Université d'Ottawa • University of Ottawa

# Video Indexing and Summarization Service for Mobile Users

By

Mohamed A. Ahmed, B. Sc., M. Sc.

A thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements for the degree of

**Doctorate of Philosophy**

in

Electrical and Computer Engineering

Supervisor

Dr. Ahmed Karmouch

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
University of Ottawa

# Acknowledgement

I would like to thank my research supervisor, Dr. Ahmed Karmouch, for his steadfast help and support all through my Ph.D. program. I enjoyed the time spent in his laboratory under his guidance. He was always very helpful, caring and understanding, both academically and personally. I count myself fortunate indeed to have been one of his research students. I am grateful to my colleagues in the lab for their co-operation and fruitful discussions. I also thank my Ph.D. committee members — Dr. Shikharesh Majumdar and Dr. Luis Orozco-Barbosa — for their suggestions, criticisms and guidance that helped to shape this thesis into its final form. I am of course grateful to my family for their many sacrifices that made my education possible. They were a constant source of encouragement during my studies and research. I would like to thank Dr. Tom Gray and Sergei Mankovski from Mitel Corporation, Dr. Suhayya Abu-Hakima and Dr. Ramiro Liscano for their helpful discussions and suggestions while they were at the National Research Council of Canada. Special thanks go to the Faculty of Graduate and Postdoctoral studies at the University of Ottawa for their financial support throughout my Ph.D. program.

# Abstract

Image processing, video analysis and computer vision techniques are presently developing rapidly because of the availability of acquisition, processing and editing tools which use current hardware and software systems. However, problems still remain in conveying this video data to the end users. Limiting factors are the resource capabilities in distributed architectures, and the features of the users' terminals. The efficient use of image processing, video indexing, and analysis techniques can provide solutions or alternatives.

This thesis presents a new algorithm for video segmentation, indexing and key framing tasks. The algorithm is based on color histograms, and uses a binary penetration technique. Although a lot of work has been done in this area, most does not adequately consider the optimization of timing performance and processing storage. This is especially the case when the techniques are designed for use within run-time distributed environments. A main contribution of this thesis is to blend high performance and storage criteria with the need for effective results. The algorithm uses the temporal heuristic characteristics of the visual information in a video stream. It considers the issues of detecting false cuts and missing true cuts due to the movement of the camera, the optical flow of large objects, or both. We discuss the merits of the new algorithm compared to the existing one, supporting the discussion both with results from experiments and from the implementation of our application.

We also propose a video event modeling mechanism to intelligently parse, analyze and extract the significant content information from digital video libraries or video mails. This also requires an adaptation stage in order to react to the status, policies and configuration of the end user environment.

In order to build robust and extendable systems capable of dealing with future new devices that may have new specifications, we consider devices by their characteristics rather than their type (PDA, PC, cellular phone, etc). We designed and developed a video key framing and summarization service within an overall agent-based architecture that negotiates the different factors autonomously and dynamically at run-time in order to provide the service to the user in an efficient and secure manner.

# Table of Contents

# List of Figures

# List of Tables

# Glossary

| | |
|---|---|
| AVI | Audio Video Interleave |
| AVO | Audio Video Object |
| BMP | BitMaP |
| CC/PP | Composite Capabilities/Preferences Profile |
| D | Descriptor |
| DCT | Discrete Cosine Transform |
| DDL | Description Definition Language |
| DPCM | Differential Pulse Code Modulation |
| DS | Description Scheme |
| DTD | Document Type Definition |
| GPS | Global Positioning System |
| HVC | Hue, Value, Chroma |
| JPEG | Joint Photographic Experts Group |
| LDAP | Lightweight Directory Access Protocol |
| MPEG | Moving Picture Experts Group |
| MP3 | MPEG audio layer 3 |
| OCI | Object Content Information |
| OCR | Optical Character Recognition |
| PDA | Personal Digital Assistant |
| QoS | Quality of Service |
| QTW | Quick Time for Windows |
| RDF | Resource Description Framework |
| RGB | Red, Green, Blue |
| TTS | Text To Speech |

| | |
|---|---|
| **VBR** | **Variable Bit Rate** |
| **VOP** | **Video Object Plane** |
| **XML** | **eXtensible Markup Language** |

# CHAPTER 1

# INTRODUCTION

## 1.1. Motivation

Video information is a very useful way of conveying messages or sharing ideas. It is used in different aspects of life such as entertainment, digital broadcasting, digital libraries, interactive-TV, video-on-demand, computer-based education, video-conferencing, and video email. Content-based media indexing, analysis and summarization services are currently becoming more useful, and at times indispensable due to the growing quantity, volume, and complexity of multimedia information and because of access to an ever-increasing number of devices.

However, we know that streaming of video has limitations. First, quality of service (QoS) over the Internet is not guaranteed. We have to take into account resource limitations such as unavailability of network bandwidth, especially in mobile and wireless environments. Users may not have the time or tools to browse in order to select the required video contents. Users may well be employed by organizations with policies and constraints that govern the employee traffic and activities allowed over its local Intranet.

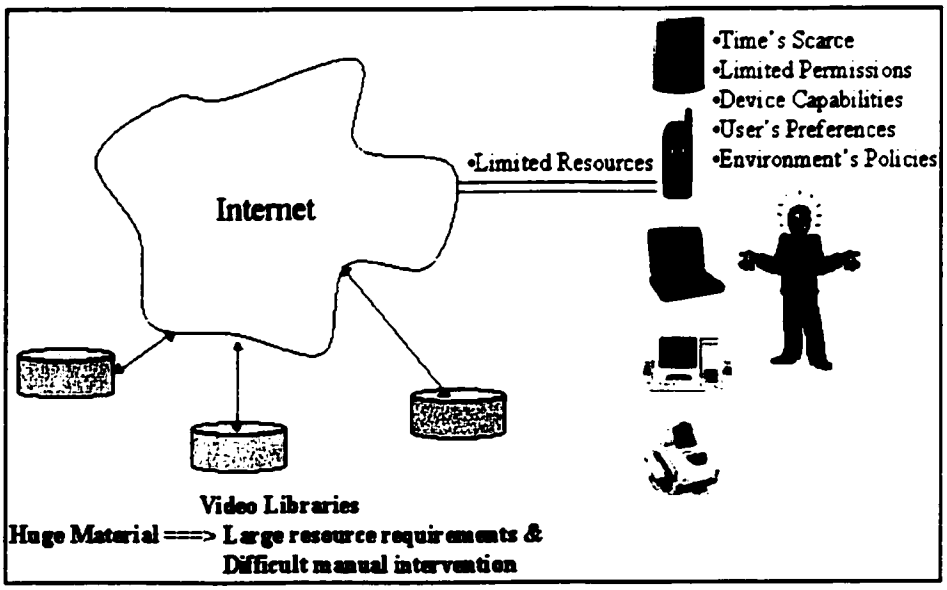



**Figure 1-1.** Current and future telecommunications problems to access video libraries.

Another aspect of the multimedia communication problem is the boom in use of web-enabled wireless appliances such as Personal Digital Assistants (PDAs), pagers, cellular phones, and set-top boxes. This current boom is expected to continue into the future. These devices have features that do not necessarily support video transmission and rendering. As an alternative, we can offer short summary messages to users anytime anywhere on their mobile or stationary devices, delivered according to their preferences.

Figure 1-1 explains the need for automatic or semi-automatic media abstraction or summarization services for current telecommunication environments. This need impels us to provide new requirements for major applications, and a new breed of multimedia applications, in order to cope with the limitations.

Ideally, these new services should be cheap, fast, reliable, opportunistic and dynamic with the environment. We believe that the design and development of video-processing tools should be based on heuristic and opportunistic approaches in order to meet the goals of improved processing speed and adaptability. While the problems shown in figure 1-1 do not necessarily have to occur simultaneously, our approach should be able to anticipate them all, and to react accordingly.

As mentioned, we consider video summarization to be one of the most useful and urgent application services. This service actually represents a high level *"semantic"* quality of service that will not only handle network bandwidth, delay and jitter parameters but also factors such as device capabilities, users' permissions and organizational policies. The video service would be integrated with eXtensible Markup Language (XML), now increasingly used as a common document language. Figure 1-2 illustrates the goal of our work: facilitating and accommodating the retrieval of multimedia content in a variety of conditions. We attempt to convey the same message information in different run-time forms customized to match the resources available in the users' environment, such as organizational policies, device capabilities, and user preferences. This requires an intelligent analysis and appropriate selection of the multimedia content, and its conditioning and conversion. Our approach to multimedia indexing and summarization uses the agent-based architecture built in our laboratory for this specific purpose. We suggest that video indexing, key framing and summarization services could also be useful in multimedia data base management systems and would provide more efficient browsing, query results, and management of multi-format video archives.

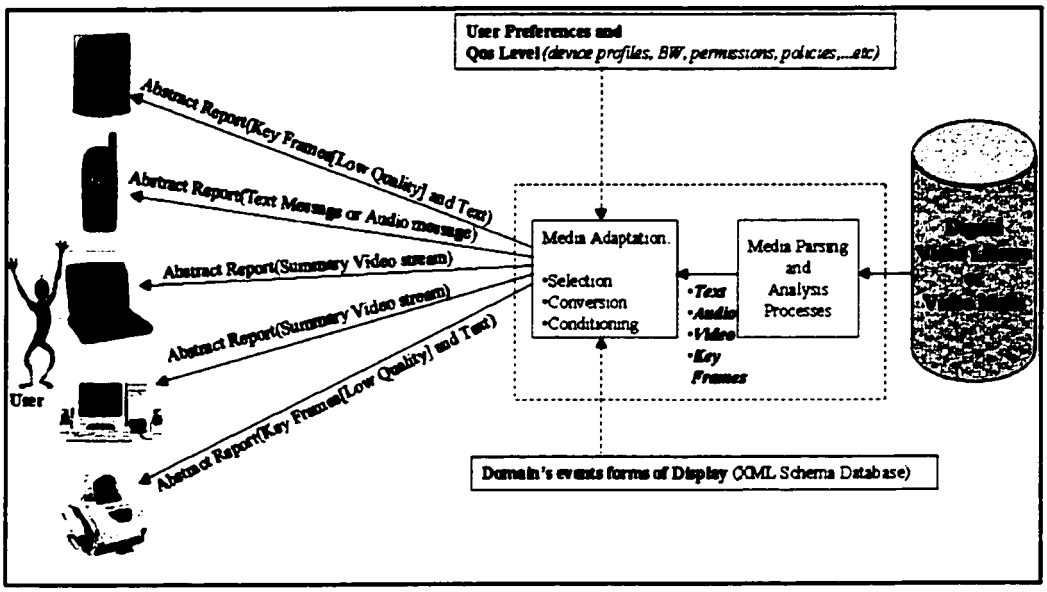



**Figure 1-2.** The goal: delivering the video message in an adapted form.

## 1.2. Approach

We designed and implemented a modular video server that can be extended to accommodate new video services and to interact with other modules to support mobile computing applications. The main objective is to provide content-based video summarization services according to different run-time conditions and limitations, whether these limits are imposed by bandwidth and processing resources, the need to respect the policies of an organization, or the run-time capabilities of the users' devices.

We started first by designing and developing functions for multi-format video segmentation, indexing and key framing. We reviewed current approaches and found that most need extensive processing power, memory and storage requirements. Current algorithms and methods of video analysis and summarization were mainly developed to perform specific image and video processing functions. They are not necessarily suitable for the Internet's high volume of requests and the requirement for run-time customization. The existing algorithms do not perform well when processing time and storage requirements are measured alongside their accuracy and efficiency.

We therefore developed a new algorithm called the Binary Penetration Algorithm for video indexing, segmentation and key framing, whose performance is measured in terms of processing time and accuracy of results. The algorithm is based on visual heuristics and a high correlation of consecutive video frames, and applies a dichotomy penetration procedure to detect true video shot cuts. We tested and evaluated the processing time and accuracy of the algorithm (that is, measuring recall and precision which will be defined in section 6.2) against other existing algorithms in order to appreciate the merits and limitations of our algorithm. We will show in our testing and evaluation results that the algorithm's performance was promising. The tests allowed us to identify optimum parameter values of the algorithm; the rationale for selecting these values is described.

We then used the algorithm to provide a multi-scale video service over the Internet offering small footprint results customized to the surrounding conditions. We integrated this process in our agent-based architecture using a service representative agent in the video server. We describe how our process is positioned in the overall system, and also describe the interface mechanism needed to integrate the service with the system. We show the output document using XML elements and an associated Document Type Definition (i.e. DTD) file, to validate the results so that other organizations and smart devices can understand the semantics of the output.

We designed a model for video summarization. The model uses the video domain's knowledge or general, external knowledge. The knowledge schema could be recorded exclusively, or by applying a learning algorithm to a training set of different video editing sources. This knowledge would accommodate facts or rules of spatial, temporal and spatio/temporal relationships (between both coarse-grained domain events and fine-grained event-clues relationships) that are most likely to be typical of this domain. We use extended N-ary relationship operators to describe those relationships. Each domain is characterized by certain high-level events that we try to detect through the analysis procedure. Each event is recognized through corresponding clues and triggers. These clues and triggers could be detected through an initial parsing that uses existing low-level extraction tools or utilities. In the knowledge schema of each domain, each clue or trigger has a certain importance level relative to its associated

event.

We recognize that this video indexing and summarization server may use low-level tools that have different degrees of accuracy. Our analysis therefore takes into account the confidence of the system in each tool and the confidence of each tool in its ability to account for uncertainties before we decide on the event classification. We also designed a conflict resolution procedure to detect and resolve unclassified events within the same time frame. We expect these to arise because of noisy video multimodal inputs or problematic tool results. The final step is the selection of certain events in a customized form according to run-time conditions. As an example, we will apply our approach to a soccer game.

