Multi-method Audio-based Retrieval of Multimedia Information

MARIO MALCANGI
DICo – Dipartimento di Informatica e Comunicazione
Università degli Studi di Milano
Via Comelico 39 – 20135 Milano
ITALY
malcangi@dico.unimi.it

Abstract: - Multimedia information and embedded systems are two major technological advances that have significantly changed the way people interact with systems and information in recent years. In this context, audio proves to be the most advantageous media for interacting with embedded systems and their content. Advantages include: hands-free operation; unattended interaction; and simple, cheap devices for capture and playback. The use of embedded systems to seek information stored locally or on the web points up several difficulties inherent in the nature of multimedia-information signals. These difficulties are especially evident when palmtop or deeply embedded devices are used for such purposes. Developing a set of digital-signal-processing-based algorithms for extracting audio information is a primary step toward providing user-friendly access to multimedia information and developing powerful communication interfaces. The algorithms aim to extract semantic and syntactic information from audio signals, including voice. Extracted audio features are employed to access information in multimedia databases, as well as to index it. More extensive, higher-level information, such as audio-source identification (speaker identification) and genre (in the case of music), must be extracted from the audio signal. One basic task involves transforming audio into symbols (e.g. music transformed into a score, speech transformed into text) and transcribing symbols into audio (e.g. score transformed into musical audio, text transformed into speech). The purpose is to search for and access any kind of multimedia information by means of audio. To attain these results, digital audio-processing, digital speech-processing, and soft-computing methods need to be integrated. Neural networks are used as classifiers and fuzzy logic is used for making smart decisions.

Key-Words: - Audio features, multimedia information, speech-to-text, audio-to-score, text-to-speech, score-to-audio, digital audio processing, pattern matching, soft computing

1 Introduction

The traditional approach to searching for information in storage systems (databases) and networks (webpage content) is limited to text. Engines like Google are able to search for information where an alphanumeric string matches webpage content. Multimedia information can be retrieved only if text is embedded in it. Such text is not exhaustive of multimedia information. Therefore, much of the multimedia information will not be available to search engine unless a preprocessing action, such as indexing or transcribing text (titles), has been executed.

There is a great deal of multimedia information that cannot actually be represented by text, since most such information is not strictly semantic [2][15]. Audio and video are very rich in information content, but audio is part of the video so that video information is also related to audio. Videos can be more extensively indexed by means of audio fragments than by text indexing. For this reason, a primary step in building a multimedia search engine must focus on audio classification.

The audio component of multimedia information is characterized by certain peculiarities. It is simple to capture (microphones are very cheap, always available in most embedded devices, very simple to use in unattended mode, etc.), needs only a low sampling rate to be digitalized, and can be synthesized relatively easily.

Audio exists in three main forms, depending on its source: voice, music, and generic sounds. Each of these is a completely distinct form of audio information, which shows to what extent audio can be richer in information than any other medium. However, this is a disadvantage as far as processing is concerned, because different algorithms need to be applied for pattern-matching and synthesis purposes.

A huge amount of research has focused on speech processing (recognition, identification, synthesis, encoding, and decoding) [1][16][17]. Music has also received researchers' attention for several decades, particularly in terms of synthesis [13][14][6]. Only recently has much attention focused on genre recognition and on stream retrieval.

Far less attention has been aimed at general purpose sounds, primarily because such sounds are generated by different sources and carry different information. But this audio is no less rich in information than speech or music. One example is speech mixed with sounds such as hiccups, coughs, and so forth. Another is (modern) music mixed with generic sounds[23][24].

Speech-to-text and music-to-score are two basic strategies for empowering a traditional search engine to match text and/or string sequences on the basis of traditional text-retrieval tactics. When text and scores are not available in a multimedia database, pattern matching is the only possible strategy for successfully providing interaction and access to multimedia information in a retrieval application.

Applying pattern matching to audio is a very complex task. It consists of two main processing subtasks, feature extraction and distance evaluation. Feature extraction refers to classical digital signal-processing algorithms. Distance evaluation can be accomplished through hard-computing methods, such as dynamic time warping (DTW) or the hidden Markov model (HMM) [9]. Alternatively, it can be achieved with soft-computing methods such fuzzy logic [5][10][11][19][20] or artificial neural networks (ANNs) [4][7][8][12]. A methodology that combines of the above principles may also afford a good solution, especially when the pattern-matching task is highly complex.

Other methods can be also considered as alternative to the classical for the pattern matching[25].

2 Audio-feature extraction

Audio features are extracted from raw audio data in the time and frequency domain, by analyzing short, 50%-overlapped, Hamming-windowed frames. A frame duration of 20 milliseconds is used to measure features, thus generating a set of feature-time sequences.

The main time-domain features are RMS (root mean square) and ZCR (zero-crossing rate). Other time-domain features can be derived from these through more complex processing, so as to highlight certain specific audio proprieties.

