groups form the color model $T = \{t_1 ... t_n\}$ for the weather shot detection, where t_i is the average histogram for model group i. To test if a shot s is a weather shot, we compute the histogram $H(s)$ of its key-frame, and compare it with t_i in the color model. If the distance between $H(s)$ and one of histogram in the color model can be tolerated, the shot s is classified as a weather shot.

The motion content is analyzed for the verification of the initial detections. To verify if a candidate shot s is a true weather shot or not, we perform the following steps:

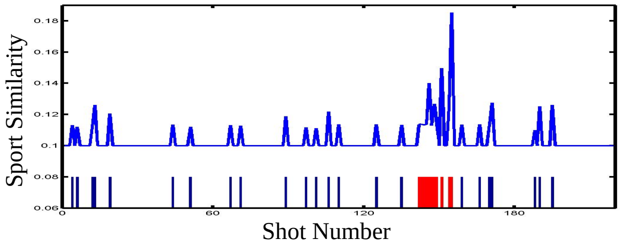


Figure 11: The plot of the sporting similarities of the shots in video. Bars in the bottom row represent the potential sport shots, and the red region represents the actual sporting story.

1. For each frame F_i in the shot, the motion field U_i between F_i and F_{i+1} is computed based on the 16x16 blocks grid X_i.

2. Estimate the Affine motion parameters A_i from U_i using the equation $U_i = A_i X_i$.

3. Apply parameters A_i on X_i to generate the re-projected motion field U_i^p.

4. Compute motion content M_i as the average magnitude of the "disagreement" between the original motion field U_i and the re-projected field U_i^p.

5. The motion content of shot s is the mean of $\{M_1 ... M_{n_s - 1}\}$, where n_s is the number of frames in the shot. If the motion content of the candidate shot s is above defined threshold, this shot is rejected as a non-weather shot.

Finally, other false detections are eliminated by taking only the largest temporally continuous section as the true weather news story.

3.2.2 Sport Detection

We utilize the normalized text similarity measurement to detect sporting shots. In sports video, we often hear the particular words that are related only to the sport games, e.g., "scoring","quarterback", "home run", "basketball", "Olympic", etc. Given such a database of sports related words, we can find the relationship between a shot and the sporting database by computing the correlation between the words spoken in the shot with the words in the database. The text information is provided by the automatic speech recognition (ASR) output of the video [4]. The ASR output contains the recognized words from the audio track of the news program and their starting times. From each candidate shot s, we extract the key-words between the time lines by applying a filter to prune the common words, such as "is" and "the". The remaining key-words form a *sentence* Sen_s

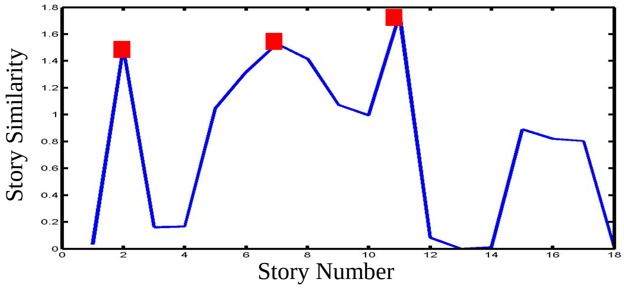


Figure 12: The story similarity plot for the stories created by phase-1. The red peaks are the stories merged into the following ones by in phase-2.

for this shot. The similarity of the candidate shot s to the sporting database is defined as:

$$SportSim(s) = \frac{K_s}{L(Sen_s)} \tag{2}$$

where K_s is the number of the key-words from shot s that also appear in the database, and $L(Sen_s)$ is the length of the key-word *sentence* of shot s. The detection system declares the shots having strong correlation with the sporting database to be the sporting shots. Similar as the technique used for weather detection, false detections can be removed by taking only the largest continuous section of the detected sporting shots as the sporting story. In Figure 11, the upper plot is the similarity of the shots to the sporting database, while the bars in the bottom row represent the potential sporting shots. The red region is the true sporting story in the video.

