VIDEO AND AUDIO BASED DETECTION OF FILLED HESITATION PAUSES IN CLASSROOM LECTURES

Vassilis Tsiaras, Costas Panagiotakis and Yannis Stylianou

Department of Computer Science, University of Crete,
P.O. Box 2208, Heraklion, Crete, GR-71409 Greece
{tsiaras, cpanag, yannis}@csd.uoc.gr

ABSTRACT

In this paper we study the detection of hesitation filled pauses in oral presentations of university lectures taught in the Greek language and recorded using a tablet PC via a specialized software. We suggest a hierarchical approach fusing video data with audio data for increasing the precision rate in our detection system. The detection method works at frame level rather than the usual segmental level for more accurate synchronization of audio and video data after removing the detected hesitations. Audio characteristics are modeled using Gaussian Mixture Models while the stationarity of the recorded video is taken into account. This efficient video and audio combination yields higher precision and recall rates comparing with other works in the literature. On a dataset of approximately 7 hours the precision rate is 99.6% while the recall rate is 84.7% when audio and video data are taken into account.

1. INTRODUCTION

Spontaneous speech contains disfluencies. These are generally defined as "phenomena that interrupt the flow of speech and do not add propositional content to an utterance" [1]. They include pauses, interruptions, repeated words and phrases, restarted sentences, words with elongated pronunciations, and filled pauses. In this paper we concentrate our attention on filled pauses, which are meaningless vocalizations that are inserted into speech when a speaker is thinking about the speech contents on the fly. They are usually found in three distinct forms: i) an elongated central vowel only, such as "ee" or "aa"; ii) a nasal murmur only, such as "mm"; iii) a central vowel followed by a nasal murmur, such as "eem" [2]. Pauses play valuable roles in oral communications, such as helping a speaker hold a conversational turn and express mental and thinking states. On the other hand, filled pauses are often considered undesirable, unnecessary and annoying to listeners. For this reason people take courses to learn how to avoid saying them and they are spliced out, usually manually, from recorded interviews, public speeches, university lectures etc.

Filled pause detection systems have mainly been suggested for improving the speech recognition accuracy and thus, rendering easier and more robust the human-machine communication. This is because current speech recognition systems fail to process efficiently spontaneous speech where the filled pauses effect is quite frequent. Automatic identification methods of filled pauses rely on *intramedia* fusion of parameters such as speech spectra, fundamental frequency and duration [3, 4, 5]. The duration of a phonetic event had to be 120 ms or more to be considered as a filled pause. When a filled pause contains an elongated vowel the tension of the vocal cords and the vocal-tract shape are unvaried, and consequently the fundamental frequency and the spectral envelope remain almost constant [3]. The existing filled pauses detection methods achieve high precision and recall rates. Goto et al. [3] reported 91.5% precision and 84.9% recall rates and Li et al. [6] reported 80.66% precision and 92.59% recall rates.

All the above approaches were only based on speech-related features in detecting filled pauses. However, there are many applications where data from more than one media are available like from audio and video and where there is an opportunity for *intermedia* fusion (or multimedia processing) for increasing the detection rate. Moreover, the task may not necessarily be speech recognition but rather a kind of audio enhancement by automatically detecting and deleting (filtering) the undesired disfluency events. In this paper we focus on multimedia data from university lectures recorded in a tablet PC during a Speech Signal Processing course. The recordings were made using the Camtasia Studio software [1] where there are options, among others, for enhancing a recording by removing the ambient noise and equalize the loudness of the recording. However, it is not possible to detect and remove specific sound events like the filled pauses. Therefore, such a task should be performed manually which is quite time-consuming. In this paper, we suggest a system for automatically detecting the filled pauses in audio-visual data. Our approach takes advantage of the fact that two modalities, speech and video, are available suggesting a multimedia approach. We make two hypotheses. First, when a speaker is uttering a filled pause, he/she is in a thinking state and he/she does not write on the tablet PC. We recall here that by filled pause we mean the time of meaningless vocalizations as those mentioned before, and not the pauses (with silence) which define the rhythm in a presentation and to some extent characterize the speaking style of the lecturer. Such pauses should remain untouched. Second, when a segment of length ≥ 140 ms is erroneously identified as a filled pause then the segment contains at least one consonant. For this reason we model not only the filled pauses but also some of the Greek language consonants. We suggest a hierarchical approach; we first detect stationary video frames and then only on these frames we perform detection of filled pauses using the audio. This approach improves the precision score of our detector while speeding up the detection process. Our method is able to detect elongated vowels inside a word as well as filled pauses that start and/or end with silence segments. The suggested system is completely trainable, which means that it can be adapted to the needs

