DETECTION OF UNIQUE PEOPLE IN NEWS PROGRAMS USING MULTIMODAL SHOT CLUSTERING

Cuneyt M. Taskiran †, *Alberto Albiol* ‡, *Luis Torres**, *and Edward J. Delp* †

†School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, 47907 U.S.A.

‡Technical University of Valencia
Valencia, Spain

*Technical University of Catalonia
Barcelona, Spain

ABSTRACT

In this paper we describe an approach that uses a combination of visual and audio features to cluster shots belonging to the same person in video programs. We use color histograms extracted from keyframes and faces, as well as cepstral coefficients derived from audio to calculate pairwise shot distances. These distance are then normalized and combined to a single confidence value which reflects our certainty that two shots contain the same person. We then use an agglomerative clustering algorithm to cluster shots based on these confidence values. We report the results of our system on a data set of approximately 8 hours of programming.

1. INTRODUCTION

In this paper we address the problem of detecting unique people appearing in television news programs. For each person, our goal is to determine the temporal regions where that person appears in the program. For this work, we have focused on video content obtained from C-SPAN networks. C-SPAN is a private, non-profit company in the United States that broadcasts meetings, sessions for the U.S. Senate and House sessions, interviews, and other news programming, without commercials as a public service. C-SPAN programs usually use 1-2 cameras and view-points and there is not much camera or object motion. For C-SPAN programs a change of speaker is generally marked with a shot boundary so each shot usually contains a single person of interest. In this case the detection of unique people may be achieved by grouping all shots containing the same speaker.

The objective of our work was to develop a system that would let indexers at the C-SPAN Video Archives generate shot-based labels for all speakers appearing in a given program. The Archives was established to record, index, and archive all C-SPAN programming. As of January 2002, the Archives contained 167,267 hours of C-SPAN programs. Although all programs are hand-indexed using a large number of descriptors, the indexing system used by the Archives currently cannot generate shot-based person labels, which would be too time consuming. However, such labels are highly desirable since they would enable searches with higher granularity based on video shots rather than complete programs. These labels can also be used to automatically generate on-screen speaker identification graphics and chapters for the DVD versions of C-SPAN programs [1]. Besides the ones mentioned above, there are many other applications of this work in video analysis. For example, the accuracy of speech recognition may be improved by performing speaker adaption over each group of shots [2].

Many systems were proposed to perform the person detection problem using only the audio information [3, 4]. An example of video indexing using multimodal features is found in [5] where audio transcripts, video captions, and face detection was used to associate faces and names in videos.

In this paper we describe an approach that uses a combination of visual and audio features to cluster shots belonging to same persons. We assume no prior knowledge about the number of persons appearing in sequences. We illustrate our technique on a collection of video content obtained from the C-SPAN Video Archives. The paper is organized as follows: In Section 2 we describe the features extracted from each shot and how pairwise shot distances are calculated based on these distances. Section 3 explains the distance normalization procedure we use to convert distance values to confidence values. Our hierarchical clustering procedure is explained in Section 4. The detection results are presented in Section 5 and Section 6 presents conclusions.

2. SHOT FEATURE EXTRACTION AND DISTANCE CALCULATION

We first segment a given video to be processed into its constituent shots. For this task only visual features, in the form of color histograms and pixel variances of frames are used. Our shot boundary detection technique is described in detail in [6]. After the video sequence is divided into shots we extract a number of features from each shot for the person detection task. We use color histograms of face regions, color histograms of shot keyframes, audio, and shot adjacency as our features. After these features are extracted from each shot, we calculate pairwise shot distances as described below.

