# MAJOR CAST DETECTION IN VIDEO USING BOTH AUDIO AND VISUAL INFORMATION

*Zhu Liu*[*]

AT&T Labs - Research
Room A5 4F04, 200 Laurel Ave. South
Middletown, NJ 07748
zliu@research.att.com

*Yao Wang*

Department of Electrical Engineering
Polytechnic University
Brooklyn, NY, 11201
yao@vision.poly.edu

**ABSTRACT**

*Major casts, for example, the anchor persons or reporters in news broadcast programs and principle characters in movies play an important role in video, and their occurrences provide good indices for organizing and presenting video content. This paper describes a new approach for automatically generating the list of major casts in a video sequence based on multiple modalities, specifically, both speaker and face information. A list of major casts is created and ordered by the accumulative temporal and spatial presence of corresponding casts. Preliminary simulation results show that the detected major casts are meaningful and the proposed approach is promising.*

## 1. INTRODUCTION

With huge amount of video data generated daily, it is indispensable for a video creator or distributor to provide content description for browsing and retrieval capability. While low level content descriptors including camera shot changes, speech or music boundaries, etc. are useful, they can not provide semantically meaningful indices. Higher level content based abstract is more desirable to help the users to grasp the synopsis effectively. Major casts, for example the anchor persons or reporters in news programs and principal characters in movies play an important role, and their occurrences provide good indices for organizing and presenting video content. The users may easily digest the main scheme of a video by skimming through clips associated with major casts.

Because manual content annotation is time consuming and sometimes inconsistent, many research efforts have been involved to automate this procedure. Most of the previous works are focusing on utilizing one type of modality, e.g. audio or visual alone, to tackle this problem. Zhang and Kuo [1] classified audio content in a hierarchical way. At the coarse level, audio data is classified into speech, music, environmental sounds, and silence, and at the fine level, environmental sounds are further classified into applause, rain, etc. Rui *et al.* [2] explored the automatic extraction of video structures from both the physical shots and the semantic scenes and developed tools that can construct table of content (TOC) to assist user's access. Since the semantics of video data are embedded in multiple forms that are usually complimentary to each other, we need to analyze all available media simultaneously. Saraceno and Leonardi [3] considered segmenting a video into the following basic scene types: dialogs, stories, actions, and generic. This is accomplished by first dividing a video into audio and visual shots independently, and then grouping video shots so that audio and visual characteristics within each group follow some predefined patterns. Huang *et al.* [4] proposed to generate content hierarchy for broadcast news programs by integrating audio, video, and text information simultaneously.

This paper presents a new approach for automatically generating a list of major casts for video based on both audio and visual information. In section 2, we illustrate the overall diagram of major cast detection algorithm. Speaker and face information extraction is described in section 3. How to combine cues from different modalities and further detect major cast is explained in section 4. In section 5 we present and discuss some preliminary results, and finally in section 6, we draw our conclusion.

## 2. MAJOR CAST DETECTION DIAGRAM

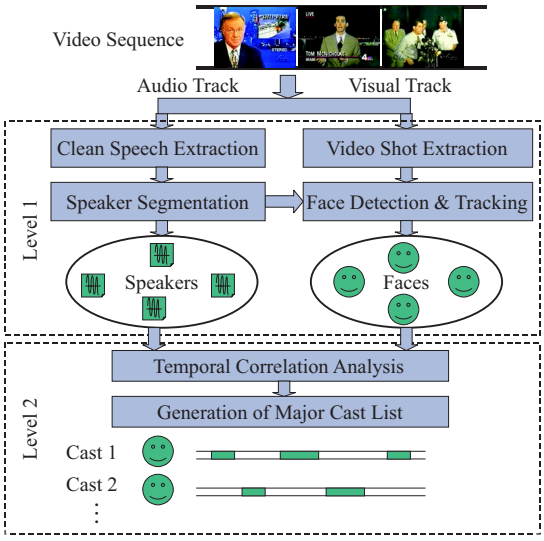


**Fig. 1**. Major Cast Detection Algorithm

Figure 1 illustrates the major cast detection algorithm we proposed. Each major cast is characterized by two attributes: face and speech. The detection procedure is to find corresponding face occurrences and speech segments by analyzing video at two levels. Audio and visual information is utilized separately at low level, and at high level cues from different modalities are combined.

