UNSUPERVISED SIGNAL SEGMENTATION BASED ON TEMPORAL SPECTRAL CLUSTERING

J. Arias, R. André-Obrecht, J. Farinas

IRIT/SAMOVA, Université Paul Sabatier
118 route de Narbonne, 31062 Toulouse, France
phone: + (33) 5 6155 7201, fax: + (33) 5 6155 6258
email: {arias,obrecht,jfarinas}@irit.fr
web: http://www.irit.fr/-Equipe-SAMoVA-

ABSTRACT

This paper presents an approach for applying spectral clustering to time series data. We define a novel similarity measure based on euclidean distance and temporal proximity between vectors. This metric is useful for conditioning matrices needed to perform spectral clustering, and its application leads to the detection of abrupt changes in a sequence of vectors. It defines a temporal segmentation of the signal. When the input to the algorithm is a speech signal, we further process the segments and achieve their labeling in one of three phonetic classes: silence, consonant or vowel. When the input signal is a video stream, the algorithm detects scene changes in the sequence of images. Our results are compared against classic unsupervised and supervised techniques, and evaluated with the phonetically labeled multi-language corpus OGI-MLTS and the video database of the french video indexing campaign ARGOS.

1. INTRODUCTION

Currently, automatic segmenting systems rely on the knowledge of a signal's statistical model, and when this is difficult to define, some model-free approaches are proposed. Existing solutions are often based on complex assumptions as ARMA transition measures, temporal or spectral autoregressive modeling, support vector machine optimisations, dynamic programming, genetic algorithms or Bayes theory [1]. The time series segmentation we propose is a fast and simple unsupervised method built on the idea of euclidean and temporal similarity among neighbor vectors and the spectral clustering theory.

Our algorithm is performed in three steps. First, parameters localised in time are generated from the input signal. These can be dominant colours in the case of video images or cepstral coefficients computed from audio signals. Second, we process these descriptors to obtain the affinity matrix necessary for performing spectral clustering. Finally, we transform this matrix to disclose stable temporal segments on the input signal.

Temporal spectral matrix construction is inspired by the fact that sequential speech/image parametric vectors can be considered as nodes of a weighted graph [2]. Edges are weighted according to the similarity and temporal order between points. Following this, similar vectors situated far away in time on the sequence are considered 'different', and dissimilar vectors close in time are used to define segment borders over the signal.

When the input to the algorithm is speech, we can use some *a priori* knowledge of its acoustic nature to classify the resulting segments. We know that speech is a continuous signal composed from stationary and transitory units, which we can loosely label as vowels, consonants or silences (we associate voicing sounds to the 'vowel' label, and unvoicing segments to the 'consonant' label, but it this is not a formal association because vowel and consonantal qualities come from some complex abstractions as dynamic behaviour). We perform this labeling step based in the low dimensional representation of the signal obtained after applying Kernel PCA to the representative vector of each discovered segment. Augmenting MFCC or LPC parametrisation can be useful in speech recognition systems, speaker verification, language identification applications and conceptual modelling of audio documents [3]. This proposal is an evolution of the system described in [4], but in this case we incorporate timing information in the spectral clustering based signal segmentation.

When the algorithm process a video stream, scene change detection is performed and the defined segments are grouped to obtain consistent stories. In this case we do not know the number of classes present in the document, so we try to discover this number automatically.

The paper is organized as follows. In Section 2 we recall basic elements about spectral clustering and Kernel PCA. In Section 3 we define the similarity measure used to transform the original affinity matrix of spectral clustering into a matrix from which we obtain a temporal segmentation. Section 4 is devoted to explain the whole segmenting process and Sections 5 and 6 give results of different evaluations. Finally, in Section 7 we present some conclusions and further work.

2. SPECTRAL ALGORITHMS

2.1 Spectral clustering

Spectral clustering methods have been mainly applied to image segmentation, using image's pixels to calculate an affinity matrix A and its eigenvectors in different ways [5]. For example, A could be constructed by fitting a radial basis function (RBF) kernel to the data, and normalised to take into account the different spread of several clusters. After diagonalization of A, the first q eigenvectors are arranged as columns in a matrix Y. The rows of this matrix are then treated as q-dimensional vectors. Desired clustering are obtained after performing k-means algorithm on these vectors.