RMS is the measurement of audio loudness. This feature is essential in tracing the presence or absence of audio or in using dynamics to characterize an audio source:

$$RMS(n) = \sqrt{\frac{1}{N}\sum_{m=0}^{N-1} s^2(m)}$$

Zero-crossing rate is a frequency measurement based on the time domain. This feature is a good indicator of the nature of the audio frame, in terms of the extent to which it is pitched or unpitched:

$$ZCR(n)=$$

$$=\sum_{m=0}^{N-1} 0.5\left|sign(s(m)-sign(s(m-1))\right|w(n-m)$$

where $w(n)$ is a rectangular window of length N and scaled by $1/N$. We can thus measure the number of zero crossings per sample.

Zero-crossing rate correlates with frequencies that have major energy levels. In speech signals, voiced sounds have low zero-crossing rate values because such sounds are pitched. High zero-crossing rate measurements reveal the absence of pitch characteristics of unvoiced utterances.

A sound fundamental frequency can also be estimated from its zero-crossing rate as:

$$F_o = (ZCR * F_s) / 2$$

where F_s is the sampling rate.

The main frequency-domain features are pitch and band spectrum. Other frequency-domain features can be derived from these to focus on specific audio characterization, e.g. to distinguish between speech and music[26][27].

Pitch (P) is measured using an autocorrelation function:

$$AC(i)=\sum_{i=1}^{N}\sum_{j=1}^{N+1-i}x(j)x(i+j-1)$$

Band spectrum (represented by *B* in the formula below) is computed using a short-term Fourier transform (STFT). The frequency spectrum is divided into bands representative of frequency grouping in audio (formants for speech, harmonic distribution for music, etc.):

$$B_1(j)_1=[0,\frac{\omega_0}{8}],$$

$$B_2(j)=[\frac{\omega_0}{8},\frac{\omega_0}{4}],$$

$$B_3(j)=[\frac{\omega_0}{4},\frac{\omega_0}{2}],$$

$$B_4(j)=[\frac{\omega_0}{2},\omega_0]$$

Other features can be also computed using more complex computations, such as Cepstrum and Linear Prediction Coding (LPC). These measurements can help extend the pattern matcher's ability to support additional applications such as speaker identification or melody tracing.

3 Hard-computing Pattern Matching

Hard-computing pattern matching refers to two main methods, primarily for speech recognition: dynamic time-warping and hidden Markov modeling.

The dynamic time-warping algorithm is employed extensively in audio pattern-matching applications to align and compare audio templates with the audio segment to be recognized. The templates consists of a combination of coefficients extracted from each audio frame. The DTW algorithm computes the matching cost for each audio item in the set of searchable audio patterns.

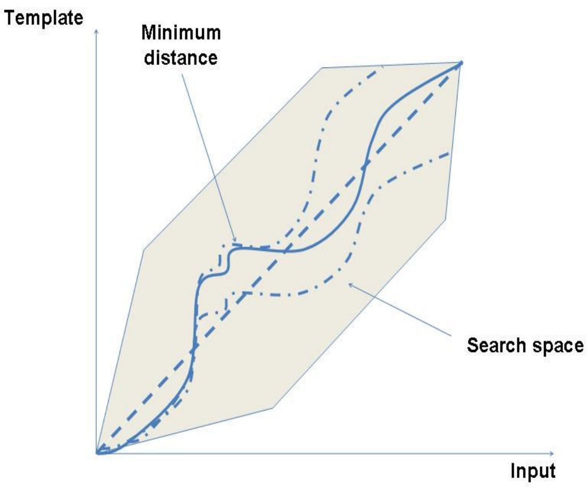

Fig. 1 DTW pattern alignment.

Euclidean distance is used to score the audio patterns to be recognized. The Mahalanobis distance measurement

$$D_i(x)=(x-\overline{x})^T W^{-1}(x-\overline{x})$$

requires that a test pattern *x* be processed with reference to the averaged feature vector $\overline{x}$ representing the audio item to be found. The distance $D_i(x)$ is used as a score for the *i-th* audio item in the pool of searchable sounds.

4 Soft-computing Pattern Matching

Soft-computing pattern matching refers to two main methods: fuzzy logic and artificial neural networks. Each of these has its own particular advantages and disadvantages, so that choosing between the two depends upon the nature of the information that has to be matched.

4.1 Fuzzy-logic-based Pattern Matching

Fuzzy logic is a novel approach to pattern matching. Its traditional domain of implementation, where it has been very successful, has been in control applications. Only in recent years has part of the research focused on exploring fuzzy logic's ability to match patterns.

A membership function is defined for each measured feature to transform the feature from crisp datum into a fuzzy value. These normalized measurements are fuzzified according to membership functions appropriate to the feature being measured.

