Audio-Visual Content Analysis for Content-based Video Indexing

Sofia Tsekeridou and Ioannis Pitas
Department of Informatics, Aristotle University of Thessaloniki
Box 451, 54006 Thessaloniki, Greece
e-mail: {sofia,pitas}@zeus.csd.auth.gr

Abstract

An audio-visual content analysis method is presented, which analyzes both auditory and visual information sources and accounts for their inter-relations and coincidence to extract high-level semantic information. Both shot-based and object-based access to the visual information is employed. Due to the temporal nature of video, time has to be accounted for. Thus, time-constrained video labelling functions are generated. Audio source parsing leads to the extraction of a speaker identity mapping function over time. Visual source parsing results in the extraction of a talking face shot mapping function over time. Integration of the audio and visual mappings constrained by interaction rules leads to more detailed video content descriptions and even partial detection of its context.

1. Introduction

Content-based video parsing, indexing and retrieval have recently grown to active research topics due to the enormous amount of *unstructured* video data available nowadays, the spread of its use as a data source in many applications and the increasing difficulty in its manipulation and retrieval of the material of interest. Many application areas have emerged such as multimedia publishing, video-on-demand (VOD), digital video libraries, entertainment, broadcasting or surveillance systems [1]. The need for content-based indexing and coding has been foreseen by ISO/MPEG that has introduced two new coding standards: MPEG-4 and MPEG-7 [2].

Content-based video parsing involves temporal video segmentation into elementary units and content extraction of those units based on visual and/or audio semantic primitives. Thus a structured organization of the video source based on content is achieved and further used for indexing and efficient retrieval or browsing. Manual content-based video segmentation and indexing is a tedious and time consuming process and becomes impractical when an immense amount of video data has to be handled. In summary, content descriptors have to be found and automatic or semi-automatic methods should be used for the structured organisation of video based on content. A multimodal approach will evidently lead to far better results. Due to the inability for a universal definition of content, different types of content have been addressed, e.g. violence detection [3], scene change detection [4], news indexing [5]. Limited efforts have been made to apply general descriptors [6]-[8]. Another approach attempts to provide visual or iconic summaries to the user to facilitate browsing [9]-[11]. A third approach aims at serving user queries by example [12].

A content analysis method is presented which analyzes both auditory and visual sources and accounts for their inter-relations and synergy to extract high-level content information. Due to the temporal nature of video, time has to be accounted for. Thus, time-constrained video representations and indices are generated. Audio source parsing leads to the extraction of a speaker identity mapping over time. Video source parsing results in the extraction of a talking face shot mapping over time. The semantic labels employed are: speech, silence, speaker identities, no face shot, face shot (containing a person), talking face shot (containing a talking person). Additional semantic labels can be inserted if further processing is performed. The interaction of audio and visual labels supplies the user with more detailed information and a degree of its context.

2. Audio Source Parsing Module

2.1. Low-level Audio Analysis

Low-level audio analysis involves the tasks of speech-silence discrimination, voiced-unvoiced discrimination, linear predictive (LP) analysis and derivation of LP cepstral coefficients of the speech-voiced classified frames, which compose the low-level audio feature vectors. In the sequel, the term *audio frame* refers to the shortest in duration audio part used in short-time audio analysis, whereas the term *segment* refers to a group of a variable number of successive frames pre-classified as speech frames.

Initially, *silence detection* is performed to discard silence periods from further analysis. Silence frames are audio frames of background noise with a relatively low energy level and high zero crossing rate compared to voice segments. In order to distinguish speech from silence, the average magnitude M_t and zero crossing rate Z_t functions are exploited [13]. Audio frames of 10msec duration are employed. A convenient approach to robust speech-silence discrimination is end point detection [13], which determines the beginning and end of words, phrases or sentences so that subsequent processing is applied only on these. Average magnitude and zero crossing rate thresholds are chosen relative to background noise characteristics of an apriori known audio interval, its average magnitude and zero crossing rate functions being $M_{t,n}$ and $Z_{t,n}$ respectively. The average magnitude thresholds used by endpoint detection are set equal to: $M_{thr,up} = \frac{E[M_t]-M_{thr,low}}{2}$ and $M_{thr,low} = \max(M_{t,n})$. The zero crossing rate threshold is set equal to: $Z_{thr} = \max(Z_{t,n})$. Such threshold selection proves robust and endpoint detection is correctly performed. Boundaries of words, phrases or entire sentences are well estimated, a useful outcome subsequently exploited in the speaker recognition phase. At this stage, a first level of audio abstraction is achieved, namely the time-dependent speech-silence mapping.

