VISUAL CONTENT BASED SEGMENTATION OF TALK & GAME SHOWS

O. Javed, S. Khan, Z. Rasheed, M.Shah

{ojaved, khan, zrasheed, shah}@cs.ucf.edu

Computer Vision Lab

School of Electrical Engineering and Computer Science

University of Central Florida, Orlando, FL 32816

Abstract

In this paper, we present a method to remove commercials from talk and game show videos and to segment these videos into host and guest shots. In our approach, we mainly rely on information contained in shot transitions, rather than analyzing the scene content of individual frames. We utilize the inherent difference in scene structure of commercials and talk shows to differentiate between them. Similarly, we make use of the well-defined structure of talk shows, which can be exploited to classify shots as host or guest shots. The entire show is first segmented into camera shots based on color histogram. Then, we construct a data-structure (shot connectivity graph) which links similar shots over time. Analysis of the shot connectivity graph helps us to automatically separate commercials from program segments. This is done by first detecting stories, and then assigning a weight to each story based on its likelihood of being a commercial. Further analysis on stories is done to distinguish shots of the hosts from shots of the guests. We have tested our approach on several full-length shows (including commercials) and have achieved video segmentation with high accuracy. The whole scheme is fast and works even on low quality video (160x120 pixel images at 5 Hz).

Keywords: Video segmentation, video processing, digital library, story analysis, semantic structure of video, removing commercials from broadcast video.

1. Introduction

We live in the digital age. Pretty soon everything from TV shows to movies, documents, maps, books, music, newspapers, etc. will be in the digital form. Storing videos in digital format removes the limitations of sequential access of video (for example *forward* and *rewind* buttons on a VCR). Videos may be more efficiently organized for browsing and retrieval by exploiting their semantic structure. Such structure consists of *shots* and groups of shots called s*tories*. A story is one coherent section of a program or commercials. The ability to segment a video into stories gives the user the ability to browse using story structure, rather than just sequential access available in analog format tapes.

In this paper we assume that the collection of shows has been digitized and address the problem of how to organize each show, so that it is suitable for browsing and retrieval. We consider the user may be interested to look at only show segments without the commercials. The reasons for automatically identifying and/or removing the commercials might be to prevent discontinuity in the program, to save disk storage space in video servers, or to digitally insert new commercial sequences in place of old ones. The user may also want to view clips which display the host talking or performing, or may want only to keep track of the guests appearing in the show.

Talk show videos are an important segment of televised programs. Lots of popular prime-time programs are based heavily on a host and guests format, for example, Crossfire, Larry King Show, Who Wants To Be A Millionaire, Jeopardy, Hollywood Squares etc. The algorithm presented in this paper has been tested on Larry King Live and Who Wants To Be A Millionaire. However the algorithm is not tailored for a specific talk show and can be applied to any of these other shows to study their structure. This should significantly improve the digital-organization of these shows for browsing and retrieval purposes.

There has been lots of interest recently in video segmentation and automatic generation of digital libraries. The Informedia Project [1] at Carnegie Mellon University has spearheaded the effort to segment and automatically generate a database of news broadcasts every night. The overall system relies on multiple cues, like video, speech, close-captioned text and other cues. Alternately, some approaches rely solely on video cues for segmentation [2, 3, 4]. Such an approach reduces the complexity of the complete algorithm and does not depend on the availability of close-captioned text for good results.

In this paper, we exploit the semantic structure of the shows to not only separate the commercials from talk show segments, but also to analyze the content of the show to detect host shots versus guest shots. All this is done using only video information and relying mainly on the information contained in shot transitions. No specific training is done for a talk show, and therefore, the scheme is generalized to all the shows that are based on a host interacting with guests format.

In related work, in [5] the authors present a heuristic approach to segment commercials and individual news stories. They rely heavily on the fact that commercials have more rapidly changing shots than programs and are separated by blank frames. The overall error reported is high. Our approach to separate commercials and non-program segments exploits scene structure rather than multiple heuristics based on shot change rate. We are able to achieve high accuracy in our results.

