

FROM DATA TO INSIGHT: THE COMMUNITY OF MULTIMEDIA AGENTS

Gang Wei
Accenture Technology Labs
161 N. Clark Street
Chicago, IL 60601
gang.wei@accenture.com

Valery A. Petrushin
Accenture Technology Labs
161 N. Clark Street
Chicago, IL 60601
valery.a.petrushin@accenture.com

Anatole V. Gershman
Accenture Technology Labs
161 N. Clark Street
Chicago, IL 60601
anatole.v.gershman@accenture.com

ABSTRACT

Multimedia Data Mining requires the ability to automatically analyze and understand the content. The Community of Multimedia Agents project (COMMA) is devoted to creating an open environment for developing, testing, learning and prototyping multimedia content analysis and annotation methods. It serves as a medium for researchers to contribute and share their achievements while protecting their proprietary techniques. Each method is represented as an agent that can communicate with the other agents registered in the environment using templates that are based on the Descriptors and Description Schemes in the emerging MPEG-7 standard. This allows agents developed by different organizations to operate and communicate with each other seamlessly regardless of their programming languages and internal architecture. A Development Environment is provided to facilitate the construction of media analysis methods. The tool contains a Workbench using which the user can integrate the agents to build more sophisticated systems, and a Blackboard Browser that visualizes the processing results. It enables researchers to compare the performance of different agents and combine them to build more powerful and robust system prototypes. The COMMA can also serve as a learning environment for researchers and students to acquire and test cutting edge multimedia analysis algorithms. Thus the efficiency of research in this area can be improved by sharing of media agents.

KEYWORDS

Multimedia content analysis; Agent; MPEG-7; XML Schema

1. INTRODUCTION

The extraction of information from multimedia data is of vital importance with the explosive growth of digitized image, audio and video data. It requires the ability to automatically analyze, understand and annotate multimedia content. A large number of approaches have been proposed in this area, ranging from simple measures like color histogram for image, pitch/energy for audio signal, to more sophisticated systems like emotion recognition in audio [1],

and automatic summarization of TV programs [2] and topic detection and tracking using audio transcripts [3]. However, the capability of the current techniques is still far from the requirement of many applications in practice, especially in term of intelligence level and robustness. For example, even the most advanced face recognition algorithms can easily be fooled by a little makeup or environmental changes. Those challenges are calling for the consolidation of the research efforts in this area. We believe that the reliable understanding of multimedia content has to be achieved by the interaction of a number of specialized, effective and relatively primitive modules (agents) that address different aspects of the content. A number of research efforts have been made in this direction, producing encouraging results, such as the TV genre classification based on face and superimposed text detection in [4], and the use of both audio and video information to analyze multimedia content [5]. To enable the cross-organization sharing and integration of agents, three major issues need to be addressed. First, the data format between the agents should be compatible to allow communication with each other. The coming standard Multimedia Content Description Interface (MPEG-7) [6] promises to provide a unified base for multimedia content description for both producers and consumers. Second, agents should not expose the proprietary techniques of the inventors. Finally, a development environment is needed to facilitate the manipulation of the agents and visualization of the processing results.

Agents are defined as active, persistent software components that perceive, reason, act, and communicate [7]. Agent-based approach proved to be very useful in many applications. We found that the concept of agent is highly valuable for multimedia analysis. Most of the multimedia processing systems uses agents (in the above mentioned sense) implicitly or explicitly [8, 9].

