Speaker Dependent Video Indexing Based on Audio-Visual Interaction

S. Tsekeridou, I. Pitas
Department of Informatics, University of Thessaloniki,
Thessaloniki, GREECE, GR-540 06
e-mail: {sofia,pitas}@zeus.csd.auth.gr

Abstract

A content-based video indexing method is presented in this paper that aims at temporally indexing a video sequence according to the actual speaker. This is achieved by the integration of audio and visual information. Audio analysis leads to the extraction of a speaker identity label versus time diagram. Visual analysis includes scene cut detection, face shot determination, mouth region extraction and tracking and finally talking face shot determination. Results from both sources are combined to improve speaker-dependent video indexing. Such a task enables flexible video retrieval or browsing in cases where queries according to speaker identities are imposed. Speaker recognition errors are reduced to 2%.

1 Introduction

Content-based video indexing and retrieval have recently grown to active research topics due to multimedia requirements for flexible video manipulation. Since video sources contain rich information, both tasks are complicated and usually focus on specific applications, e.g topic or scene discrimination [1, 2], speaker recognition [3], key-frame extraction.

A content-based video indexing method is presented that aims at temporally indexing a video sequence according to the actual speaker. Each speaker is labelled by a number $1, .., k$. Indexing is achieved by the integration of audio and visual information. The audio processing module outputs a speaker identity label versus time diagram. The video processing module successively estimates scene cuts, face shots and talking faces. Results from both sources are combined to improve indexing. Such a task enables flexible video retrieval in cases where queries according to speaker identities are imposed. If performed manually it would be a very complex and time consuming task. Simulation results exhibit the satisfactory performance of the video indexing scheme. Recognition errors are reduced to 2%.

2 Audio Processing Module

2.1 Audio Analysis

Audio analysis is performed as shown in Figure 1. In the following, the term *frame* is the one used in short-term audio analysis, whereas the term *segment* refers to a variable number of successive speech frames grouped together due to having a similar attribute. Initially, *silence detection* is performed to discard silence periods. Silence frames are audio frames of background noise with a low energy level with respect to voice segments. The signal energy of an audio frame is evaluated by: $E_i = \sqrt{\sum_{k=1}^{M} x_i(k)^2}$, $i = 1, .., N$, where M is the number of samples in a speech frame and N is the total number of speech frames. Thresholding by $E_{th} = E_{min} + 0.1(E_{max} - E_{min})$ leads to silence frame detection, being those having $E_i < E_{th}$. E_{min} and E_{max} are the lowest and greatest values of E_i. The silence detector also achieves to fairly detect the boundaries between words or phrases.

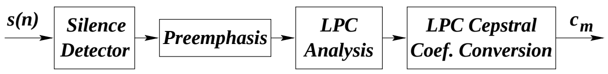

Figure 1: Audio analysis

Audio feature extraction is performed only in voice frames [4]. The speech signal is initially preemphasized by an FIR filter with transfer function $H(z) = 1 - 0.95z^{-1}$. Speech frames of 30msec are extracted with an overlap of 20msec from each other. Each frame is windowed by a Hamming window of size M. LPC analysis succeeds and the LPC parameter set α_m is later on converted to LPC cepstral coefficients: $c_m = \alpha_m + \sum_{k=1}^{m-1} (\frac{k}{m}) c_k a_{m-k}$, $1 \leq m \leq p$.

2.2 Speaker Classification

Speaker modelling is performed by an LVQ3 classifier [5, 6]. All speaker codebooks are trained in parallel using training vectors from apriori known time periods in which a certain speaker speaks. Initialization of the codebooks consisting of 32 codevectors is performed by the LBG algorithm. Classification of a training vector $\mathbf{c}$ to a speaker codebook $\mathbf{m}_i$ is performed judging from Mahalanobis codevector distance values $d(\mathbf{m}_{ik}, \mathbf{c})$, $k = 1, .., 32$:

$$d(\mathbf{m}_{ik}, \mathbf{c}) = (\mathbf{m}_{ik} - \mathbf{c})^T \mathbf{R}_i^{-1} (\mathbf{m}_{ik} - \mathbf{c}) \qquad (1)$$