In another work in [2], a scene transition graph is used to extract scene structure of sitcoms. We employ a similar data-structure in our computations. However, our work differs from their work in some important respects. In [2] all *cut edges* are treated as story boundaries. This paradigm would result in a high number of stories for non-repetitive scenes, like commercials. Their approach, therefore, would not work well in separating commercials from programs. In addition, we employ a novel weighing scheme (see Section 3) for each story to distinguish commercials from programs. We also analyze the story for its content, rather than simply finding its bounds.

In the next section, we discuss the algorithm to detect shot boundaries and to build the shot connectivity graph. In Section 3, we present our scheme to detect interview segments and separate them from commercials. In Section 4, we analyze the interview stories found by our algorithm to label host shots and guest shots. Finally we present the results in Section 5.

2. Shot Connectivity Graph

The first step in processing the input video is to group the frames into *shots*. A shot is defined as a continuous sequence captured by a single camera. We use a modified form of the algorithm reported in [7] for the detection of shot boundaries, allocating 8-bins for hue and 4-bins each for saturation and intensity values. Let the normalized histogram be denoted by H_i, where i is the frame-number. Let $D(i)$ represent the histogram intersection of frames i and the previous frame i-1. That is

$$D(i) = \sum_{j \in \text{all bins}} \min(H_i(j), H_{i-1}(j)) \qquad (1)$$

Then we define the shot change $S(i)$ measure as

$$S(i) = D(i) - D(i-1) \qquad (2)$$

Usually a threshold was applied to $D(i)$ to find shot boundaries. We, however, found out that a threshold applied to $S(i)$ does a better job in finding shot boundaries. Note that $D(i)$ is bound between [0,1], and $S(i)$ is the derivative of $D(i)$.

For each shot that we extract, we find a key frame representing the content of that shot. The key frame is defined as the middle frame between the two shot boundaries. Once shot boundaries have been identified, they are organized into a data-structure, which we call *shot connectivity graph G*. This graph links similar shots over time, thus extracting the semantic structure of video and making the segmentation task easier. The vertices *V* represent the shots. Each vertex is assigned a label indicating the serial number of shot in time and a weight *w* which is the number of frames in that particular shot.

The process of inserting edges to connect the vertices in the shot connectivity graph consists of finding the intersection of the histogram of each key frame with those of previous key frames to determine whether a similar shot had occurred before or not. However, this process is time-constrained to only a certain number of previous shots (the memory parameter, T_{mem}). Thus, shot proximity i.e. shots that are close together in time, and shot similarity i.e. shots that have similar color statistics, are two criteria to link the vertices in the shot connectivity graph. For shot *q* to be linked to shot *q-k* (where $k \leq T_{mem}$) the following condition must hold true:

$$\sum_{j \in \text{ all bins}} \min(H_q(j), H_{q-k}(j)) \geq T_{color} \qquad (3)$$
$$\text{for some } \quad k \leq T_{mem}$$

where T_{color} is a threshold on the intersection of histograms and captures the allowed tolerance between color statistics of two shots for them to be declared similar.

It is important to point out here that we have not employed a time constraint on the number of frames, as in some previous approaches. Rather, we have used a constraint on the number of shots, which makes our scheme more robust. Commercials generally have rapidly changing shots and therefore this threshold would translate into a shorter time constraint, whereas interviews would span