The results from the detections of the main anchor, sub-anchor(s), weather forecast and the sporting stories are combined and passed to the second phase of the framework.

3.2.3 Story Merging

The proposed segmentation method over segments the video in case of an anchor appearing more than once in a single story. To over come this problem, we merge adjacent segments based on the visual and text similarities. We use the histogram intersection technique to compute the visual similarity of two stories and the Normalized Text Similarity (NTS) as the text similarity measure.

Suppose stories S_i and S_j are the news sections with same topic created by phase 1. They have n_i and n_j non-anchor shots respectively. For each of the shots, we extract the middle frame as the key-frame of that shot. The visual similarity $V(i,j)$ between stories S_i and S_i is defined as:

$$V(i,j) = max(HI(s_i^p, s_j^q)), p \in [1...n_i], q \in [1...n_j] \tag{3}$$

Table 1: UCF performance on the feature extraction task with five features. The second column represents the average precisions and the last column is the relative standing among all the runs for each feature.

Feature	***Avg. Prevision***	***Rel. Standing***
(29). Albright	0.0899	8 / 56
(30). Clinton	0.0536	48 / 72
(32). Beach	0.0635	1 / 59
(33). Basketball	0.2632	25 / 63
(35). Walking	0.0752	23 / 49

where $HI(s_i^p, s_j^q)$ is the histogram intersection between shots s_i^p and s_j^q. This means if there are two visually similar shots in the adjacent stories, these two stories should have the focus on the similar news topic.

Sometimes, the semantic similarity is not always reflected in the visual appearance. For example, in some news program which is related to a taxation plan, the news program may first show the interviews with the middle-class citizens. Then, after a brief summary by the anchor, the program switches to the congress to show the debate between the political parties on the same plan. In this case, the visual appearances of these two adjacent stories are not similar at all. However, if any two stories are focused on the same story, there is usually a correlation in the narrations of the video. In our framework, this narrative correlation between stories S_i and S_i with *sentences* Sen_i and Sen_j is calculated by the Normalized Text Similarity (NTS):

$$NTS(i,j) = \frac{K_{i\rightarrow j} + K_{j\rightarrow i}}{L(Sen_i) + L(Sen_j)} \tag{4}$$

where $K_{i\rightarrow j}$ is the number of words in Sen_i that also appear in Sen_j, and similar definition for $K_{i\rightarrow j}$. $L(Sen_i)$ and $L(Sen_j)$ are the lengths of Sen_i and Sen_j respectively.

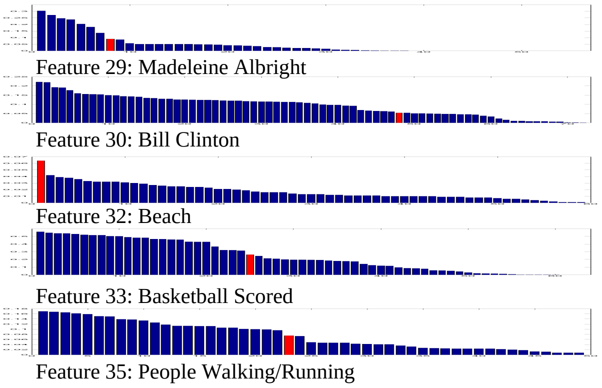
The final similarity between stories S_i and S_j is a fusion of the visual similarity $V(i,j)$ and the normalized text similarity $NTS(i,j)$ (Figure 12),

$$Sim(i,j) = \alpha_V \times V(i,j) + \alpha_{NTS} \times NTS(i,j) \tag{5}$$

where α_V and α_{NTS} are the weights to balance the importance of two measures. If $Sim(i,j)$ for the two adjacent stories S_i and S_j is above the defined threshold, these two stories are merged into a single one.

4. Evaluation Results and Discussions

The results of the feature extraction task is shows in Table 1, including the average precision and relative standing among corresponding runs of the features. Since the TRECVID 2004 data set contains a variety of news related videos and


Feature 29: Madeleine Albright

Feature 30: Bill Clinton

Feature 32: Beach

Feature 33: Basketball Scored

Feature 35: People Walking/Running

Figure 13: Relative standings of UCF's feature extraction methods comparing with all the runs in the task. The runs are sorted by the mean average precision (MAP), and red bars represent UCF's standing.

