

DEVELOPING AUDIO PROCESSING AGENTS FOR MULTI-AGENT MPEG-7 ENABLED ENVIRONMENT

Mingkun Li⁺, Gang Wei^{*}, Valery A. Petrushin^{*}, Ishwar K. Sethi⁺

⁺Department of Computer Science and Engineering
Oakland University
Rochester, MI 48309, USA
{li,iseti}@oakland.edu

^{*}Accenture Technology Labs
161 N. Clark Street, Chicago, IL 60601, USA
{gang.wei,valery.a.petrushin}@accenture.com

ABSTRACT

This paper presents a methodology for developing audio processing agents for a multi-agent environment that is known as the Community of Multimedia Agents. The Community's philosophy, objectives and architecture are described. The methodology is illustrated using audio feature extraction agents as example. The algorithms used for extracting audio features are classical and work in general audio domain. The agents have the standard MPEG-7 interface for better interoperation and wide usage. Two low level tools - the MPEG-7 audio descriptor wrapper classes and the MPEG audio decoder - are also presented. An example of agent aggregation for an annotation system prototyping is provided.

1. INTRODUCTION

Multimedia data is growing explosively and the techniques to find the desired content is lagged far behind despite of the amount of researcher efforts. Challenges in video, audio and image content annotation are making the collaboration between researchers increasingly important, which is however held back by a number of factors, such as the variety of platforms, programming languages, and data exchange formats and unwillingness of corporations to distribute their intellectual property unprotected. For example, when one wants to build a multimedia analysis system that requires face detection, very likely he or she will end up re-implementing some existing algorithm or creating his/her own, even though dozens of them have already been developed. On the other hand, when one invents a new tool, the chances of it to be used by others depend heavily on its exposure to web search engines and

other indexing tools. Academic researchers would usually be happy to get ready their tools for sharing if it does not take too much effort. This situation triggered the Community of Multimedia Agents project at the Accenture Technology Labs [1].

2. THE COMMUNITY OF MULTIMEDIA AGENTS

The Community of Multimedia Agents is a virtual community of researchers and students, who exchange their multimedia analysis tools and test data. The Community's objective is to improve the cross-organization collaboration, consolidate efforts and expedite research and education in the field of multimedia analysis and annotation.

The Community's organization is based on the following principles.

- **Web-based Portal.** The Community is located on the World Wide Web at <http://community.techlabs.accenture.com> and accessible from any Internet-able workstation. Beside the tools (see below) the Community Portal provides information about related conferences and workshops, business and academic news, links to related projects, book and paper recommendations, tutorials, etc.
- **Free Membership.** The Community's membership is free.
- **Media Library.** The Community provides links to media files that could be used for multimedia annotation research. The Community members provide these links to the collections of media data. The copyright belongs to the authors of the collections or/and their organizations

- **Agent Library.** The Community provides a library of multimedia analysis and annotation tools in a form of agents. An agent is a relatively simple module that does a particular task, such as a feature extraction, an object detection or classification. Agents are represented in an executable form (executable programs, compiled classes, etc.), thus protecting the proprietary details of agents' design and the agent author's intellectual property. Any community member can submit an agent and download the agent library. All rights for a particular agent belong to the agent's authors or their organizations. The community members are granted a license for free non-commercial usage of the agent library.
- **MPEG-7 Interface.** The agent output could be a transformed media file or an annotation file. The annotation outputs are represented in MPEG-7 language. MPEG-7 or Multimedia Content Description Interface is an ISO/IEC International Standard 15938 [2]. It is an XML-based language that provides a rich set of descriptors and description schemas for annotating, indexing and organizing multimedia documents, such as videos, audio recordings, still images, electronic ink, text, and their combinations. The Community provides templates for agents' outputs that facilitate both agent development and communication among agents and allow building hierarchies of agents.
- **Development Tools.** The Community provides open source tools for creating agents and visualizing their performance. These tools can be freely downloaded from the Community Web site. The tools include:
 - The agent development tools that consist of agent developing classes for a specific programming language (C++, Java, Perl) and low-level routines, such as MPEG video and audio decoders, filtering, etc.
 - The agent aggregation and visualization tools that serve for rapid prototyping media analysis and annotation systems using agents as blocks.