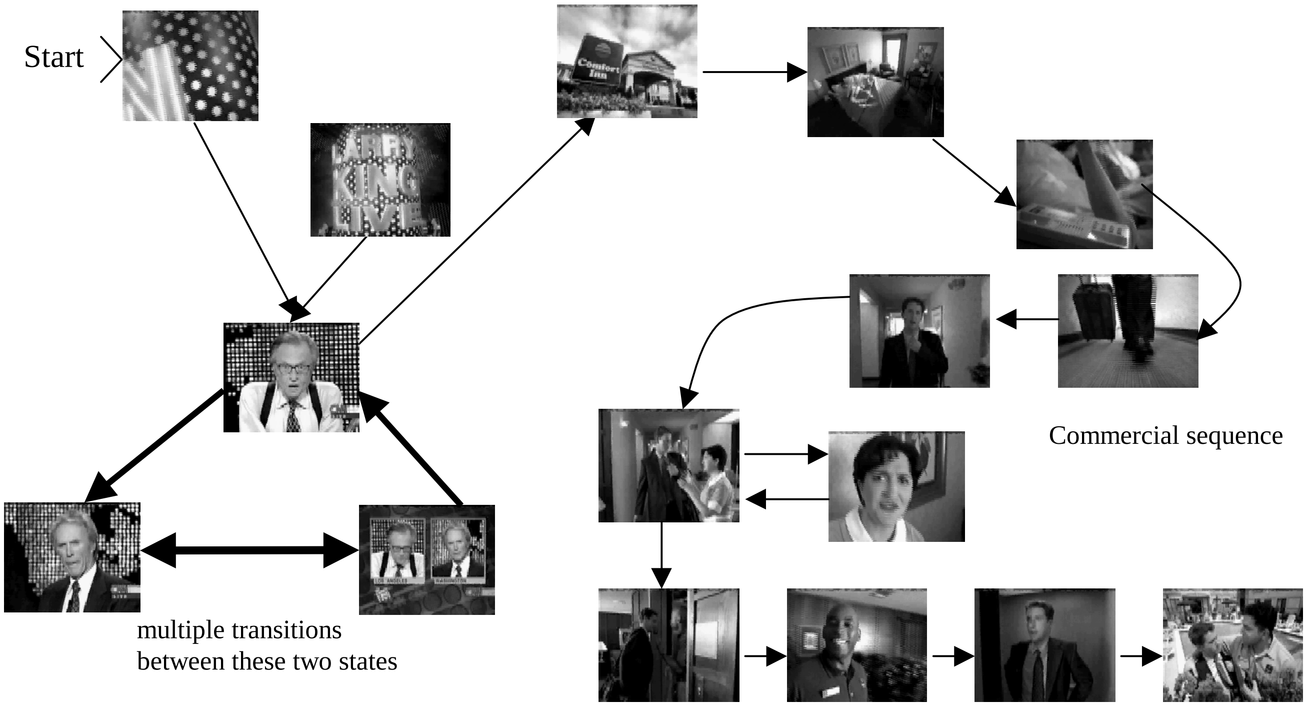


Figure 1: Shot Connectivity Graph: Note the high repetitive structure of the show segment (shown by thick arrows), versus the linear structure of the commercial sequence (shown by thin arrows). Even though commercials also have cycles (as shown), our algorithm is able to separate them from the interview segment.

more frames within the same number of shots. This results in a larger time constraint for interviews, which yields a more meaningful segmentation.

Significant story boundaries (for example that between the show and the commercials) are often separated by a short blank sequence. This is done to provide a visual cue to the audience that the following section is a new story. These blanks can be found by putting a test on the histogram H_i to check if all the energy in the histogram is concentrated into a single bin.

We utilize these blanks to avoid making links across a blank in our shot connectivity graph. Thus two vertices v_p and v_q, such that $v_p, v_q \in V$ and $p<q$, are adjacent, that is they have an edge between them, if and only if

- v_p and v_q represent consecutive shots or
- v_p and v_q satisfy the shot similarity, shot proximity and blank constraints.

The shot connectivity graph exploits the structure of the video selected by the producers in the editing room. Interview videos are produced using multiple cameras running simultaneously in time, recording the host and the guest. The producers switch back and forth between them to fit these parallel events on a sequential tape. By extracting this structure, different story segments can be differentiated from each other. In addition, we can achieve understanding of the story *content* by looking closely at the structure. This follows from the fact that scene structure is not arbitrary, but is carefully selected by the producers for best user perception. An example of the shot connectivity graph, automatically computed by our system for a section of Larry King Live show, is shown in Figure 1.