Our video indexing procedure, using the Binary Penetration algorithm, depends greatly on the predefined threshold parameter used in the algorithm. We therefore analyzed the sensitivity of the algorithm against this parameter using different video domains and editing techniques. This gave us an effective range of threshold values and allowed a high level of efficiency in accuracy and processing time. We performed evaluation and testing procedures to validate the merits of our approach and its limitations. We designed the data structures and the reasoning necessary to analyze the video contents accurately at the beginning, and to establish selection criteria for the video summary at the end.

The selected content can be adapted according to its semantics. For example, a goal-scoring event in a soccer game can be represented in a few frames, while a yellow card can be delivered in just one frame.

We must then deal with the user's device profile. We can characterize the device by certain features irrespective of its type (PC, laptop, cellular phone, PDA, etc). This profile could be preserved within the user profile on Lightweight Directory Access Protocol (LDAP) servers. Or it could reside on the device itself as an XML description file, similar to smart cards used for security systems. We therefore describe the device features that match the services, including video, that are possible in our overall architecture and project requirements. These device profile attributes are used to guide the system components to transcode, condition and adapt the video service results as required.

## 1.3. Summary of Contribution

To date, this work has been reported in several conference papers, journal articles, and book chapters. These publications are listed at the beginning of the references chapter. We feel that the research contributes to the field of multimedia telecommunications and services as described below.

1. High performance and high accuracy video segmentation and indexing technique [AHM02]

Video processing systems include video segmentation, indexing and key framing over the Internet. The resulting high volume of access and requests and access has more requirements than other stand-alone processing engines. The systems need to be fast, reliable and extendable. We therefore built a video segmentation algorithm based on opportunistic processing of the characteristics of video content. We call this a Binary Penetration Algorithm. Testing and evaluation of this algorithm showed promising results in timing performance and accuracy. This algorithm provides an orthogonal approach that can be used to improve the performance of other sequential and linear algorithms in aspects such as processing time, memory, and storage requirements.

2. Customization of video presentation [AHM99a]

Using the Binary Penetration Algorithm, we built a key framing process for use over the World Wide Web. This process is capable of seamlessly handling different video formats, either compressed or uncompressed, without changing the main algorithm itself. It can therefore be applied to new video formats or to new processing algorithms. The process can provide different result reports for the same request, with results adapting to the conditions of the surrounding environment. It can provide the chosen segment in a video stream or in small key frames, whose size could decrease by the use of gray-scale frames with different compression ratios. The process also measures the processing time of the

request for billing and accounting purposes. We built thin client interfaces with a small footprint using Java classes and applets for use in appliances supporting Java Virtual Machine (JVM).

3. Model-based Approach for video indexing, analysis and summarization that respects uncertainties of different video processing elements [AHM01]

Our approach to video indexing, analysis and summarization is based on an external knowledge of different domains. This could be used for sports, news, films, or even general-purpose video. Once the system is built, this knowledge provides a long-term memory of video segments of the same domain. The same approach and algorithms can then be used for different domains, or to accommodate new knowledge in the same domain. The process of analysis uses clues in the domain, called triggers that quickly guide the analysis towards its objectives. However, analysis of video content is a complex, multi-layered problem. First, there is the amount of multimodal information involved. Second, there are temporal relationships as well as spatial relationships between the different video contents. The problem is therefore more than the normal pattern recognition and stationary event classification. After initial recognition, the temporal relationships between the coarse events must be accounted for before the final decisions on event classification. We therefore provide a mechanism that respects the known features of the domain.

Other researchers tend to assume guaranteed results of their tools or algorithms. We can adjust for the confidence levels and uncertainties of the tools we use. We expect to get noisy results that do not recognize certain characteristics of the video domain, be it news, sport, movies, or documentaries. We therefore provide a mechanism to resolve conflicts, using high-level rules to make event decisions, and low-level rules to analyze the clues of each event.

4. Integration of video summarization service as a web service within a distributed architecture [HAR01a]

We envision building the video summarization sub-system using a video service agent integrated with the agent-based infrastructure designed and implemented in our laboratory. The objective is for users to have access to their home site services while they are roaming or nomadic, such as when visiting another organization, staying in a hotel, or travelling. One of these services is video summarization. We therefore describe a standard interface allowing the video service to interact with other modules of our architecture so that its function can adapt to run-time limitations or other constraints. This standard integration interface is delivered with the other modules of the overall system. The video server is built as an extendable system capable of accommodating future video services, while still interacting with the other modules, using standard notations in XML format. The results of requests submitted to the video server could then be parsed and understood at run-time, or later, by foreign components sharing the XML schema.

## 1.4. Outline

The thesis is organized as follows. In Chapter 2, we survey related work. This survey includes related aspects of multimedia processing techniques, as well as the evolution of digital video encoding techniques towards content- and description-based encoding formats. We also refer to other tools or algorithms in the appropriate chapters. Chapter 3 describes our Binary Penetration Algorithm, capable of visually segmenting video content using a new high-performance approach with no loss of accuracy. In Chapter 4, we describe the model designed for the media summarization process. We try to analyze video contents using any multimodal features that may be useful. These include audio, visual and text-caption contents. We believe that each video domain has its own criteria and related information that can improve the summarization process. Since the service is designed for use in distributed environments, Chapter 5 first briefly describes our architecture, designed and developed for agent technology. The architecture seeks to provide nomadic users with a seamless environment that provides service even as they roam from one environment to another. We then describe where we position the media summarization service and how we use and interface it in the telecommunication architecture. Chapter 6 provides the results of the testing and evaluation of different algorithms that we designed and

implemented, followed by a discussion of the results. Chapter 7 presents our conclusions, discusses limitations of the research and suggests future directions. Already-published papers on this research are listed at the end, together with other references.

# CHAPTER 2

# RELATED WORK

## 2.1. Video Indexing, Segmentation, Key Framing and Summarization Techniques

A first step to achieve video summarization objective is usually to segment the visual content. In the past decade, researchers have made considerable contributions in the area of image analysis and recognition to allow partitioning of a video source into separate segments. In addition, they developed algorithms to index videos in multimedia databases, allowing users to query and navigate by content through the database. In [BOR96], the authors define a shot as *an unbroken sequence of frames from one camera. Thus, a movie sequence that alternated between views of two people would consist of multiple shots. A scene is defined as a collection of one or more adjoining shots that focus on an object or objects of interest. For example, a person walking down a hallway into a room would be one scene, even though different camera angles might be shown.* One approach of video segmentation is to use Pixels-Pair wise comparison [ZHA93] as a simple way to detect a quantitative change between a pair of images by comparing the corresponding pixels in the two frames to determine how many pixels have changed. Such algorithms evaluate the total percentage of the pixels changed and if this percentage exceeds some

preset threshold, the algorithm decides that a frame change has been detected. The value of this method is its simplicity. However, its disadvantages exceed the advantages. One disadvantage is that it involves a large processing overhead in comparing all consecutive frames. It does not accommodate some cases, such as a situation where we have large objects moving within the shot before terminating the continuous shot.

The Spatial, Temporal Skips (Histogram Analysis) [SWA91] [ZHA93] [BOR96] [AHM99a] [AHM99b] methods benefit from the redundant characteristic of the video frames either in the spatial dimension or the temporal dimension. Such approaches verified that the use of color histograms shows more robustness and reliability against objects and camera movements within the same shot. We could temporally compare every defined number of frames, as frame samples, instead of all the consecutive frames and/or spatially use a number of pixels less than the total frame size. Thus, we could save a significant processing time and resources utilization during the analysis. Figure 2-1 shows a histogram function of some image. There are various histogram difference measures that could be used to detect a shot cut.

For example:

1) Difference = $\Sigma \mid H_i(j) - H_{i+1}(j) \mid$ for j = 0 To N-1

Where, $H_i(j)$ is the histogram for color j in Frame i,

N = Number of the different possible 6 bits RGB bits values (i.e. N = 64)

OR

2) Difference = $\Sigma ( \mid H_i(j) - H_{i+1}(j) \mid^2 / H_{i+1}(j) )$ for j = 0 To N-1

Where, $H_i(j)$ is the histogram for color j in Frame i,

N = Number of the different possible 6 bits RGB bits values (i.e. N = 64)

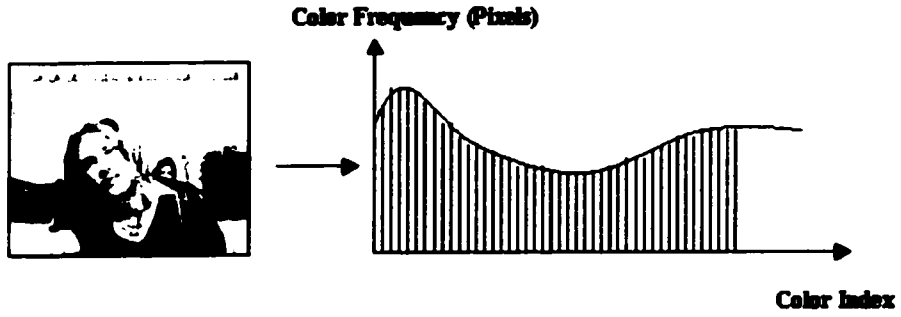



**Figure 2-1.** Color Histogram Function

Then, using a preset threshold, we could discover the existence of a frame change and thus the shot cut operation is detected. Hirzalla [HIR97] has described a new detailed design of a key frame detection algorithm using the HVC (hue, value and chroma) color space. First, for every two consecutive frames, the system converts the original RBG coloring space of frame pixels into its equivalent HVC histogram coloring representation because the HVC space mirrors the human color perception. The system uses the hue histogram information for performing the comparison instead of the intensity distribution to reduce, to some extent, the variations in intensity values due to light changes and flashes. Some of the terms definitions stated within the approach are as follows:

- ***Scene:*** *a scene consists of one or more frames representing a continuous action in time (i.e. no camera breaks) and space forming a piece of video.*

- ***Cut:*** *defined to be a discontinuity between scenes. A cut is sharp if it can be located between two frames, and it is gradual if it takes place over a sequence of frames.*

- ***Histogram:*** *A histogram of a frame or an image reflects the number of pixels in that frame having the same parameter value for all possible values of that parameter.*

- ***Hue, Value and Chroma:*** *HVC are image parameters forming three-dimensional space that represents trio-attributes of psychological human color perception. The hue of a color refers to its 'redness', 'greenness' and so on.*

- ***Frame size:*** *the number of pixels in one frame of the video clip, and equals to its width times its height.*

Now, we want to conclude some remarks regarding improving the performance of that algorithm which we followed and adopted in our approach for video indexing and segmentation function particularly. First, the system doesn't use any temporal or spatial skip (i.e. it doesn't save time and processing power due to redundant information within the video stream) taking into consideration that we also need to perform the histogram analysis within all the sub-areas as well. We could imagine the difficulty to do the algorithm for medium or long video clip, which could represent about 54,000 frames for about only half an hour. So, we could improve the performance, memory and storage usage of the algorithm significantly using both temporal and spatial skips, as we will describe in chapter 3.

In [KAN99], the author apply certain fuzzy rules to select key frames based on meaningful regions within the frames and using the temporal variations within the shot to detect and classify video shots into either panning, tilting, zooming or no camera motion. Jian et al. [JIA98], Xiong et al. in [XIO97], Otsuji et al. in [OTS93], Gargi et al. in [GAR96] and [GAR98] reviewed different techniques for scene change detection techniques. It is realized that the different techniques didn't deal with measuring and optimizing their processing power performance, which gives our work a major contribution in that area. In [HUA97] and [HUA99], the authors used color correlograms instead of color histograms and handle problems of image sub-region querying, object localization, object tracking and cut detection. Their approach makes use of spatial correlation between pairs of colors. However, they didn't study and compare the run-time performance of their algorithm relative to using color histograms. It is obviously expected that their algorithm would require large processing and statistical power than using color histograms. Similarly in [PAS96], to improve the lack of spatial information of color histogram, the authors evaluates a Color Coherence Vector (CCV) for each pixel. They try to classify each pixel with a

certain classification either as coherent or not within a large similarity-colored region. However, they don't take the temporal characteristics of visual information into account to improve the performance, which would be intensive for such calculations.

In addition, some work has been done in the area of detecting video scene cuts for compressed format such as MPEG format. They usually utilize the use of the DC coefficients of the Discrete Cosine Transform (DCT) in MPEG sequences such as in [YEU96] [TAS98] [GAR00]. However, first their techniques are mainly suitable for these video-encoding formats as they utilize the video encoding coefficients such as the DCT coefficients. Secondly, they neither discuss or compare the issue of the performance of the algorithm in their work nor measure the effect of using such approach as a service over the Internet which needs to process large volume of requests of different video file formats and using different device characteristics..

Other image processing algorithms try to utilize the texture information within the frame sequence of the video for video analysis. The images texture is characterized be three features: coarseness, direction and contrast. Gabor filters [WEL94] [WEL96] [DUN97] are bandpass filters which have been successfully applied to many image-processing applications, such as texture segmentation, document analysis, edge detection, fingerprint processing and image representation. They could be used as single or multi bank filters. Generally speaking, researchers use Gabor Filters when trying to solve problems involving complicated images comprised of textured regions. They essentially transfer adjacent regions into high-contrast regions that can be isolated by thresholding. However in our opinion, It is not suitable for medium and long video analysis because it depends on extensive mathematical processing and the difficulty of adjusting the filter parameters to design the filter bank. Figure 2-2 [WEL96] represents an image processing block diagram using gabor filters.