At low level, video sequence is segmented independently in

[*]The author performed the work while he was at Polytechnic University. This work was supported in part by the National Science Foundation through its STIMULATE program under Grant No. IRI-9619114.

both audio and visual tracks. In audio track, clean speech chunks are extracted, within which speaker boundaries are then identified. On the other hand, visual track is segmented into homogeneous shots, and face detection and tracking are applied within each shot. At high level, we exploit both audio and visual information based on temporal correlation among different faces and speakers. All speaker segments and face tracks are grouped using an integrated clustering method such that segments containing the same speaker and tracks consisting of the same face are merged. A list of major casts is then constructed by associating faces and speakers to certain characters. The order of the list reflects the importance of each characters, which is determined based on corresponding accumulative temporal and spatial presence.

## 3. SPEAKER SEGMENTATION AND FACE TRACKING

### 3.1. Speaker Segmentation

Besides speech signal, there are other kinds of sound in audio track, for example, music, speech with music, noise, speech with noise, etc. To separate and compare different speakers, we want to extract speaker information based on clean speech only. Therefore the speaker segmentation algorithm includes two steps: 1) Extract the clean speech chunks from the audio track. 2) Locate the speaker boundaries in clean speech audio chunk.

To extract clean speech, we segment the audio stream into adjacent clips, which are about 2 seconds long, compute 14 audio features for each clip, and then classify each clip by Gaussian Mixture Model (GMM) classifier into two classes: clean speech and non-clean speech. Detailed information can be found in [5, 6].

The aim of speaker segmentation is to find the switching of speakers in audio track. In [7], an approach for segmenting, modeling, and comparing general audio content was proposed. Here we follow similar approach to segment speaker in clean speech chunk. The speaker segmentation scheme is composed of three steps: feature computation, splitting, and merging. The audio track is divided into frames, each 32 ms long and overlapping with the previous frame by 16 ms. For each frame, an audio feature vector is computed, which includes 13 Mel-Frequency Cepstral Coefficients (MFCCs) and 13 delta MFCCs. During the splitting stage, for each frame whose volume is a local minimum, we compute the Kullback Leibler distance (KLD) between $N$ previous frames and $N$ future frames. If the distance is high, we find a possible speaker boundary. During merging stage, we compute the KLD between adjacent candidate segments and merge them if their distance is small. When we compute KLD, we only consider those frames for which pitch is detectable. These frames normally correspond to voice, and reflect the characteristics of speaker's vocal track.

To group scattered segments of the same speaker, we can apply a clustering algorithm on all segments. We build a GMM model for the features of each segment, and then use the distance between corresponding GMMs to measure the difference of two speaker segments. Only those frames that have pitch values are used to build the model, such that it represents more speaker dependent information. Here, we employ the distance proposed in [7] to compute the model distance, and adopt KLD as the element distance.

### 3.2. Face Detection and Tracking

In [8], we developed a template matching based face detection and tracking algorithm. Instead of tracking faces directly on the entire video, we first segment the video sequence into shots, then track faces in each shot independently. The face detection algorithm finds the best warping functions between the test region and the face template following an iterative dynamic programming procedure. The same algorithm can also determine the difference between two faces by using one as template and computing the matching error of the other. Two stages are involved for face tracking within each shot: detecting frontal faces in all frames and expanding face tracks in surrounding frames. In the first stage, an average face model is used to detect faces in each frame, where only frontal faces can be effectively detected. In the second stage, we use detected faces as new face templates to search faces in neighboring frames bidirectionally. By using a detected frontal face as the template, we can usually detect slightly tilted/turned faces of the same person, which are typically missed in the first stage.

Clustering algorithm can be used to merge the face tracks of same person in different shots. To measure the distance between two detected face tracks, we first find a representative face for each face track, which is the face detected with the highest matching score in the first stage, and then measure the distance of corresponding representative faces.

## 4. MAJOR CAST EXTRACTION

In current study, we only consider detection of major cast appearances that are accompanied by both speech and face. Satoh et al. used visual and text information to associate faces with names [9]. Our approach is to associate faces with speech for major casts based on the temporal correlation between faces and speakers. In this section, we first give the definition of speaker face correlation matrix. Based on this matrix, we show the integrated speaker segment and face track clustering algorithm, and major cast selection and ordering method.