The spectral clustering algorithm defined in [6] may be explained informally considering the 'ideal' case, where the vectors are ordered according to the cluster they belong to, and where the different clusters are considered infinitely far

apart from each other. In this case, A is block diagonal and each block A^{ii} represents the intra-cluster affinities for cluster i, $i = 1,\ldots,q$, with $q =$ *number of clusters to be discovered*. For ex., if $q = 3$ we obtain the next representation for A.

$$A = \begin{bmatrix} A^{11} & 0 & 0 \\ 0 & A^{22} & 0 \\ 0 & 0 & A^{33} \end{bmatrix} \quad (1)$$

Since A is block diagonal, its eigenvalues and eigenvectors are the union of the eigenvalues and eigenvectors of its blocks padded appropriately with zeros. If we take the principal normalized eigenvector of each block A^{ii}, we will have the $n \times q$ matrix Y.

$$Y = \begin{bmatrix} Y^1 \\ Y^2 \\ Y^3 \end{bmatrix} = \begin{bmatrix} v_1^{(1)} & 0 & 0 \\ 0 & v_1^{(2)} & 0 \\ 0 & 0 & v_1^{(3)} \end{bmatrix} \quad (2)$$

Each row Y_i correspond obviously to the true clustering of original data. In the general case there are not infinite distances between clusters, but we expect to recover a stable cluster configuration whether a large *eigengap* between the retained and discarded eigenvalues exists.

2.2 Kernel Principal Component Analysis

Kernel PCA performs principal component analysis in feature space by the mean of a kernel function κ such that $\kappa(y_i, y_j) = \langle \Phi(y_i) \cdot \Phi(y_j) \rangle$ [7]. Kernel PCA finds a subspace that seems to be the result of an information extraction process, helped by the implicit non-linear transformation $\Phi(y)$.

Eigenvectors u_p in feature space lie in the span of $\alpha_i^p \Phi(y_i)$. The weights α_i^p are obtained from the relationship between eigenvalues and eigenvectors of kernel and covariance matrices of centered feature space data. Projections of test vectors $\Phi(y)$ onto principal eigenvectors in feature space are computed using a kernel function:

$$P_{u_p} = u_p' \Phi(y) = \left\langle \sum_{i=1}^{n} \alpha_i^p \Phi(y_i), \Phi(y) \right\rangle = \sum_{i=1}^{n} \alpha_i^p \kappa(y_i, y) \quad (3)$$

3. EUCLIDEAN/TEMPORAL SIMILARITY MEASURE

We find inspiration in the ideal case of spectral clustering described in eq. 2 to take into account temporal information between vectors in times series and discard some components far from the main diagonal of the original affinity matrix.

After computing the affinity matrix A of a sequence $X = \{x_i,\ i = 1,\ldots,n\}$ with a RBF kernel, we analyze the dissimilarity between each diagonal element a_{ii} of the matrix and their forward neighbours $[a_{i,i+1}, a_{i,i+2}, a_{i,n}] \quad i = 1,\ldots,n$.

When the difference between a_{ii} and a neighbour a_{ij} is superior to a predefined threshold ε, the rest of the elements $a_{i,j+1},\ldots,a_{in}$ are set to 0 (see figure 1). In this manner we isolate a 'pseudo stable' temporal segment $S_i = [x_i, x_{i+1},\ldots,x_j]$ from the diagonal of A, associated to the element x_i.

The temporal spectral clustering matrix $\widetilde{A}$ is defined as :

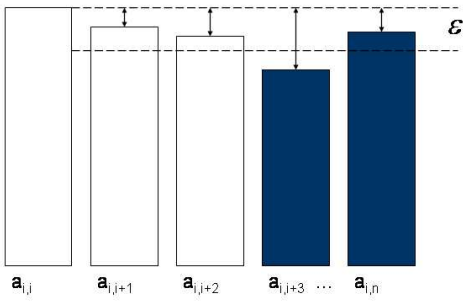

Figure 1: *Metric for the modification of the affinity matrix. Starting from the diagonal element a_{ii}, if the difference respecting a forward neighbour $a_{i,i+j}$ is superior to a threshold, the rest of the sequence is considered 'infinitely' far from a_{ii} (frames filled in black in the figure).*

$$\text{if } i < k$$

$$\widetilde{a}_{ik} = \begin{cases} \exp^{-\frac{||x_i - x_k||^2}{2\sigma^2}} & \text{if } x_k \in S_i \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$\widetilde{a}_{ki} = \widetilde{a}_{ik}$$

The matrix $\widetilde{A}$ is a block diagonal matrix, consisting of p blocks. Figure 2 shows a standard affinity matrix computed using a sequence of cepstral speech vectors. It also shows the diagonal symmetric blocks formed in the corresponding temporal spectral clustering matrix. Eigenvectors of $\widetilde{A}$ identify stable temporal units in the input sequence, as shown in figure 3.