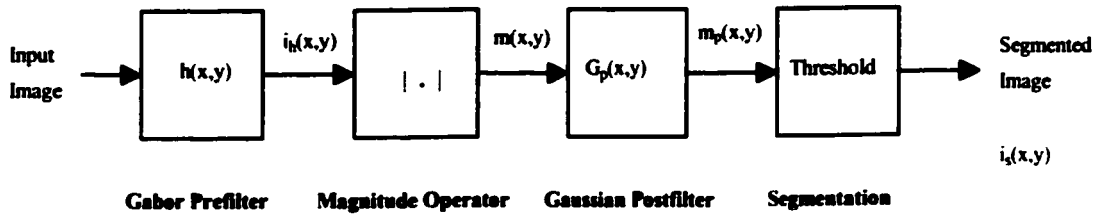



**Figure 2-2.** Image processing block diagram using gabor filters. from [WEL96]

Some edge detection algorithms, such as in [ZAB95], have been utilized as well to achieve reliable video segmentation. They are based on the changes in the detected object edges using color gradients. However, they obviously still impose extensive processing and storage requirements. Thus, they could be suitable for such applications that require object detection, object tracking and video surveillance for example rather than recognizing video segmentation and indexing tasks.

Among the work done in the area of the video summarization and abstraction, in [LIE99], the author provided some approaches to shorten large home video captured from personal camcorders. He utilizes the characteristics of the home videos. They usually could be separated into distinct events such as birthdays, trips, ...etc and using the time stamps of the video data, we could segment the video into hierarchical distribution of events and activities. The provided algorithms are suitable mainly to home video characteristics and can't be generalized to other forms of videos such as film genres, sports, news, ...etc.

In [UCH99] the authors try to summarize video segments using few key frames that are associated with certain importance values. Their summary resembles cosmic books with the size that are suitable to be published in graphical web page layouts. They enhance the summary by including text captions derived from OCR or other methods. They provided a frame-packing algorithm to build the frame summary. However, the authors don't utilize the audio contents in their analysis. Thus, they don't make use of the spatio/temporal relationships among the multimodal video contents within the video segment (such as audio, visual and the text-captions).

In [SAT99], the authors built an automatic extraction and recognition system of superimposed news captions and annotations. They try to solve two main problems. These problems are: the low-resolution characters and the very complex backgrounds. They use interpolation filters, multi-frame integration and character extraction filters. Thus, their approach again doesn't build on any spatio-temporal relations among the multimodal video content. In [KAN97] [NAK97], the authors try to analyze the media contents semantically to associate natural language clues with key image clues to segment the video contents into meaningful segments to identify the segment into certain types such as: speech/opinion, meeting/conference, crowd, visit/travel and location using certain clues. However, their work concentrates on segmenting and identifying the contents rather than summarizing these contents into short messages. They don't use temporal and spatio/temporal correlation among the contents of video.

In [YOW95], the authors use image processing and object recognition techniques to highlight soccer goal events with soccer games. They use motion estimation algorithms, panoramic reconstruction, soccer ball detection and tracking, and detection of goal posts to discover the goal events. However, their technique is suitable to detect soccer goal event only. It is not a general-purpose engine to detect other possible events even within this soccer domain itself. Secondly, no use of audio analysis or text captions in their approach. In addition, the approach lacks modeling the temporal and spatio/temporal relationships among the events and activities of the soccer game.

Other complex approaches try to achieve the summarization goal using the visual and audio contents of the video segments. For example, Smith et al. in [SMI97] [CHR98] [WAC00] use image and language understanding techniques to skim large digital video databases of documentaries and newscasts. They define *video skimming* as *"temporal, multimedia abstraction that incorporates both video and audio information from a longer source"* [CHR97] [CHR98]. Pfeiffer et al. in [LIE97] [PFE96] [FIS95] propose an abstraction technique for film movies. They built their method for events definition based on *Contrast Perception, Color Perception, Content Perception* and *Stimuli Processing*. Nam et al. in [NAM99] provides a "dynamic" video abstraction technique based on an adaptive nonlinear sampling of video contents for movies. They mainly try to recognize two *semantic events: emotional dialogues* and *violent* featured actions to represent the abstract. In [DAW99], the authors proved that the use of shot

content description modeling, rather than classical video source models, improved the performance of specifically the variable bit rate (VBR) MPEG video traffic in terms of means transmission delay. They classify each shot according to its texture and motion complexity into nine classes and described each class with an autoregressive model.

Another recognized issue is that these approaches only embed certain knowledge about the structure of the video and the expected analysis and summary results. They all neglect the uncertainty factor of the used algorithm accuracy and the confidence of the algorithms and tools in their results. In addition, these content-based processing systems are mainly used for interactive and indexing video library databases and not for Internet requests and messaging services. This certainly needs adaptation consideration as the Internet still has certain limitations such as limited bandwidth, organizational policies and heterogeneous user's device capabilities. Thus, in our vision, we try to address and solve these different issues in addition to the video indexing and summarization engine itself.

## 2.2. Digital Video Encoding Standards: Towards Content-Based Encoding, Search and Retrieval

Nowadays, another parallel directions are being developed to provide new standards for content-based video encoding, search and retrieval that aim to provide and facilitate new and more reliable intelligent multimedia systems in near future such as more robust video summarization service. Two of these promising standards are Moving Picture Experts Group's MPEG4 and MPEG7 standards.
We will describe briefly the major advances and new capabilities within both standards. Meanwhile, we will comment how these new features will lead to new breed of multimedia applications such as our video summarization application.

### 2.2.1. MPEG-4

The initial objective for MPEG-4 [SIK97] work is to provide algorithms and tools to support very low bit rate coding of video data. However, it was extended to suit the new type of interactive multimedia

applications that requires interactivity with individual audiovisual objects within the video data, high degree of scalability and hybrid encoding of natural and synthetic objects. MPEG-4 supports these requirements through efficient compression encoding of individual audiovisual objects, higher user interaction with individual objects, object-based and temporal random access, generic encoding of natural and synthetic objects and spatial, temporal, quality and object based scalability techniques.

MPEG-4 uses object-based representation for Audio Visual Objects (AVOs) to encode a scene. Each AVO could be a visual object only, audio object only or a combination of both components. Examples of AVOs are synthesized speech of plain text, recorded sound trough a microphone, 2-D visual object, ...etc. Each AVO is coded into separate bit stream. To decode the scene at the receiver, another distinct stream is used that contain scene description information that describes the spatial and temporal coordinates of the different AVOs in order to enable re-composing the scene at the end user. The scene description information stream is multiplexed together with the original AVOs bit stream. At the user's terminal, the streams are demultiplexed and the primitive AVOs streams are decompressed and the scene is recomposed and rendered using the scene description information. This approach allows the user to interactively access, manipulate certain objects, and change the scene display scenario without changing the AVOs contents themselves. For example, the user would be able to change the position of certain objects, rotate them, ...etc. He could change the pitch or frequencies of certain sounds. This could permit the change of the temporal, spatial and quality of certain objects according to different applications. For example, for mobile teleconferencing applications, higher spatial resolution and higher frame rate could be assigned to the foreground objects such as a talking person than the background objects.

MPEG-4 provides four different representation and compression coding tools to suit different applications, bit rates and formats. These coding tools include:

- **Video object coding** to code rectangular, arbitrarily shaped, synthesized and/or natural objects. A video object is defined as an arbitrarily shaped segment that has a semantic meaning. A 2-D snapshot of the video object at specific time is called video object plane (VOP). A VOP is defined by its shape

and texture (luminance and chrominance values). MPEG-4 allows content-based access to the video objects or a certain VOP at certain instant. MPEG-4 encodes the shape, motion and texture of each VOP. If the VOP is the whole frame rectangle, it is encoded similar to the encoding methods in MPEG-1 and MPEG-2.

In MPEG-4, and using available automatic or semi-automatic image segmentation tools [HAR85], we could separate different objects from each other and from the background. This segmentation process could be done on line in real time, or offline. Other segmentation techniques use chroma keying [KAT98] approaches by using a unique color to separate a video object from the background.

Thus, MPEG-4 video coding consists of shape coding of arbitrarily shaped video objects, motion compensated prediction to make use of temporal redundancies, and texture coding using DCT-based encoding of the motion compensation prediction error to make use of the spatial redundancies. VOPs are partitioned into macroblocks such that a complete VOP could be included by a minimum number of macroblocks inside a rectangle boundary that include only that video object as shown in figure 2-3. Video coding in MPEG-4 is implemented at the macroblock level. As in MPEG-1 and MPEG-2, MPEG-4 uses Intracoded (I), forward-predicted (P) and bidirectional (B) VOPs in the same manner as used with video frames in the predecessor MPEG standards.

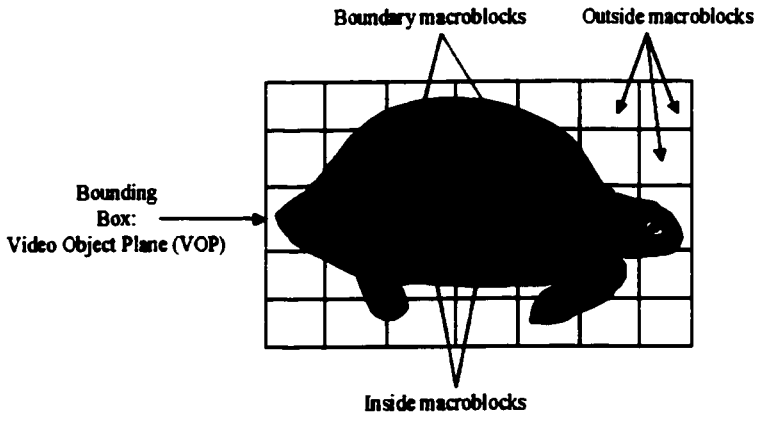



**Figure 2-3.** Example of a VOP and macroblock partitioning.

MPEG-4 uses sprite coding to represent static video objects within a scene or other video objects such as backgrounds that could be approximated by warping the main video object planes. They are coded similar to intra VOPs and transmitted once at the beginning of the scene, stored in a buffer, decoded and used to reconstruct the video scene. Meanwhile, few transformation parameters could be employed to describe zooming and camera motion during the video scene. An example of a sprite is the panoramic background shown on the upper left side in figure 2-4 [ISO00a]. As seen, it includes the pixels occluded by the human object extracted in the top right picture. Obviously, the compression efficiency is enhanced using sprites. The bottom picture in the figure is a video snapshot frame example reconstructed using arbitrarily shaped foreground object representation and sprite background coding.

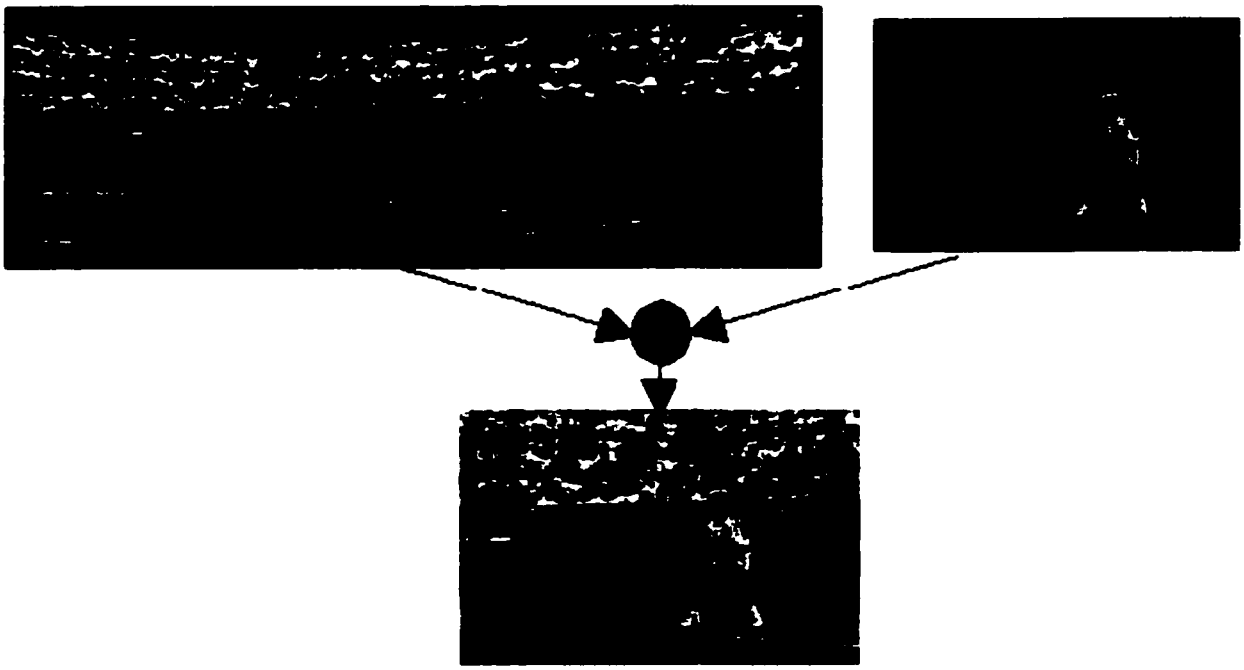


**Figure 2-4.** Utilization of sprite coding to reconstruct video scenes. from [ISO00a]

- **Model-based coding** is utilized to encode the human face, body description and animation features. It is used to model the behavior of a human being using limited number of parameters. Thus, it is efficient for very low bit rate video coding applications instead of the transmission of video frames. MPEG-4 supports two human models; *face object model* using 3-D polygonal meshes to synthesize

human faces and animate face expressions and movements and *body object model* using 3-D polygonal meshes to simulate human body description and movements.