3. Story Segmentation and Removal of Commercials

Talk shows have a very strong semantic structure that relates them in time. Typical scenes of such shows have alternating shots of the host and the guests, including shots of single or multiple guests in the studio, split shots of guests in the studio with guests at another location, and shots of both the host and the guests. These shots are strongly intertwined back and forth in time, and prove to be the key cue in discriminating them from other stories.

Commercials on the other hand have weak structure and rapidly changing shots (see Figure 1). There might still be repetitive shots in a commercial sequence, which appear as cycles in the shot connectivity graph. However, these shots are not as frequent, or as long in time, as those in the interview. Moreover, since our threshold of linking shots back in time is based on the *number of shots*, and not on the total time elapsed, commercial segments will have less time memory than talk shows.

We contend here that simply relying on the hypothesis that commercials have more rapidly changing shots than programs for segmenting commercials [5] is not enough. Even good stories might occasionally have a high rate of change of shots, due to either video summaries shown within the program or just multiple people trying to speak simultaneously within the talk show. Exploiting scene structure, however, is more robust and takes care of these situations.

Our scheme to differentiate commercial sequences from program sequences relies on analysis of the shot connectivity graph. Commercials generally appear as a string of states, or small cycles in the graph. To detect them, we find *stories*, which are collection of shots linked, backed in time. To extract stories from the shot connectivity graph *G*, we find all the strongly connected components in *G*. A strongly connected component $G'(V', E')$ of *G* has the following properties

- $G' \subseteq G$
- There is a path from any vertex $v_p \in G'$ to any other vertex $v_q \in G'$.
- There is no $V_z \in (G - G')$ such that adding V_z to G' will form a strongly connected component.

Each strongly connected component G' in *G* represents a story. We compute the likelihood of all such stories being part of a program segment or not. Each story is assigned a weight based on two factors; *the number of frames in a story and the ratio of number of repetitive shots* to the total *number of shots in a story*. The first factor follows from the observation that long stories are more likely to be program segments than commercials. Stories are determined from strongly connected components in the shot connectivity graph. Therefore, a long story means that we have observed multiple overlapping cycles within the story since the length of each cycle is limited by T_{mem}. This indicates the strong semantic structure of the program. The second factor stems from the observation that programs have a

Results	Total	Show Segments	Show Segments	Misclassified Frames	Misclassified Frames	Total Error	Overall Correct Classification
	Frames	Ground truth	found	(False +ve)	(False –ve)	%	%
Video 1	34611	8	8	12	890	2.61	97.39
Video 2	12144	6	6	4	17	0.17	99.83
Video 3	17157	8	9	6	1804	10.55	89.45
Video 4	13778	6	6	105	265	2.69	97.31
Video 5	19700	7	7	41	904	4.80	95.20
Video 6	17442	7	7	10	351	2.07	97.93

Table 1: Detection of interview segments. Video 1 was digitized at the frame rate of 10 Hz, all other videos were digitized at 5 Hz. Video 1-4 are Larry King Shows and Videos 5 and 6 are Who Wants To be a Millionaire.

large number of repetitive shots in proportion to the total number of shots. Commercials, on the other hand, have a high shot transition rate. Even though commercials may have repetitive shots, this repetition is small compared to total number of shots. Thus program segments will have more repetition than commercials, relative to total number of shots. Both of these factors are combined in the following likelihood of a story being an program segment:

$$L(G') = \sum_{\forall j \in G'} w_j \frac{\sum_{\forall E'_{ji} \in G' | j > i} 1}{\sum_{\forall j \in G'} 1} \Delta t \qquad (5)$$

where G' is the strongly connected component representing the story. w_j is weight of the jth vertex i.e. the number of frames in shot j. E' are the edges in G'. Δt is the time interval between consecutive frames. Note that the denominator represent the total number of shots in the story.

This likelihood forms a weight for each story, which is used to decide on the label for the story. Stories with $L(story)$ higher than a certain threshold are labeled as program stories, whereas those that fall below the threshold are labeled as commercials. This scheme is robust and yields accurate results, as shown in Section 5.