### 4.1. Speaker Face Correlation

Suppose there are M speaker segments, $S_1, S_2, ..., S_M$, and N face tracks, $F_1, F_2, ..., F_N$. Different speaker segments or face tracks may correspond to the same person. To make our approach general, we assume that speaker segment $S_i$ has $L_i$ discontinuous sub-segments: $s_1^i, s_2^i, ..., s_{L_i}^i$, each sub-segment has two attributes: starting time(ST) and ending time(ET). Similarly, face track $F_i$ has $l_i$ discontinuous sub-tracks: $f_1^i, f_2^i, ..., f_{l_i}^i$, each sub-track has three attributes: starting time, ending time, and face size(FS). Here we use the representative face of each face sub-track to determine the face size. Then the speaker face correlation(C) matrix is an $N \times M$ matrix, whose item $C(i,j)$ is defined as:

$$C(i,j) = \sum_{m=1}^{L_i} \sum_{n=1}^{l_j} OL(s_m^i, f_n^j) \times FS(f_n^j), \qquad (1)$$

where $OL(x,y)$ is the overlapping duration of speaker sub-segment x and face sub-track y, and FS(y) is the face size of y.

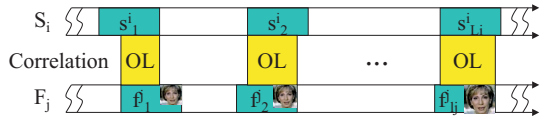


**Fig. 2**. Illustration of Speaker Face Correlation.

Figure 2 illustrates the correlation between speaker segment $S_i$ and face track $F_j$. Our definition not only considers the temporal overlapping among speaker segments and face tracks, but also takes into account the effect of face size. The consideration of face size is helpful when more than one face show up during a speech

segment, where the face with bigger size is more likely to be the real speaker.

### 4.2. Integrated Speaker Face Clustering

While speaker segments or face tracks can be clustered independently, performing such clustering jointly will help improve the performance. For example, suppose there are two speaker segments of the same person, one with clean speech, one with light background noise, then the speaker alone clustering may fail to merge these two segments. If we know that the two face tracks that shown in these segments are very similar, we can confidently merge these two segment of speakers. Here we propose a new integrated approach that cluster face tracks and speaker segments simultaneously.

Suppose after speaker segmentation and face tracking, we have M speaker segments, N face tracks, denoted in the same way in the last section. The distance matrix among speaker segments is $D_S$, where the distance between two segments is defined as the maximum distance among all possible pairs of two sub segments from each segment. Similarly, we define the distance matrix of face tracks $D_F$. The idea of integrated clustering is to define an augmented distance matrix for speaker segments $D'_S$ (resp. face tracks $D'_F$) based on not only the distance among speaker segments (resp. face tracks) but also distances among corresponding face tracks (resp. speaker segments). The item in $D'_S$ and $D'_F$ can be computed as,

$$D'_S(i,j) = \lambda_f \frac{\sum_{1\le m,n\le N} C(i,m)C(j,n)D_F(m,n) + T_f\epsilon}{\sum_{1\le m,n\le N} C(i,m)C(j,n) + \epsilon} + D_S(i,j), \quad 1 \le i,j \le M, \tag{2}$$

$$D'_F(i,j) = \lambda_s \frac{\sum_{1\le m,n\le M} C(m,i)C(n,j)D_S(m,n) + T_s\epsilon}{\sum_{1\le m,n\le M} C(m,i)C(n,j) + \epsilon} + D_F(i,j), \quad 1 \le i,j \le N, \tag{3}$$

where $\lambda_f$ and $\lambda_s$ are ratios that determine the weighting of distance effect from different modality, $\epsilon$ is a small constant to prevent division by zero, and $T_f$ and $T_s$ are two thresholds that are used in face tracks/speaker segments independent clustering. The detailed integrated clustering procedure is shown as follows.

1. Starting with $M^{(0)}$ speaker segments, $N^{(0)}$ face tracks, distance matrix $D_S^{(0)}$, $D_F^{(0)}$, and correlation matrix $C^{(0)}$. Set $i = 0$.
2. Compute the augmented distance matrix: $D_S^{\prime(i)}$ and $D_F^{\prime(i)}$.
3. Merge speaker segment/face track pairs with minimum augment distance if they are less than certain thresholds.
4. Set $i = i + 1$, and update distance matrix $D_S^{(i)}$, $D_F^{(i)}$, and $C^{(i)}$.
5. If no merge happens, then stop, otherwise, go to the second step.