Face properties such as its texture and shape geometry are defined by Face Definition Parameters (FDPs). The decoder that supports face object decoding should already have a default face model that could be customized then using the transmitted FDPs. Each face object is described through a group of nodes within it. These nodes are called feature nodes. Face animation Parameters (FAPs) control the animation of the facial expressions of the face using displacement and angles of the face feature points. MPEG-4 defines 82 feature points for the teeth, cheek, eyes, hair, ear, ...etc. In addition, it has 68 low-level facial animation features such as mouth opening and head rotation. High-level facial expressions are also defined in MPEG-4 such as sadness, joy, surprise, ...etc. Each high-level expression is defined through a collection of certain low-level expressions. For example, a joy high-level expression is recognized by open mouth, relaxed eyebrows and mouth corner displacement towards the ears.

Similarly, body object model has two groups of parameters: Body Definition Parameters (BDPs) to construct the body through its surface, texture and dimensions and Body Animation Parameters (BAPs) which define the movement and posture of the body model.

- **Mesh object coding** is used to represent and encode any visual objects with a mesh structure. Mesh encoding of a frame or a video object means partitioning it into small polygonal patches. It was proved that it is a good and efficient modeling and rendering techniques of 3-D objects in computer graphics. Figure 2-5 [ISO98] represents an example of an object mesh representation that could be efficiently used for encoding object animation. Mesh encoding allows different functions such as: efficient content-based retrieval using object trajectory information rather than bitmap search and more efficient and smooth continuous motion representation without getting blocking artifacts that are associated with block-based representations such as the DCT representation. MPEG-4 employs a two dimensional mesh representation of video objects or still texture objects using connected triangular patches. Each patch has three vertices, which are called node points. Using the corresponding spatial textured patches, we could have a good model of continuous motion

compensation fields. Similar to VOPs, samples of mesh objects represent Mesh Object Planes (MOPs). The encoding of the node points locations is performed through differential encoding relative to previous encoded points. The encoding starts usually with the boundary node points then the inside node points. However, the choice and tracking of these node points are not standardized in MPEG-4. The texture itself of the video object is encoded separately.

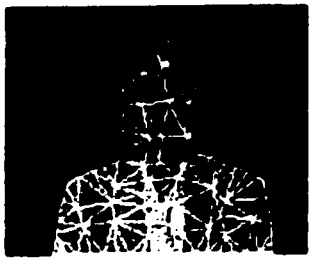


**Figure 2-5.** An example of object mesh representation. from [ISO98]

- **Still texture coding** is performed by wavelet encoding. The texture is first decomposed using a two-dimensional separable wavelet transform. The lowest frequency subband coefficients are encoded using Differential Pulse Code Modulation (DPCM). Meanwhile the rest of the subbands are encoded using zero-tree modeling. Zero-tree encoding is performed by encoding the locations of non-zero wavelet coefficients. It takes advantage of the properties of wavelet encoding. In wavelet encoding, if a wavelet coefficient is quantized to a zero value, it is most likely that all the wavelet coefficients at the same spatial locations and same orientation in finer wavelet scales will be quantized to zero as well.

Finally, MPEG-4 supports both frame-based and object-based encoding. Object-based encoding will result in better compression performance than the frame-based coding if the video sequence, foreground and background objects shapes vary slowly. For example, a sprite and few parameters of camera motion will represent a complete panoramic background. However, if they are changing rapidly, frame-based encoding will be more efficient because for example, the shape of the objects in object-based encoding

will need more bits to accommodate the quick change of the object shape. However, frame-based coding techniques will obviously limit the accessibility and manipulation of the video objects.

In addition, MPEG-4 allows associating information to objects about their content. Therefore, video authors can use this Object Content Information (OCI) data stream to send textual information along with the MPEG-4 content. Further possibilities are giving unique labels to the content, and storing camera parameters.

Regarding audio encoding in MPEG-4 audio (ISO/IEC 14496-3), we could summarize the main features of it as follows:

1) Composition and coding of synthetic and natural audio objects.
2) Text-To-Speech Interface (TTSI): An interface for text-to-speech conversion utilities.
3) Structured Audio: A universal language for score-driven sound synthesis from an existing description.
4) Scalability of the decoder or encoder complexity.
5) Scalability of the bit rate of the audio bit stream.

### 2.2.2. MPEG-7

MPEG-7 standard [ISO00b] official name is "multimedia content description interface", a means of providing meta-data for multimedia. The call for proposals for MPEG-7 started in October 1998. MPEG-7 doesn't replace the predecessor standards such as MPEG-1, MPEG-2, MPEG-4. It is supposed to add complementary functions to them especially with MPEG-4 standard, which defines object-based representation, modeling and access. MPEG-4's encoding syntax supports the inclusion of user data in the bit stream. MPEG-7 objective is to standardize the description of multimedia data for efficient and effective access, manipulation and retrieval. MPEG-7 tries to unify the descriptions of the multimedia contents so that they could be accessible and interoperable between different multimedia applications. Nevertheless, MPEG-7 is not considered to standardize the extracting and processing algorithms and

tools themselves but only focuses on the descriptions and specifications of the outputs of such algorithms and tools.

Currently, access and retrieval systems employ either text-based or feature-based search methods. Text-based methods use keyword description of multimedia data. However, this text description is usually performed using human intervention to describe visual or audio content. For example, HTML language could associate text descriptor with image, audio and video files. There are some text-based search engines available for visual contents (e.g. Yahoo Image Surfer, Icon browser and Image Surfer).
Feature-based search methods could use low-level features such as shape, texture and color. Others use high-level features such as composition information. Low-level features could be extracted automatically using different available tools. However, high-level features are usually defined using human assistance. Different research techniques are utilized to extract each low-level feature. For example, shape methods include region-based techniques (e.g. roundness, area, ...etc) and boundary-based techniques (e.g. fourier, geometric or chain codes). Similarly, texture analysis methods vary from statistical, structural, multi-resolution to stochastic random field models. Color search methods are used widely because of its invariance to picture scaling and rotation using dominant color, average color or local/global color histograms. However, sometimes more than one of these low-level features, which might represent a high-level feature, is employed to generate more accurate results.

Thus, MPEG-7 standardization process has to address some major challenges such as different media types (e.g. graphics, still images, audio, video, ...etc) which are described by various feature representations for each type of them and the difference of media coding (e.g. uncompressed or compressed). Figure 2-6 shows the focus of the MPEG-7 activity. MPEG-7 would standardize the content description interface through defining a group of Descriptors (Ds), Description Schemes (DSs), Description Definition Language (DDL) and the coding schemes of the descriptions. However, it will neither standardize the tools used for description generation such as feature extraction and segmentation tools nor the tools or the applications that utilize these descriptions such as search engines and recognition systems.

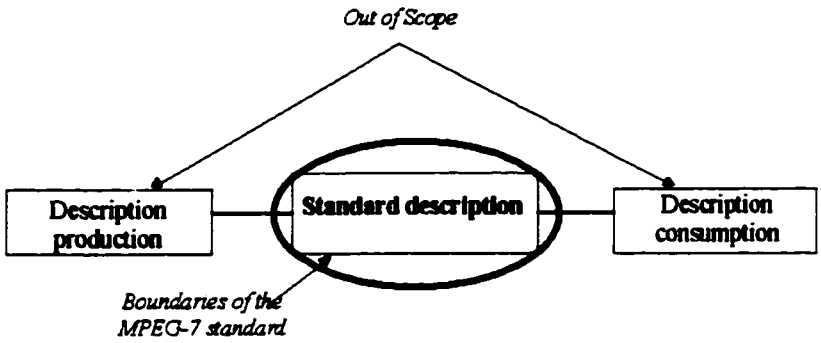



**Figure 2-6.** Scope of MPEG-7 activity.

Now, we will explain briefly the main components of MPEG-7: Ds, DSs, DDL and the coding schemes.

**Descriptors (Ds):** The media content (image, audio, video, ...etc) could be described by a set of features. An example, an image could be described by its shape, texture and color. Then, each of these features could be specified using several parameters called descriptors. For example, some of the descriptors associated with texture features are: coarseness, directionality, contrast, wavelet coefficients, DCT coefficients. Examples of descriptors of the shape are chain code, geometrical, fourier coefficients. Therefore, the media content could be projected by feature vector. Each feature in that feature vector could be described by one of several descriptors. The descriptor should be effective and relevant (i.e. it should guarantee that the feature is described completely and accurately). Thus for example, no descriptor is needed to describe the texture feature of a complete white object in an image. MPEG-7 considers prioritizing the importance level of certain descriptors for an object. For example, shape descriptors could have more importance level than color descriptors for certain visual object within the scene.

**Description Schemes (DSs):** A description scheme consists of set of several components and the relationships among these components. Each component could be a descriptor, a description scheme or both descriptors and other description schemes. Similar to descriptors, DS should be effective and relevant. In addition, it should provide extensibility, scalability and expression efficiency. The effectiveness and relevance properties of DSs is realized if the components and relationships within it

are also effective and relevant. DS should support multilevel representation of the media content as well as the descriptors. Expression efficiency is realized by following the parsimony principle (i.e. the DS should be defined by minimum possible number of components and relationships among the components).

**Description Definition Language (DDL):** A language is required to specify the description schemes and descriptors. This language should be platform-independent. MPEG-7 committee looks for a description definition language that should have unambiguous grammar. It should also support compositional operations such that existing standardized DSs could be expanded and new DSs could be created. The current approach for defining this language is to use extendable XML language so that it could support the utilization of the multimedia content descriptions rather than defining a complete new language. The relationship among the DDL, DSs and Ds in MPEG-7 and the extensibility requirement for new DSs and new Ds is shown in figure 2-7 [ISO00b].

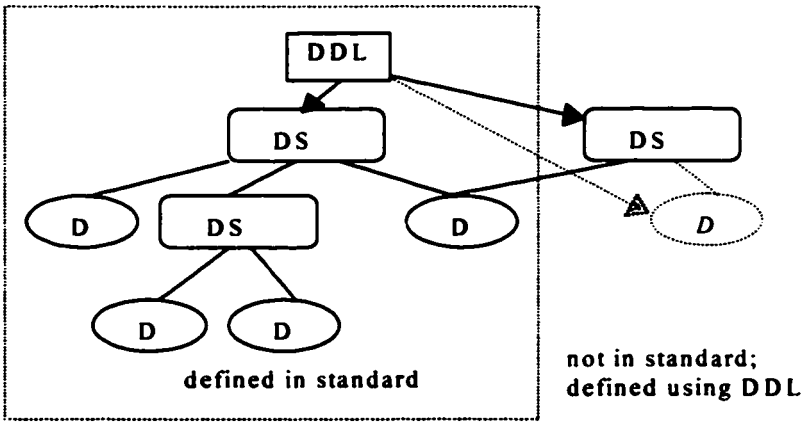



**Figure 2-7.** Relationships of DDL, DSs and Ds in MPEG-7. from [ISO00b]

**Coding Schemes:** The coding of the content description should be efficient for multimedia transmission, storage processes and processing. MPEG-7 is looking to standardize these coding schemes through low complexity and error resilient methods.

It is expected that a decoder or parser that is MPEG-4 and MPEG-7 compliant will allow the user to access and manipulate a multimedia database efficiently and effectively without the expensive decoding, segmentation and feature extraction as these expensive processes would be done once at the encoder site. However, these expensive decoding operations should be done with the other older standards such as MPEG-1 and MPEG-2. A sample of a query using MPEG-7 could be to find the scenes of persons "running". Such query using MPEG-4 mesh models, if standardized in MPEG-7, which easily define a continuous motion would allow the search of moving objects with similar trajectories. Another example is to find persons who are angry using MPEG-4 Face Animation Parameters (FAPs) that describe the mood of the human being (e.g. sad, angry, happy) without doing expensive decoding, segmentation and complex recognition processes.

MPEG-7 encoding of the description could be embedded with the multimedia content files themselves or be as a separate file that is linked to the original multimedia files it describes. It could be used then by search engines or other multimedia applications that utilize this description information.

An important concern, which is out of the scope of MPEG-7 standard, is how the MPEG-7 descriptions will be generated. Descriptors could be generated using the following options:

1) Automatic or semi-automatic extraction (e.g. texture, shape, color parameters and maybe with some human assistance or even feedback).
2) Utilization of existing descriptive data (e.g. scripts, meta-data) through the production/delivery chain.
3) Automatic generation by the capture devices (e.g. GPS location or time in a camera).
4) Manual production (e.g. for legacy material such as those already existing film archives and video digital databases).

Thus though it is recognized that there is more than one way to perform the task of content description, it is expected that users will adopt the ways that match their own abilities and requirements.

Regarding MPEG-7 audio (ISO/IEC 15938), MPEG-7 will provide standardized descriptors and description schemes of sound content, audio structures and a language to implement such descriptors and description schemes. Four sets of audio description tools, which could be useful for many application areas are:

- Sound effects descriptors and description schemes are a group of descriptors that will be utilized for indexing, classification and categorization of general sound effects so that sound recognizing tools could efficiently index and segment sound tracks.

- Spoken content DS is used for indexing, querying and retrieval of audio speech streams and for indexing of multimedia objects annotated with speech. Such DS would solve the problem of out-of-vocabulary words because of the use of a DS that includes combined word and phone lattices for each speaker in an audio stream.

- Musical instrument timbre descriptors intend to describe some perceptual features of the sound of musical instruments. Timbre is defined as the perceptual features that let two sounds that have the same loudness and pitch sound different from each other. The aim of the timbre DS is describing these perceptual features using a limited group of descriptors. These descriptors would formalize these features as the "attack", "richness" and "brightness" of the sound.

- Melody contour DS is a compact description of melodic information that will allow efficient and reliable search matching using melodic similarity. For example, this could be used for query-by-humming.