[1] http://www.techsmith.com/camtasia.asp

of a lecturer; the lecturer provides to the suggested system a training set of filled pauses which are considered undesirable. For example, in this paper, the undesirable filled pauses were defined by the 3rd author of the paper who was the lecturer.

The paper is organized as follows. In Section 2 the suggested detection system based on audio and visual features is described. In Section 3 we present the dataset where the suggested system was evaluated while in Section 4 the detection results are presented. Future work and conclusions are provided in the last section of the paper.

2. AUDIO AND VIDEO ANALYSIS

2.1 Audio Features

An example of a typical filled pause is depicted in Fig. 1. During a hesitation filled pause, the fundamental frequency, F0, declines very slowly and very smoothly and this is an observation that is used in most of filled pauses detectors [3, 4].

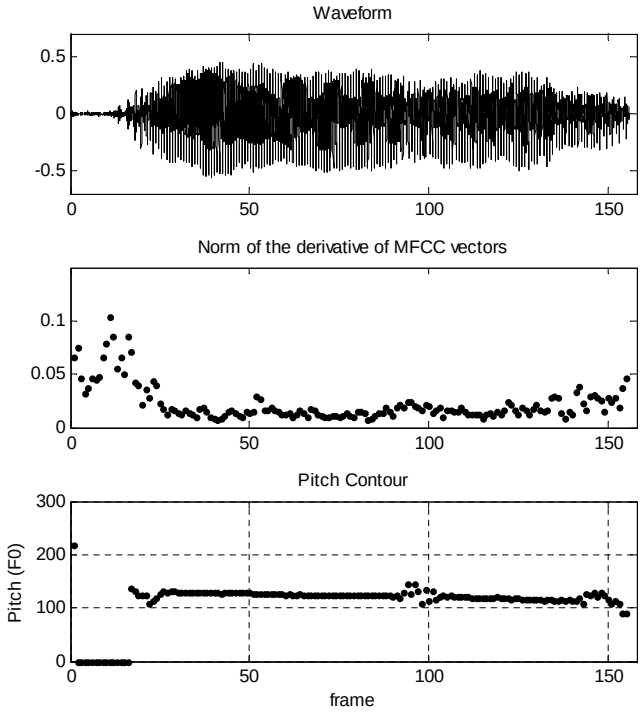

Figure 1: Waveform (upper panel), norm of the derivative of MFCC vectors (middle panel) and pitch contour (lower panel) of a typical "ee" hesitation filled pause.

We also make use of F0 and more specifically the first derivative (velocity) of F0 is calculated using regression analysis, as [7]:

$$\partial \mathrm{F0}_t = \frac{\sum_{k=-K}^{K} k \cdot \mathrm{F0}_{t+k}}{\sum_{k=-K}^{K} k^2} \quad (1)$$

where $\partial \mathrm{F0}_t$ is the first derivative of F0 at time frame t, and $\mathrm{F0}_{t+k}$ is the fundamental frequency at time frame $t+k$. Hanson at al. [8] recommend a width of $K=2$. For the estimation of fundamental frequency we use a standard autocorrelation based method [9].