Indeed, temporal segmentation is achieved with the diagonalization of $\widetilde{A}$, normalizing and thresholding its principal eigenvectors.

Scaling parameters σ and ε control how the affinity is measured between vectors. They are chosen using cross validation.

4. SEGMENTING ALGORITHM

We obtain a temporal segmentation from $\widetilde{A}$. Consider that the rank of $\widetilde{A}$ provides the number of potential clusters in the matrix ($\text{rank}(\widetilde{A}) \geq q$). We extract eigenvectors of $\widetilde{A}$ associated to non-zero eigenvalues and superior to a certain threshold, because spectral clustering theory shows that only the q most important eigenvalues of the affinity matrix are relevant for clustering.

Each eigenvector defines a temporal segment S_q. The union of these boundaries give the final temporal segmentation. The relation among the signal and its temporal segments is shown in fig. 4. Transient periods in the signal are registered as a chain of short segments. We fusion these small units to create transitory segments.

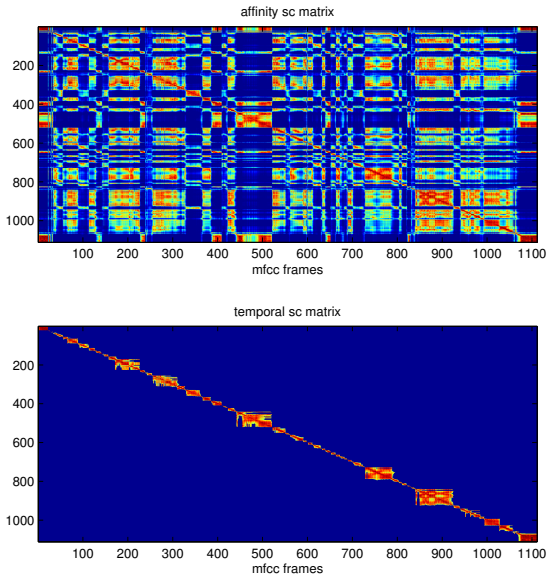

Figure 2: *Spectral clustering matrix of speech vectors. Affinity matrix A (top) computed with a typical spectral clustering algorithm. Modified matrix $\widetilde{A}$ (bottom) computed with temporal spectral clustering.*

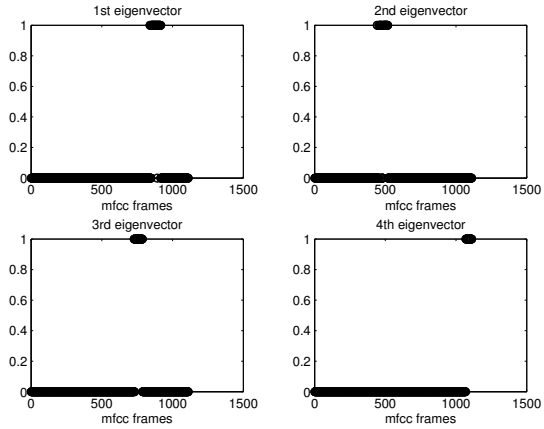

Figure 3: *First eigenvectors of $\widetilde{A}$. The nonzero interval of each eigenvector defines a temporal segment.*

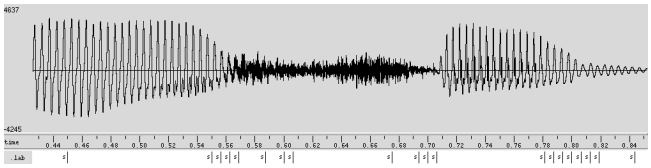

Figure 4: *Speech signal segmented with the temporal spectral clustering algorithm (before smoothing).*

In conclusion, the algorithm consists of the following steps:

1. Parametrization of the sequence (for example, cesptral coefficients for audio frames or histograms for images).
2. Affinity matrix A construction using a RBF function with parameter σ.
3. Modification of A to isolate its diagonal blocks using ε.
4. Diagonalization of $\widetilde{A}$. Normalization and thresholding of the eigenvectors corresponding to non zero eigenvalues.
5. Association of each eigenvector to a temporal segment in the signal.
6. Smoothing phase to fusion very short segments.