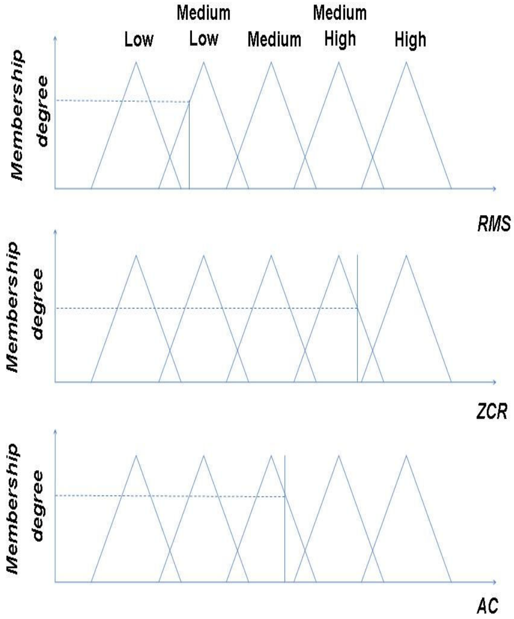

Fig. 2 Membership functions.

A set of rules is defined to model the way a specific class of audio patterns might be recognized by a human being. An expert in audio perception transfers her or his knowledge into these rules, thus tuning the fuzzy logic engine [21] so that it recognizes such sounds the way the human expert does. A typical rule has the following format:

IF f_1 IS (fuzzy value) AND

IF f_2 IS (fuzzy value) AND …

THEN the item IS (fuzzy value)

4.2 Neural-network-based Pattern Matching

Artificial neural networks prove very adept at matching an isolated audio pattern as a result of their ability to compensate for the time variation in the input vis-à-vis the template. A three-layer, feed-forward, back-propagation artificial neural network (FFBP-ANN) was used for this purpose. Such an ANN has its inputs fully connected to all the nodes of the hidden layer. The hidden layer is also fully connected to the output nodes.

All the input lines have a linear activation function. A non-linear, sigmoid activation function connects hidden-layer nodes to output-layer nodes:

$$s_i = \frac{1}{1+e^{-I_i}}$$

$$I_i = \sum_j w_{ij} s_j$$

s is the output of the i-th unit
E_i is the total input
w_{ij} is the weight from the j-th to i-th unit

The ANN's input is the binary-encoded audio features. This input refers to a given audio frame, and can be sequenced so that only stationary frames will stimulate the ANN.

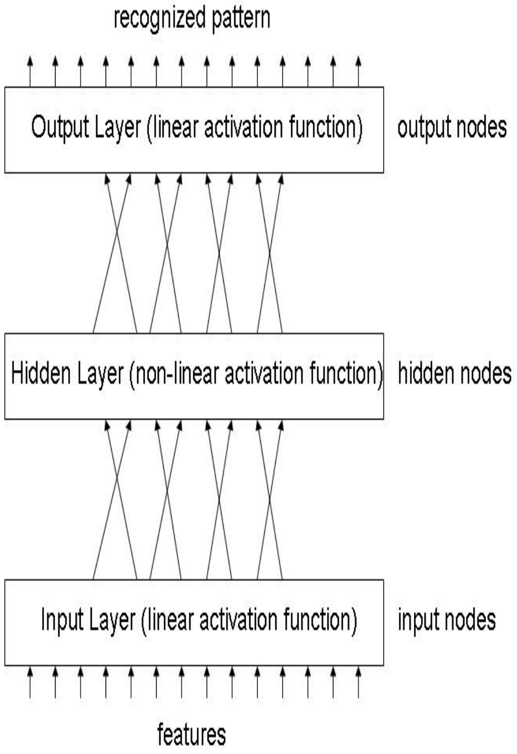
recognized pattern
Output Layer (linear activation function)
output nodes
Hidden Layer (non-linear activation function)
hidden nodes
Input Layer (linear activation function)
input nodes
features

Fig. 3 Architecture of the FFPB- ANN

The ANN's output encodes the audio frame that corresponds to the features that have been fed into it as input. Isolated, as well as continuous, audio frames can be identified by the ANN, depending on how it is trained.

5 Audio Spotting

Audio spotting might be considered the initial level of combining hard-computing and soft-computing methods to meet the challenge of accessing multimedia information by means of audio input.

Audio spotting means detecting occurrences of an audio pattern in continuous audio streams. Words to be spotted are represented by their features and by named templates. The template's features are aligned with the features extracted from the audio stream.

Hard-computing methods, such as DTW and HMM, can be used for this purpose, but soft-computing methods have proven more effective. This is due to the ability of the ANN to work like a window that moves along the signal. A combination of hard-computing and soft-computing methods yields better performance because soft computing, principally fuzzy logic, enables solutions to several problems that arise in regard to the decision thresholds brought into play by hard-computing methods.

Time normalization is the process needed to normalize the features' time variations so the template and the pattern can be correctly aligned to evaluate the distance between them. This is basically a stretching or compressing process that forces the template and the pattern to be equal in length. Due to inter-pattern time variability, time normalization is a mix of time stretching and time compressing.