We also consider the speech spectral envelope represented by Mel-Frequency Cepstrum Coefficients (MFCC). For each frame, we calculate a vector c of 13 MFCC using the Slaney's Auditoty Toolbox [10]. The 0th cepstral coefficient $c[0]$ is not used in the feature vector. The first derivative, ∂c, of vector c is computed in the same way as the first derivative of F0 by substituting in Eq. (1) c for F0. Then the norm of ∂c is used as feature. As it is indicated by the middle panel of Fig. 1 the norm of ∂c provides evidence for detecting the filled pause shown in the figure. To sum up the feature vector that we use is $[c[1:12]^T,\ \partial \mathrm{F0},\ |\partial c|]^T$. The feature vector for consonants consists of 12 MFCC $c[1:12]$.

2.2 Audio Model Description

To model the acoustic space of the filled pauses we use Gaussian Mixture model (GMM). A Gaussian mixture density is a weighted sum of M component densities as given by:

$$p(x|\Theta) = \sum_{i=1}^{M} \alpha_i p_i(x|\theta_i) \quad (2)$$

where x is a D-dimensional random vector, the parameters are $\Theta = (\alpha_1, \ldots, \alpha_M, \theta_1, \ldots, \theta_M)$ such that $\sum_{i=1}^{M} \alpha_i = 1$ and $\theta_i = (\mu_i, \Sigma_i)$. Each component density is a D-variate Gaussian function

$$p_i(x|\theta_i) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)} \quad (3)$$

with mean vector μ_i and covariance matrix Σ_i. In our experiments we use diagonal covariance matrices while the estimation of the GMM parameters is obtained by using a standard Expectation-Maximization (EM) algorithm [11]. Two critical factors in estimating a GMM are selecting the order M of the mixture and initializing the model parameters prior to the EM algorithm. In our experiments we set $M=25$ for "ee" and "eem" trying to model the various recording conditions (i.e., room acoustics, distance from the microphone, and angle of recording) while for each of the 10 consonants we set $M=5$. All the recordings were made using the same tablet PC and always the microphone of the PC was used. For the initialization, the training data of each GMM was clustered, using the BIRCH algorithm [12], into M classes. The class means and variances then served as the initial model for EM training. BIRCH clustering algorithm runs faster than LBG and K-means algorithms since it performs one scan of the data and according to our experiments its results are comparable to those of LBG and K-means in terms of quality and superior in terms of stability.

2.3 Video Analysis

The goal of video analysis is to determine the frame images where the video has been changed in content which mainly occurs when the lecturer writes on its tablet PC. In our case, the camera is static and the video shows the contents of the screen of the tablet PC of the lecturer (see examples in Fig. 2). By analysis of the audio-visual content, it holds that a hesitation filled pause appears when the lecturer does not write on the tablet PC. Therefore, the estimation of frames

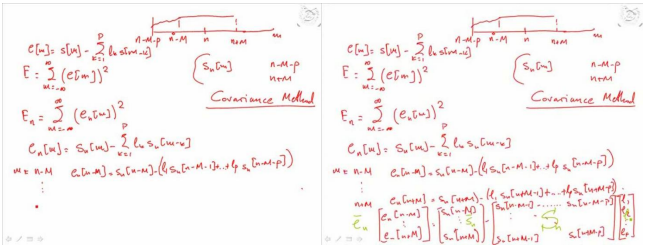

Figure 2: Two examples of frames from video data.