where $\mathbf{m}_{ik}$ represents the k-th codevector of codebook $\mathbf{m}_i$ and $\mathbf{R}_i$ denotes the global covariance matrix derived from the training vectors of $\mathbf{m}_i$. Those $\mathbf{m}_{ik}$ of every $\mathbf{m}_i$ that were assigned the minimum $d(\mathbf{m}_{ik}, \mathbf{c})$ are selected. Finally, $\mathbf{c}$ is classified to $\mathbf{m}_i$ if the selected $d(\mathbf{m}_{ik}, \mathbf{c})$ is the minimum of all others.

In succession, correct classification and misclassification cases are appropriately used for speaker codebook updating and total distortion minimization. If a training vector $\mathbf{c}$, known to belong to $\mathbf{m}_j$, is classified at iteration t to $\mathbf{m}_i$ and $\mathbf{c}$ lies in a predefined "window", then updating of $\mathbf{m}_i$ and $\mathbf{m}_j$ at iteration $t + 1$ is performed by:

$$\begin{aligned} \mathbf{m}_i(t+1) &= \mathbf{m}_i(t) - \alpha(t)[\mathbf{c} - \mathbf{m}_i(t)] \\ \mathbf{m}_j(t+1) &= \mathbf{m}_j(t) + \alpha(t)[\mathbf{c} - \mathbf{m}_j(t)] \qquad (2) \end{aligned}$$

$\mathbf{c}$ is assumed to lie in the "window" if $\frac{s_j - s_i}{s_j} > w$, where w is a constant set equal to 0.25 and s_j and s_i are the Mahalanobis distances of $\mathbf{c}$ from the true codevector of $\mathbf{m}_j$ that it belongs to and from the codevector of $\mathbf{m}_i$ that it has been classified to respectively. In the case that $\mathbf{c}$ is classified correctly to $\mathbf{m}_j$ then updating at iteration $t + 1$ is performed by:

$$\mathbf{m}_j(t+1) = \mathbf{m}_j(t) + \epsilon\alpha(t)[\mathbf{c} - \mathbf{m}_j(t)] \qquad (3)$$

It is noted that updating is performed only to the true codebook $\mathbf{m}_j$. $\alpha(t)$ is a linearly decreasing function depending on the number of iterations which controls the learning rate. The constant ϵ is set usually to a small value (0.25 in our case). Training is performed iteratively until the overall average distortion falls below a small value or the maximum number of iterations has been reached.

During the recognition process, the distortion $s_i(\mathbf{c}_t)$ of an input test vector $\mathbf{c}_t$ when it is quantized using the speaker codebook $\mathbf{m}_i$ is calculated by:

$$s_i(\mathbf{c}_t) = \min_{\mathbf{m}_{ik} \in \mathbf{m}_i} [d(\mathbf{m}_{ik}, \mathbf{c}_t)], k = 1, .., 32 \qquad (4)$$

Index i spans the number of speaker classes. $\mathbf{R}_i$ has been evaluated during the training process.

In our approach, instead of considering input test vectors independently, entire speech segments are used consisting of a variable number of input vectors derived from successive speech frames lying between two silence frames. A coarse detection of words or phrases is performed in this way. Such an approach is adopted based on the fact that between two silence frames only one speaker is bound to be talking. Thus, the average distortion of the whole speech segment is evaluated by:

$$S_i = \frac{1}{T} \sum_{t=1}^{T} s_i(\mathbf{c}_t) \qquad (5)$$

where T is the number of speech frames consisting the current speech segment. The unknown speaker is identified as the reference speaker whose model gives the smallest quantization error:

$$SpeakerID = \underset{1 \leq i \leq N_{Speakers}}{\arg\min} \; S_i \qquad (6)$$

Finally, the time interval that corresponds to the speech segment is assigned to the identified speaker. Thus, *a speaker identity label versus time diagram* is extracted. Speaker interchangeability is also tested. The training algorithm is computationally intensive. However, training and even testing could be performed in non-real time since the only medium used for further speaker dependent video retrieval is the stored labelling time diagram.