The architecture of the Community's is presented on Figure 1. The Development Environment Tool consists of two major parts: the Graphical Agent Composition Workbench and the Agent Result Browser (Blackboard). The Workbench allows a user to select and combine existing agents as building blocks to construct multi-agent systems. The Blackboard Browser visualizes the results produced by the agents.

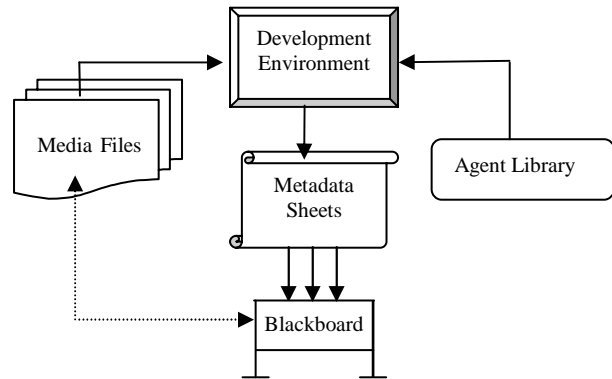


Figure 1. The Community's architecture.

3. AGENT DEVELOPMENT

To facilitate agents development the following methodology has been proposed. We assume that a researcher already has an algorithm and the problem is to convert it into an agent. If the algorithm is implemented as an executable program, then the researcher has to create a "wrapper" program that prepares data for the processing program, calls the processing program and converts resulting files into MPEG-7 annotation files. If the algorithm is implemented as a procedure then the researcher has to create an executable program that reads input data, calls the procedure and generates results. In any case an agent consists of the following blocks (Figure 2):

- Reading and decoding raw data (signal) and annotations (features). The complexity of this step depends on the complexity of data coding method, for example, such coding standards as MPEG-1, MPEG-4 or JPEG are rather sophisticated.
- Do pre-processing, such as filtering, normalizing or pre-emphasizing the signal, or re-arranging annotations.
- Do the main agent's function, such as face detection in image or gender identification in audio stream.
- Do post-processing, such as calculate feature statistics that is required by the MPEG-7 standard.
- Write transformed media data and/or annotation files.

The Community provides tools that help the researcher to accomplish all data input/output and pre- and post-processing tasks and allows the researcher focusing on the main function of the agent, which is depicted by a shaded box on Figure 2.

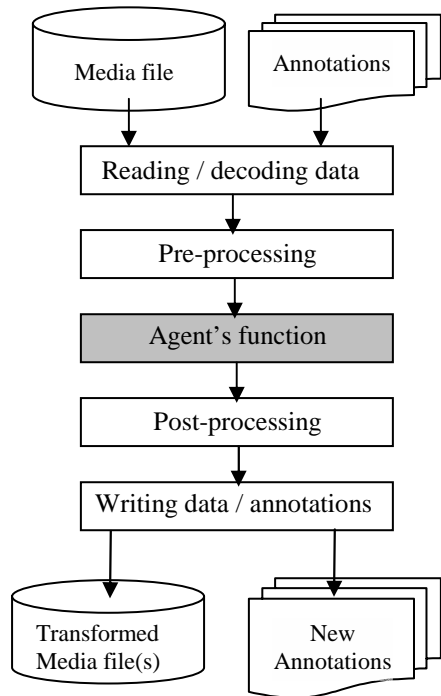


Figure 2. Agent's structure.

4. AUDIO AGENTS

4.1. MPEG-7 Audio Descriptions

The MPEG-7 standard contains of a number of documents that specify descriptors and description schemas that cover different media features, different aspects of media temporal and spatial structures, and media usage. The detailed information about MPEG-7 standard can be found in [2,3,4,5].