Low-level feature vectors are evaluated on voiced classified frames only [13]. *Voiced-unvoiced discrimination* is based on the fact that unvoiced sounds exhibit significant high frequency content in contrast to voiced ones. Thus, the energy distribution of the frame signal is evaluated in the lower and upper frequency bands (the boundary is set at 2kHz with a sampling rate of 22kHz). High to low energy ratio values greater than 0.25 imply unvoiced sounds, which are ignored in subsequent processing.

For *audio feature extraction* in voiced frames, the speech signal is initially pre-emphasized. Speech frames of 30msec duration each are extracted with an overlap of 20msec from each other. Each frame is windowed by a Hamming window of size M. LP analysis follows and the LP parameter set α_m, $m = 1,..,p$, (p: order of the LP model), is later on converted to the LP cepstral coefficients c_m, $m = 1,..,p$ according to $c_m = \alpha_m + \sum_{k=1}^{m-1}(\frac{k}{m})c_k a_{m-k}$, $1 \leq m \leq p$.

2.2. Speaker Classification

After the generation of the sequence of feature vectors on voiced frames, speaker modelling and recognition succeeds aiming at extracting a time-constrained speaker identity mapping.

Speaker modelling is performed by an LVQ3 classifier [14]. All speaker codebooks are trained in parallel using training vectors from apriori known time periods in which a certain speaker speaks. Misclassification cases are compensated for. Initialization of the codebooks, each composed by 32 codevectors, is performed by the LBG algorithm. Classification of a training vector $\mathbf{c}$ to a speaker codebook $\mathbf{m}_i$, $i = 1,..,N_{speakers}$ 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 speaker codebook $\mathbf{m}_i$ and $\mathbf{R}_i$ denotes the global covariance matrix estimated from the training vectors of $\mathbf{m}_i$. The codevectors of every class that were assigned the minimum codevector distances are selected. Finally, $\mathbf{c}$ is classified to $\mathbf{m}_{ik}$ of the speaker codebook $\mathbf{m}_i$, which produces the minimal $d(\mathbf{m}_{ik}, \mathbf{c})$.

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 the codevector $\mathbf{m}_{jk}$ of the speaker codebook $\mathbf{m}_j$, is classified at iteration t to the codevector $\mathbf{m}_{ik}$ of the speaker codebook $\mathbf{m}_i$ and $\mathbf{c}$ lies in a predefined "window", then updating of the respective codevectors of both speaker classes at iteration $t+1$ is performed in the following way:

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

A training vector $\mathbf{c}$ is assumed to lie in the "window" if $\frac{s_j - s_i}{s_j} > w$ [14], 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 codevector of the codebook $\mathbf{m}_j$ that it belongs to and from the codevector of the codebook $\mathbf{m}_i$ that it has been classified to respectively. In the case that $\mathbf{c}$ is classified correctly to $\mathbf{m}_{jk}$ of the speaker codebook $\mathbf{m}_j$ then updating at iteration $t+1$ is performed by:

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

It is noted that updating is performed only to the appropriate codevector $\mathbf{m}_{jk}$. $\alpha(t)$ is a linearly decreasing function depending on the number of iterations which controls the learning rate. Constant ϵ is usually set equal 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 *speaker recognition* process, the minimal codevector distance $s_i(\mathbf{c}_t)$ between an input test vector $\mathbf{c}_t$ and the codevectors $\mathbf{m}_{ik}$ of the speaker codebook $\mathbf{m}_i$ is calculated by $s_i(\mathbf{c}_t) = \min_{\mathbf{m}_{ik} \in \mathbf{m}_i}[d(\mathbf{m}_{ik}, \mathbf{c}_t)]$, $k = 1,..,32$. Index i spans the number of speaker classes. $d(\mathbf{m}_{ik}, \mathbf{c}_t)$ is evaluated by (1). The global covariance matrix $\mathbf{R}_i$ has been evaluated during the training process.