3 Video Processing Module

The video processing module is shown in Figure 2. The system input $v(n)$ is the entire video sequence. Its output is a time sequence $S(t)$ mapping the different shot boundaries detected, which is processed by the face shot detection algorithm in order to estimate the time sequence $F(t)$ containing the additional information of identified face shots. Mouth regions are detected by a facial feature extraction method and tracked along face shots to estimate a Mean Square Error ($MSE(t)$) time sequence denoting the variations of the mouth area among successive frames of the same face shot.

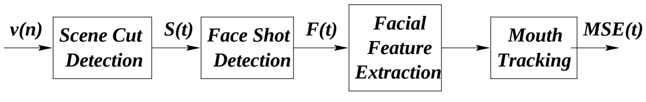

Figure 2: Video Processing module

Scene cut detection is performed by evaluating color frame differences $FD(t)$ between successive frames and thresholding by $FD_{th} = E[FD(t)] + 0.3 * (FD_{max} - E[FD(t)])$. Frame differences above FD_{th} indicate a possible scene change and thus a shot boundary. Although the proposed scene cut detection method may result in oversegmentation of the video sequence, especially in single shots where abrupt changes occur, shots containing faces are well characterized due to their rather slow varying nature, even in complex background.

In order to *detect and isolate face shots*, the first frame of every shot is processed by a face localization algorithm, which is based on the assumption that faces could be characterized by their skin-like color, their elliptical shape and the x/y ratio of their surface. Since skin-like colors are better represented in the HSV color space, color transformation from the RGB to the HSV color space is performed [7]. Pixels are afterwards selected having skin-like colors by setting appropriate thresholds to H, S and V values, specifically $0^0 \leq H \leq 25^0$, $335^0 \leq H \leq 360^0$, $0.2 \leq S \leq 0.6$ and $V \geq 0.4$. The range of H restricts segmentation in reddish colors and the range of S ensures exclusion of pure red or very dark red colors. Both account for small variations in lighting conditions. V thresholding is introduced to discard dark colors (such as dark hair). This color segmentation process proves to be adequately robust even with complex background or slightly varying lightning conditions. An initial map of all pixels having colors close to the ones corresponding to skin is created. Refinement of this map is successively attempted in order to eliminate secluded assigned pixels or add non-assigned ones if they are appropriately surrounded by assigned ones.

The next task is to *detect different skin-colored objects* in the frame. Object determination is performed by an iterative merging algorithm which considers 4-neighbourhood connectivity and which attempts to fill the "holes" that possibly exist in a compound object (e.g. the regions of eyes that color segmentation has discarded). The merging algorithm starts from the left-most assigned pixel of the currently considered object and proceeds by assigning the next pixels of the same row to the current object, if they are assumed to be connected with previously assigned to this object pixels. Start and end points are saved and search in the next row is initialized from these points on. The algorithm is performed iteratively until no more connected pixels to the current object are detected. The process is computationally efficient and only a few iterations are required.

After the different skin-colored objects have been extracted, *a decision is made on which of them possibly represents a face.* The decision is based on the size of the object with respect to the size of the image as well as on the ratio of its width to height which must be in a certain range considering the facial x/y ratio. In cases where only one person appears in the scene, such a decision method performs satisfactorily. After the decision is made, all other objects are discarded from the initial map, leaving only the face region. This final map is called *face region map.* It is noted that video processing up to now was performed at a coarse resolution level of the initial frame.

Coarse facial feature extraction by detecting only the eyes, mouth and chin aiming at locating the mouth region follows. Facial feature candidates are initially extracted by minima analysis in the vertical profile of the face estimated using only the pixels assigned to the face region map. Information from all three color channels is used, which proves to enhance the discrimination between the features. Minima locations in the y direction are extracted using the first and second derivatives of the vertical face profile. At this point, our aim is to roughly extract the y-position of the eyes. Eyes are searched for in the upper part of the face. Once a rough estimate of the y-position of the eyes is derived, blocks with dimensions relevant to the face object size are initialized from either side of the center between the eyes, set equal to the middle-x point of the face object, and search regions are defined to further estimate the eyes by a block matching approach. Due to the mirror-symmetry of the face in frontal or semi-frontal views, such a process will lead to a good estimation of the eye locations and to the evaluation of the rotation angle of the head, i.e. $\theta = \arctan(D_y/D_x)$, where D_x and D_y are the distances between the centers of the matched blocks in the x- and y-directions, respectively.