4. Host Detection: Analysis of Shots within an Interview Story

We perform further analysis of program stories, extracted by the method described in the pervious section, to differentiate host shots from guest shots. Note that in most talk shows a single person is host for the duration of program but the guests keep on changing. Also the host asks questions which are typically shorter than answers. These observations can be utilized for successful segmentation. Note that no specific training is used to detect the hosts. Instead, the host is detected from the pattern of shot transitions, exploiting the semantics of scene structure.

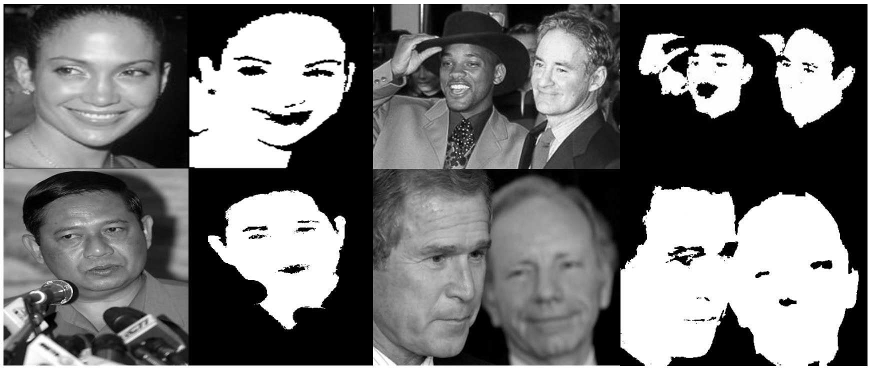


Figure 2: Example images and their binary masks used to train the system for skin detection. Portion of the images containing the skin are manually marked in the binary images.

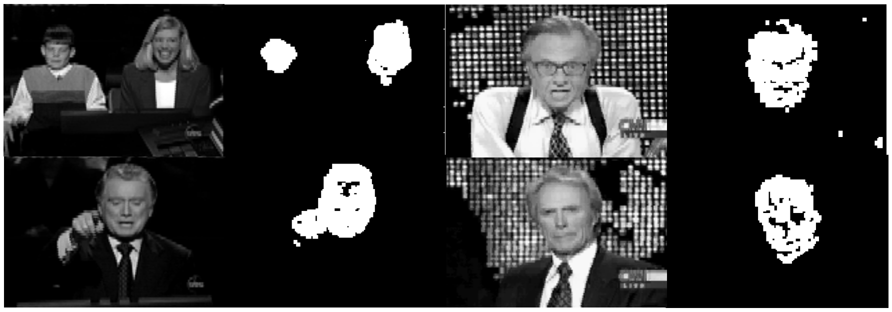


Figure 3: Some results of skin detection. White area in the images shows region where skin is detected.