2. MOTIVATION

Multimedia content analysis requires expertise in a number of fields such as image and video processing, audio processing, speech recognition, linguistics, information retrieval and knowledge management. The range of

expertise spans from DSP techniques for feature extraction to methods for knowledge representation, integration and inference. Unlikely a researcher or a research laboratory can cover the required range of expertise to develop a multimedia analysis system from scratch. Usually, each lab concentrates on its own research agenda using commercial tools (if available) or borrowing some experimental tools from other researchers to develop a rounded-up multimedia analysis prototype. Borrowing from the others is not easy due to the variety of platforms, programming languages, data exchange formats and unwillingness of companies to disseminate their intellectual property unprotected. A lucky researcher can get a tool that covers a particular task, for example, face detection; an unlucky researcher has to implement a tool by himself. In any case, the researcher will have only one (or two, if any) face detector, in spite of his awareness that two dozens of such tools exist in the world. This scarcity of media analysis tools and difficulty finding them motivated our COMMA project. The project's general objective is to create a virtual community of researchers, who exchange their multimedia analysis tools and test data. The Community's objective is to consolidate efforts and expedite research and education in multimedia analysis. To facilitate exchanging and combining media analysis tools the following requirements are held:

- The Community provides a library of multimedia analysis agents. Any community member can submit and download agents.
- Agents exist in formats that can be directly used as modules to build larger systems, however the proprietary techniques are hidden from the user.
- Copyrights belong to the agents' authors or their organizations.
- The Community is located on the World Wide Web and agents are program-accessible from any Internet-able workstation.
- The Community provides templates for agents' outputs that facilitate communication among agents and allow building hierarchies of agents.
- The Community provides open source tools for creating agents and visualizing their performance. These tools can be freely downloaded from the Community Web site.

Currently we foresee the following stages in developing the COMMA project.

Stage 1. Simple Agents. Agents at this stage perform the tasks assigned by the human users. The objectives is to:

- Develop tools for creating agents and visualizing their work.

- Create the development environment. Users can deploy agents and build more sophisticated high-level systems by connecting them together.
- Develop templates for the communication between agents' based on MPEG-7.
- Accumulate initial "critical mass" of agents.

Now the Accenture Technology Labs have released a first version of the agent development and visualization tools for Windows 2000/XP platform. And we collaborate with several Universities to create an initial library of agents. After this we shall launch the Community's Web site.

The Community at this stage can serve to both researchers and students. A researcher can compare his/her approach to the known approaches presented in the agent library, combine agents to create a high-level agent, and do a rapid prototyping of a system that solves a particular problem. A student can learn about different approaches to solve a problem, get experience in building media analysis algorithms and systems, and learn up-to-date data representation technologies, such as XML and MPEG-7.

Stage 2. Intelligent Agents. Agents will not only be able to act on assigned tasks, but also automatically synthesize by themselves to solve a specified problem. This will require the description of the agent at the knowledge level, and we plan to use techniques such as Resource Description Framework (RDF) as in [10] or the emerging DARPA Agent Markup Language (DAML) as in [11] to represent the ontology of the agents.

Stage 3. Distributed Agents. The further step is to develop formal specifications, interfaces and tools that allow distributed agents to find each other on the Web to communicate and solve a specified problem. At this stage the Community of researchers will be extended to the Community of Multimedia Agents to justify the title of the project. Some research steps have been made in this direction for simple business-oriented agents [12].

3. ARCHITECTURE

Figure 1 shows the architecture of the system. The Community of Multimedia Agents provides the user two components: the Agent Library and the Development Environment. The agent library contains a set of agents, preferably in executable form and an agent description file, which describes the set of agents in XML. The Development Environment is an application for Windows ME/2000/XP platforms. It consists two parts, namely the Workbench and the Blackboard Browser, responsible for the creation of multimedia analysis processes with agents and the visualization of the results, respectively. The user provides the multimedia files to be processed. Three types of media are allowed: still images, audio files, and video files. Each media object is associated with a "Metadata

Sheet” in XML format, which is a directory of the processing results produced by the agents. When an agent is applied to the media file, the Workbench updates the corresponding Metadata Sheet by adding a record. The Blackboard visualizes the results to the user by the interpreting of the Metadata Sheet.

To start using the system a COMMA member should download the Development Environment application and the agents to a local computer. Then the user can build multi-agent media analysis processes in the Workbench by loading media files and connect agents.