Several technologies are needed to provide a low-level audio descriptor framework. One of the techniques that could be utilized is the scale tree, which lets temporal series of descriptors to be described in a scalable way. The low-level audio descriptors generated using existing Digital Signal Processing tools, which fit into this technique, include fundamental frequency, spectral centroid, temporal envelope, spectral envelope and harmonicity of audio segments.

An example of a simple and still useful high-level audio descriptor in MPEG-7 is the silence descriptor such that a certain audio segment could be labeled as a semantic silent segment (i.e. no significant sound). It may be used to assist in further indexing and segmentation of the incoming audio stream or as an indication not to process this segment or to assist indicating a semantic description (e.g. suspense or context change) of a certain scene within a movie for example.

Thus in summary, while MPEG-1 and MPEG-2 concentrate almost entirely on the compression algorithm and improving the compression ratios with still good video quality, MPEG-4 moves to a higher level of semantic descriptions through encoding separate objects and utilizing other content-oriented techniques to perform a content-based coding. MPEG-7 jumps to a higher semantic level to standardize the multimedia content representation. MPEG-1, MPEG-2, and MPEG-4 are designed to represent the information itself. Meanwhile, MPEG-7 intends to represent information about the information. Another argument is that MPEG-1, MPEG-2, and MPEG-4 make the content available while MPEG-7 permits to describe and hence reliably find, retrieve and analyze the content we have. Finally, although there are certain object identifiers and other simple meta-data descriptions within MPEG-4, their functions would still be much more limited relative to MPEG-7.

# CHAPTER 3

# VIDEO VISUAL SEGMENTATION

## 3.1. Introduction

A first step to achieve the video summarization goal is to segment the video's visual contents. Thus, we have designed, developed and evaluated a new approach to segment the video content into separate shots and then extract key frames within visual video information. The main contribution of this new approach is improving the performance of the visual segmentation of video content into separate shots in a high-performance manner and without sacrificing the accuracy of the results, as we will describe. High-performance systems are needed to process such high volume of data found within the video. The problem gets more challenging when we use such system in distributed environments over telecommunication infrastructures. We concentrated on detecting video cut camera operation, as it is normally the most used technique for shot transition in video editing.

## 3.2. Video Cut Detection Problem

For the process of segmenting video streams into separate shots, we designed and implemented a video segmentation system. Our system first extracts consecutive frames to detect the camera cut events. The system could define various configurations for the extracted frames. They could be color or gray frames, different image qualities and different image formats (either JPG or BMP). In addition, there is a temporal skip parameter, so that we don't need to extract and analyze all the consecutive frames within the required segment.

Figure 3-1 represents a sequence of video frames with the key frames extracted outside. We realize that this problem is a two-fold problem. It handles the spatial analysis in the two spatial dimensions, x and y and the temporal dimension in time as well.

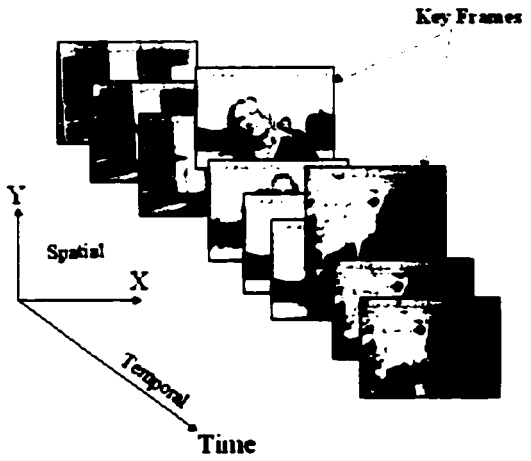



**Figure 3-1.** Key Framing Process Dimensions.

For cut detection, the system already implements a spatial skip parameter as well, to improve the performance without sacrificing the accuracy of the detection. This will benefit from the redundant information and high correlation among the neighboring pixels. We will describe first the original algorithm, which we adopted after few refinement procedures. We call this algorithm: the six most

significant RGB bits with the use of Blocks Intensity Difference. Then, we will present our new algorithm, which we call Binary Penetration algorithm. This new approach improves further the performance of the cut detection procedure, taking into account the temporal heuristics of the video information.

### 3.3. The Six Most Significant RGB Bits with the Use of Blocks Intensity Difference Algorithm

First, the system makes use of the 24 bits RGB color space components of each of the compared pixels (each component has 8 bits representation). However, to speed the performance considerably, the system exploits only the 2 most Significant bits [ZHA93] of each color (using the masking operation), which means that we actually define only 64 ranges of color degrees for the entire RGB color space. The system evaluates the histogram of the corresponding two frames, taking the temporal skip into account. Then, the following histogram difference formula, between two frames $f_1$ and $f_2$, is used:

$$\text{Histogram Difference}(f_1, f_2) = \tfrac{1}{2} \cdot \Sigma \, | H_2(i) - H_1(i) | \; / \; \Sigma \, H_1(i) \qquad \text{for } i = 0 \text{ To } N\text{-}1$$

Where, $H_1(i)$ is the RGB Histogram for frame M,

$H_2(i)$ is the RGB Histogram for frame (M + Temporal Skip),

N = Number of the different possible 6 bits RGB bits values (i.e. N = 64)

The ½ factor is used in the equation to normalize the difference to be in the range of 0% to 100%.

Then, if this Histogram Difference exceeds some defined threshold, the system decides that the 2 frames represent a camera cut operation between two consecutive shots.

Originally, following the results of the initial trials, the system gave poor results in few cases. One problem was that this first trial algorithm makes use only of the global color distribution information. This results in the system not discovering some cut detection even when there is an actual cut detection in the compared frames. The two examples, shown in figure 3-2, illustrate this problem, indicating that previous algorithm ignores the locality information of the colors' distribution, especially for the case of

two frames of different shots within the same scene (i.e. the background color information is normally similar). That means that the 2 consecutive 6 MSBs color histograms are similar, despite the fact that the actual spatial distributions of the colors are so different especially, as mentioned, when two frames are extracted from two different shots within the same scene. Therefore, we extended the algorithm to suit these circumstances by making use of partitioning each frame into a number of disjoint blocks.

This allowed us to make use of the locality information of the histogram information and in a quick decision making manner. The system evaluates every corresponding two-block histogram difference between the two compared frames. We use the following equation so that the system evaluates the mean of every corresponding two-block histogram difference between the compared frames to represent the overall two-frame histogram difference ($f_1$ , $f_2$). Figure 3-3 shows the solution to the mentioned problem.

Histogram Difference ($f_1$ , $f_2$) = $\Sigma\ \Sigma$ Histogram Difference ($b_{1ij}$ , $b_{2ij}$) / $L^2$ for i , j = 1 To L

Where, Histogram Difference ($b_{1ij}$ , $b_{2ij}$) is the histogram difference of each two corresponding blocks (sub-images) $b_{1ij}$ , $b_{2ij}$ of the two frames $f_1$ , $f_2$, evaluated similar to the previous equation,

L = number of horizontal blocks = number of vertical blocks

Another issue, which we handle, is the false detection problem. It occurs here mainly because of the use of the temporal skip during processing. If the temporal skip is significantly high, along with a quick change in the same continuous shot due to object movement or camera operation, such as tilting, panning or zooming, the algorithm will mistakenly recognize the frames as significantly different, and so we use an additional step. After the first cut detection process, we need to analyze the specific changed frames more comprehensively [ZHA93], provided that the temporal skip is already greater than one. We do this in order to compensate for the speed of possible object and/or camera movements, taking into consideration that the use of block difference magnifies the effect of any object or camera movement. Subsequently, we re-analyze the frames within this region but with the temporal skip equal to one. Hence, the system needs to extract all the frames included in the current analyzed region. The algorithm

thus became more accurate in rejecting false cuts obtained from the first process while maintaining true camera cut accuracy.

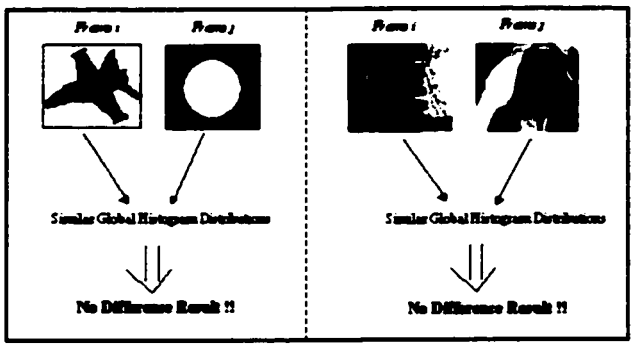


**Figure 3-2.** Wrong Frames Unchange Decision !

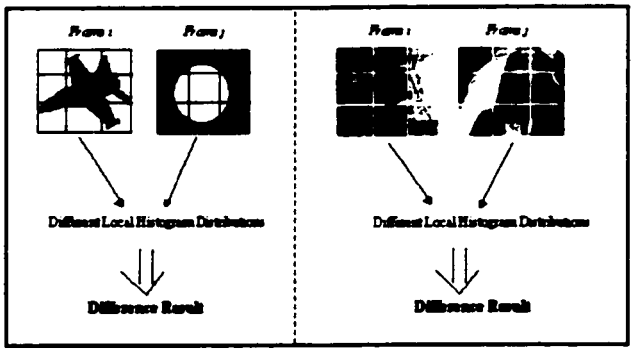


**Figure 3-3.** Correct Frames Change Decision.

Now, the algorithm could be described as follows for every two frames, taking the temporal skip into consideration (i.e. separated by (the temporal skip – 1) number of frames):

```
Total_Blocks_Difference = 0 ;

For BlockX = 0 To (H_Number_of_Blocks – 1)
For BlockY = 0 To (V_Number_of_Blocks – 1)
// where,       H_Number_of_Blocks is the Horizontal Number of Blocks
// and          V_Number_of_Blocks is the Vertical Number of Blocks

{
      Histogram1(j) = 0 ;                    for j = 0 to 63
```

```
        Histogram2(j) = 0 ;                    for j = 0 to 63

        For Row = BlockX . (Frame_Height / H_Number_of_Blocks)  To (BlockX + 1) .
                (Frame_Height / H_Number_of_Blocks) Step Spatial_Skip
        For Column = BlockY . (Frame_Width / V_Number_of_Blocks)  To (BlockY + 1) .
                (Frame_Width / V_Number_of_Blocks) Step Spatial_Skip
        {

        Frame1_Pixel_Color(Row,Column)=      Six_MSB(R1(Row,Column),G1(Row,Column),
        B1(Row,Column)) ;
        Frame2_Pixel_Color(Row,Column)=      Six_MSB(R2(Row,Column),G2(Row,Column),
        B2(Row,Column)) ;
// where, Six_MSB(R, G, B) Function is used to get the 6 MSB equivalent of the R, G, B
// input parameters (i.e. return a value between 0 and 63)

        Histogram1(Pixel_Color) = Histogram1(Frame1_Pixel_Color(Row,Column)) + 1 ;
        Histogram2(Pixel_Color) = Histogram2(Frame2_Pixel_Color(Row,Column)) + 1 ;
        }

Block_Difference = ½ . Σ | Histogram2(i) – Histogram1(i) |  /  Σ Histogram1(i) ; // for i = 0 to 63

Total_Blocks_Difference = Total_Blocks_Difference + Block_Difference ;
}

Final_Difference_Avg = Total_Block_Difference / (H_Number_of_Blocks .
                        V_Number_of_Blocks) ;
IF Final_Difference_Avg > Threshold AND Temporal_Skip >1 THEN
{
        Temporal_Skip = 1 ;

        Extract all the current region included frames ;
        Repeat the process again using Temporal_Skip =1 ;          // re-analyze again
}
IF Final_Difference_Avg > Threshold AND Temporal_Skip = 1 THEN
{
        Decision of Cut Detection ;
        Exit ;
}
Else
{
        Decision of NO Cut Detection ;
        Exit ;
}
```

### 3.4. Binary Penetration Algorithm

Although the use of the six most significant RGB bits, with the use of Blocks Intensity Difference algorithm, provided us with efficient and robust results, it lacks the performance objective required in these kind of high-volume processing systems, especially in distributed architectures. For this reason, we updated the algorithm with a new approach using the temporal dependencies within the visual information. Thus, we designed and implemented the Binary Penetration algorithm.

The idea behind this algorithm is to relax the step of analyzing all the consecutive frames of a certain region, which means that the algorithm suggests that they may have a potential cut. The previous algorithm extracts and analyzes all the frames (i.e. equals to the temporal skip) when the histogram difference exceeds the threshold. However, in our new approach, we extract and analyze the frames in a binary penetration manner as shown in Figure 3-4.