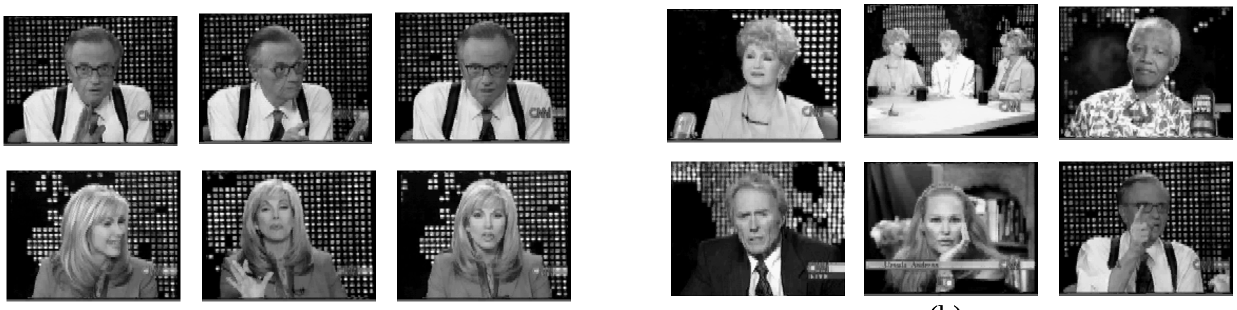
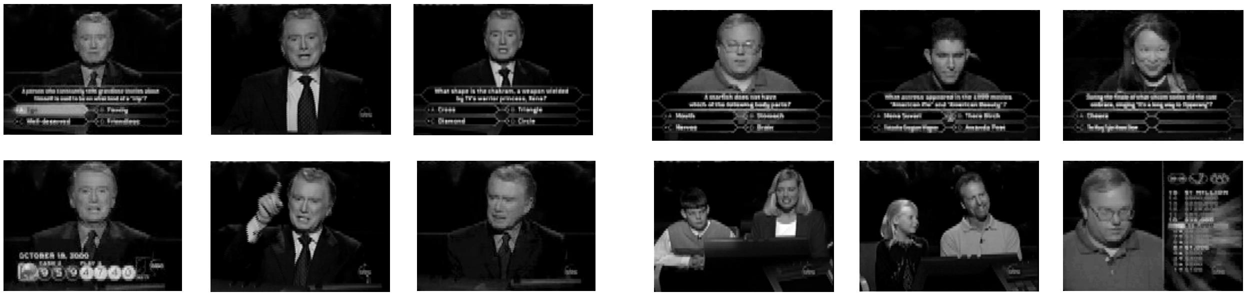

(b)

Figure 4: Examples of host detection In Larry King Show: (a) Correct host detection Leeza Gibbons substituting for Larry King in one show). Correct classification is achieved even for varying poses. (b) Guest shots; Larry King shot is misclassified due to occlusion of the face.


(a)

(b)

Figure 5: Examples of host detection In Who Wants To Be A Millionaire: (a) Correct host shot detection. Correct classification of show host achieved for a variety of poses. (b) Guest shots.

For a given show, we first find N shortest shots in the show containing only one person. To determine if a shot has one person or more, we use the skin detection algorithm presented in [6]. A skin color predicate is first trained on a few training images, by manually marking skin regions and building a 3D-color histogram of these frames. Figure 2 shows some of the training images used to train the system for skin detection. A binary mask is made for each image marking the presence of skin. For each positive example, the histogram is incremented by a 3D Gaussian distribution, so that colors similar to the marked skin color also get selected. For each negative training example, the histogram is decremented by a narrower Gaussian. After incorporating information from all training

Name	Correct *HostID* ?	Host Detection Accuracy
Video 1	Yes	99.32%
Video 2	Yes	94.87%
Video 3	Yes	96.20%
Video 4	Yes	96.85%
Video 5	Yes	89.25%
Video 6	Yes	95.18%

Table 3: Accuracy of Host Detection: Column 2 defines whether the correct host was found in the story or not. Column 3 gives the overall error in labeling host shots.

images, the color predicate is thresholded to a small positive value, and thus essentially forms a color lookup table. Including persons of various ethnic backgrounds in training images makes this color predicate robust for a variety of skin tones. For detection, the color of each pixel is looked up in the color predicate to be labeled as skin or non-skin. If the image contains only one significant skin colored component, then it is assumed to have one person in it. Figure 3 shows some results of skin detection.