The MPEG-7 audio standard defines the structures to describe the audio content. It has two sets of descriptors. The first set consists of low level and generic features, which can be extracted directly from audio samples and used in a wide range of applications. The second set defines some high level tools aiming for specific applications like sound recognition, music indexing and retrieval, and speech recognition.

The low level descriptors are divided into six groups called MPEG-7 Audio Framework, which depict audio sample temporal, parametric, and spectral properties. The first group is called basic and includes audio waveform and audio power descriptors. The second group (basic spectral) contains the descriptors for the audio spectral features like spectrum centroid, spectrum spread, spectrum envelope, and spectrum flatness. Signal parameter descriptors include audio harmonicity and audio fundamental frequency. Two groups – timbral spectral and

timbral temporal – provide tools for describing musical timbre spectrum and temporal properties. Spectral basis gives the low dimensional representation of audio spectrum by providing the basis and projection. The low level descriptors have one useful and simple semantic feature called silence descriptor, which denotes whether an audio segment is classified as silence or not.

The high level tools target five common and established applications, namely, sound recognition, audio matching, timbre matching, melody matching, and spoken content. They provide a set of canonical and condensed descriptors to facilitate interoperation and interchangeability. The audio signature scheme, which uses audio spectral flatness, can be used for robust audio matching. Timbre description tools standardize the description of two classes of musical instrument sounds: harmonic sustained sound (the *HarmonicInstrumentTimbre* Descriptor) and non-sustained percussive sound (the *LogAttackTime* Descriptor). Melody description tools have a set of melody features to support melody matching and representation. MPEG-7 also provides a set of tools for general sound recognition and indexing, such as music and speech classification, genre classification, etc. These tools include statistical models (the *SoundModel* Description Scheme), classification schemes (the *SoundClassificationModel* Description Scheme), and state paths (*SoundModelStatePath* Descriptor) that are nothing but sequences of states. Recognizing that the speech is of primary importance in audio and that current automatic speech recognition is inadequate, MPEG-7 offers a set of tools to help index and retrieve speech audio by recording the speaker information and intermediate automatic speech recognizer results.

4.2. Tools for developing audio agents

The Community's tools for developing audio agents include a MPEG audio decoder class, several audio descriptor wrapping classes and auxiliary procedures for pre- / post-processing audio signal.

The MPEG audio decoder class is based on the Microsoft DirectX API. A component of DirectX called DirectShow provides high-quality capture and playback of multimedia streams. Even though DirectShow was not design for MPEG audio decoding, it is possible to use it for developing an MPEG audio decoder. The trick is to develop an audio capture class to intercept the audio samples from the commercial MPEG players. The resulting decoder can uncompress many audio compression formats, such as MPEG layers 1, 2 and 3. In general, it can get audio samples from any audio files that can be played back in the computer.

The audio decoder class includes functions that allow getting information about audio format, controlling the audio streams and accessing the audio samples both in direct and sequential mode.

The audio descriptor wrapper classes provide tools for reading and writing MPEG-7 annotations according to the Community's templates. Using these classes assures the compatibility of agents' outputs and makes the programmer's work more efficient excluding such tasks as memory management and detailed replication of MPEG-7 format. Additionally, following the object-oriented design methodology allows the users building other audio descriptor wrapper classes using the provided basic classes.

MPEG-7 annotation files can be divided into two parts. The first part contains the media and content management information, including media location, media format, and media creation information. The second part contains the content description, which contains different kinds of descriptors and description schemas. The descriptors have a similar structure. The first part contains the descriptor name and the parameters used to compute the descriptors. This part is descriptor dependent. The second part gives the data, which is either a series of vector or a series of scalar. Considering the specific structure of MPEG-7 files, a basis class called *CAudioSegment* has been developed. It encodes all the header information, and every refined class inherits it. We also have two data container classes – *CSeriesOfScalar* and *CseriesOfVector* – to perform general data reading and writing. Thus, each descriptor wrapper class inherits from *CAudioSegment* class and contains either a *CSeriesOfScalar* or a *CSeriesOfVector* member. In addition to the inherited *CAudioSegment* class and the *CSeriesOfScalar/Vector*, a descriptor wrapper class reads and writes according to its specific parameters (usually, a class has two or three parameters).