2.1. Distance based on face histogram features

Faces provide powerful cues in comparing different speakers. We detect faces in shots and use the histogram of the face regions as one of our features. For face detection we examine every 10th frame, which reduces the computational load without decreasing detection accuracy for most video programs. In this work we have used the face detector described in [7], which is based on a boosted cascade of simple classifiers and yields a highs detection rate. The result of face detection is a list of face regions that are represented by their bounding boxes. The pixels within each face box are converted from the original RGB space to normalized RGB values

This work was partially supported by a grant from the C-SPAN Archives. Address all correspondence to E. J. Delp, ace@ecn.purdue.edu or telephone: +1 765 494 1740.

using the formulas

$$R_n = \frac{R}{\Sigma},\ G_n = \frac{G}{\Sigma},\ B_n = \frac{B}{\Sigma}, \tag{1}$$

where $\Sigma = R + G + B$. Since one of the normalized components is redundant, we use only the R_n and G_n color components. This color normalization step is important to reduce changes in pixel values caused by the automatic gain control available in many video cameras.

After the bounding boxes for faces are detected, we derive a normalized histogram for the R_n and G_n color components of the pixels within the face region. We then obtain average face histograms, $HF^i_{R_n}$ and $HF^i_{G_n}$, for shot s_i by averaging the face histograms for the frames in s_i. The face histogram distance between two shots, s_i and s_j, both containing faces, is calculated using

$$\begin{aligned} d_f(i,j) \quad &= \tfrac{1}{2}\textstyle\sum_{k=1}^{BF} |HF^i_{R_n}(k) - HF^j_{R_n}(k)| \\ &+ \tfrac{1}{2}\textstyle\sum_{k=1}^{BF} |HF^i_{G_n}(k) - HF^j_{G_n}(k)|, \end{aligned} \tag{2}$$

where BF is the number of bins for face histograms. Currently we use the value $BF = 256$, which produced good results for our data set.

2.2. Distance based on keyframe histogram features

There are cases where face detection fails to detect a person appearing in a shot or the detected face region is too small to provide reliable comparison between different persons. In order to handle these cases we use an additional color histogram feature, which is derived from the keyframe of each shot. For simplicity, we have selected the middle frame of each shot as the keyframe. We calculate the normalized histograms for the R_n and G_n color components for each keyframe. When calculating the color histograms, an area that is approximately the bottom one-third of the keyframe is ignored. In C-SPAN programs, as well as most other news programs, this area contains on-screen graphics, with similar color content for different people, illustrated in Figure 1. In order to make the keyframe histogram feature more sensitive to differences between shots containing different people, we do not include this region in the histogram. For each shot s_i we obtain the keyframe histograms $HK^i_{R_n}$ and $HK^i_{G_n}$. The keyframe histogram distance between shots is computed similar to the face histogram distance using

$$\begin{aligned} d_k(i,j) \quad &= \tfrac{1}{2}\textstyle\sum_{k=1}^{BK} |HK^i_{R_n}(k) - HK^j_{R_n}(k)| \\ &+ \tfrac{1}{2}\textstyle\sum_{k=1}^{BK} |HK^i_{G_n}(k) - HK^j_{G_n}(k)|, \end{aligned} \tag{3}$$

where BK is the number of bins for keyframe histograms. Currently we use the value $BK = 128$ that provided a good compromise in our tests. However, the performance of the system is not very sensitive to the particular choice of BK and a smaller number of bins can be used.

2.3. Shot Similarity based on audio features

For each shot we extract acoustic vectors consisting of the 20 mel-frequency cepstral coefficients obtained every 10 ms using a 20 ms Hamming window. Let $\mathbf{x}^i = \{x^i(k) \in \Re^n, k = 1, \ldots, N_i\}$ and $\mathbf{x}^j = \{x^j(k) \in \Re^n, k = 1, \ldots, N_j\}$ be acoustic vectors extracted from shots s_i and s_j, respectively, where N_i denotes the number of 10ms segments for shot s_i and n is the dimensionality of the acoustic vectors, in our case $n = 20$. We define $\mathbf{x}^{i,j} = \{\mathbf{x}^i \cup \mathbf{x}^j\}$ to be the combined set of acoustic vectors for shots s_i and s_j.