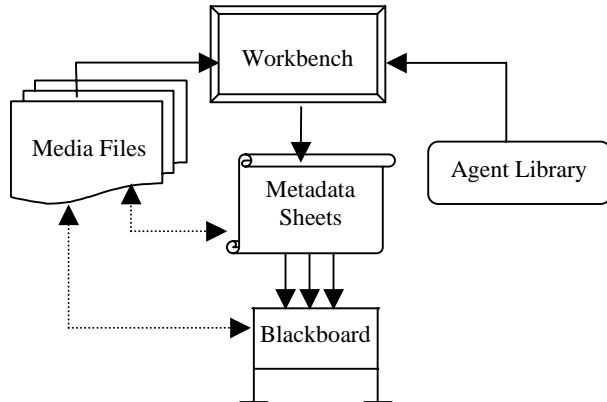


Figure 1. The COMMA Architecture.

4. AGENT LIBRARY

COMMA provides a library of multimedia processing and analysis agents that serve as building modules for more sophisticated, powerful and robust systems. Each agent exists as an individual executable application developed by different researchers and organizations. To enable the agents to communicate and collaborate with each other, we defined the specifications of the agent interface and the XML-based schema for agent description.

4. 1. Agent Interface

The agent interface specification includes two aspects, namely the *syntactic* interface and the *signature* interface. The former addresses the lower-level “physical” characteristics of the agents. The signature interface, in contrast, represents relatively higher-level features of the data to be processed or results that are produced by the agents.

The syntactic interface requires each agent to be an application that can be invoked through a command line, e.g., a console executable program. Any programming language can be used for developing an agent. The system allows also using any interpretive language for agent development, but installation of the interpreting program should be done separately.

Seen at the signature level, an agent in COMMA is a filter that either takes the raw data of the media directly or the

processing results produced by other agents as input, and generates its own processing results that can be used for the possible consumption by other agents. As shown in Figure 2, the signature interface of an agent contains three visible parts, namely Input Pins, Output Pins and Tuners.

An agent must have one or more input pins and output pins for data flow. There are different types of pins depending on the natures of the data. For example, if an agent performs face detection on MPEG video, it has one input pin of type “MPEG” and an output pin of type “Visual Object Information”. Pins of the same type are considered to be compatible with each other. In the Workbench, the user can build multi-agent systems by connecting the input pin of one agent to a compatible output pin of another agent. Thus the agents can collaboratively process the media content by sharing data. We created templates for the data format different pin type based on MPEG-7 standard so that agents with compatible pins can communicate with each other.

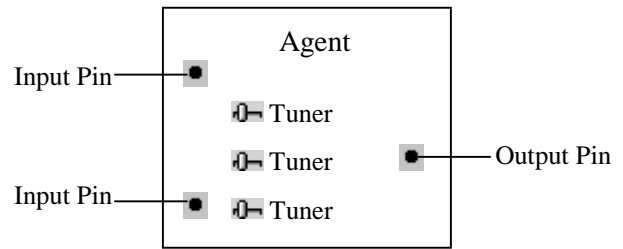


Figure 2. Signature Interface of an Agent

Tuners are used for adjusting technical configurations of agents to give them flexibility. An agent may include zero or more Tuners. Each tuner has a default value recommended by the inventor of the agent to ensure good performance in general cases, while the users can change it to meet their particular needs. For example, when a researcher designs an agent that detects traffic signs on the road for driving assistance, he may prefer to have a balanced recall (the ratio of detected signs among all signs) and precision (the ratio of real signs among all claimed signs), while in practice it is usually desirable to detect as many sign as possible, even though at the cost of producing more false alarms.

4. 2. Agent Description

The executable agents are not self-describing, and thus for the Development Environment to know how to manage them, we defined the XML schema to describe their characteristics, under which the agents are represented in a formalized way understandable not only to human users, but also to the Development Environment.