Head rotation compensation and re-estimation of the vertical face profile follow in order to obtain a better estimation of the exact location of the mouth area. Minima analysis in the same way as described above is performed restricting the selection of minima to those lying in the upper/lower part of the rotated face and their distance being in a certain range with respect to the face width based on biometric facial analogies. After feature extraction in the rotated frame, feature locations are rotated back to the initial coordinate system and a mouth block is initialized around the mouth center estimated beforehand. It is obvious that compensation for rotation enhances the accuracy of facial feature extraction.

A mouth tracking process is further initialized inside a single face shot using the previously estimated mouth region as reference. A block matching approach is adopted and the locations of the mouth region in successive frames are estimated by evaluating the minimum MAE between the current and reference frames. A search region is appropriately initialized to account for motion of the head besides the mouth movement. Thus, the mouth regions in the detected face shots are extracted and the MSE between successive mouth blocks inside each shot is calculated to further detect movement of the mouth. Such movement implies a talking face. Obviously, errors in the estimation of the mouth region would result in errors concerning the decisions for talking or non-talking faces. A final time diagram of talking face, non talking face and no face shot indices is derived to be combined with the results of the audio processing module.

4 Audio-Visual Interaction

A combination of the labeling time diagrams obtained by audio and visual processing is achieved by simple decision rules. Boundaries of a face shot ensure the existence of a person. Mouth movement detection in this shot implies that this person speaks. Non face shot durations cannot be used for speaker detection, since interchangeability between speakers cannot be detected by the visual information. Consequently, refinement of the speaker dependent indexing achieved by the audio processing module is performed in face shots with talking faces.

The refinement process involves estimation of speaker presence likelihoods in every face shot. The presence likelihood of speaker SP_i in a face shot FS_k is evaluated by:

$$P(SP_i|FS_k) = \frac{M_i}{L_k} \tag{7}$$

where M_i is the number of speech frames assigned to speaker SP_i and L_k is the total number of speech frames in face shot FS_k. The speaker that exhibits the maximum presence likelihood is the winner. All speech frames in face shot FS_k are indexed with the winner's identity. This approach enables quite accurate detection of transition points between speakers in different successive face shots.

5 Simulation Results

The speaker dependent video indexing technique has been tested on various video sequences. We show here its application to a news sequence of approximately 6min duration. 5 different speakers exist but the faces of only three of them appear in the sequence. Two of the speakers appear in different parts of the sequence. The video sequence has been recorded with a frame rate of 25fps at 184×136 resolution. The audio signal has been sampled at 22kHz and each sample is a 16bit signed integer. Synchronization of the audio and video sources was attained during recording.

At the audio processing module, training has been performed on 40secs of continuous speech for each speaker. The silence detector exhibits a detection error of 0.6% of silence frames. It fails to detect a silence segment of high background noise. The indexing map achieved at this stage is shown in Figure 6, middle plot. Figure 6, bottom plot, represents the actual one. Silence segments have been removed to enable better assessment of the results. A recognition error of 5.23% of speech frames is obtained.

At the video processing module, Figures 3(a), (b) and (c) show results obtained by the face object detection, the eye block determination and the head rotation compensation techniques, respectively. Examples

(a) (b) (c)

Figure 3: Results from: (a) the face object detection, (b) the eye blocks determination, and (c) the head rotation compensation.

of mouth tracking can be seen in Figures 4(a)-(d). The scene cut mapping (bottom) along with the talking face shot mapping (top) are shown in Figure 5. The shot boundary mapping function is 1 if a scene cut is detected and 0 otherwise. A value of -1 has been asiigned to a video segment where over-segmentation has occured due to abrupt scene changes. However, face shots with talking persons are perfectly extracted. The talking face-no face mapping function is 1 if a talking face shot is present, -1 if a no face shot exists and -0.5 at the end of a face shot.