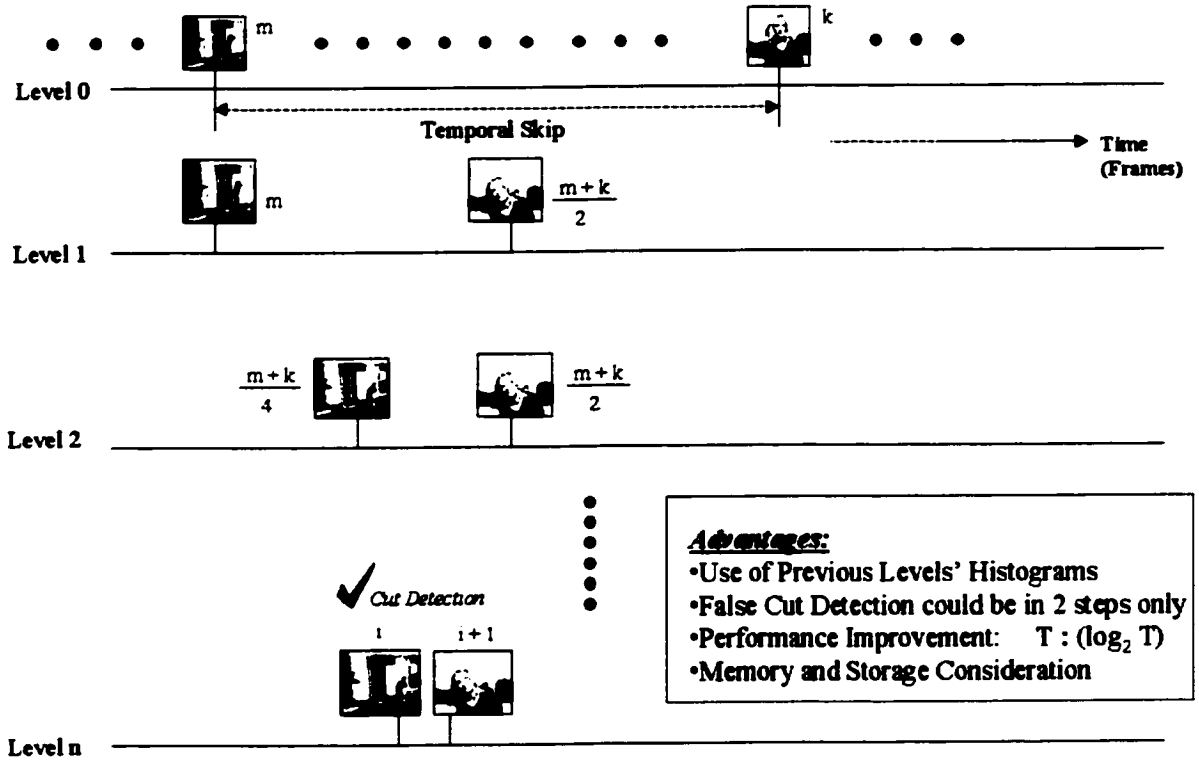



**Figure 3-4.** A Camera Cut Detection Scenario Using the Binary Penetration.

For example, initially the algorithm compares frames m and k which are separated by the temporal skip of frames. Then, if the difference in the two frames exceeds the threshold, we extract the middle frame and compare it to both ends. If both differences are less than the threshold, we conclude that there is actually no cut within this region and continue processing for the following regions. However, if one of the two differences exceeds or even if both of them exceed the threshold, we take the greater difference to represent a potential possibility of finding a cut within its half region. We need to stress that the temporal skip should be a moderate value, as will be evaluated later in the testing and valuation chapter, so that it represents a possibility of finding only one camera cut operation at maximum within it. Then, we continue the same procedure with the selected half in the same manner. This continues until we get the difference of two consecutive frames exceeding the given threshold, which represents a true camera cut operation. In addition, the procedure stops when the difference in both halves, at a certain level, are lower than the threshold. That could mean that there is actually no camera cut operation in this whole region as there could be a large object or camera movement, which happened within the given region.

Thus, taking into consideration the use of the temporal skip, the difference could pass this cut detection test in high levels but this procedure will recognize the false cut in any of the lower levels. Thus applying the binary penetration algorithm as seen in figure 3-4 to the video cut detection problem as shown in figure 3-1 could be modeled in figure 3-5.

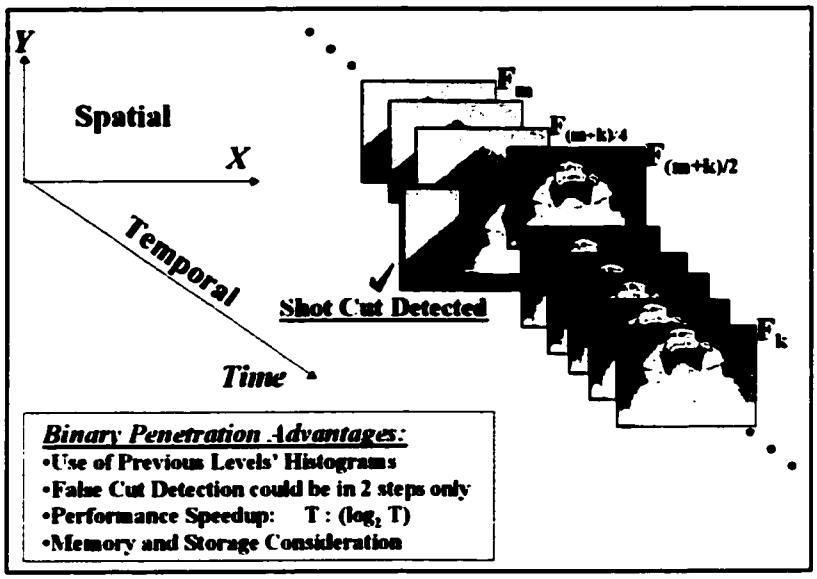



**Figure 3-5.** Applying the Binary Penetration Algorithm for the Video Cut detection problem.

Figure 3-6 shows a flowchart of this binary penetration algorithm. The terms used in the flowchart are defined as follows:

| | | |
|---|---|---|
| initialF | = | starting frame number of the entire processed video segment. |
| finishF | = | ending frame number of the entire processed video segment. |
| startR | = | starting frame number of each processed region within the video segment. |
| endR | = | ending frame number of each processed region within the video segment. |
| Finish | = | boolean variable to denote the end of the overall processing. |
| lowF | = | intermediate starting frame number in each level in the binary algorithm. |
| midF | = | intermediate middle frame number in each level in the binary algorithm. |
| highF | = | intermediate ending frame number in each level in the binary algorithm. |
| KeyF | = | detected key frame number which represents a camera cut effect. |
| Temporal_Skip | = | temporal skip parameter value. |
| threshold | = | assigned histogram threshold parameter to detect histogram changes. |

Following the algorithm's flowchart shown in figure 3-6, although we use a binary penetration search approach, the selection of the temporal skip parameter, in figure 3-4, does not necessarily have to be a multiple integer exponent of 2 (i.e. the temporal skip parameter value does not have to be equal to $2^n$, where n is positive integer number). The binary penetration algorithms, both the original and the generalized, will still work smoothly with any integer temporal skip value.

## 3.5. A Generalized Binary Penetration Algorithm

The previous algorithm description handles the most likely cases for detecting video cuts of consecutive shots. However, the use of the temporal skip parameter makes other scenarios possible. Figure 3-7 shows the important cases of camera editing within a video segment. We could define these cases as follows:

- Case 1: the entire temporal skip region of processing is included within one continuous shot.
- Case 2: one true camera cut exists within the temporal skip period.
- Case 3: more than one camera cut exists within the temporal skip period. The editing operation includes the transition between at least three different camera angles
- Case 4: again, more than one camera cut is found within the temporal skip period. However, in this case, the transition returns to the first camera angle after one or more other camera shots. An example is the interview-like scenario in news or documentary videos.
- Case 5: the start and/or end frame of a processing region coincides with the camera cut effect.

The "binary penetration" algorithm will recognize cases 1 and 2 easily. In case 5, the algorithm still works smoothly, using any temporal skip value, because the last frame of the previous processed region is itself the starting frame of the following processing region. So, even if we cannot detect the shot cut effect from the previous region, the algorithm will still detect the change effect in the following region. For cases 3 and 4, care is needed; as a result, we made a simple generalization to the algorithm.

This generalization simply just changes one step. Instead of choosing only the half region of the higher evaluated differences in any level and which exceeds the defined threshold, we need to continue the penetration in each half of the level. This is done if both differences exceed the threshold, not necessarily only the higher of them. Therefore, these modifications allow the system to discover all the cuts within the temporal skip frames, even if there is more than one such video cut.

In case 4, it is more likely that camera cuts will be missed, thereby reducing the recall accuracy. We could decrease this effect by reducing the histogram threshold value to pass the first level and discover the actual cuts in the next levels. However, we still need to remember that, in both cases 3 and 4, the problem is avoided if the temporal skip parameter has a moderate value, not large enough to include more than one cut effect.

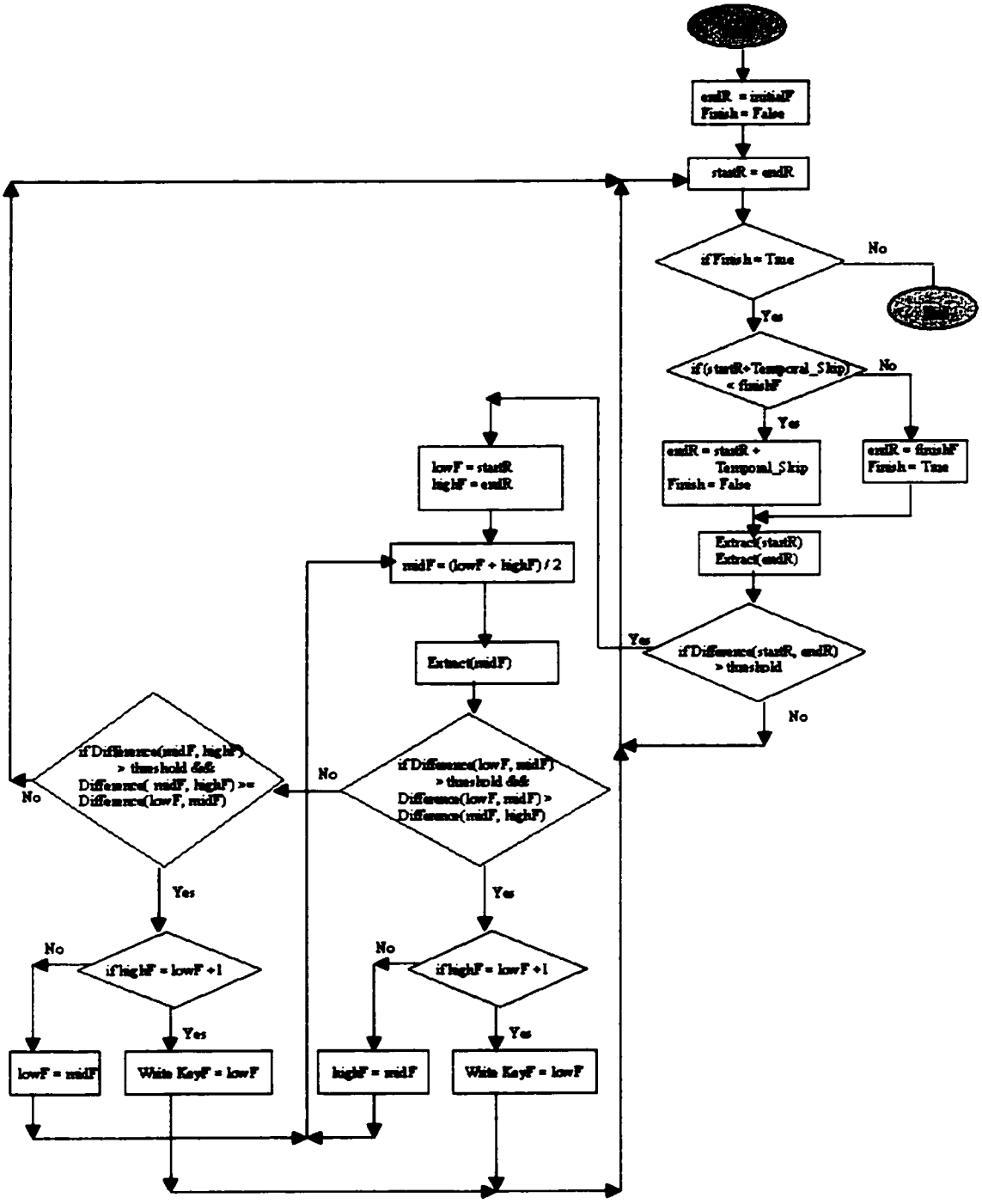



**Figure 3-6.** The Binary Penetration Algorithm.

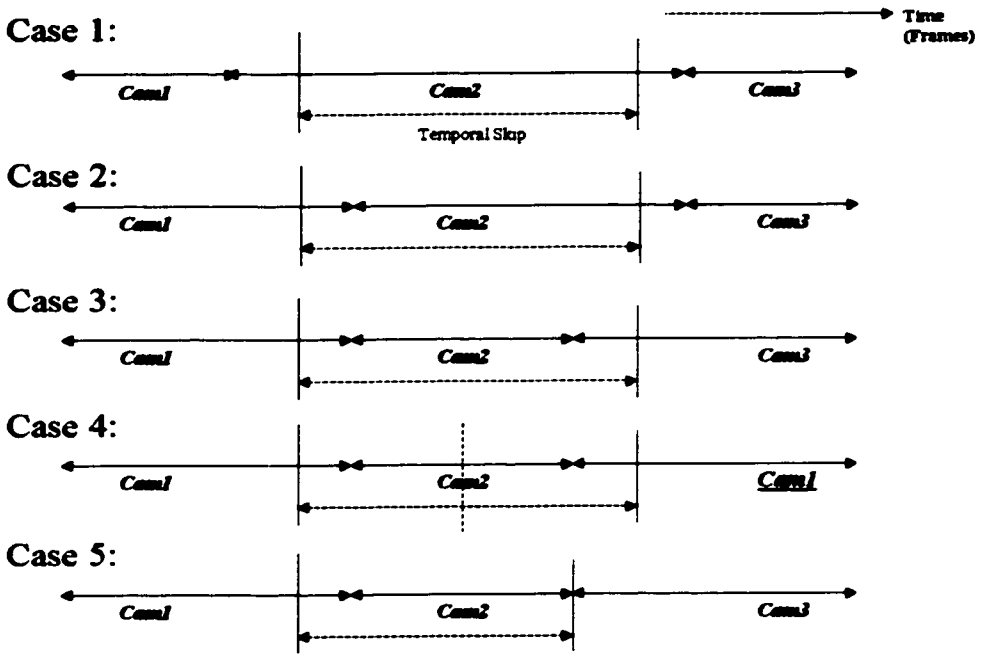



**Figure 3-7.** Consecutive Shots Cut Possibilities.

## 3.6. Key Framing System Architecture

We have developed a prototype system called MediABS addressing multiple video format visualization and analysis. The system aims to discover the different cut changes within a video file of different video encoding formats with high accuracy and better performance.