Time warping, applied to the word-spotting problem, is a practical method for finding the end point of an audio pattern that is embedded in an audio stream [22]. Continuous path generation highlights the pattern that best matches the template. The global cost (the sum of the measured discrepancy between each couple of aligned samples):

$$D(C) = \sum_{k=1}^{K} d[c(k)]$$

$$where$$

$$d[c(k)] = (a_{i(k)} - b_{j(k)})^2$$

is then minimized by considering the lattice of points.

A threshold comparison is needed to complete the identification process. It prevents the detection of false positives. This solution is very effective but is highly sensitive to any noise that might have contaminated the audio stream. Smart logic (fuzzy logic) is then needed to develop a decision logic robust enough to withstand data variation due to noise.

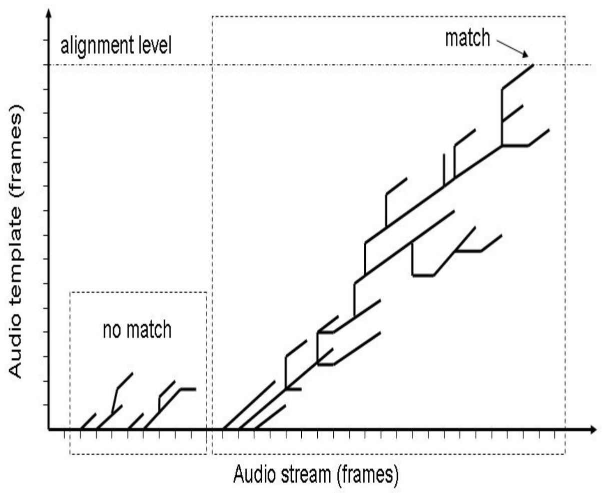

Fig. 4. DTW pattern alignment during the word spotting process

HMM can be also a valid alternative to DTW in audio spotting because its offers performance comparable to that of dynamic programming. The advantage of HMM is its low computational cost compared to that required by the DTW algorithm. As does applying DTW to the problem of audio spotting, the HMM audio-spotting technique uses a modified left-to-right model with an extra state used to represent the audio pattern when it doesn't match the start of the target pattern. This state also corresponds to the state of the target pattern's end point.

HMM, again like the DTW method applied to audio spotting, needs a threshold decision logic. Good performance is obtainable when thresholds can be computed in advance, one for each template [x]. This solution limits the applicability of audio spotting to a few application categories.

Artificial neural networks can be combined with DTW or HMM to overcome the limits of application due to the thresholding. A combination of a time - delay neural network (TDNN) and a dynamic programming (DP) model for time warping was successfully tested; it performed at a figure of merit (FOM) rate of 82.5% (FOM is the average detection rate from 0 to 10 false alarms per keyword hour).

Better performance can be achieved using a combination of an ANN and a fuzzy logic engine (FLE). The ANN is used as a classifier and the FLE is used to evaluate, frame by frame, whether the frame belongs to the target or not. The target is mapped onto the FLE by a set of rules tuned specifically to represent the audio pattern to be spotted.

The ANN is a SOM (self-organizing ROM) that receives as input a set of audio features and yields as output an audio-frame classification using a two-dimensional map. When an audio stream is windowed and analyzed, a frame is generated. Each frame (with its measured features) activates the class to which it belongs. A sequence of audio windows generates a transition pattern by activating a different class area of the map. The fuzzy logic engine is tuned to work as an end-point detector. When the end-points of the target audio pattern are detected, the spotting process is thus completed.

6 Mapping sound features

Mapping the multidimensional feature space onto two-dimensional space is a good strategy for achieving an effective solution to the problem of seeking information using audio input. The Kohonen feature-map (KFM) is an ANN that can map multi-dimensional space onto two-dimensional space. Such mapping is more practical for use as data input to a categorization layer, specifically a fuzzy-logic-based categorization layer.

We use the ability of the KFM to map an n-dimensional input-vector space onto a layer where neurons are organized according to similarities in input values to align the audio target with the audio patterns to be searched. The mapping done by the KFM is very effective, as shown in several works concerning speech recognition based on phonetic recognition rather than word recognition.

The main contribution of the KFM is the clustering, as part of its mapping activity, of input measurements. It clusters them in a matrix of nodes according to their similarities. A further functionality is the topological distribution of the clusters, so that input measurements with greater similarity are mapped proportionally closer together. Otherwise, if the input measurements are dissimilar, they are mapped farther apart, again in proportion to their dissimilarity. This topological mapping ability can be processed by an upper layer to categorize audio information. The categories then serve when seeking other information in a multimedia-data context.

Training is needed to teach the KFM to recognize input-space topology. To this purpose, the map is initialized randomly beforehand, so that, once trained, it cannot be biased by a previous mapping or pseudo-mapping.