Currently, seven audio wrapper classes have been implemented. For example, *CAudioWaveform* and *CAudioPower* classes wrap the MPEG-7 audio waveform and audio power descriptors respectively. Some wrapper classes for widely used features, such as mel-frequency cepstral coefficients (MFCC), zero crossing, and formants, have been implemented (in spite that they are not defined in the MPEG-7 standard).

4.3. Audio Feature Extraction / Classification Agents

Using the above-described development tools several feature extraction and classification agents have been implemented. Table 1 presents the list of currently implemented audio agents.

The first three agents work in the time domain and calculate features and their statistics for signal waveform, power, and zero crossing. Using the length of sliding window as a parameter, the audio waveform agent extracts samples of raw signal and statistics. This data can be used to display the signal. Audio power is the sum of squared

sample values in the window. Zero crossing is defined as the number of sign changes in the processing window and serves as a rough estimate of the frequency content of a speech signal. All three agents can extract raw data and/or statistics, such as minimum, maximum, mean, variance, or select a representative sample using random, first or last modes as specified in the MPEG-7 standard.

Table 1. MPEG-7 audio agents

Feature extraction agents	Audio waveform agent Audio power agent Audio zero crossing agent Audio fundamental frequency agent Audio formants agent Audio MFCC agent
Classification agents	Audio silence detector Audio gender classification

Fundamental frequency, formants are important suprasegmental features that are widely used to perform speech classification and speech synthesis tasks. The classical autocorrelation method for computing fundamental frequency has been implemented due to its robustness and accuracy [6]. The well-known approach that is based on finding the roots of linear prediction polynomial has been chosen for estimation formant and their bandwidths [7].

Mel Frequency Cepstral Coefficients (MFCC) are widely used in speech recognition and speaker identification systems [8]. MFCC uses Mel scale to simulate the human auditory system frequency selection capability. For estimating MFCC, first, the triangle filter bank analysis on the Mel frequency scale is performed and, then the filter values are converted to MFCC using discrete cosine transform (DCT) [9].

Silence is a natural choice for audio stream partition. The first step in any audio indexing and retrieval system is to perform silence detection. The simplest silence detector is built using a threshold for audio power values. More advanced silence detector can use both power and zero crossing features.

5. AGENT AGGREGATION

The principal idea that lies behind the Community software architecture is that we can build content annotation systems using a hierarchy of agents. Low-level agents could be rather simple feature extraction or classification agents that can be used as modules to build more sophisticated agents. The Community provides a user-friendly tool to integrate the agent together and