Once we define the signal temporal segmentation, we can choose a vector representing each segment and perform their labeling according to the contents and segment classes we are dealing with. Further clustering can be done with spectral clustering [8], k-means clustering [9] or Kernel PCA [10].

5. SPEECH EXPERIMENTS

5.1 The OGI-MLTS corpus

The speech database OGI-MLTS is a reference in the language identification community [11]. The corpus consists of spontaneous telephonic speech presented in sequences of around 45 seconds length sampled at 8 kHz. It is phonetically labeled by experts following the CSLU rules [12] .

We use a six languages subset of the OGI-MLTS database to perform tests: English, German, Hindi, Mandarin, Japanese and Spanish. For testing purposes we use eight files per language which represents almost 40 minutes of speech. We perform cepstral parametrisation of the corpus using 14 coefficients plus energy, derivatives and acceleration. The signal is decomposed into 16 ms frames with a frame overlap of 12.5 ms. Cepstral features are normalized by cepstral subtractions.

System evaluations are made using one tool issued from NIST campaign for audio indexing. This tool calculates the accuracy of raw audio segmentation.

5.2 Temporal segmentation and SCV labeling

Using the segmenting algorithm of Section 4, we define temporal sub-phonetic segments in speech. We perform Kernel PCA in vectors identifying each segment to obtain a 3-dimensional representation of data. In this space we can further apply k-means algorithm. As a first labelling study, we propose to detect the speech most important phonetic classes: silences, consonants and vowels. After applying the 3-means algorithm, phonetic identification of the clusters is achieved automatically based in the mean energy of each cluster. Highest energy correspond to vowel (V) cluster, lowest energy is associated to silences (S) and consonant (C) cluster is in between.

In figure 5 we show an example of the SCV labeling in the clustering space. The low dimensional embedding of Kernel PCA orders silences, consonants and vowels in a smooth and separable configuration. We use a RBF kernel with $\sigma = 3$.

We evaluate our system against OGI manual segmentation and labeling (see Table 1). Three algorithms are implemented for comparison purposes: two unsupervised approaches and one supervised system.

Baseline system [3] uses the forward-backward divergence (fbd) algorithm for temporal segmentation. Fbd algorithm looks for changes in the auto-regressive model of two shifted sliding windows to define temporal borders in the sig-

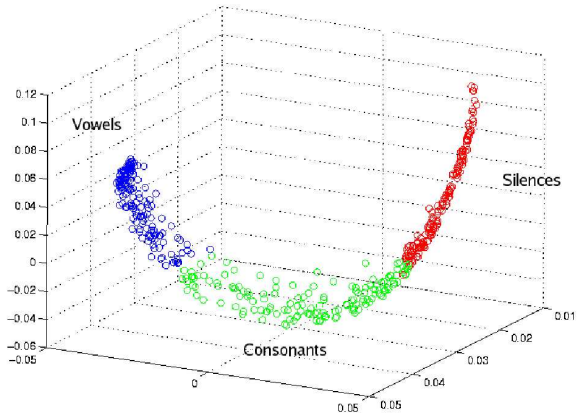

Figure 5: *Low dimensional embedding with the three most relevant components of Kernel PCA. One vector (typically the vector of the middle of the segment) representing the segment it belongs is used to perform the embedding. Phonetic classes (SCV) are grouped with k-means algorithm.*

nal. It utilizes the energy in spectral band of 0.35-1kHz to classify the segments [13].

The second baseline system [4] uses also fbd algorithm for temporal segmentation, but it utilizes a low dimensional projection of the vectors representing segments obtained with standard spectral clustering for the SCV labeling.