Figure 3-8 depicts the general architecture of the system. The system has four main components. They are:

1) The *Media Preparation module* recognizes and verifies the input media file format, dimensions, resolution, ...etc. We also use this module to handle different video formats seamlessly. Therefore, we implemented this preparation function to unify the next processing functions irrespective of the input media format. A third responsibility of this module is browsing using different normal VCR operation such as slow motion, fast forward, ...etc.

2) The *Media Analysis module* implements two different key framing algorithms to study their relative merits, namely six Most significant RGB bits with the use of Blocks Intensity Difference through blind search and the other implementation through binary penetration as we explained. It also updates the key framing index list of the required input media segment. The *Media Analysis module* provides the resulting key framing index list to the *Key Frames Handling module.*

3) The *Frames Management module* handles the media format segment, which is used for the analysis. The system utilizes this module to extract certain frames from the input segment. The *Media Analysis module* requests certain frames within a defined segment from the *Media Analysis module.* The *Media Analysis module* specifies the region of frames, temporal skip, frames quality, color/gray condition and frames format. The *Frames Management module* executes the frame extraction function itself. It then replies to the *Frames Analysis module* request with the extracted frame references (i.e. their locations on the local storage). We make use of partitioning the file into discrete regions to be analyzed separately to compensate for the storage requirements.

4) The *Key Frames Handling* module could use the *Frames Management module* services again later to extract the actual frames from the media file by passing their index received in the Key Frames list from the *Media Analysis module.* The *Key Frames Handling module* is responsible of rendering the key frames result report, including the total processing time and the key frames themselves to the user.

We tried to separate the different modules as most as possible according to their functions to facilitate future updates in any module without affecting the other ones. The mechanism of processing is that each module could request services from the other modules in consistent interfaces of coordination. Thus, for example, we could support new media formats to be processed without changing the media analysis algorithm. Another benefit is to change the media analysis algorithm itself for the current supported media formats.

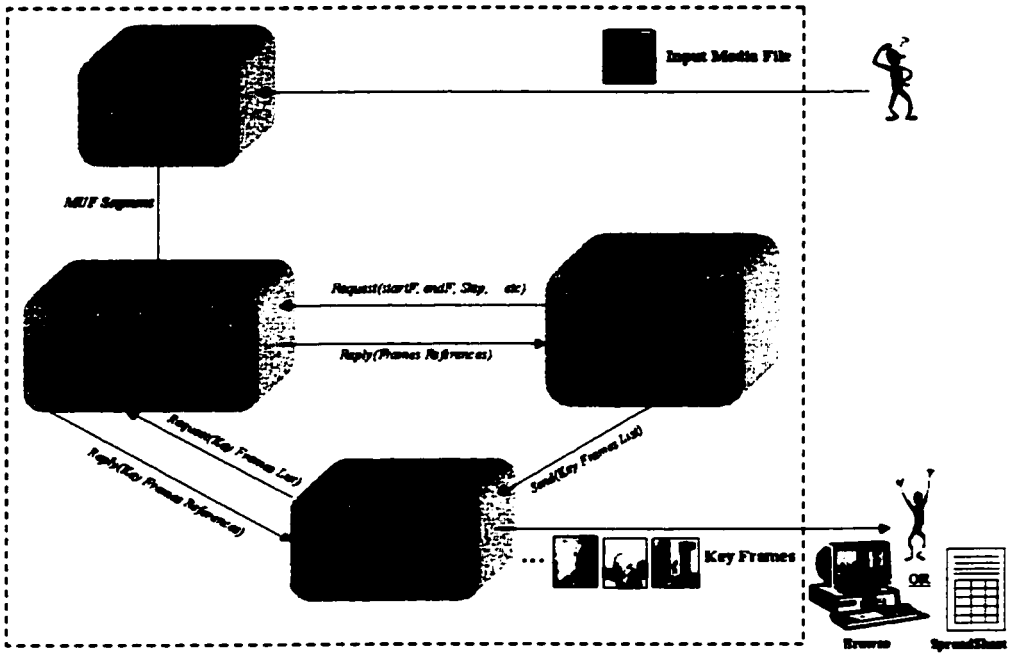



**Figure 3-8.** Media Key Framing System's Architecture.

## 3.7. Media Unification Pre-Processing Stage

The system uses a media unification phase for different media encoding standards before analyzing the video file. These standards are the Audio Video Interleaved (AVI), Apple Quick Time for Windows (QTW) and Motion Picture Experts Group (MPEG). The use of this process allows us to simplify the media analysis management afterwards, as we don't have to worry about the different characteristics of each encoding format for the following processing. Thus, irrespective of the various video-encoding formats, we use the same processing modules. This simplifies managing and analyzing the video for the different formats. The different components of this pre-processing stage are illustrated in figure 3-9.

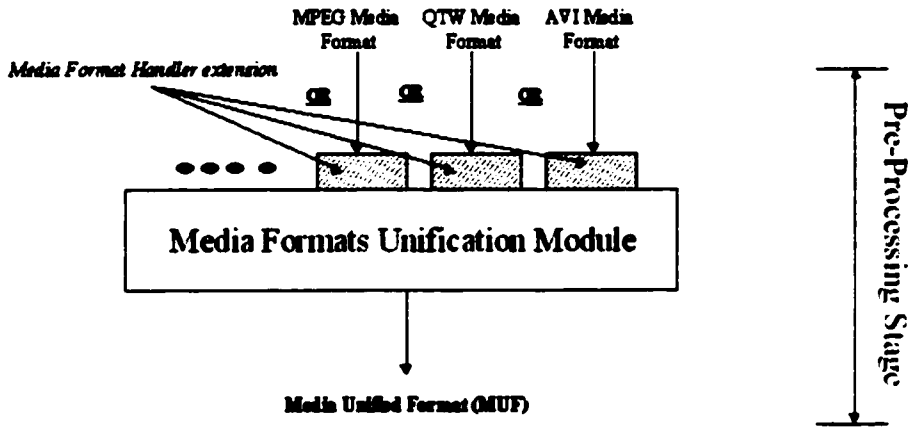



**Figure 3-9.** Media Unification Pre-Processing Stage.

The components of this stage are:

- *Media Format Handler Extension:* This component could be regarded as the media driver of each media format. They are implemented using media control interface drivers for Microsoft Windows. It handles the special format coding of the corresponding format representation. This layer could be extended to handle other media formats and devices (e.g. CD-ROM, CD-I, the new DVD standard, ...etc). Streaming media could be already processed in the same manner as well without adjusting the implemented and approved processing mechanisms of the next stages. We could verify the input media format and recognize the input media features such as: dimensions, resolution, ...etc.

- *Media Formats Unification Module:* This module is used to unify the environment and the operations that will be used for the media processing in the next stage. For example, some of the high-level function calls of this module (irrespective of the actual video encoding format) are controlling the speed of video display, enabling or disabling the audio component of the video, adjusting the audio volume level, the ability to specify certain video segments to analyze and browse, ...etc.

The uncompressed output of this media pre-processing stage is regarded as a *Media Unified Format (MUF)* segment that will be used in further media handling operations. The use of this format will reduce the complexity of handling different input standards since we don't need the special characteristics of these formats in the current implemented media processing algorithms. Thus, for example, we don't have to worry about the format to access certain frames within the video with certain temporal skip though we know that the frames coding of the information is very much different between MPEG and AVI video formats for instance.

### 3.8. MediABS Function Description

The functions and capabilities of the above interactive video segmentation and key framing system are:

- Media Segment:
  -Play, Pause, Resume, Stop, Repeat and Audio Volume Control
- Media Speed Adjustment
- Sound on/off
- Video Frames Random Access:
  -GoTo, Next, Previous, Begin, Middle, End, Skip and Scroll
- Verifying the Input Media Format
- Specified Frames regions Extraction and Analysis
- Temporal, Spatial skip Processing- Defaults: 5 frames, 5 pixels respectively
- Frames Save Formats: JPG, BMP: Color, Grey-scale, various Qualities.
  -Each compressed frame size is about only 4KB
- Frames Analysis:
  - 6 Most significant RGB bits with the use of Blocks Intensity Difference
  - The use of Binary Penetration algorithm
- Exporting Video analysis configurations and results to a Spread Sheet file

The main user interface window of the interactive system is shown in figure 3-10. The user could define a certain video segment to be analyzed. Using the temporal and spatial information redundancy within the video, we could improve the performance without sacrificing the accuracy of the algorithm results.

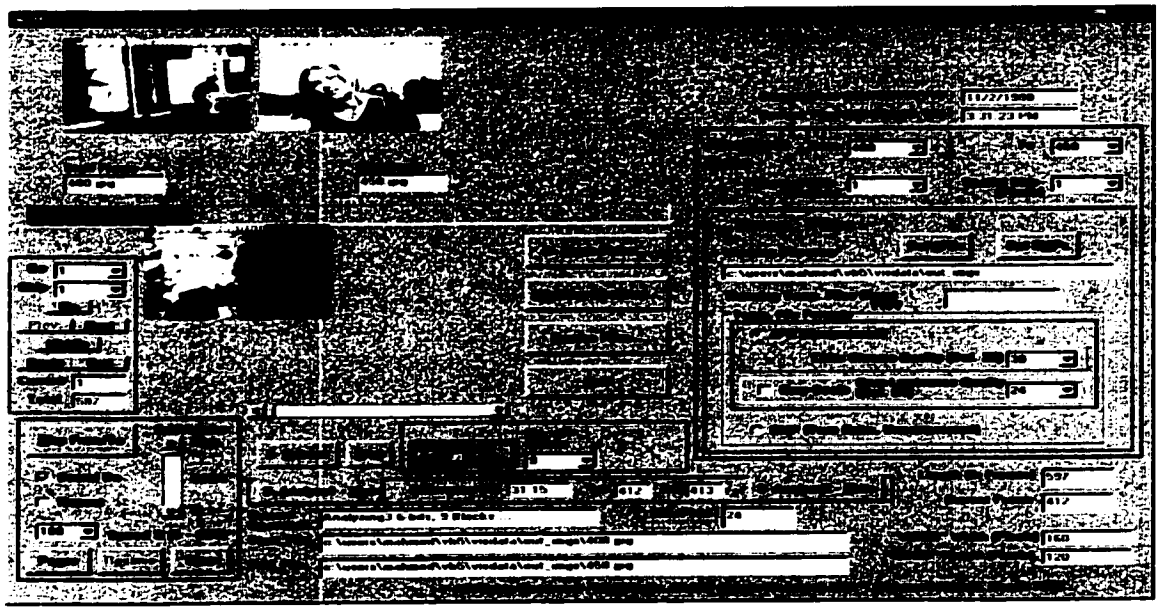


**Figure 3-10.** MediABS Main Window.

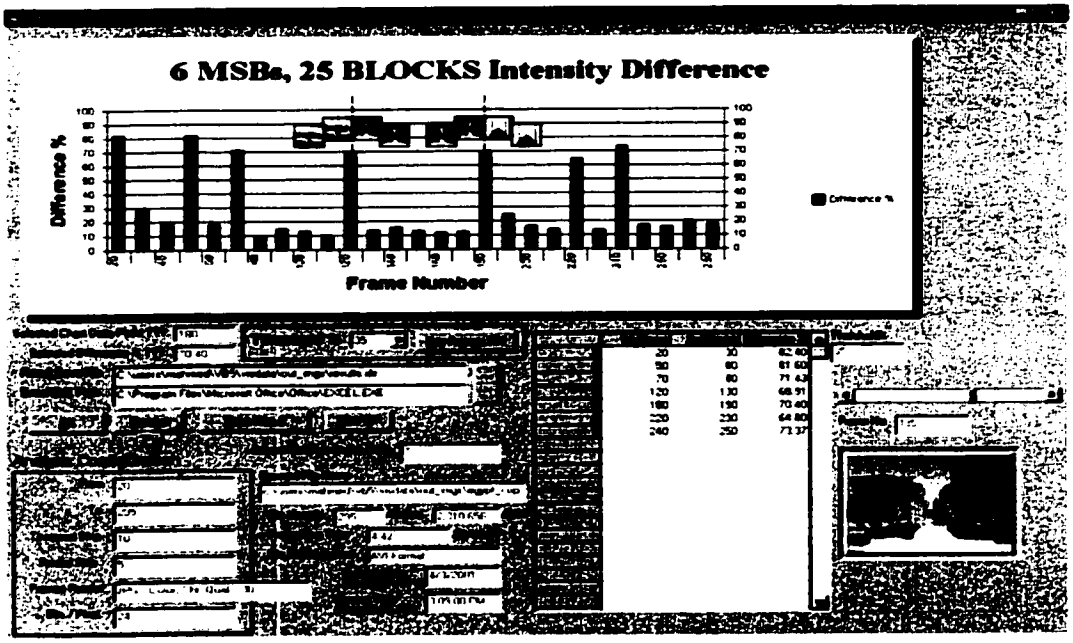



**Figure 3-11.** MediABS 6 Most significant RGB bits with the use of Blocks Intensity Difference.

Figure 3-11 presents a snapshot of the processing result using the "6 Most significant RGB bits with the use of Blocks Intensity Difference" algorithm.