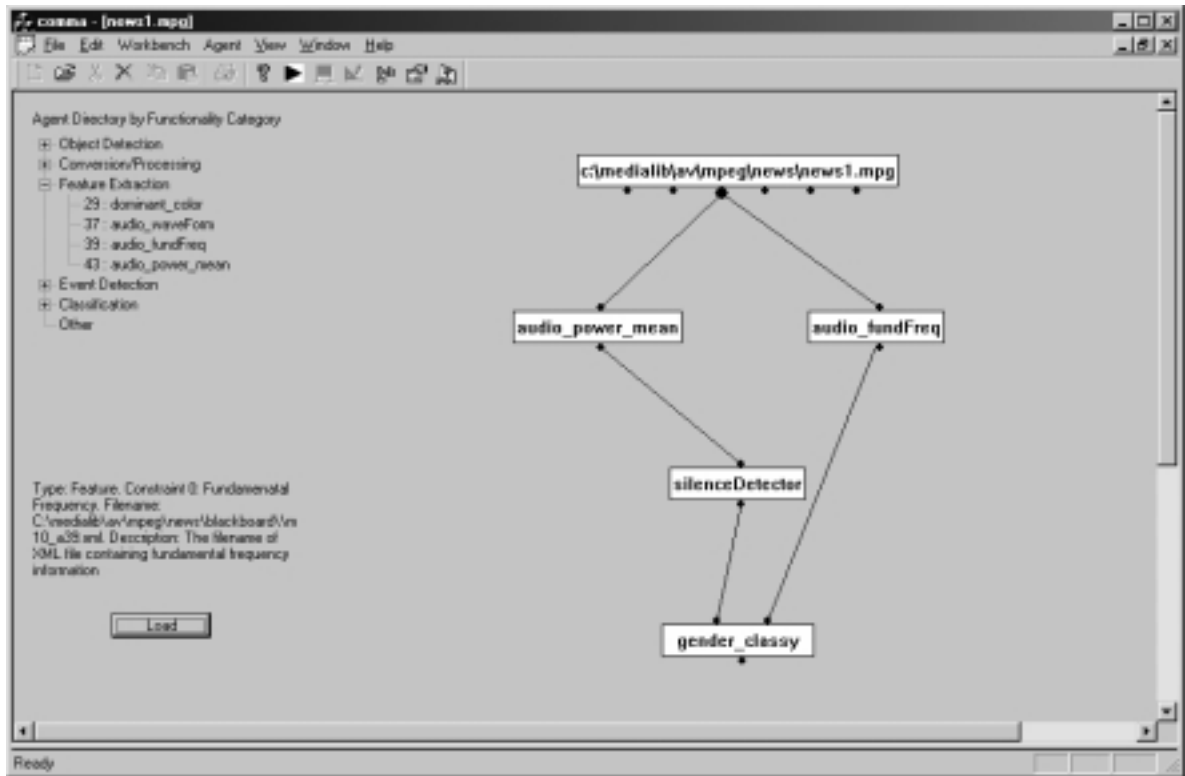


Figure 3. Constructing the speaker gender classification prototype using the Workbench.

visualize their results. In this section we shall give an example how to use these tools for building a prototype for a speaker gender classification system, i.e. the system that identifies the gender of speakers in an audio clip. Statistically, female voices have higher pitch (fundamental frequency) than male voices. Let us make a prototype of a system that exploits this difference in mean pitch values just to test the idea. A sketch of the system's algorithm looks like below:

1. Calculate the mean audio power value for each 10 ms window of the audio clip.
2. Use a threshold on the mean power values for the windows. Windows with power higher than the threshold are labeled as non-silent audio intervals; otherwise labeled as silent intervals.
3. Calculate the mean fundamental frequency value for each 10 ms window of the audio clip
4. Use a threshold on the fundamental frequency values for the windows within the non-silent audio intervals. Windows with mean fundamental frequency that is higher than the threshold are classified as female speech; otherwise they are classified as male speech.

Without using the Community's tools the researcher may have to develop modules to calculate the audio power and fundamental frequency, which may take a while. However, as the Community provides agents and related tools, the process of building the prototype could be made much

easier. Step 1 can be done by using the audio power agent. The audio fundamental frequency agent covers step 3. Suppose, we don't have an agent that does silence detection. Therefore, the researcher needs to write two programs. The first one takes the result of the audio power agent and performs simple threshold operations to decide if the audio signal in the window represents silence or not. Because the output of the audio power agent conforms to the MPEG-7 template, and the Community provides open-source tools to handle the templates, the programming effort is minimal. Let us call this program silence detector agent. Similarly, the second program takes the results of both the fundamental frequency agent and the silence detector agent as input, and decides if the audio windows are classified as male or female speech based on the mean of fundamental frequency values of the windows and the threshold value. Let us call this program a gender classification agent.

If we compare the efficiency of developing the system prototype using the Community's tools to doing the development from scratch, we can see that the source codes of both the silence detector agent and the gender classification agent takes less than 100 lines. This significantly reduces the time and efforts need to build an audio analysis system prototype. Therefore, the Community improves the efficiency of research. When the agents are coded the researcher can integrate them using the Workbench (Figure 3).



Figure 4. Visualizing Results in the Blackboard Browser.