Input vectors used to train the map are ordered randomly. At training time, the mapping process is ruled by a mechanism that imposes order by sorting the input measurements as follows:

- The node that measures the least distance from the input value is the winner and adjusts its weight to be as close as possible to the input value it maps;

- Neighbors of the winning node adjust their weights to be closer to the same input-data vector.

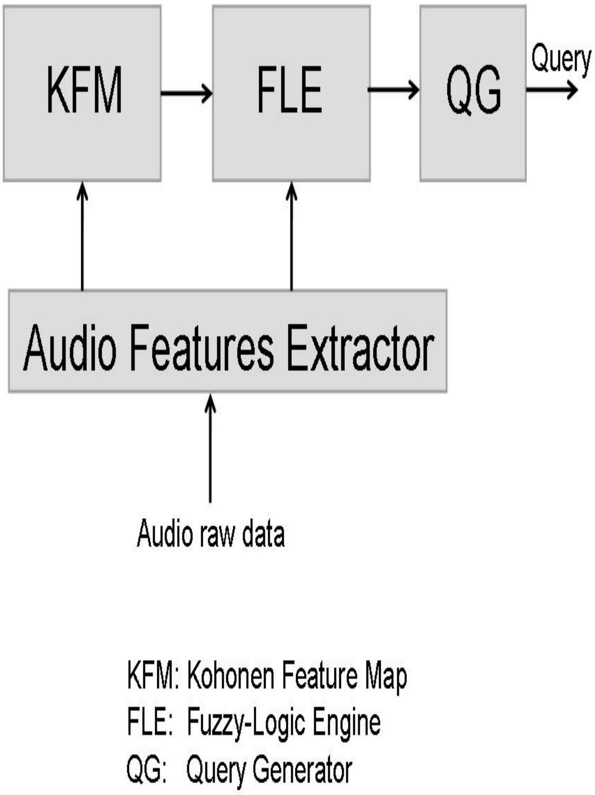

Fig. 5 KFM combined with an FLE to generate a query from an audio pattern.

In determining the winning node in the map, the Euclidean distance can be a very effective indicator, and it is computationally advantageous:

$$D_i = |X - W_i| =$$

$$= \sqrt{(x_1 - w_{i1})^2 + (x_2 - w_{i2})^2 + ... + (x_M - w_{iM})^2}$$

where:

X is the input-data vector
Wi is the vector of the weights of the *i-th* node.

A side-effect of the initial randomization of the map is that some nodes end up representing too much of the input data. This effect tends to introduce some clustering artifacts, altering the effectiveness of the mapping. To minimize this, a mechanism of a "conscience" is then applied.
How often each node wins is recorded and such information, at learning time, is used to bias the distance measurement, according to the following rules:

- when a node wins more than 1/N times (N is the number of Kohonen nodes), its distance is adjusted upward to attenuate its chance to win;
- for nodes that wins less than 1/N times, the distance is adjusted downward to make them more likely to win.

The distance adjusting factor is computed using the following:

$$B_i = \gamma(1/N - F_i)$$

F_i is the win frequency for *i-th* node

Initially F_i is equal to 1/N and γ starts with a high value between 2 and 10.

The adjusted distance D'_i is then computed as:

$$D'_i = D_i - B_i$$

thus obtaining a trained map that is highly representative of the audio feature to be categorized.

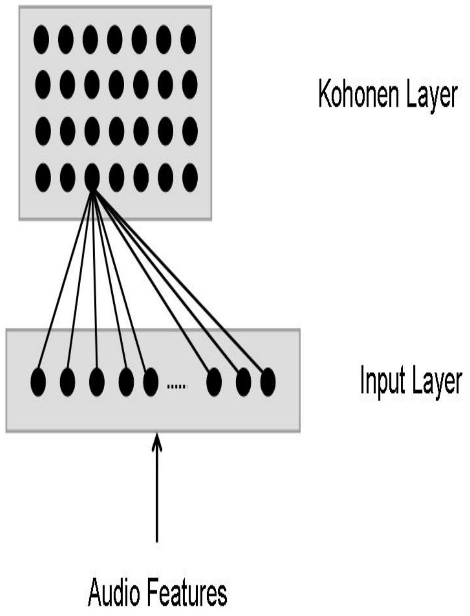

Fig. 6 Two-dimensional mapping of multi-dimensional features by a Kohonen feature map.

7 Fuzzy-logic-based categorization

Categorization of the mapped audio features is an important step that leads to a successful application of the KFM approach to the problem of searching for information in a multimedia-information repository. Through categorization, it is possible to organize classes of audio patterns that have the similarities demanded by a specific query to be executed on the multimedia-information repository.

Following a full ANN approach, the KFM can be extended with an upper layer to serve as a categorization layer. Due to the training dependency of such a layer, it is not the appropriate as a solution on which to build a search engine based on audio queries. A fuzzy-logic-based categorization layer can be a more flexible solution, because it doesn't need training by means of large amount of data patterns and because it can be linguistically programmed and tuned by an expert. If an expert in not available, fuzzy logic setup can also be executed automatically in several ways.