Fig. 1. A typical frame extracted from a C-SPAN program. The lower part with on-screen graphics is ignored when calculating the keyframe histogram feature.

The extracted acoustic vectors are modeled using multivariate Gaussian distributions. Three different models $\mathcal{M}_i = \{\mu_i, \Sigma_i\}$, $\mathcal{M}_j = \{\mu_j, \Sigma_j\}$, $\mathcal{M}_{i,j} = \{\mu_{i,j}, \Sigma_{i,j}\}$ for $\mathbf{x}^i$, $\mathbf{x}^j$ and $\mathbf{x}^{i,j}$ are built, where μ_i and Σ_i are the mean vector and covariance matrix for the vectors in shot s_i, respectively. The dissimilarity between two audio segments corresponding to two shots is based on the comparison of their statistical models using the Bayesian Information Criterion (BIC) [8]. The BIC, which has been introduced in the statistics literature for model selection, is a likelihood criterion penalized by model complexity, which is represented by the number of model parameters. The key idea behind BIC is that, although a higher likelihood can be obtained using separate models for audio segments i, j, this is done at the expense of a greater complexity, i.e., more parameters, that reduces the BIC value.

We use a similar approach to [8] and define the audio distance between two shots using BIC as follows

$$\begin{aligned} d_a(i,j) \quad &= N_i|\Sigma_i| + N_j|\Sigma_j| \\ &+ \tfrac{1}{2}(n + \tfrac{1}{2}n(n+1))\log(N_i + N_j) \\ &- (N_i + N_j)|\Sigma_{i,j}|. \end{aligned} \tag{4}$$

One important restriction of this distance is that enough data must be available to model the acoustic vectors. Based on the results given in [8], we calculate the audio distance, $d_a(i,j)$, only when both the audio segments corresponding to shots s_i and s_j are longer than 2 seconds.

3. FEATURE NORMALIZATION

For each pair of shots in a given sequence, we use the distance values described in Section 2 to derive a single confidence value for each pair of shots, which is a continuous value in $[0, 1]$, where 0 indicates that the two shots contain the same speaker with high confidence, and 1 indicates that the shots contain different speakers. This distance normalization step is necessary in comparing shot distance values that have different ranges.

Let $p(d|S)$ and $p(d|D)$ be the probability density functions (pdfs) of the distance between two shots, $d(s_i, s_j)$, when they contain the same person and different persons, respectively. Empirical pdfs obtained from our training data set for the three shot distances introduced in Section 2 are shown in Figure 2. We then can obtain the cumulative distribution functions for shot distances for the case

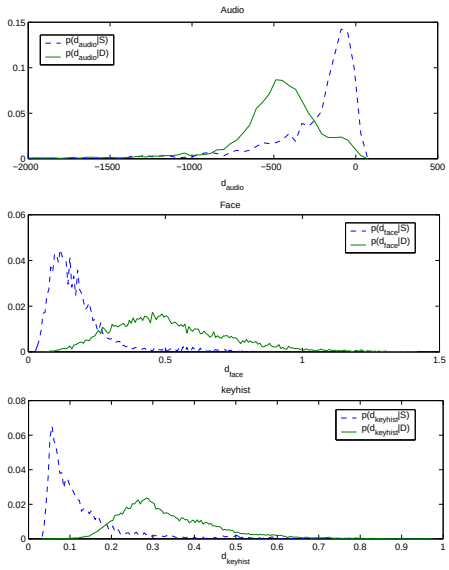

Fig. 2. Conditional probability density functions for the audio, face histogram, and keyframe histogram distances.

of same and different speakers as

$$\begin{aligned} P(d\,|S) &= \int_{-\infty}^{d} p(y\,|S)\,dy \\ P(d\,|D) &= \int_{d}^{\infty} p(y\,|D)\,dy. \end{aligned} \tag{5}$$