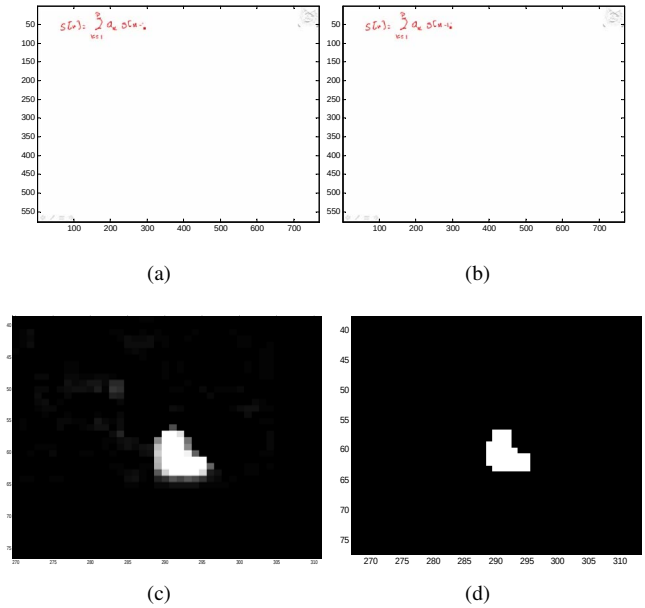

Figure 3: **(a)(b)** Two sequential frames from video data ($I(t)$, $I(t+1)$), **(c)** the single frame-difference $D(t)$ and **(d)** the estimated bitmap image $B_1(t)$.

where the lecturer writes on the tablet PC will automatically exclude all these frames from being in the set of frames containing filled pauses. The video analysis method is described hereafter.

First, the absolute value of single frame-difference image ($D(t)$) is obtained by subtracting the current frame image ($I(t+1)$) from the previous-frame image ($I(t)$),

$$D(t) = |I(t+1) - I(t)|, \tag{4}$$

where t denotes the frame number. Single frame differences encode the visual content changes. Especially when the camera is static, frame-differences have been used successively on background subtraction techniques [13], object tracking [14, 15] and on video representation [16]. Next, we estimate the maximum value, $m(t)$ of image $D(t)$.

A binary image $B_0(t)$ is estimated by thresholding $D(t)$. If a pixel intensity change is higher than a threshold (e.g., 40) then it is classified as changed (white color), otherwise it is classified as unchanged (black color). Morphological operations (erosion and dilation) [17] are applied on $B_0(t)$ in order to reduce noise due to coding effects (e.g., salt and pepper noise) resulting $B_1(t)$ (see Fig. 3). When there is a content change, the lecturer writes at a specific point on the table, meaning that the change is a compact region which is not affected by the procedures of erosion and dilation. Then, the number of pixels, $n(t)$, that change at frame t can be robustly given by the sum of white pixels of $B_1(t)$. In order to increase the robustness of the method, $n(t)$ can be computed by the sum of pixels inside the largest compact white region (region of the maximum area). This is useful when the video has been recording in very low quality. Highest area object estimation has been successfully used on human tracking under bad quality athletics video captured from TV [18]. Fig. 3 illustrates an example of $B_1(t)$ estimation. The last two images of Fig. 3 are cropped close to the white region, showing in high detail the region of interest. The remaining pixels of these images are black.

Finally, an empirical probability $p(t)$ of visual change at frame t is estimated using the proposed features $n(t)$ and $m(t)$:

$$p(t) = (1 - e^{-\frac{m^2(t)}{\sigma_m^2}}) \cdot (1 - e^{-\frac{n^2(t)}{\sigma_n^2}}) \tag{5}$$

where it is assumed that $m(t)$ and $n(t)$ are independent random variables. The parameters σ_m and σ_n were estimated to be 40 and 15, respectively, by statistical analysis of our data. The left picture in Fig. 4 illustrates $n(t)$, $m(t)$ and $p(t)$ of a 28 seconds video. Note that $m(t)$ contains small spikes (mostly at the end of the sequence) while $n(t)$ is more smooth. The spike character of $m(t)$ is caused by video coding.

After calculating $p(t)$, a threshold of 0.5 was used on $p(t)$ to detect the frame with change ("writing activity"). Using Eq. (5) the areas with no writing activity are easily detected as it is shown for this example in the right picture in Fig. 4. In our dataset, we found that the combination of $n(t)$ and $m(t)$ features results in 99.7% detection of video content changing with 0.5% of false alarm.