The organization of the Agent Description Schema is presented in Figure 3. Under the schema, each agent has a

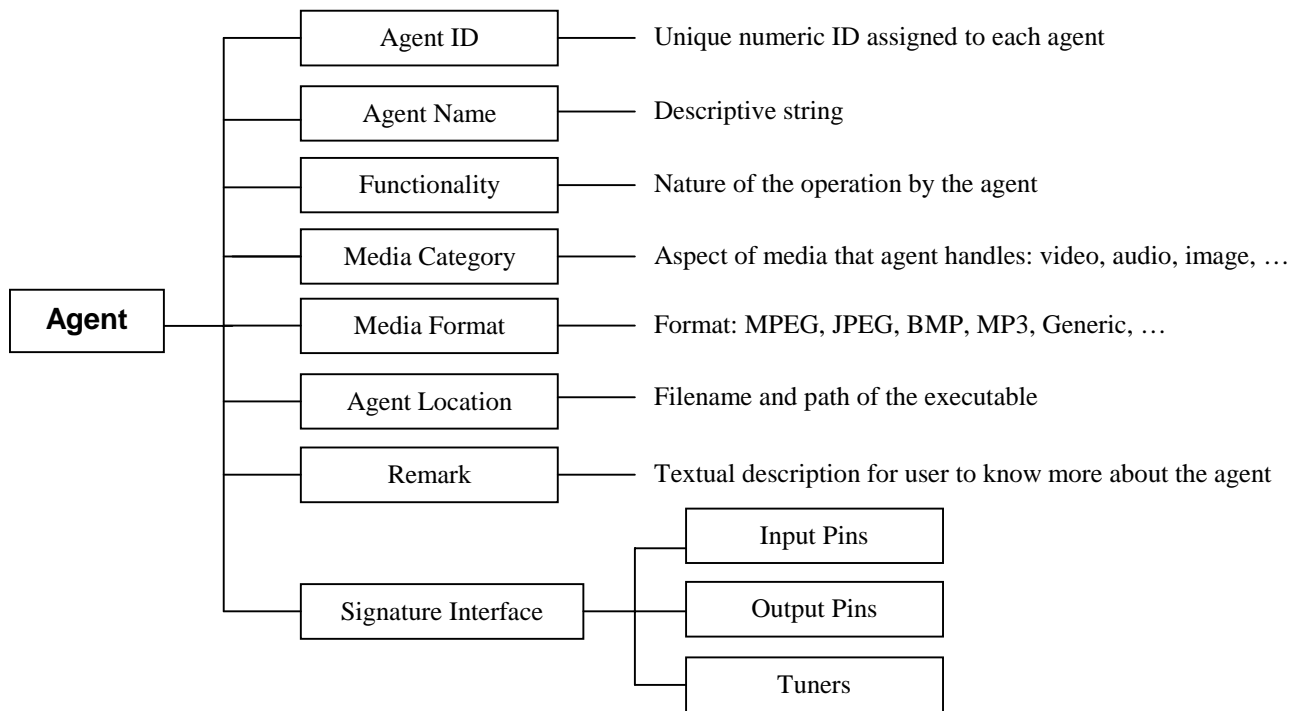


Figure 3. Major Components of the Agent Description Schema

unique numerical ID for retrieval purpose. Other major elements include *Functionality*, *Media Category/Format*, *Agent Location* and *Remark*. The *Functionality* is based on the nature of the operation conducted by the agent, e.g., classification (assign media data into predefined categories), event detection (find certain events in video or audio segments) and object tracking. The *Media Format* attribute indicates what formats of the media files can be processed by the agent, such as MPEG, AVI, BMP, or WAV. The *Media Category*, in contrast, illustrates the general aspect of media the agent deals with, e.g., video, audio or image. For example, consider two agents that both apply to MPEG clip. The first one classifies the camera motion and the second one performs speech recognition. The *Media Category* of the first agent is “video” while that of the second one is “audio”. The *Agent Location* is the path and filename of the executable file corresponding to the agent. The *Remark* attribute provides a brief introduction about the agent in plain words to let the user know about the agent in a more natural way. The agent description schema also includes the signature interface, including the input, output pins and the tuners, which has been mention above. Each agent is represented as an XML node in the agent directory. The Development Environment of COMMA contains a GUI tool through which the agent contributor can register new agents by filling out a form. The tool automatically encodes the information provided into the XML description.