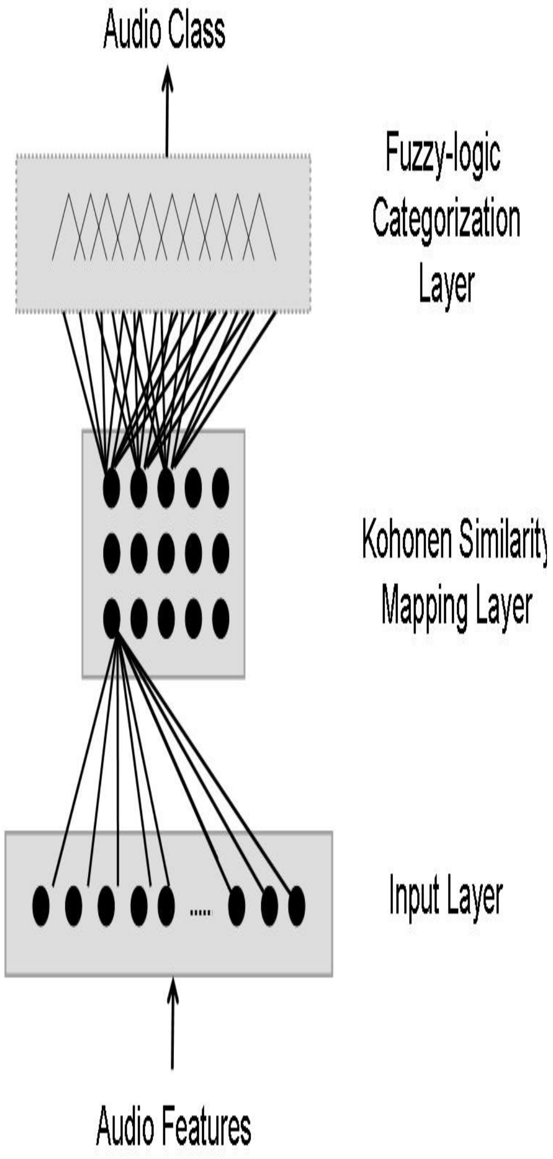
Audio Class
Fuzzy-logic Categorization Layer
Kohonen Similarity Mapping Layer
Input Layer
.....
Audio Features

Fig. 7 Two-dimensional mapping of multi-dimensional features by a Kohonen feature map.

A fuzzy logic upper layer is added to the Kohonen layer. It consists of a fuzzy-logic engine (FLE) tuned to categorize sounds into types.

Tuning the FLE for sound categorization is a challenging issue, because it needs to be general enough to cover all sound types while at the same time simple to tune. For this purpose, a hierarchical classification was chosen because it quickly leads to the appropriate class.

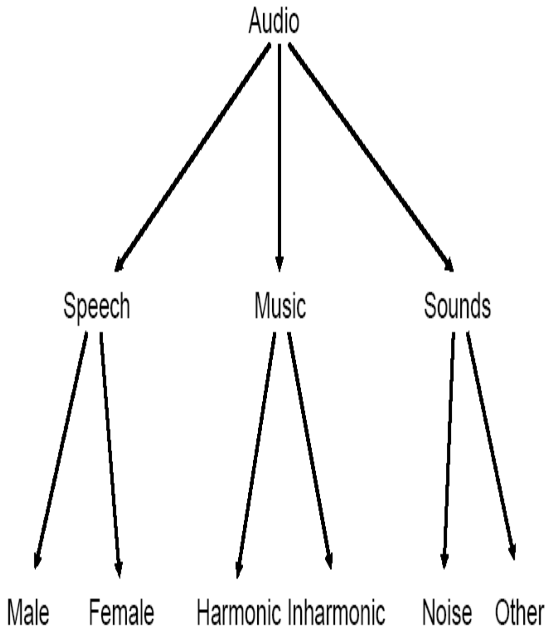

Fig. 8 Hierarchical classification of audio for fast search in multimedia database.

The advantage of the fuzzy logic approach to categorization is that input from different sources can be easily combined. In this specific case, input to the FLE comes from the Kohonen mapping layer and from the audio feature extractor.

One way to fuzzify crisp information from the Kohonen layer and from the audio feature extractor that leads to automatic setup of the membership functions is based on the statistics for each feature and how they are clustered by the Kohonen layer.

A simple way to automatically derive the membership functions to fuzzify input consists of superimposing the membership function on the distribution of the crisp feature measurement, normalizing the crisp feature measurement in amplitude and in range [1][13]. Triangular membership functions, for example, best fit narrow band-spectrum distribution; trapezoidal membership functions can adequately accommodate large band-spectrum distribution; class and categories can be represented by singleton membership functions.