A supervised HMM system is implemented as a reference for unsupervised approaches. HMM system is trained with a different 20 minutes subset of the speech corpus, and it is tested with the same files than the unsupervised systems. It consists of trigram (three states connected from left to right) phoneme models : silence, consonant and vowel. Each state is modeled with an eight-component gaussian mixture. Its accuracy is 81.22%, and is considered as a reference for automatic systems. The advantage of our proposal is that if an unknown language is incorporated to the corpus, we do not need a training phase as the HMM system will do.

Table 1: *Accuracy of segmentation and SCV labeling of three unsupervised systems.*

System	Accuracy
F-b divergence + 0.35-1kHz energy	72.66 %
F-b divergence + Standard spectral clustering	73.14 %
Temporal spectral clustering + Kernel PCA	**74.66** %

6. VIDEO EXPERIMENTS

6.1 The ARGOS corpus

ARGOS evaluation campaign [14] is aimed to develop resources for a benchmarking of video content analysis. Their corpus consists of three video sources: TV news journals, documentaries and video surveillance scenes. For the system tests we use 2 hours of TV news, which are files encoded in MPG-1 format with a resolution of 352 x 288 pixels.

The corpus annotation is fulfilled according to the production rules. We compare the performance of our segmenting algorithm against the shot boundary detection ground truth provided with the corpus.

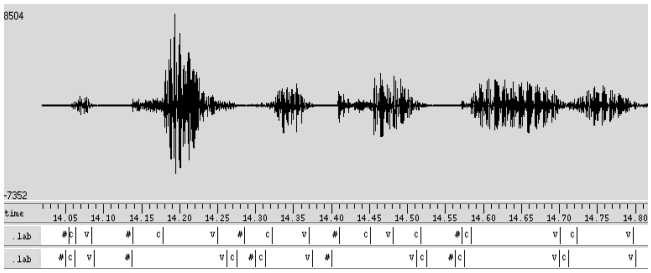

Figure 6: *Temporal segmentation of speech and labeling of 3 phonetic classes (SVC). The first row shows labels from manual OGI labeling and second row shows our automatic unsupervised labeling.*

In this case, input videos are transformed into sequences of images. Each image is then represented with nine parameters: average luminance, first and second HSV dominant colors, contrast and movement quantity.

6.2 Temporal segmentation and grouping

We use the algorithm described in Section 4 to perform temporal segmentation in image sequences. According to the evaluation metric, transient segments in video represent plane transitions so they are skipped for computing the segmentation accuracy. Then, 'holes' in the video timeline are tolerated. Results of temporal segmentation using ARGOS corpus are presented in Table 2.

Table 2: *Segmentation accuracy for different TV news journals of the ARGOS database.*

Program	Accuracy
INAO1 - 35 mins	55 %
INA02 - 35 mins	61 %
SFR01 - 30 mins	58 %
INA06 - 35 mins	62 %

For the grouping step of the algorithm, we do not know the number of classes representing the key images extracted to represent the segments. We then run the cluster detecting method described in [8], which is based in a modified version of the spectral clustering algorithm [6]. The main idea is to consider that the low dimensional representation issued from spectral clustering should not be normalized. Considering this, a modified version of k-means clustering based on Mahalanobis metric will discover the clusters. An example of this algorithm running on key images representing segments is shown in figure 7.

We can put together the segments belonging to the same class to construct homogeneous video stories. In the example shown in figure 8 we separate different topics from a TV journal.

7. CONCLUSIONS AND FURTHER WORK

The new temporal segmenting algorithm presented in this paper is derived from a modification on the known spectral clustering procedures and is aimed to process multimedia