5. DEVELOPMENT ENVIRONMENT

The Development Environment provides means for registering media files and agents, and two major tools: a *Workbench* for developing media annotation processes, and a *Blackboard Browser* for visualizing results.

5.1. The Workbench

The Workbench allows a user to select and combine existing agents as building blocks to construct multi-agent systems. The user starts by selecting a media file. 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 recorded in the Metadata Sheet for the media file and can be used as inputs to other agents to avoid repeated computation and significantly reducing overhead, especially for time-consuming video processing algorithms. The Workbench filters the agent library and displays only the agents that can process the media. The agents are organized by their functionality in a tree structure in the top-left area as shown in Figure 4. The user can load an agent to the working space by highlighting it and clicking the “Load” button. Each agent is represented as a rectangle with input and output pins displayed as dots at the top and bottom, respectively.

The user can build media annotation processes by connecting the media and agents. Figure 4 gives an example

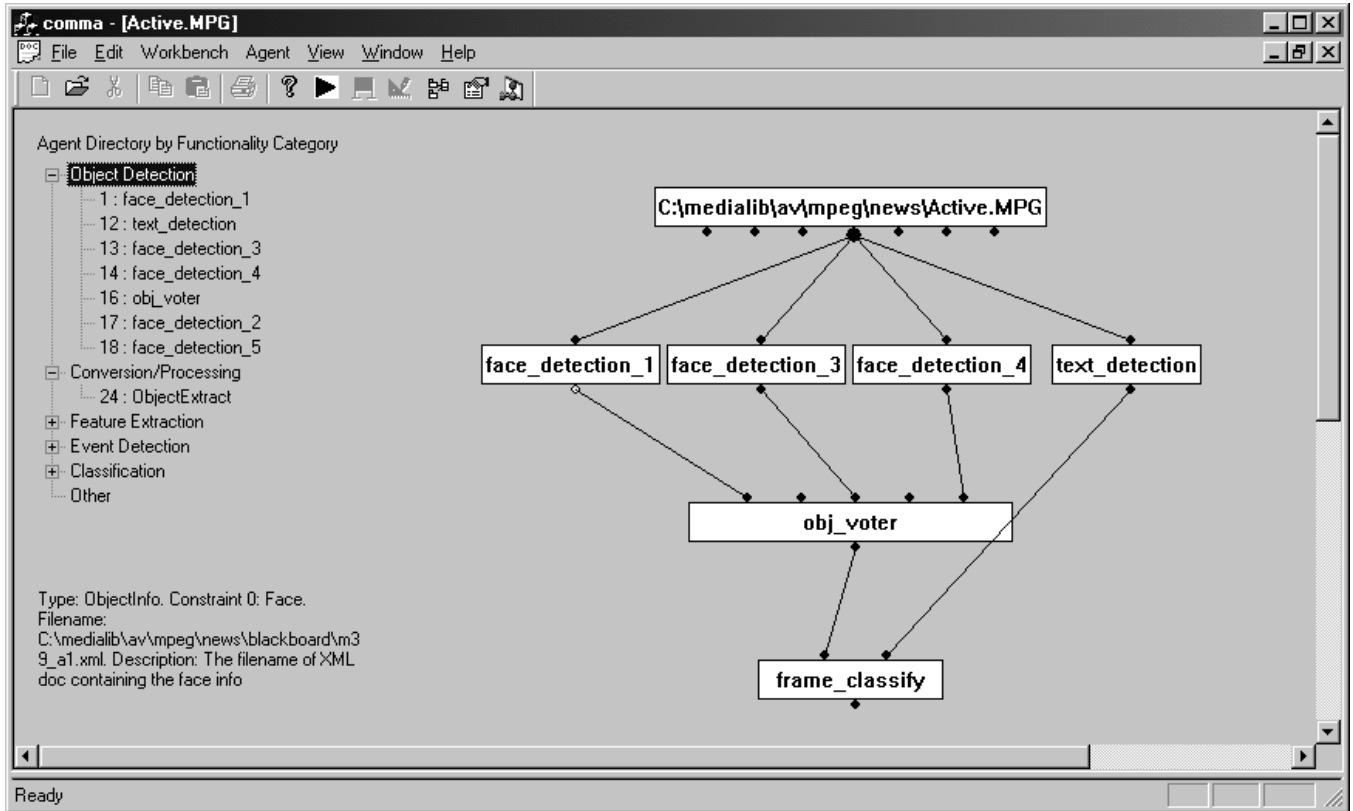


Figure 4. Working in a workbench window.

of integrating agents to build more intelligent and robust system. Consider the scenario where a researcher needs to create an agent that assign the video frame into predefine categories (e.g., “frame with face only”, “frame with text only”, “frame with both text and face”). Without the Community of Multimedia Agent, the researcher may have to re-implement some face and text detection algorithms or creating his own. In the environment of COMMA, he can simply design an agent that takes the results of face and text detection agents as input, and produces classification labels, like the “*frame_classify*” agent in Figure 4. Compared with developing every component from the scratch, a lot of time and efforts can be saved. The user can also save the system composed of agents as a script and later load it as a “macro-agent”.

On the other hand, with the availability of more than one face detection agents, their results can be combined to obtain more reliable performance. Since the face agents may employ various algorithms, e.g., neural network, color-shape analysis, each may have its own strength and weakness at different occasions, and we can expect to improve the overall accuracy by having a voting committee among them. This can be accomplished by the “*obj_voting*” agent in Figure 4, which accepts the results of up to 5 object-detection agents. It has a parameter (tuner) that

specifies the mode of “voting”, which could be “or” (a frame has a face detected if at least one of the agents detects a face), “and” (if all agents detect a face) or “majority” (if the majority of agents detect a face). It has been proved that a voting committee can produce more accurate results than any of its members when the errors of the members are uncorrelated with each other [13]. Therefore with the growth of the agent library, COMMA users are better equipped to address for the complexity of the problem, and we can eventually overcome the challenges in the area of multimedia processing research.