Combination of both audio and video source enhances indexing reducing the recognition error to 2.13% of speech frames. Figure 6, top plot, shows the achieved speaker dependent indexing diagram.

6 Concluding remarks

Content-based video indexing systems offer a flexible and efficient tool for further video retrieval and

(a) (b) (c) (d)

Figure 4: Examples of mouth region tracking.

browsing. A speaker dependent video indexing system has been presented that combines information from both audio and visual sources to enhance indexing, a fact experimentally proved. Estimated speaker transitions converge to the actual ones. However, background noise or the nature of continuous speech may limit the success of speaker recognition based on audio information. Furthermore, very bad lightning conditions could result in failure of the face detection method or abrupt changes in a single shot in scene-based over-segmentation. Such imperfections are compensated for when results are combined.

References

[1] C. Saraceno and R. Leonardi. Audio as a support to scene change detection and characterization of video sequences. In *Proc. of ICASSP'97*, volume IV, pages 2597–2600, Munich, Germany, 21-24 April 1997.

[2] J. Nam and A. Tewfik. Combined audio and visual streams analysis for video sequence segmentation. In *Proc. of ICASSP'97*, volume IV, pages 2665–2668, Munich, Germany, 21-24 April 1997.

[3] J. Luettin, N.A. Thacker, and S.W. Beet. Speaker identification by lipreading. In *Proc. of ICIP'96*, Philadelphia, USA, 1996.

[4] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition.* Englewood Cliffs, N.J.: Prentice Hall, 1993.

[5] T. Kohonen. The self-organizing map. *Proc. of the IEEE*, 78(9):1464–1480, September 1990.

[6] I. Pitas, C. Kotropoulos, N. Nikolaidis, R. Yang, and M. Gabbouj. Order statistics learning vector quantizer. *IEEE Trans. on Image Processing*, 5(6):1048–1053, June 1996.

[7] K. Sobottka and I. Pitas. A novel method for automatic face segmentation, facial feature extraction and tracking. *Image Communication, Elsevier*, 1997. to appear.

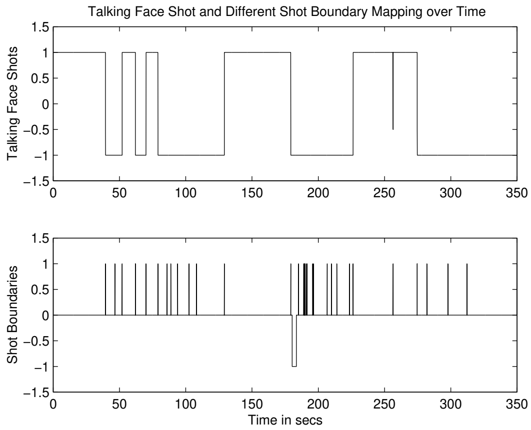

Figure 5: Scene Cut Detection and Talking Face-No Face Mapping over time.

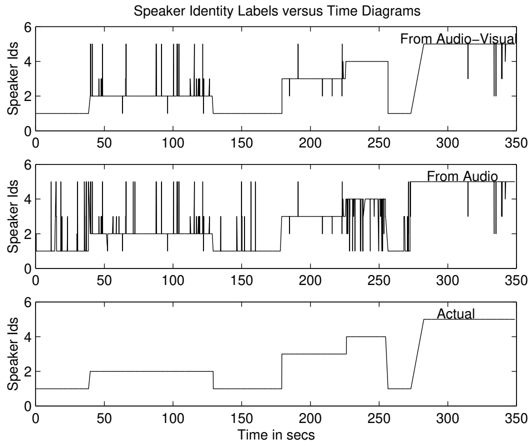

Figure 6: Speaker identity label over time diagram. Bottom: Actual, Middle: Estimated from the audio source, Top: Estimated by combining both sources.