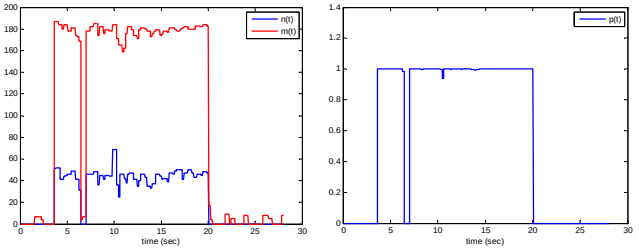

Figure 4: Results of the video analysis method. Left, the estimated number of pixels, $n(t)$, and the maximum value, $m(t)$, of the image. Right, the probability of content change, $p(t)$.

2.4 Frame Classification

The classification procedure is hierarchical giving priority to the video data. All the video frames where content change was detected are excluded by the detection algorithm since no hesitations are expected there. An audio frame of length 20ms is considered with a frame rate of 10ms providing 100 frames per second. The video frame rate is 20 frames per second. The ratio between the video and audio frame rate is used to synchronize the information into these two media

streams. The rest of the classification process only uses the audio information. The hesitation filled pause detection task can be stated as a basic hypothesis test between two hypotheses:
H0: (audio) frame with features x belongs to a filled pause
H1: frame with features x does not belong to a filled pause
We decide between these two hypotheses using a likelihood ratio test given by:

$$\frac{p(x|H0)}{p(x|H1)} \begin{cases} > 1 & \text{accept H0} \\ \leq 1 & \text{accept H1} \end{cases}$$

Likelihood $p(x|H0)$ is evaluated for feature vector $x = [c[1:12]^T,\ \partial\text{F0},\ |\partial c|]^T$.

$$p(x|H0) = \sum_{i=1}^{M} \alpha_i p(x|\theta_{ee,i}) \quad (6)$$

For likelihood $p(x|H1)$ we compute the likelihood $p(x|\theta_{cons_j})$ for $j = 1,\ldots,10$ for feature vector $x = c[1:12]$. Then we set

$$p(x|H1) = max(p(x|\theta_{cons_1}),\ldots,p(x|\theta_{cons_{10}}),p_{other}), \quad (7)$$

where p_{other} is a small numerical value (e.g., 0.0005) that accounts for other sounds (i.e., non modeled consonants and vowels).
If $p(x|H0) > p(x|H1)$ the frame is labeled as filled pause candidate and then we continue with the next frame by repeating the above procedure. When a maximal sequence of adjacent frames, labeled as filled pause candidates, has length of at least 14 frames (140 ms) then this sequence is identified as an "ee" filled pause segment. Next we try to extend this sequence as far as possible to the right by appending frames that may correspond to nasal murmur "mm". For this, we check whether the next right neighboring frame is an "eem" and we append this frame to the filled pause segment if $p(x|\theta_{eem}) > p_{eem}$, where $x = c[1:12]$ and p_{eem} is a small numerical value. We repeat the extension procedure until we find a frame with $p(x|\theta_{eem}) \leq p_{eem}$.

3. DATA

Our analysis has been applied to data from recorded lectures of a course entitled "Speech Signal Processing", which was taught in the Greek language at the Computer Science Department of University of Crete during the Fall term of academic year 2007-2008. The course was recorded using a tablet PC. Each recording sessions has a variable duration, ranging from 60 to 90 minutes. All the recorded classes were taught by the same teacher, a male speaker. The teacher's speech overlaps with the "tics" from the tablet PC pen. Apart from the teacher's voice the recordings contain the voices of students who ask or answer questions, and background noise. About 5% of the speech segments were saturated. Long silence epochs and sections of the recordings that have no educational value were manually removed. Hesitation filled pauses, especially long "ee" and "eem", appear too often to be removed manually.