### 4.3. Major Cast Generation

After clustering, each speaker segment corresponds to one speaker, and each face track corresponds to one face. We need to further determine the major casts by linking the faces to corresponding speakers. Then, an importance score is assigned to each major cast, so that a list of sorted major casts is extracted.

Association of faces to speakers entirely depends on the speaker face correlation matrix. The value of speaker face correlation reflects both the temporal (time span) and the spatial (face size) importance of the major cast. In the following algorithm, we perform the speaker-face association and major cast ordering at the same time. Suppose after integrated speaker and face clustering, we get $M$ different speakers and $N$ different faces, and an $M \times N$ C matrix. The algorithm is as follows:

1. Set $i = 0$.
2. Find an entry in the C matrix with maximum correlation value; denote the row and column indices of this entry by $s_i$ and $f_i$, respectively.
3. Assign the speaker corresponding to row $s_i$ and the face corresponding to column $f_i$ to major cast i.
4. Remove row $s_i$ and column $f_i$ in C.
5. Set $i = i + 1$, and go to step 2 unless the maximum value in C is smaller than a threshold.

This algorithm produces a list of major cast with corresponding correlation values, which are used as temporal-spatial importance scores.

## 5. EXPERIMENTAL RESULTS

The experimental data consists of 8 half-hour news broadcasts collected from NBC Nightly News off the air in 2000. The audio track is sampled at 16 KHz with resolution 16 bits per sample. The visual track is digitized at 10 frames per second, with size $240 \times 180$. We use 4 broadcasts for training the clean and non-clean speech models and the rest are used for testing, denoted as sequence test1, test2, test3, and test4. The acquired data is manually segmented, tagged as clean speech or non-clean speech. Speaker identification in clean speech segments and frontal view face identification for each visual shot are also annotated. These labels are used as ground truth to train the required models and measure the performance of proposed algorithms.

In clean speech extraction, GMM with 2 mixtures achieves a raw error rate of 6.2%. If we smooth the classification results by considering those of neighboring clips, the error rate can be further reduced.

Table 1 gives the performance of speaker segmentation. If the detected speaker boundary is within 2 seconds away from the real boundary, we count it as a correct one, otherwise, a falsely detected one. The reason for high false alarms is due to the following two reasons. First, we intentionally set the detection thresholds low since we do not want to miss the real speaker boundaries. Second, some of the speaker segments of the same person have different background sound and different speaking styles.

| Test Data | test1 | test2 | test3 | test4 |
|---|---|---|---|---|
| Correctly Detected Segments | 75 | 75 | 73 | 89 |
| Falsely Detected Segments | 12 | 11 | 8 | 11 |
| Missed Segments | 3 | 2 | 2 | 1 |

**Table 1**. Speaker segmentation results.

Table 2 shows the results of integrated face speaker clustering. The second row shows the total number of speaker segments, and the third row gives the number of different speakers manually labeled. *Speaker splits* counts how many speakers are split into different clusters. If one speaker is distributed into N clusters, then this speaker contributes $N - 1$ in the final count. *Speaker mixes*

measures how many different speakers are mixed in one cluster, where, for each cluster, the number of different speakers minus one is counted. The measurements of face clustering results are similarly defined.

During the clustering, we intentionally tune the thresholds to reduce speaker/face mixes, which leads to higher speaker/face splits. By observing the speaker clustering results in detail, we find that the effect of speaker splits is not serious, since many of split speaker segments are short, about 3 to 5 seconds. Considering that the average duration of anchor person or reporter segments is more than 20 seconds, the influence of split segments is tolerable. The results of integrated speaker face clustering are consistently better than those based on speaker and face alone, which are not presented here.

| Test data | test1 | test2 | test3 | test4 |
|---|---|---|---|---|
| Total speaker segments | 87 | 86 | 81 | 100 |
| Different speakers | 34 | 41 | 36 | 42 |
| Speaker splits | 29 | 23 | 23 | 30 |
| Speaker mixes | 2 | 5 | 1 | 2 |
| Total face tracks | 30 | 18 | 21 | 23 |
| Different faces | 18 | 16 | 15 | 17 |
| Face splits | 1 | 1 | 3 | 2 |
| Face mixes | 0 | 2 | 1 | 5 |

**Table 2**. Integrated speaker face clustering results.

For the four test sequences, we detect 8, 9, 6, and 8 major casts respectively. Among all these characters, the most important ones are consistently the anchor persons, followed by different reporters and interviewees. Figure 3 shows the face images of the eight major casts detected in test1 in the order of their importance values. The top major cast is the anchor person: Tom Brokaw. The third, the forth and the last major casts are news reporters, and the rest are interviewees. The reason why some reporters earn low importance scores is that their faces only appear occasionally even when their speech is present during an entire reporting or interview period.