This normalization is performed on all three of the shot distances. For each value of the shot distance, d, these cumulative distribution functions give the conditional probabilities that the shots contain same or different speakers, respectively. Using these conditional probabilities we define the normalized distance between shots as

$$d_n = \frac{P(d|S)}{P(d|S) + P(d|D)}. \tag{6}$$

Once shots distances corresponding to different features are calculated and normalized using Equation 6, we derive a single distance value that is used to cluster similar shots together. Note that, as described in Section 2, not all features are available to calculate distance values for all pairs of shots s_i and s_j. Therefore, for each pair we derive the single shot distance values using the available shot feature distances, as illustrated in Figure 3. In our analysis of news programs we have found that consecutive shots very rarely contain the same person. For this reason, we set the confidence value between consecutive shots to be equal to 1.

4. SHOT CLUSTERING

After pairwise shot distances are calculated for a sequence, we use agglomerative or bottom-up clustering to group shots containing the same person together. Let N be the number of shots in the given video sequence to be processed. At the start of the algorithm all shots are placed in separate clusters so we start with N clusters. Then, at each iteration of the agglomerative clustering algorithm we search through the distance matrix, $\{d_{ij} = d(i,j)\}, 1 \leq i < j \leq N$, and find the two clusters, r and s, having the minimum distance. These two clusters are merged into a new cluster k

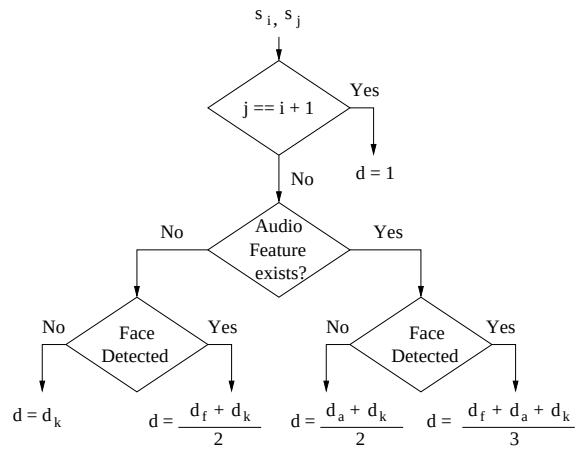

Fig. 3. Scheme used to derive a single shot distance value from multimodal feature distance values.

and the distances from the newly formed cluster k to all remaining clusters are updated. We have used the complete-link update rule [9] for updating distances, which is given as

$$d_{hk} = \max(d_{hr}, d_{hs}),\ 1 \leq h \leq N,\ h \neq k. \tag{7}$$

The algorithm stops if either the number of clusters is equal to N_{min} or the smallest distance found is less than a threshold τ. We assume that there will be at leat two people in all programs, so we used the value $N_{min} = 2$. For the threshold on cluster distances we have used the value $\tau = 0.5$. Note that this value of the threshold is not an arbitrary choice, but reflects the upper bound on the normalized confidence value for shots containing the same person. This is due to our normalization process, described in Section 3.

5. RESULTS

For our speaker detection experiments we have used 13 C-SPAN sequences that contain more than two persons. Information about these sequences is listed in Table 1. For each sequence, we cluster the shots belonging to the sequence to obtain the unique persons appearing in the program. In evaluating the accuracy of our person detection system we have used the following rules

1. Only keyframes containing people of interest were considered in our error analysis, that is, key frames containing the audience, wide shots, etc. were ignored.
2. A *false split* (FS) was declared if the keyframes belonging to the same speaker are split into two different clusters. For example, if the same person is split into three clusters, we count this as two false splits. However, shots of the same person obtained using different camera angles were not considered as false splits if they are grouped in different clusters.
3. A *wrong grouping* (WG) was declared if a keyframe for a person is grouped with that of another one in the same cluster. For example, if a cluster contains keyframes from three different people, we count this as two wrong groupings.