In our approach, instead of assigning every input test vector separately to a speaker class, groups of input test vectors corresponding to entire speech segments composed of a variable number of successive speech frames are all together classified. Such an approach is adopted based on the fact that only one speaker is bound to be talking between two silence segments. Thus, the average distortion of the entire speech segment is evaluated by $S_i = \frac{1}{T}\sum_{t=1}^{T} s_i(\mathbf{c}_t)$, where T is the number of speech frames included in the current speech segment and $s_i(\mathbf{c}_t)$ are defined above. The unknown speaker is identified as the reference speaker whose model gives the minimum average distortion:

$$SpeakerID = \arg\min_{1 \leq i \leq N_{Speakers}} S_i \quad (4)$$

Finally, the time interval that corresponds to the speech segment determined beforehand is assigned to the identified speaker. Thus, *a speaker label mapping over time* is made possible thus offering a second level of audio abstraction. Information about the presence of a speaker in certain parts of the video source can thus be indexed and retrieved. Speaker alternation can also be monitored and retrieved.

3. Visual Source Parsing

The visual analysis module involves tasks such as shot boundary extraction, face detection, facial feature extraction and mouth region tracking. In the sequel, the term *shot* denotes a sequence of successive frames that corresponds to a single camera start and end session. The term *scene* refers to one or more successive shots combined together due to exhibiting the same semantically meaningful concept, e.g. a scene that addresses the same topic although many shots may be involved. Scene characterization is different when different content is searched for.

Shot boundary detection is initially performed. A color frame difference metric is estimated and appropriately thresholded in such a way that camera motion or simple edit effects like zoom-ins/outs are not mistaken for possible scene cuts. Color differences, $FD(t)$, are defined by:

$$FD(t) = \frac{1}{3X_{dim} \times Y_{dim}} \sum_{\mathbf{x}} ||\mathbf{I}(\mathbf{x};t) - \mathbf{I}(\mathbf{x};t-1)||_1 \quad (5)$$

where $\mathbf{I}(\mathbf{x};t) = [I_r(\mathbf{x};t) I_g(\mathbf{x};t) I_b(\mathbf{x};t)]^T$ represents the vector-valued pixel intensity function composed of the three color components: $I_r(\mathbf{x};t)$, $I_g(\mathbf{x};t)$ and $I_b(\mathbf{x};t)$. By $||.||_1$ the L_1-vector norm metric is denoted. $\mathbf{x} = (x, y)$ spans the spatial dimension of the sequence (each frame is of size $X_{dim} \times Y_{dim}$) whereas t its temporal one. Frame differencing is computationally intensive but no limitations on processing time are imposed when the task is performed offline. In order to detect possible shot breaks, local color frame difference mean values on a $1d$ temporal window W of size 3 are obtained ($FD_{m,t\in W} = E[FD(t)|_{t\in W}]$). Then, the current value of $FD(t)$ is examined to specify whether it is the maximum value of those inside the temporal window (possibility of a scene break at t). If this is the case and $FD(t)/FD_{m,t\in W} > 2$, a scene cut is declared at t. Such window-based thresholding offers the means of adaptive thresholding according to local content and proves flexible and efficient in gradual camera movements and simple edit effects as zoom-ins and outs (no false positives, over-segmentation is avoided). Abrupt changes are directly recognised.

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 can 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, HSV color segmentation is performed to isolate skin-colored regions. The employed thresholds are: $0^0 \leq H \leq 25^0$, $335^0 \leq H \leq 360^0$, $0.2 \leq S \leq 0.6$ and $V \geq 0.4$. This hue range corresponds to reddish colors. The saturation range ensures exclusion of pure red or very dark red colors. Both account for small variations in lighting conditions. Value 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. It is performed at a coarse frame resolution to create an initial map of all pixels with skin-like colors. Refinement of this map is successively attempted in order to eliminate spurious pixels.

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 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 onwards. The algorithm is performed iteratively until no more pixels connected to the current object are detected. The process is computationally efficient and requires only a few iterations.

Afterwards, *a decision is made on whether one of them represents a face*. The decision is based on the size of the object with respect to the size of the frame (very small objects, even faces, are of no significance) and (mainly) on the ratio of its width to height which must be within a certain range of the facial x/y ratio. After a face object is detected, all other objects are discarded from the initial map, leaving only the face region. This final map is called *face region map*.