We selected 10 sound files from the recorded lectures for training and testing. The total duration of files was 6 hours and 51 minutes. For training, the first 30 minutes of the 5 first files were transcribed for filled pauses and for the 10 selected consonants of the Greek language. The rest of data were transcribed only for filled pauses and they were used for testing. In total, 1124 filled pauses were annotated with 399 of them being in the training data set. Labeling was done on the basis of listening and visual inspection of the waveform, spectrogram, F0 and intensity contours, using WaveSurfer[2].

Methods	Precision	Recall
Proposed (Audio)	98.5%	80.6%
Proposed (Video and Audio)	99.6%	84.7%
Goto et al. [3] (Audio)	91.5%	84.9%
Li et al. [6] (Audio)	80.66%	92.59%

Table 1: Precision and recall rates using only audio and both video and audio streams and comparizons with other methods.

4. RESULTS

Our main hypothesis that the lecturer does not write on tablet PC during a filled pause event is supported by the statistical data since only 5 of the 1124 annotated filled pause events ($\sim 0.45\%$) overlap with writing on tablet PC. Additionally we tested this hypothesis on two randomly chosen lectures (different from the 10 selected lectures) and we found that only one event out of more that 300 filled pause events overlaps with writing.

Using only the audio stream, the suggested detector has precision rate 98.5% and recall rate of 80.6%. If both video and audio streams are included into the detector the precision rate is 99.6% and the recall rate is 84.7%. Goto et al. [3] reported 91.5% precision and 84.9% recall rates and Li et al. [6] reported 80.66% precision and 92.59% recall rates using only audio stream. Therefore, the proposed method outperforms the other works in the literature (see Table 1), due to the efficient combination of video and audio features.

The above measurements concern not only isolated filled pauses that are characterized by silence segments before and/or after the filled pause but also elongated "ee" that are inside words (see Fig. 5). If in the statistics we consider only very long filled pauses (of duration greater than 230ms), which are the ones that are considered to be unpleasant to listeners, then the recall rate is 99.3%. The missing filled pauses are either within saturated areas or they have very low energy compared to the mean energy of filled pauses.

5. CONCLUSION

We suggested a method for the detection of hesitation filled pauses "ee" and "eem" in oral presentations using multimedia (audio and visual) data. The removal of hesitation filled pauses is a useful task, since they are annoying to listeners. By analysis of the audio-visual content, it holds that a hesitation filled pause appears when the lecturer does not write on the tablet PC. Therefore, the combination of audio-visual information increases both the precision and the recall score, yielding higher precision and recall rates comparing with other works in the literature. Audio characteristics are modeled using Gaussian Mixture Models while the stationarity of the recorded video is taken into account based on single frame-difference images.

[2] http://www.speech.kth.se/wavesurfer/

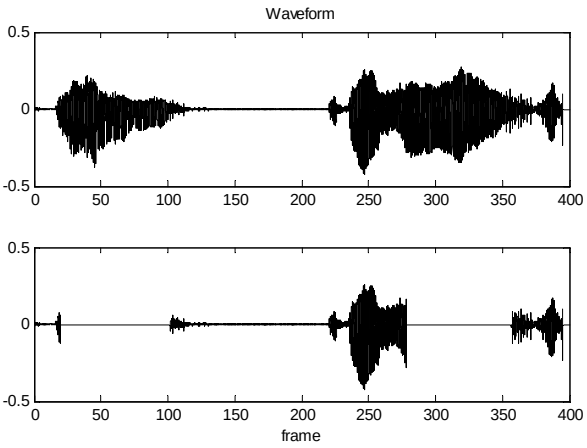

Figure 5: Waveform of a speech signal before and after the removal of detected filled pause segments. In filled pause segment from frame 20 to frame 120 a few frames have not been detected. These frames are surrounded by silence frames and can easily be detected and removed in a post-processing step. The filled pause segment from frame 275 to frame 375 is an elongated "ee" that is uttered after a Greek word.