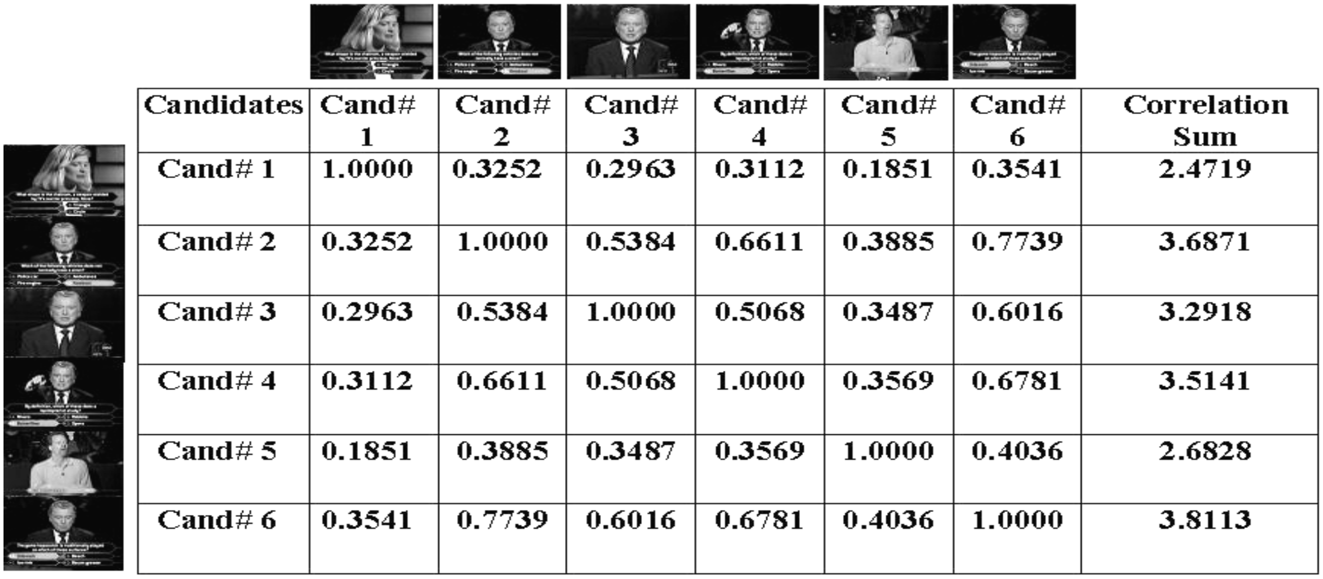
The key frames of the *N* shortest shots containing only one person are correlated in time to find the most repetitive shot. Since questions are typically much shorter than answers, host shots are typically shorter than guest shots. Thus it is highly likely that most of the *N* shots selected will be host shots. An *N*-by-*N* correlation matrix **C** is computed such that each term of **C** is given by:

$$C_{ij} = \frac{\sum_{r \in \text{allrows}} \sum_{c \in \text{allcols}} \left(I_i(r,c) - \mu_i\right)\left(I_j(r,c) - \mu_j\right)}{\sqrt{\left(\sum_{r \in \text{allrowd}} \sum_{c \in \text{allcols}} \left(I_i(r,c)\right)^2\right)\left(\sum_{r \in \text{allrows}} \sum_{c \in \text{allcols}} \left(I_j(r,c)\right)^2\right)}} \qquad (6)$$

where I_k is the gray-level intensity image of frame *k* and μ_k is its mean. Notice that all the diagonal terms in this matrix are 1 (and therefore do not need to be actually computed). Also, **C** is symmetric, and therefore only half of non-diagonal elements need to be computed. The frame which returns the highest sum for a row is selected as the key frame representing the host. That is,

$$HostID = \arg\max_{r} \sum_{c \in \text{allcols}} C_{rc} \quad \forall r \qquad (7)$$

Figure 4 shows key host frames extracted for our test videos. Note that the correct host is identified in video 3 because she was substituting for Larry King. We identified the correct host for all our test videos using this scheme. Guests are the shots which are non-host. Figure 5 shows similar results for Who Wants To Be A Millionaire.