Coarse facial feature extraction follows by detecting only the eyes, mouth and chin through minima analysis of the vertical face profile estimated on the face region map only. 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 considering facial biometric analogies. Once a rough estimate of the y-position of the eyes is derived, *finer estimation of eye locations* is attempted by a weighted block matching approach based on facial mirror-symmetry principles in frontal or semi-frontal views. Specifically, initial blocks of size $b_x \times b_y$ relevant to the face object size are centered around the initial coarse eye locations (l_x, l_y), (r_x, r_y), respectively. Search regions of size $(2s_x + 1) \times (2s_y + 1)$ are subsequently defined around the initial block locations. Minimization of the weighted block matching error, $E(dx, dy)$, follows:

$$E(dx,dy) = \sum_{x=0}^{b_x}\sum_{y=0}^{b_y}\sum_{i}[|W(x,y)I_i(l_x - \frac{b_x}{2} + x + dx, l_y - \frac{b_y}{2} + y + dy)$$

$$+W(b_x - x, y)I_i(r_x + \frac{b_x}{2} - x - dx, r_y - \frac{b_y}{2} + y + dy)|]$$

$i \in \{r, g, b\}$, $dx \in [-s_x, s_x]$, $d_y \in [-s_y, s_y]$. The $2d$ weighting function $W(x, y)$ is defined by $W(x,y) = \frac{\sin((x^2+y^2)/c)}{(x^2+y^2)/c}$ attempting to emphasize the center regions (iris), and inverse weight the surrounding regions, thus leading to a simple model of the eye structure. c is set proportional to the eye block width. Accurate estimation of eye locations leads to precise evaluation of the head rotation angle by calculating the angle between the eye block centers, i.e. $\theta = \arctan(D_y/D_x)$, where D_x and D_y are the pixel distances between the eye block centers in the x- and y-directions, respectively.

Head rotation compensation and re-estimation of the vertical face profile follows in order to obtain a better estimation of the exact location of the mouth area. Eye positions, face center and face map are known in the new coordinate system (rotated accordingly). Minima analysis of the vertical face profile is again employed in order to estimate the mouth center y-location. The mouth horizontal profile is then extracted in order to locate the mouth edges and thus be able to initialize a rectangular mouth template with respect to the mouth width. All feature locations are subsequently rotated back to the initial coordinate system. Thus exact location of the mouth region is estimated and a rectangular mouth template is extracted to be used for mouth template tracking along the remaining frames of the face shot. Obviously, head rotation compensation enhances the accuracy of facial feature extraction.

A mouth template tracking process follows aiming at extracting information on the presence of a talking face in the current shot. Mouth locations in successive frames are estimated by minimizing L_1 norms between the current frame and the reference template. Search regions are appropriately initialized to account for motion of the head besides the mouth movement. After mouth regions have been tracked throughout the entire face shot, mean square errors between successive tracked mouth regions are evaluated to further detect mouth movements. Such movement implies a talking face. Large error variance denotes mouth movement. Obviously, errors in mouth region estimation would result in errors concerning the decisions for talking or non-talking faces. The output of the visual analysis module is a time-dependent mapping function $M(t)$ containing information about detected talking or non-talking face shots.

4. Audio-Visual Interaction

Multimodal interaction can serve two purposes: (a) enhance the "content findings" of one source by using similar content knowledge extracted from the other source(s), (b) offer a more detailed content description about the same video instances by combining the content descriptors (semantic labels) of all data sources based on interaction rules and coincidence concepts.

The refinement process in our case involves estimation of speaker presence likelihood in every face shot. Speaker recognition performed on the audio source is prone to recognition errors which can be limited by utilizing the related visual information, e.g. presence of talking people, as described subsequently. The presence likelihood of speaker SP_i in a face shot FS_k is evaluated by:

$$P(SP_i|FS_k) = \frac{M_i}{L_k} \tag{6}$$

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. Refinement of the time dependent speaker identity indexing function is performed only in face shots with talking faces which contain respective content (speaker - talking face).

The interaction of audio and visual semantic labels (speech, silence, speaker identity, face presence, face absence, talking face presence) supplies the user with more detailed information and a degree of the underlying context, e.g. person X talking and being present in scene Y (characterized by its duration) or a news story is shown in scene Z (characterized by its duration) while reporter W is telling the story without being present in the scene. Separate audio and visual mappings containing different content information can be combined in different ways based on such interaction rules. Another example is the combination of successive shots into a single story unit by observing that the same speaker speaks throughout the duration of those shots. Furthermore, dialogue situations can be detected by setting appropriate interaction rules, such as the frequent alternation of the same speakers.