As future work, we plan to improve the audio part by working at the segmental level instead of the frame level and by using sequence models (e.g., Hidden Markov Models) to explore the differences in sequential structure among filled pauses, elongated words and normal speech. We also plan to extend our method to summarize an oral presentation, adding more audio-visual features, so making possible searching and indexing tasks.

REFERENCES

[1] J.E. Fox Tree, "The effects of false starts and repetitions on the processing of subsequent words in spontaneous speech," *J. Memory & Lang.*, vol. 34, pp. 709–738, 1995.

[2] H. Moniz, A.I. Mata, and M.C. Viana, "On filled pauses and prolongations in european portuguese," in *Interspeech 2007*, pp. 1246–1249.

[3] M. Goto, K. Itou, and S. Hayamizu, "A real-time filled pause detection system for spontaneous speech recognition," in *Eurospeech99*, 1999, pp. 227–230.

[4] D. 0'Shaughnessy and M. Gabrea, "Automatic identification of filled pauses in spontaneous speech," in *CCECE 2000*, 2000, vol. 2, pp. 620–624.

[5] F. Stouten, J. Duchateau, J.-P. Martens, and P. Wambacq, "Coping with disfluencies in spontaneous speech recognition: Acoustic detection and linguistic context manipulation," *Speech Communication*, vol. 48, pp. 1590–1606, 2006.

[6] Y.-X. Li, Q.-H. He, and T. Li, "A novel detection method of filled pause in mandarin spontaneous speech," in *ICIS 08*, 2008, pp. 217–222.

[7] F. Bimbot, J. F. Bonastre, C. Fredouille, G. Gravier, Magrin I. Chagnolleau, S. Meignier, T. Merlin, Ortega J. Garcia, Petrovska Delacretaz, and Reynolds, "A tutorial on text-independent speaker verification," *EURASIP Journal on Applied Signal Processing*, vol. 4, no. 4, pp. 430–451, 2004.

[8] B. Hanson and T. Applebaum, "Robust speaker-independent word recognition using static, dynamic and acceleration features: experiments with lombard and noisy speech," in *ICASSP-90*, 1990, vol. 2, pp. 857–860.

[9] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," in *Speech Coding and Synthesis (W. B. Kleijn and K. K. Paliwal, eds.)*, pp. 495–518. Elsevier, 1995.

[10] M. Slaney, "Auditory toolbox version 2," Tech. Rep. 1998-010, Interval Research Corproation, 1998.

[11] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statist. Soc. Ser. B (methodological)*, vol. 39, no. 1, pp. 1–22 et 22–38 (discussion), 1977.

[12] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: A new data clustering algorithm and its applications," *Data Min. and Knowl. Disc.*, vol. 1, no. 2, pp. 141–182, 1997.

[13] B. Han, D. Comaniciu, and L. Davis, "Sequential kernel density approximation through mode propagation: Applications to background modeling," in *ACCV 2004*, 2004.

[14] A. Caplier, L. Bonnaud, and J.M. Chassery, "Robust fast extraction of video objects combining frame differencesand adaptive reference image," in *ICIP01*, 2001.

[15] I. Grinias and G. Tziritas, "Foreground object localization using a flooding algorithm based on inter-frame change and colour," in *IEEE, AVSS 2007*, 2007.

[16] M.J. Lee, A.S. Lee, D.K. Lee, and S.Y. Lee, "Video representation with dynamic features from multi-frame frame-difference images," in *Motion07*, 2007, pp. 28–34.

[17] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, 1992.

[18] E. Ramasso, C. Panagiotakis, M. Rombaut, D. Pellerin, and G. Tziritas, "Human shape-motion analysis in athletics videos for coarse to fine action/activity recognition using transferable belief model," *Electronic Letters on Computer Vision and Image Analysis*, vol. 7, no. 4, pp. 32–50, 2009.