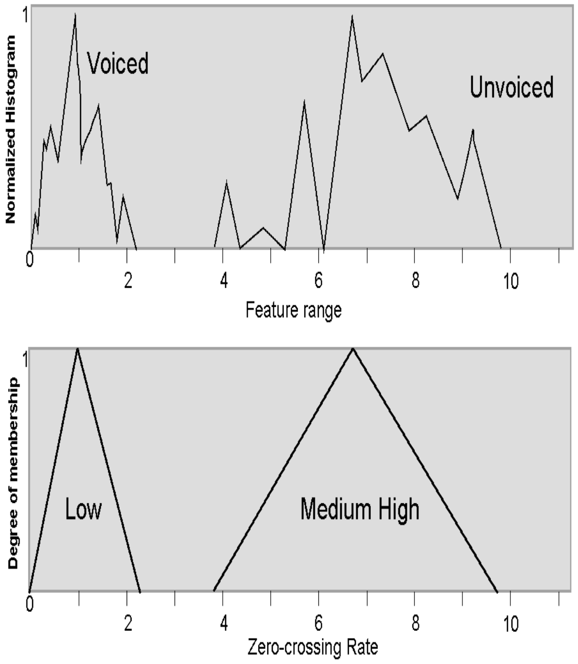

Fig. 9 Membership modeling by superimposition

The same feature clustered by the KFM is also represented as a function of the parameter used to categorize it (e.g. area, centroid position, intensity, etc.). The shape of the distribution of this parameter is then used to model the appropriate membership function.

Fuzzy rules for categorization have the following format:

IF (Condition 1) AND (Condition 2)
THEN (Category)

Condition 1 evaluates audio measurement
Condition 2 evaluates KFM mapping

Condition uses the fuzzy measurement from the membership function: very low, low, medium, high, very high (e.g. *RMS is medium, ZCR is low*, etc.).

One or more rules can be compiled and tuned for each audio category. The result of the evaluation (minimum operator) of every single rule is then OR processed (maximum operator) in one batch to obtain a fuzzy evaluation of the audio object. A singleton function is used to defuzzify each audio object, thus determining its degree of belonging to an audio category.

The fuzzy logic engine needs to be tuned for best performance. Two options are available for the purpose: manual tuning or automatic tuning. Manual tuning relies on an audio expert, who chooses among different membership functions. The audio expert may also create rules for best categorizing audio, based on her or his knowledge. A graphic user interface (GUI) is available for this task.

Automated tuning uses only a triangular membership function to fit the audio-feature distribution shape and fixed format rules. Automatic tuning can also be assisted by a genetic-like process, so that a large number of rules are generated at tuning-time, but only those used most often are kept at run-time.

8 Conclusion

The multi-method approach to accessing multimedia information by means of audio proved to be effective.

Hard-computing methods, such as DTW and HMM, can be very effective in applications like audio spotting. ANNs and FLEs can be integrated with the above, and can even replace them with optimal results.

An optimal solution was shown to be classification by means of Kohonen feature maps, combined with a categorization layer based on a fuzzy-logic engine that had been automatically tuned using feature distribution and class shapes. This solution works well because of its ability to map knowledge into the system even from information that is neither directly measurable by feature-extraction algorithms nor available for hard-computing pattern-matching method.

Audio contains very rich semantics. Therefore, retrieval accuracy is highly sensitive to the effectiveness of feature extraction. Future work will concern the issue of audio-feature extraction with a high degree of semantics.

References:

[1] K. Bosteels, E.E. Kerre, Fuzzy Audio Similarity Measures Based on Spectrum Histograms and Fluctuation Patterns, in *Proceedings of the International Conference Multimedia and Ubiquitous Engineering 2007,* Seoul, Korea, 27-28 April, 2007.

[2] F. de Jong, R. Ordelman, M. Huijbregts, Automated Speech and Audio Analysis for Semantic Access to Multimedia, In: *Y. Avrithis et al. (Eds.), SAMT 2006, LNCS 4306,* Springer-Verlag, Berlin, Heidelberg, pp. 226-240, 2006.

[3] D. Desieno, Adding a Conscience to Competitive Learning, In: *Proceedings of the International Conference on Neural Networks,* Vol.1, 1988, IEEE Press, New York, pp.117-124.

[4] B. Feiten, S. Günzel, Automatic Indexing of a Sound Database Using Self-organizing Neural Nets, *Computer Music Journal*, Vol.18, No.3, 1994, pp. 53-65.

[5] C. Hale and C. Nguyen, *Audio Command Recognition Using Fuzzy Logic,* Wescon 95, San Francisco, CA, November 7, 1995.

[6] N. Kosugi, Y. Nishihara, S.Kon'ya, M. Yamamuro, and K Kushima, *Music Retrieval by Humming,* in Proceeedings of PACRIM'99, pp. 404-407, IEEE, August 1999.

[7] T. Kohonen, The self-organizing map, *Neurocomputing*, Vol. 21, No. 1-3,1998, pp. 1-6.