**Fig. 3**. Faces of major casts detected in test1.

We developed a major cast based video presentation system, which is shown in Figure 4. The panel on the left side shows the video, and the right panel displays the list of major casts in an intuitive and user friendly way. Speech segments of different major casts are painted in different colors. Major casts are presented row by row. For each major cast, we present the face image on the left, then a vertical bar representing the importance score, and finally a time streamline identifying the occurrences of speech. By this way, the user may easily get the impression of who are the major casts, and where do they appear in the entire video. The user can browse all portions of the video associated with a detected cast, or some specific portion of one major cast's appearance by clicking on the block in speech time line. We believe this presentation provides a good audio-visual summary of the underlying content of the video and enables fast browsing and retrieval of video databases.

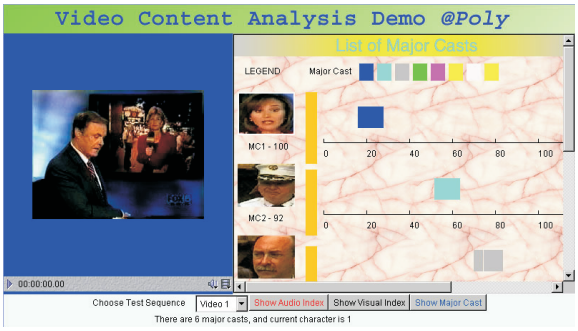


**Fig. 4**. Major cast presentation.

## 6. CONCLUSION

This paper proposes a new approach to detect major casts in a video based on both audio and visual information. The focus is on how to combine multiple cues in a problem that can not be reliably solved based on single modality. Specifically, the temporal correlation among speakers and faces are utilized to find the major casts. The preliminary experimental results show that the automatically generated list of major casts is meaningful and the proposed algorithm is promising.

## 7. REFERENCES


[1] T. Zhang and C.-C. Jay Kuo, "Hierarchical classification of audio data for archiving and retrieving," in *Proc. ICASSP*, Phoenix, AZ, Mar. 15-19 1999, vol. 6, pp. 3001–3004.

[2] Y. Rui, T. S. Huang, and S.-F. Chang, "Image retrieval: current technologies, promising directions, and open issues," *Journal of Visual Communication and Image Representation*, vol. 10, no. 1, pp. 39–62, Mar. 1999.

[3] C. Saraceno and R. Leonardi, "Identification of story units in audio-visual sequencies by joint audio and video processing," in *Proc. ICIP*, Chicago, IL, Oct. 4-7 1998, vol. 1, pp. 363–367.

[4] Q. Huang, Z. Liu, A. Rosenberg, D. Gibbon, and B. Shahraray, "Automated generation of news content hierarchy by integrating audio, video, and text information," in *Proc. ICASSP*, Phoenix, AZ, Mar. 15-19 1999, vol. 6, pp. 3025–3028.

[5] Z. Liu, Y. Wang, and T. Chen, "Audio feature extraction and analysis for scene segmentation and classification," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 20, no. 1, pp. 61–79, Oct. 1998.

[6] Z. Liu and Q. Huang, "Detecting news reporting using audio/visual information," in *Proc. ICIP*, Kobe, Japan, Oct. 1999, vol. 3, pp. 324–328.

[7] Z. Liu and Q. Huang, "Content-based indexing and retrieval-by-example in audio," in *Proc. IEEE Int. Conf. Multimedia & Expo*, New York, NY, July 30 - Aug. 2 2000.

[8] Z. Liu and Y. Wang, "Face detection and tracking in video using dynamic programming," in *Proc. ICIP*, Vancouver, Canada, Sept. 10 - 13 2000.

[9] S. Satoh, Y. Nakamura, and T. Kanade, "Name-it: Naming and detecting faces in news videos," *IEEE Multimedia Magazine*, vol. 6, no. 1, pp. 22–35, Jan-Mar 1999.