The speaker detection results for our data set are given in Table 2. From the results we observe that our system is accurately able to

sequence name	program type	number of shots	duration (min)
cspan3	house committee	14	20
cspan4	senate committee	23	20
cspan7	forum	26	20
cspan11	senate proceeding	9	40
cspan12	senate proceeding	22	40
cspan15	house proceeding	21	40
cspan17	house proceeding	42	40
cspan18	house committee	36	40
cspan20	forum	23	40
cspan21	house committee	41	40
cspan22	house proceeding	85	80
cspan25	forum	61	60
cspan27	panel	59	55
TOTAL		**462**	**535**

Table 1. Information about the C-SPAN sequences used in our experiments.

sequence name	number of speakers	FS	WG
cspan3	3	0	0
cspan4	4	0	0
cspan7	3	2	1
cspan11	4	0	0
cspan12	4	0	0
cspan15	9	0	1
cspan17	12	0	1
cspan18	6	4	2
cspan20	6	0	0
cspan21	5	1	0
cspan22	11	4	2
cspan25	8	2	3
cspan25	8	0	0
TOTAL	**82**	**13**	**10**

Table 2. Unique person detection results.

detect unique people appearing in a wide variety of programs. The errors are mostly localized to three sequences: $cspan18$, $cspan22$, and $cspan25$. The main source of errors for these sequences is what we call "voice overs," which occur when a shot of a person contains the audio for another person. Currently we are working on an updated version of our system that uses audio continuity to get rid of these errors.

6. CONCLUSIONS

In this paper we have described a system that uses multimodal features to detect unique people appearing in news programs and illustrated our results using data obtained from C-SPAN. For most programs our system is able to accurately detect the people in programs. However, errors may arise when the audio for a shot does not belong to the person shown in the shot. We are implementing measures based on audio continuity to solve this problem. We are also looking into learning algorithms to automatically determine weights for the three different kinds of shot distances to increase performance.

Fig. 4. Shots belonging to clusters 12, 13, and 14 from the sequence *cspan17* First and third clusters form a false split and the third cluster has a wrong grouping error.

7. REFERENCES

[1] Cuneyt Taskiran, Anthony Martone, Robert X. Browning, and Edward J. Delp, "A toolset for broadcast automation for the C-SPAN networks," in *5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2004)*, Lisboa, Portugal, April 21 – 23 2004.

[2] Zhi-Peng Zhan, Sadaoki Furui, and Katsutoshi Ohtsuki, "On-line incremental speaker adaptation with automatic speaker change detection," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey, 5 – 9 June 2000.

[3] H.J. Nock, G. Iyengar, and C. Neti, "Speaker localisation using audio-visual synchrony: An empirical study," in *Proceedings of the International Conference on Image and Video Retrieval (CIVR 2003)*, Urbana-Champaign, IL, 2003, pp. 488 – 499.

[4] Lie Lu and Hong-Jiang Zhang, "Speaker change detection and tracking in real-time news broadcasting analysis," in *Proceedings of the 10th ACM International Conference on Multimedia*, Juan-les-Pins, France, December 1 – 6 2002, pp. 602 – 610.

[5] 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, January 1999.

[6] Cuneyt Taskiran, Jau-Yuen Chen, Alberto Albiol, Luis Torres, Charles A. Bouman, and Edward J. Delp, "ViBE: A compressed video database structured for active browsing and search," *IEEE Transactions on Multimedia*, vol. 6, no. 1, pp. 103–118, February 2004.

[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Conference on computer vision and pattern recognition*, Kauai, HI, December 8-14 2001.

[8] S. S. Chen and P. S. Gopalakrishnan, "Speaker, environment, and channel change detection and clustering via the Bayesian Information Criterion," in *DARPA Speech Recognition Workshop*, Landsdowne, VA, February 1998, pp. 127–132.

[9] Anil K. Jain and Richard C. Dubes, Eds., *Algorithms for Clustering Data*, Prentice Hall, New Jersey, 1988.