5.2. Blackboard Browser

The Blackboard Browser visualizes the results produced by the agents to provide insight about the media content and let the user have an intuitive evaluation of the performance of the agents. Each agent can generate one or more XML files through its output pins, and the data formats conform to the MPEG-7 based templates associated with the pin types. The location of these result files are recorded in the Metadata Sheet of the media file, and thus the Blackboard Browser can retrieve and visualize them by parsing the Metadata Sheet.

Figure 5 shows a Blackboard window for a video file. It contains video browser on the right side, a current frame



Figure 5. Blackboard Browser window.

image on the left side that presents agents' results, and a summary of agents' findings for the current frame in the middle of the screen. The user can watch the results for any frame using the navigation buttons. Below the frame and time scales are the summaries of agents' findings for the whole clip. For example, the summary of a face detection agent is presented in a form of the categorical color bar. Each frame can be categorized as "no faces detected" (white color), "one face detected" (blue color), and "multiple faces detected" (red color). The same color code is used for the text detection agent's results. A user can explore how a particular detection agent works by clicking on the agent's summarization strip and watch the results represented on the current frame picture as a rectangular that frames a detected face or text. Or by clicking on the time scale the user can watch the results of all agents simultaneously on the same picture.

6. COMMUNITY OF LEARNERS

One of the COMMA project main objectives is to create a community of researcher and students in the multimedia processing problem domain. This social aspect of the project is very important for its success. The environment should encourage people to interact, exchange agents and

ideas, discuss topics of interest, and advertise relevant events, such as workshops, conferences, training sessions that target both academic and business research and development. That is why we are paying a great attention to information that is provided by the COMMA Web site. This information includes related business and academic news, overviews of achievements of lead laboratories and researchers, event and job announcements, book and paper recommendations, tutorials, and glossary of specialized terms. It also includes a directory of community member e-mail addresses and chat rooms for real-time discussions. Altogether the tools and information form a socio-technical learning environment that could be beneficial for researchers, teachers and students.