5. Simulation Results

The audio-visual content analysis scheme has been tested on many news sequences. Two of them are shown as examples in this paper, of approximately 6min and 4min duration respectively. The story units from the second news sequence have been cut out leaving only the anchorperson units (two reporters appear alternatively many times) in order to evaluate the performance of the face localization and mouth tracking methods involved in the visual parsing process. In the first news sequence, five different speakers exist but the faces of only three of them appear (two reporters in a studio on a chroma-keyed background and one reporter outdoors). Two of the speakers appear in different parts of the sequence. The video sequences have been digitized with a frame rate of 25fps at PAL spatial 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.

Speech-silence discrimination exhibits a remarkable performance achieving to locate entire phrases or sentences. Rare occasions of unvoiced speech frames being classified as silence frames have only been observed. There was no case of background noise being classified as speech. For the speaker classification task, the novel recognition method evaluating segment-based distances leads to remarkable recognition results for both news sequences exhibiting only 0.19% and 0.068% recognition error rates respectively. Such good recognition performance is also reflected to the speaker identity mapping function (from the audio source only) shown in Figure 1(b). For simplicity, speakers are labelled with numbers in the figure. Speaker identity string labels are actually stored and later on retrieved. For the first news sequence, the only failure of the method was in being unable to handle an unvoiced part of speech because it was surrounded by silence segments. This is a case where silence detection errors lead to propagation of errors to further classifications.

In Figure 1(b) the frame color difference function of the first news sequence can be viewed as well as the shot boundary mapping function. Abrupt changes are directly recognised as shot breaks. In Figure 1(a) the first frame of each detected shot of the first news sequence is illustrated. Face detection results can be viewed in Figure 2. The face region is correctly detected in almost all cases. There is only a sole instance of failure due to bad lightning conditions. Normalization methods to deal with this problem could be incorporated in the method. Facial feature extraction and mouth region estimation results are illustrated in the same figure. Eye locations are very well estimated, head rotation angles correctly evaluated and mouth region templates correctly determined. As far as mouth template tracking is concerned, some examples are given in Figure 3. Although the approach does not account for head rotation, tracking is well performed and decisions based on talking faces are correctly made as can be seen by the talking face shot mapping function illustrated in Figure 1(b).

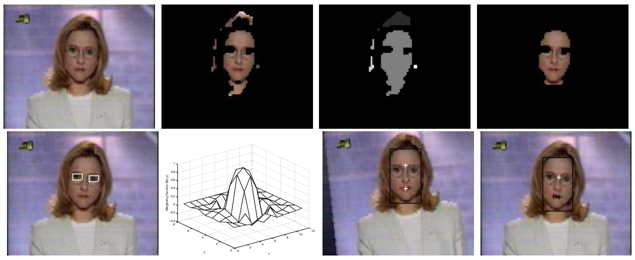


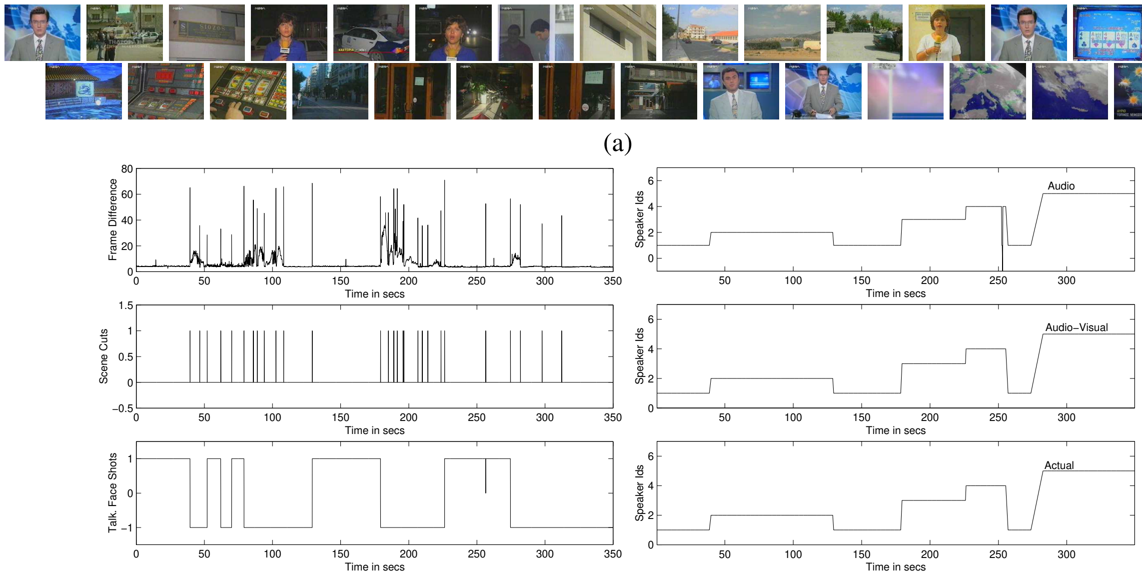
Figure 2. Original frame of face shot. After (top-left to bottom-right): skin color segmentation, skin-colored object detection, face object extraction, eye localization, weighting function for eye location estimation, head rotation compensation, mouth region extraction.