[8] T. Kohonen, The "Neural" Phonetic Typewriter *IEEE Computer*, Vol.21, No.3, 1988, pp. 11-22.

[9] Y.L.Lin and G. Wei, "*Speech Emotion Recognition Based on HMM and SVM*", in Proceedings of ICMLC 2005Piscataway: IEEE Computing Society, 2005, pp.4898-4901.

[10] M. Liu, C. Wan, L Wang, A Fuzzy Logic Approach for Content-based Audio Classification and Boolean Retrieval, In: *Fuzzy Logic and Internet, V. Loia, M. Nikravesh, L. A. Zadeh, Studies in Fuzziness and Soft Computing,* Springer, Vol.137, 2004, pp. 135-156.

[11] M. Malcangi, Improving Speech Endpoint Detection Using Fuzzy Logic-based Methodologies, in: *Proceedings of the Thirteenth Turkish Symposium on Artificial Intelligence and Neural Networks,* Izmir, Turkey, June 10-11, 2004.

[12] M. Malcangi, Soft-computing Approach to Fit a Speech Recognition System on a Single-chip, in *2002 International Workshop System-on-Chip for Real-Time Applications Proceedings*, Banff, Canada, July 6-7, 2002.

[13] M. Malcangi and Alessandro Nivuori, *Beat and Rhythm Tracking of Audio Musical Signal Processing for Dance Synchronization of a Virtual Puppet",* in Proceedings of the XIV Colloquium on Musica Informatics (XIV CIM 2003), Firenza, Italy, May 8-9-10, 2003.

[14] R. J. McNab, L. A. Smith and I. H. Witten, *Signal Processing for Music Transcription,* Proceedings of the 19th Australasian Computer Science Conference, Melbourne, Australia, January 31–February 2 1996.

[15] R. Neumayer, T. Lidy, A. Rauber, Content-based Organization of Digital Audio Collections, in: *Proceedings of the 5th Open Workshop of Musicnetwork,* Vienna, Austria, July 4-5, 2005.

[16] D. O'Shaughnessy, Speech Communication – Human and Machine, *Addison-Wesley,* Reading, MA, 1987.

[17] G. Richard, M. Goirand, D. Sinder and J. Flanagan, *Simulation and Visualization of Articulatory Trajectories Estimated from Speech Signals,* Proceedings of the International Symposium on Simulation, Visualization and Auralization for Acoustic Research and Education (ASVA97), April 1997, Tokyo, Japan.

[18] C. Spevak, R. Polfreman, in *proceedings of COST G-6 Conference on Digital Audio Effects (DAFX-00),* Verona, Italy, December 7-9, 2000.

[19] Y. Ying and P. Woo, *Speech Recognition Using Fuzzy Logic,* IJCNN '99 – International Joint Conference on Neural Networks, volume 5, 10-16 July 1999, Pages: 2962-2964, vol. 5.

[20] X. Zhao, Y. Zhuang, J. Liu, F. Wu, Audio Retrieval with Fast Relevance Feedback Based on Constrained Fuzzy Clustering and Stored Index Table, In: *Y.C. Chen, W. Chang and C.T. Hsu (Eds.) PCM 2002, LNCS 2532,* Springer-Verlag, Berlin, Heidelberg, pp. 237-244, 2002.

[21] S. Mitra, S. K. Pal and S. Banerjee, *Tuning of Class Membership Using Genetic Algorithms,* In Proceedings Third European Conference on Intelligent Techniques and Soft Computing, (EUFIT'95), pages 1420-1424, Aachen, 1995.

[22] T. Zeppenfeld, A.H. Waibel, "A hybrid neural network, dynamic programming word spotter," ICASSP, vol. 2, pp.77-80, Acoustics, Speech, and Signal Processing, 1992. ICASSP-92 Vol 2., 1992 IEEE International Conference on, 1992.

[23] G. Costantini, D. Casali, *Recognition of Musical Chord Notes* , WSEAS Transactions on Acoustics and Music , Issue 1, Volume 1, January 2004, pp. 17-20.

[24] G. Costantini, D. Casali, *Recognition of Musical Instruments by Statistical Classification* , WSEAS Transactions on Acoustics and Music , Issue 1, Volume 1, January 2004, pp. 21-24.

[25] P. Gardner-Stephen, G. Knowles. DASH: A New High Speed Genomic Sequence Search and Alignment System. WSEAS Transactions on Biology and Biomedicine. January, 2004. Vol. 1, Issue 1, 59-64.

[26] Hadar O., Bykhovsky D., Goldwasser G., and Fisher E., *A musical source separation system with lyrics alignment*, WSEAS Transactions on Systems, Issue 10, Vol. 5, pp. 2464-2467, October 2006.

[27] D. Politis, D. Mrgounakis**,** A synopsis of sound: image transforms based on the chromaticism of music, WSEAS Transactions on Computers, Volume 7 , Issue 8, pp.1113-1127, August 2008.