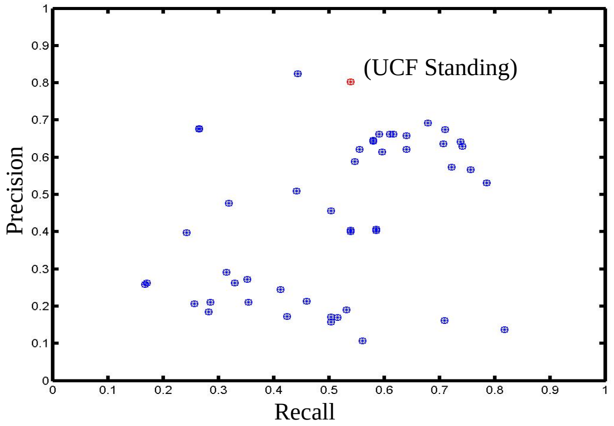


Figure 14: Precision and recall plot of the average performance of all the runs. The red spot represents the standing of UCF framework.

commercials, we believe that the techniques used are relevant to most multimedia content analysis and search tasks. Particularly, they can help any multimedia search system developer not only in the selection of suitable features, but also in determining tradeoffs between accuracy vs. computational efficiency for the semantic concept detection.

The overall precision and recall for the story segmentation task is shown in Figure 14. The precision and recall for UCF system are 0.5390 and 0.8030, respectively. Since the proposed method is motivated by the shot connectivity graph, it is biased towards more structured news videos. For instance, in ABC videos, it often following the pattern described in section 3.1. The initial segmentation is able to provide the closed solution to the true segmentation of the video. On the other hand, in CNN videos, the structure sometime is not as expected same as in ABC. It is possible for multiple stories appearing in a single shot, and the anchor person sometimes appears more than once in a single story, therefore, causing over-segmentation. The merging technique described in section 3.2.3 is useful in this type of situations.

References

[1] Intel Open Source Computer Vision Library. URL *http://www.intel.com/research/mrl/research/ opencv/.*

[2] Y. Freund and R.E. Schapire, "A Decisiontheoretic Generalization of Online Learning and an Application to Boosting.", *JCSS*, 55(1):119139, August 1997.

[3] B. Adams, C. Dorai, S. Venkatesh, "Novel Approach to Determining Tempo and Dramatic Story Sections in Motion Pictures", *ICIP*, 2000.

[4] J.L. Gauvain, L. Lamel, and G. Adda. "The LIMSI Broadcast News Transcription System", *Speech Communication*, 37(1-2):89-108, 2002.

[5] A. Hanjalic, R.L. Lagendijk, and J. Biemond, "Automated High-Level Movie Segmentation for Advanced Video-Retrieval Systems", *CSVT*, Vol.9, Issue.4, 1999.

[6] O. Javed, S. Khan, Z. Rasheed, and M. Shah, "Visual Content Based Segmentation of Talk and Game Shows", *IJCA*, 2002.

[7] R. Lienhart, S. Pfeiffer, and W. Effelsberg, "Scene Determination Based on Video and Audio Features", *IEEE Conf. on Multimedia Computing and Systems*, 1999.

[8] Y. Li, S. Narayanan, C.-C. Jay Kuo, "Movie Content Analysis Indexing and Skimming", *Video Mining*, Kluwer Academic Publishers, Chapter 5, 2003.

[9] C.W. Ngo, H.J. Zhang, R.T. Chin, and T.C. Pong, "Motion-Based Video Representation for Scene Change Detection", *IJCV*, 2001.

[10] P. Viola and M. Jones, "Robust Real-Time Object Detection", *IJCV*, 2001.

[11] M. Yeung, B. Yeo, and B. Liu, "Segmentation of Videos by Clustering and Graph Analysis", *Computer Vision and Image Understanding*, vol.71, no.1, pp. 94-109, July 1998.