Figure 3. Mouth template tracking.

The integration of the mapping functions based on audio-visual interaction remains to be evaluated. As it can be seen in Figure 1(b), refinement of the speaker label mapping function extracted from the audio source by exploiting concurrent relative visual content in the way discussed in section 4 leads to elimination of the existent recognition errors and convergence to the actual mapping function.

6. Conclusions

Content analysis and indexing systems offer a flexible and efficient tool for further video retrieval and browsing. When such tasks combine semantic information from different data sources (auditory, visual, textual) through multimodal interaction concepts multi-level video abstraction is possible. Besides, due to the temporal nature of video sources, time dependent representations and indices should also be a supplement to other kinds of indexing. The presented audio-visual content parsing method aims at extracting semantically meaningful video scenes and assign semantic label(s) to them with respect to their content. Semantic labels employed are: speech, silence, speaker identities, no face shot, face shot, talking face shot. The coincidence or inter-relations of semantic labels can be captured by audio-visual interaction tasks based on specific rules.



(a) (b)

Figure 1. (a) First frames of detected shots, (b) Time-dependent mapping functions: color frame differences, shot boundaries, talking face shots, speaker labels from audio and after audio-visual refinement.

User interactivity to the process enables higher content description levels.

References

[1] B. Furht, *Multimedia Tools and Applications*, Kluwer Academic Publishers, 1996.

[2] P. Correia and F. Pereira, "The role of analysis in content-based video coding and indexing", *Signal Processing, Elsevier*, vol. 66, no. 2, pp. 125–142, 1998.

[3] J. Nam, M. Alghoniemy, and A. Tewfik, "Audio-visual content-based violent scene characterization", in *Proc. of ICIP'98*, 1998, vol. 1, pp. 353–357.

[4] C. Saraceno and R. Leonardi, "Audio as a support to scene change detection and characterization of video sequences", in *Proc. of ICASSP'97*, 1997, vol. IV, pp. 2597–2600.

[5] Y. Nakamura and T. Kanade, "Semantic analysis for video contents extraction - spotting by association in news video", in *Proc. of 1997 ACM Int. Conf. on Multimedia*, 1997, pp. 393–401.

[6] M.R. Naphade et al., "Probabilistic multimedia objects (multijects): A novel approach to video indexing and retrieval in multimedia systems", in *Proc. of ICIP'98*, 1998, vol. 3, pp. 536–540.

[7] T-S. Chua and L-Q. Ruan, "A video retrieval and sequencing system", *ACM Trans. on Information Systems*, vol. 13, no. 4, pp. 373–407, 1995.

[8] A.M. Ferman, A.M. Tekalp, and R. Mehrotra, "Effective content representation for video", in *Proc. of ICIP'98*, 1998, vol. 3, pp. 521–525.

[9] S.W. Smoliar and H. Zhang, "Content based video indexing and retrieval", *IEEE Multimedia Magazine*, vol. 1, no. 2, pp. 62–72, 1994.

[10] M. Irani and P. Anandan, "Video indexing based on mosaic representations", *Proceedings of the IEEE*, vol. 86, no. 5, pp. 905–921, May 1998.

[11] M.M. Yeung and B-L. Yeo, "Video visualization for compact presentation and fast browsing of pictorial content", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 5, pp. 771–785, Oct. 1997.

[12] B. Guensel and A.M. Tekalp, "Content-based video abstraction", in *Proc. of ICIP'98*, 1998, vol. 3, pp. 128–132.

[13] L. Rabiner and R.W. Schafer, *Digital Processing of Speech Signals*, Englewood Cliffs, N.J.: Prentice Hall, 1978.

[14] T. Kohonen, "The self-organizing map", *Proc. of the IEEE*, vol. 78, no. 9, pp. 1464–1480, September 1990.