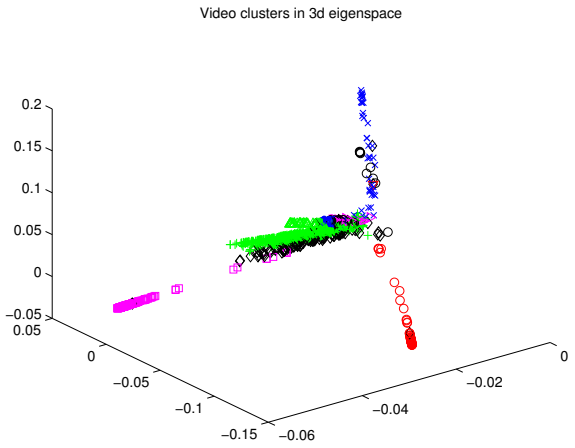

Figure 7: *Plot of eigenvectors representing stories in 3D clustering space.*

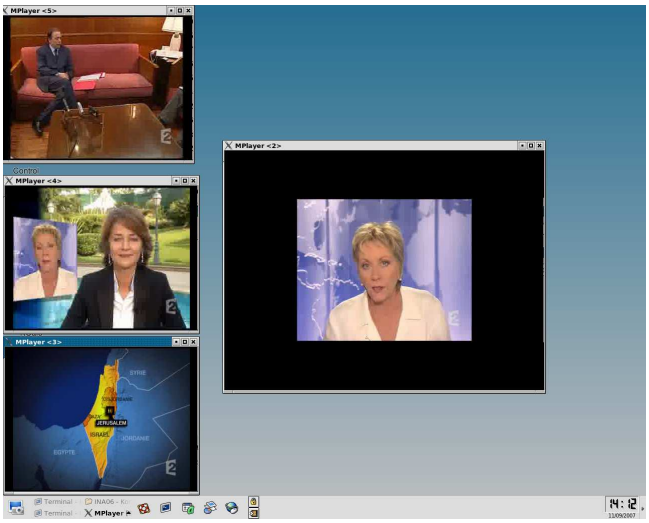

Figure 8: *Result of video segmentation and clustering. Detected video segments with similar characteristics are grouped to construct homogeneous stories.*

time series. The algorithm is unsupervised, with only two parameters to define.

In the case of speech processing, phonetic class labeling is also performed. Our test corpus consist of conversational and multi-language speech. We believe that applications of this kind are necessary to extract information that can be used in tasks ranging from prosodic analysis to language identification.

Tested with video frames, the segmenting and grouping algorithm decomposes TV programs into consistent stories.

Spectral methods are suitable for practical tasks in multimedia processing. Future research directions will exploit eigenvectors extracted from sequences for comparison purposes.

8. ACKNOWLEDGEMENTS

José Arias work is supported by a SFERE-CONACyT scholarship of the Mexican National Council of Science and Technology.

REFERENCES

[1] M. Carr and P. Pierrick, "Indexation audio : un état de l'art," *Annales des télécommunications*, vol. 55, no. 9-10, pp. 507–525, 2000.

[2] Y. Weiss, "Segmentation using eigenvectors: an unifying view," in *Proceedings of the International Conference on Computer Vision*, 1999.

[3] R. André-Obrecht, "A new statistical approach for automatic speech segmentation," *Transactions on Audio, Speech, and Signal Processing*, vol. 36, no. 1, Jan. 1988.

[4] J. Arias, "Unsupervised identification of speech segments using kernel methods for clustering," in *9th European Conference on Speech Communication and Technology*, Lisbon, Portugal, Sept. 2005.

[5] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the nystrm method," *Transactions on pattern analysis and machine intelligence*, vol. 26, no. 2, 2004.

[6] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, 2001, vol. 13.

[7] B. Schölkopf, A. Smola, and K. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, , no. 10, pp. 1299–1319, 1998.

[8] G. Sanguinetti, J. Laidler, and N. Lawrence, "Automatic determination of the number of clusters using spectral algorithms," in *Proceedings of the IEEE Workshop on Machine Learning for Signal Processing*, 2005.

[9] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, UK, 2004.

[10] B. Schölkopf and A. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.

[11] Yeshwant Kumar Muthusamy, Ronald A. Cole, and B. T. Oshika, "The ogi multilanguage telephone speech corpus," in *International Conference on Speech and Language Processing*, Oct. 1992, vol. 2, pp. 895–898.

[12] T. Lander, "The cslu labeling guide," *Internal Report. Center for Spoken Language Understanding*, 1997.

[13] F. Pellegrino, J. Farinas, and R. André-Obrecht, "Comparaison of two phonetic approaches to language identification," in *European Conference on Speech Communication and Technology*, Budapest, Hongrie, Sept. 1999, pp. 399–402, 5-9 sep.

[14] P. Joly, E. Kijak, G. Quénot, and J. Benois-Pineau, "Argos: French evaluation campaign for benchmarking of video content analysis methods," in *SPIE 19th Annual Symposiun on Electronic Imaging, Multimedia Content Access: Algorithms and Systems*, Jan. 2007.