The user starts using the Development Tool with selection of a multimedia file or a collection of files to be processed. The media file is represented as a rectangle with a number of dots at the bottom. The largest dot corresponds to the raw media data. The other smaller dots, if any, are the processing results previously produced by agents. Those results are archived and can be used as inputs to other agents to avoid repeated computation and significantly reducing overhead, especially for time-consuming video processing algorithms. Agents are displayed in a tree structure based on their operation types in the left corner of the window. The user can load agents to the Workbench and connect them together to build the audio analysis process. For example, as shown on Figure 3, to tell the gender of the speakers in the input audio clip, the user can first load the audio power agent (*audio_power_mean*) to calculate the mean power of the audio windows. Then the fundamental frequency agent (*audio_fundFreq*) is loaded to calculate the fundamental frequency. Then the user adds the silence detector agent (*silenceDetector*) and the gender classification agent (*gender_classify*). After this the user links the media data and the agents' input/output pins to create a prototype. Finally, the user can click the Run button (black triangle) to start the process. The Workbench will ensure that the agents are invoked in the correct order. As shown in this example, the researcher can create the audio processing method just like drawing a flow chart, which is very straightforward and intuitive.

The Development Environment tool also contains an Blackboard Browser to visualize the results produced by the agents. Figure 4 shows a Blackboard window for a video file. It allows the user to view the agent results at different granularities. The summary of each agent's findings is represented as a colored horizontal bar aligned with the timeline, with the segments painted in different colors based on the results from the corresponding frames. For example, the result of the gender classification agent (*gender_classify* in Figure 4) is represented as the bar at the bottom of the Blackboard Browser Window, where the segments corresponding to male speech are painted in yellow, while those for female speech are painted in red. This gives the user an insightful view of the result of all agents.

6. SUMMARY

The Community of Multimedia Agents is a community of researchers and an open environment that allows researchers to share their achievements in multimedia annotation field while protecting their intellectual property. Based on MPEG-7 standard, the Community allows agents created by different researchers to communicate and collaborate with each other to build more intelligent and robust multimedia content analysis and annotation processes. The paper presents the audio processing agents of the Community and illustrates how a

researcher can take advantage of the agents and the tools to accelerate prototyping and testing new audio analysis algorithms.

The Community's short-term objective is to create the "critical mass" of agents and a set of reliable tools to be useful for researchers and students. We are currently collaborating with several research labs and Universities to develop more agents.

7. REFERENCES

- [1] Wei, G., Petrushin, V. A., Gershman, A. V., From data to insight: the community of multimedia agents, *3rd Intl. Workshop on Multimedia Data Mining in Conjunction w/ The 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery & Data Mining*, July 2002, Edmonton, Alberta, Canada
- [2] Martinez, J.M., MPEG-7 Overview.
<http://mpeg.telecomitalia.com/standards/mpeg-7/mpeg-7.htm>
- [3] Martinez, J.M., Koenen, R., Pereira F., MPEG-7: the generic multimedia content description standard, part 1, *IEEE Multimedia*, v9, n2, pp. 78-87
- [4] Martinez, J.M. Standards - MPEG-7 overview of MPEG-7 description tools, part 2, *IEEE Multimedia*, v9, n3, p83-93
- [5] B.S. Manjunath, Ph. Salembier, and Th. Sikora (Eds.) *Introduction to MPEG-7 Multimedia Content Description Interface*. John Wiley & Sons, New York: NY, 2002.
- [6] Boersma, P., Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound, *IFA Proceedings 17*, 1993
- [7] Atal, B. S., and Hanauer, S. L., Speech analysis and synthesis by linear prediction of the speech wave, *The journal of the acoustical society of America*, **50**, No. 2, August 1971
- [8] Li, D., Sethi, I., Dimitrova, N., McGee, T., Classification of general audio data for content-based retrieval, *Pattern recognition letters*, **22** (2001), pp 533-544
- [9] John R. Deller, J. G. Proakis, and J.H.L. Hansen, *Discreet-time processing of speech signals*, Macmillan, 1993.