7. SUMMARY AND FUTURE WORK

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. Our work has three major contributions. First, its agent library of gives researchers access to tools to handle the complexity of multimedia data and absolves them from implementing existing algorithms. Second, the

Development Environment facilitates the development of multimedia analysis methods by enabling the researchers to link agents without concerning about low-level technical issues; it also visualizes the agent result to give the user insight about the media content and agent performance. Third, by improving the accessibility and reusability of multimedia processing agents, the value of each research achievement is maximized.

The future extension of our work will go in three directions. First, we are projecting a change in the interaction mechanism between agents. Presently in COMMA the data flow between agents is one-way, and thus the error made by one agent will propagate to others. A promising solution is to allow agents to confirm or negate the results of each other and reach an “agreement” that is the most consistent to the context [14]. Second, we will introduce intelligence to the agents so that they may not only be assembled by human, but also integrate by themselves to generate a solution to a problem. Third, the agents will be distributed as web services, which will give better control of the agents to the inventors and facilitate their upgrade.

REFERENCES

- [1] V.A. Petrushin. Emotion Recognition in Speech Signal: Experimental Study, Development, and Application, In Proc. 6th International Conference on Spoken Language Processing (ICSLP 2000), Beijing, 2000. Vol. IV, pp 222-228
- [2] M.T. Maybury (Ed.) *Intelligent Multimedia Information Retrieval*, AAAI Press/MIT Press, Menlo Park, CA / Cambridge, MA, 1997.
- [3] O.V. Ibrahimov, I.K. Sethi, and N. Dimitrova. Clustering of Imperfect Transcripts using a Novel Similarity Measure, In Coden A.R., Brown E.W. and Srinivasan S. (Eds.), *Information Retrieval: Techniques for Speech Applications*, LNCS vol. 2273, Springer-Verlag, 2002, pp. 23-35.
- [4] N. Dimitrova, L. Agnihotri, and Gang Wei, Video Classification using Object Tracking, *International Journal of Image and Graphics*. Vol. 1, No. 3 (2001), pp. 487-505.
- [5] Yao Wang, Zhu Liu, and Jin-Cheng Huang, “Multimedia Content Analysis Using both Audio and Video Clues”, *IEEE Signal Processing Magazine*, IEEE Inc., New York, NY, pp. 12-36, vol. 17, No 6, November 2000.
- [6] José M. Martínez, Overview of the MPEG-7 Standard, <http://mpeg.telecomitalia.com/standards/mpeg-7/mpeg-7.htm>
- [7] M.N. Huhns and M.P. Singh, “*Agents and Multiagent Systems: Themes, Approaches, and Challenges*”, In Huhns M.N. and Singh M.P. (Eds.), *Readings in Agents*, Morgan Kaufman, San Francisco, CA, 1998.
- [8] A.J. Hauptmann and M.J. Witbrock, “InforMedia: News-on-Demand Multimedia Information Acquisition and Retrieval”, In [2], pp. 215-239.
- [9] B. Merialdo and F. Dubois, “An Agent-based Architecture for Content-Based Multimedia Browsing”, In [1], pp. 281-294.
- [10] W3C Candidate Recommendation, “Resources Description Framework (RDF) Schema Specification 1.0.”, March 2001
- [11] W3C Notes, “DAML+OIL (March 2001) Reference Description “, March 2001
- [12] J. Heflin and J. Hendler, “A Portrait of the Semantic Web in Action”, *IEEE Intelligent Systems*, vol. 16, No. 2, pp. 54-59, March/April 2001.
- [13] L. K. Hansen and P. Salomon. “Neural network ensembles”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990
- [14] D. Li. “Integrated Multimedia Analysis”. Ph.D. Dissertation. Wayne State University, 2001