Candidates	Cand# 1	Cand# 2	Cand# 3	Cand# 4	Cand# 5	Cand# 6	Correlation Sum
Cand# 1	1.0000	0.3252	0.2963	0.3112	0.1851	0.3541	2.4719
Cand# 2	0.3252	1.0000	0.5384	0.6611	0.3885	0.7739	3.6871
Cand# 3	0.2963	0.5384	1.0000	0.5068	0.3487	0.6016	3.2918
Cand# 4	0.3112	0.6611	0.5068	1.0000	0.3569	0.6781	3.5141
Cand# 5	0.1851	0.3885	0.3487	0.3569	1.0000	0.4036	2.6828
Cand# 6	0.3541	0.7739	0.6016	0.6781	0.4036	1.0000	3.8113

Table 4: Correlation chart. 6 key frames were selected as candidates for being the host. Right column shows the sum of correlation results of each candidate with others. Note that summations for candidate number 2, 3 4 and 6 are noticeably higher than the rest since all of them contain the host. Candidate 6 with the largest value of correlation sum it is declared the host of the show.

The key host frame is then correlated against key frames of all shots to find all shots of the host. Table 4 shows that the correlation sum for the host is largest for a given set of host-candidates. For this algorithm, the same correlation equation (eq.6) is used. Results of this algorithm are compared against ground-truth marked by a human observer, and show high accuracy of this method (see Section 5 and Table 3).

5. Results

Our test suite was 4 full-length Larry King Live shows and 2 complete Who Wants To Be A Millionaire episodes. The videos were digitized at 160x120 size at 5 Hz. This is fairly low spatial and temporal resolution, but is sufficient to capture the kind of scene structure that we want to exploit. For each data set, we digitized a short segment before and after the show, so that the start and the end of the actual show is also captured within our data set. In one Larry King Show there was a substitute host for Larry King, who was identified correctly. The shows had guests varying from one to three. The thresholds in algorithms were kept the same, and the same skin color predicate was used for all data-sets.

Table 1 contains the talk show classification results. A human observer established the ground truth i.e. classifying frames as either belonging to a commercial or a talk show. Correct classification rate is over 95% for most of the videos. The classification results for video 3 (Larry King Show) are not as good as others. This particular show contained a large number of out door video clips that did not conform to our assumptions of the talk show model. The over all accuracy of talk show classification results is about the same for both the Larry King Live show and Who Wants To Be a Millionaire even though these shows have quite different lay out and production style

Table 3 contains host detection results with the ground truth established by a human observer. The second column shows whether the host identity was correctly established by Eq. 7. The last column shows the overall rate of misclassification of host shots. Note that for all six videos, very high accuracy and precision is achieved by our algorithm.

6. Conclusions

We have used the information contained in shot transitions to differentiate between commercials and program segments for several Larry King Live and Who Wants To Be a Millionaire shows. We have also segmented stories into host shots and guest shots. This creates a better organization of these shows than simple sequential access. The user may browse just the relevant areas of interest to extract a meaningful summary of the whole show in a small amount of time.

We have demonstrated that shot transitions of video alone are sufficient to perform these tasks to a high degree of accuracy, without using speech or close-captioned text. We also perform minimal image content analysis. The entire scheme is efficient and works on low spatial and temporal resolution video.

References

[1] Wactlar, H., Kanade, T., Smith, M., “Intelligent Access to Digital Video: Informedia Project”, *IEEE Computer*, Vol. 29, No. 5, May 1996, pp. 46-52

[2] Yeung, M., Yeo, B.-L., and Liu, B., “Extracting Story Units from Long Programs for Video Browsing and Navigation” in *International Conference on Multimedia Computing and Systems*, June 1996

[3] Kender, J. R. and Yeo, B. L., “Video Scene Segmentation via Continuous Video Coherence”, in *Proceedings of Computer Vision and Pattern Recognition*, 1998

[4] Rui, Y., Huang, T. S., Mehrotra, S., “Exploring Video Structure Beyond the Shots”, in *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, 1998

[5] Haupmann, A. G. and Witbrock, M. J., “Story Segmentation and Detection of commercials in Broadcast News Video”, in *Proceedings of the Advances in Digital Libraries Conference*, 1998

[6] Kjedlsen, R., and Kender, J., “Finding Skin in Color Images”, in *Face and Gesture Recognition*, pp. 312-317, 1996

[7] Niels Haering, “*A Framework for the Design of Event Detectors*”, Ph.D. Thesis, School of Computer Science, University of Central Florida, 1999