## 3.9. More Performance Issues

Currently, in the era of worldwide telecommunications and wireless communications, we have to find new ways to utilize the algorithms and tools that exist and proved efficient. In spite of their effectiveness, other issues should be considered. One of these issues is the performance in terms of processing time and resource requirements. In this thesis, a special care has been adopted towards achieving adequate results and higher performance in the domain of video segmentation and visual indexing of video data. Some of the related issues concerning the performance and efficiency of video segmentation and indexing are:

### 1) Spatial Skip:

We utilize the redundant and high correlated spatial content of the visual track of video segments. Within the frames, we improve the processing time by handling part of the available visual data instead of the whole available data. Thus, we could select number of the pixels within the frames for the comparison instead of processing all the pixels. That is achieved using a spatial skip parameter.

### 2) Temporal Skip:

The nature of video data has great redundancy and correlation, especially among the consecutive frames within the temporal axis. Usually, the adjacent frames within a continuous shot have large similarity and correlation. Thus to segment the video into separate shots, instead of using all the frames within the video, we extract and use every certain number of frames. The number of skipped frames is determined by a temporal skip parameter.

### 3) Binary Penetration Approach:

Using the heuristics of video's visual information, more improvement is realized in the temporal domain. Thus, in this chapter, we introduced the Binary Penetration algorithm. Instead of processing the video in a random blind manner to detect the instances of video cut editing, we apply the binary penetration to utilize only the promising regions. Thus, we penetrate the video contents in a binary fashion till we approach the existed visual cut as described in the algorithm. However, we have to understand that this approach is useful in detecting sudden cuts within the video and not for detecting transition effects such as zooming, panning, fading, ...etc. However, we have to remember that we work in a limited environment of wireless resources and huge video libraries. In addition, it is found statistically that the video editing of more than 80% video effects are sudden cuts, which follow successfully with the algorithm. Otherwise, we could augment the system with the available complex video transition detectors. However, the testing results in chapter 6 will provide good effective and accurate results in addition to the processing time improvement for a random set of video files from different domains, contexts and video editing operations.

### 4) Dynamic Binary Penetration Approach:

In our design and implementation of the binary penetration algorithm, we assume using a fixed optimal temporal skip parameter. However, there is another argument that we could use a dynamic temporal skip while processing the content of the video. That means that the temporal skip parameter varies while processing the video. One method to achieve that is using the separation of the last two detected cuts separation in terms of number of frames to be the new temporal skip to detect the next video cut and so on. Thus, the parameter is becoming dynamic with the contents of the video. However, we eliminated this idea because of two reasons. Firstly, using this approach, we have to use the generalized binary penetration approach, which is previously described, instead of the original binary penetration algorithm. That would impose more timing and processing overhead on the algorithm. Secondly, the control mechanism of using the dynamic binary penetration will add more overheads as well because we expect

a high correlation between the consecutive frames rather than the consecutive shots or scenes. Thus, it is expected that the use of dynamic parameter will lead to less efficient algorithm than the proposed one.

**5) Parallel Processing:**

Additional space of performance improvement for video processing still exists. For example, the visual information is usually consisting of separate sequential shots. This could allow us to partition the main segmentation process into parallel processes without affecting the integrity of the results. As shown in figure 3-12, the video segmentation process could be done faster if we apply parallel processing by segmenting the video data into separate chunks and apply the video segmentation algorithm to each process. This operation combined with the use of the binary penetration algorithm could give faster results without affecting the accuracy of these results although it will need more control procedure to partition the job, arrange, synchronize and aggregate the results. However, such control time is relatively small relative to the video analysis consumption time itself. The same analysis could be applied as well for the other possible video processing algorithms regarding the audio processing and text caption detection and analysis. However, an important point here is that the use of parallel processes means more difficult debugging, more memory and storage requirements to handle the extracted frames in each process. Thus it is a trade off between the performance speed and memory, storage, inter process communication (IPC), synchronization points overheads and the difficulty to debug. Nevertheless, if the memory and storage capacity is large enough along with small IPC and synchronization points overheads, we better use the parallel processing approach using number of available processors according to the available capacity.

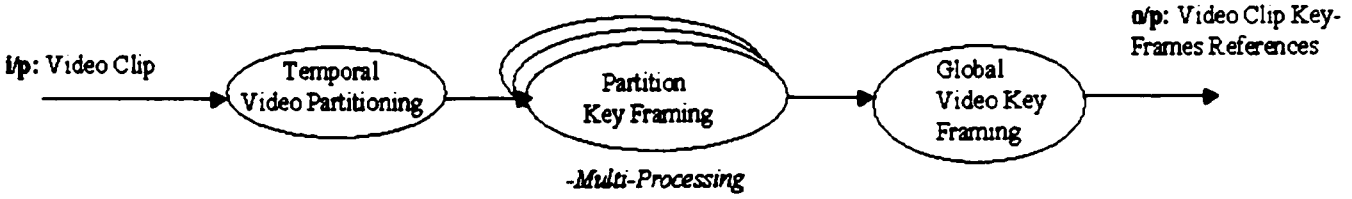



**Figure 3-12.** Applying parallel processing approach in video segmentation.

The speed of a program is the time it takes the program to execute. This could be measured in increments of time. The speedup parameter is defined as the time it takes a program to execute in serial (with one processor) divided by the time it takes to execute the same program in parallel (with many processors). Thus, the formula for speedup is:

$$\text{Speedup} = \frac{T(1)}{T(M)}$$

Where T(M) is the time it takes to execute the program using M processors. Another parameter is the parallel efficiency, which equals the observed speedup, divided by the number of processors used. This is an important parameter to consider. Due to the cost of multi-processor super computers, an organization usually wants to get the most efficiency for the system's cost.

Now, we'll derive a speedup formula. If there are M processors working on a program code, we may ideally assume that each one would be able to do its job in 1/M time of one processor working alone. However, if we assume the serial part of the program is performed in K . T(1) time units, that means that the parallel part is performed in [ ((1-K) . T(1)) / M ] time units. Hence, with some simple substitution, we get the formula for speedup as:

$$\text{Speedup} = \frac{M}{(K \cdot M) + (1-K)}$$

This formula is called Amdahl's law [GUS88]. As an example:

If M = 8 processors = ideal speedup, K = 15% = percentage of serial code

Then, $$\text{Speedup} = \frac{8}{(0.15)(8) + (1 - 0.15)}$$

$$= 3.9$$

However, the observed speedup could be less than this number because of IPC overheads, parallelism control, imperfect concurrency of synchronization points and time sharing of these processors with other unrelated processes.

## 3.10. Summary

In this chapter, we described the original algorithm that we initially used in our system for video segmentation, indexing and detecting key frames within a video segment. We adopted the approach of partitioning the selected frames into exclusive blocks for resolving the missed cuts problems. The system implements the use of spatial and temporal skips for improving the performance of the analysis operation. However, to compensate for the use of the temporal skip, the original algorithm re-analyzes all video region frames if there is a potential to find a cut within it in the first test. Then, we explained the updated algorithm using the binary penetration mechanism. The algorithm, which is based on dichotomy ad hoc image processing, makes use of the temporal correlation heuristics of the visual information within a video stream. The new approach, as we will show from the testing results in chapter 6, results in a better performance improvement over the original algorithm without sacrificing the accuracy results of the analysis.

We presented an extendable system that we developed for the purpose of video browsing and detecting camera cut operation to represent the visual key frames within video visual information. The architecture and the modules within the system were described along with few snapshots of the interactive video segmentation and key framing system. We provided the functions and services of each module of the video segmentation architecture. In addition, we explained the scenario of data and information flow among the modules of the system and we stated the implemented function list of the interactive prototype as well.

# CHAPTER 4

# MEDIA ABSTRACTION PROCESS DESCRIPTION

## 4.1. Introduction

In this chapter, we will describe our approach to realize the multimedia abstraction function as a service for our agent-based project. We will use the video segmentation algorithm and its sub-system, which were described in the last chapter, within the video summarization procedure. Important contribution in our work is that we will account for the uncertainties of the different factors required to perform video parsing, analysis and then summarization. Another contribution is to resolve the conflicts that may arise due to noisy data or uncertain decisions.

First in the following section, we will provide our abstraction model.

## 4.2. Media Abstraction Model

The use of shot cut detection and consequently video key frames, as described in the previous chapter, is regarded in some literature as a low-level summary of the video stream. However, we could enhance the semantic recognition of the analysis and deliver higher level of summarization, as we will describe within this and the following sections. In figure 4-1, we present our high-level approach to achieve the high-level media analysis and abstraction process. We mainly need three general steps to achieve the summarization goal. They are defined as follows:

1- **Parsing:** There are different available automatic tools that are useful to drill down the video contents to parse them. We mainly handle three types of information within the video segment. They are:

   a) *Phrase Segmentation* [BEY98] *and audio effects detection* [ZHA98] [ZHA99]: using speech recognition tools for the audio information. Within a certain domain, some words could be defined as topic markers for different events in the domain. Thus, this could be useful in indexing the video contents more semantically. In addition, the classification of certain acoustic features such as: applause, cheers, ...etc could lead us to interesting segments in the video. There are different tools that could be utilized such as SPHINX-III [PLA97], Dragon NaturallySpeaking [COR98], ...etc.

   b) *Shot Segmentation and object recognition:* we use the visual information of the video data to segment the video into consecutive shots. In chapter 3, we presented our high-performance algorithm to segment video data into separate shots using visual temporal reasoning. Also, in each domain there are certain object templates [SWE95] that are useful to be recognized. We will provide examples in the soccer game domain in a next section.

   c) *Text-Caption Recognition:* We believe that the text contents within the visual information, specially the computer-generated text editing, represents good opportunities of recognizing useful information, events' description, comments or concurrent statistics. However, the nature of the text within the video could be classified as either computer-generated such as the score of a game or comments within a news segment, ...etc. or natural texts such as product names or text

on shirts, ...etc. It is not just a high contrast text on different blank background but rather they are text with different fonts and size and orientation on complex background. Thus, normal Optical Character Recognition (OCR) systems are not reliable. Thus, some researchers took these effects within their algorithms to recognize the embedded text within complex backgrounds with some high accuracy such as in [WU97] [SAT99].

However, there is an important issue that we take into consideration that most researchers unfortunately ignore when performing the summarization task. It is the confidence in current available tools. Even, the well-reputation tools don't generate 100% perfect results and moreover their result accuracy may vary from certain domain to another or even from different video segments of the same domain. Thus, in our reasoning we take the historical or manufacturer's confidence level of our tools into consideration to judge the results out of them. Hence, if we later put more robust and accurate tool into the system to perform a certain function, we raise the confidence of its function results relatively.

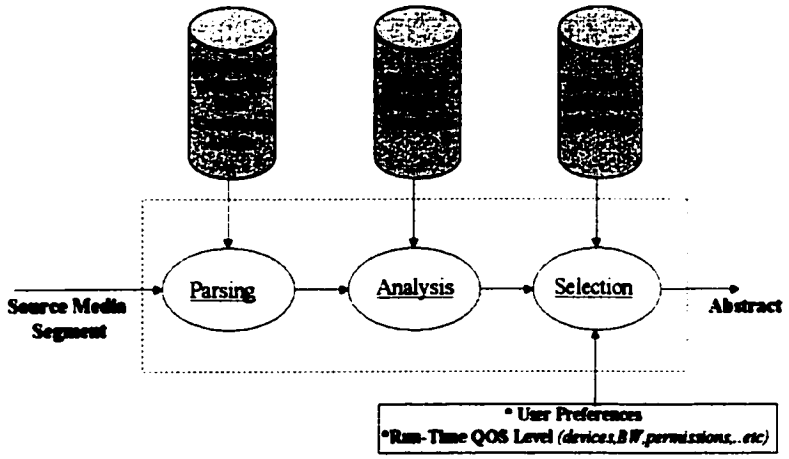



**Figure 4-1.** Steps of media abstraction solution.

Meanwhile, as we presented in chapter 2, a lot of related work nowadays aims to standardize descriptions of multimedia content. Therefore, this will make the decoding and parsing processes more efficient and accurate. One of these standards is MPEG-7. MPEG-7 standard [ISO00b] official name is "multimedia content description interface". As we said, MPEG-7 doesn't replace predecessor standards

such as H.263, MPEG-1, MPEG-2, MPEG-4. It is supposed to add complementary functions to them especially with MPEG-4 standard [SIK97] [ISO00a], that already defines object-based representation, modeling and access. In addition MPEG-4's encoding syntax supports the inclusion of user data in the encoded bit stream. MPEG-7 objective is to standardize the description of multimedia data for efficient and effective access, manipulation and retrieval. It tries to unify the description schema of the multimedia contents so that they could be accessible and interoperable between different multimedia applications and to enable brand new types of tools and applications.

MPEG-7 will standardize the content description interface and representation of visual and audio contents and their relationships using different possible levels of details (low-level features to high level descriptions) through defining a group of Descriptors (Ds), Description Schemes (DSs), Description Definition Language (DDL) and the encoding schemes of the descriptions. However, it will neither standardize the tools used for description generation such as feature extraction and segmentation tools nor the tools or the applications that utilize these descriptions such as search engines and recognition systems. It will be open for each user of the standard to adopt the parts and descriptions that satisfies his application objectives.

It is expected that a decoder that is MPEG-4 and MPEG-7 compliant will allow the user or other applications to access and manipulate a multimedia database efficiently and effectively without the expensive decoding, segmentation and feature extraction as these expensive processes would be done already once at the encoder site. Figure 4-2 provides a scene description example within a video stream using MPEG-7 interface DDL, description schemes and descriptors. As shown, there could be high-level descriptors that provide a title, composition information, or other meta-data about the scene that could guide us to more accurately index, segment and later summarize the entire video segment.

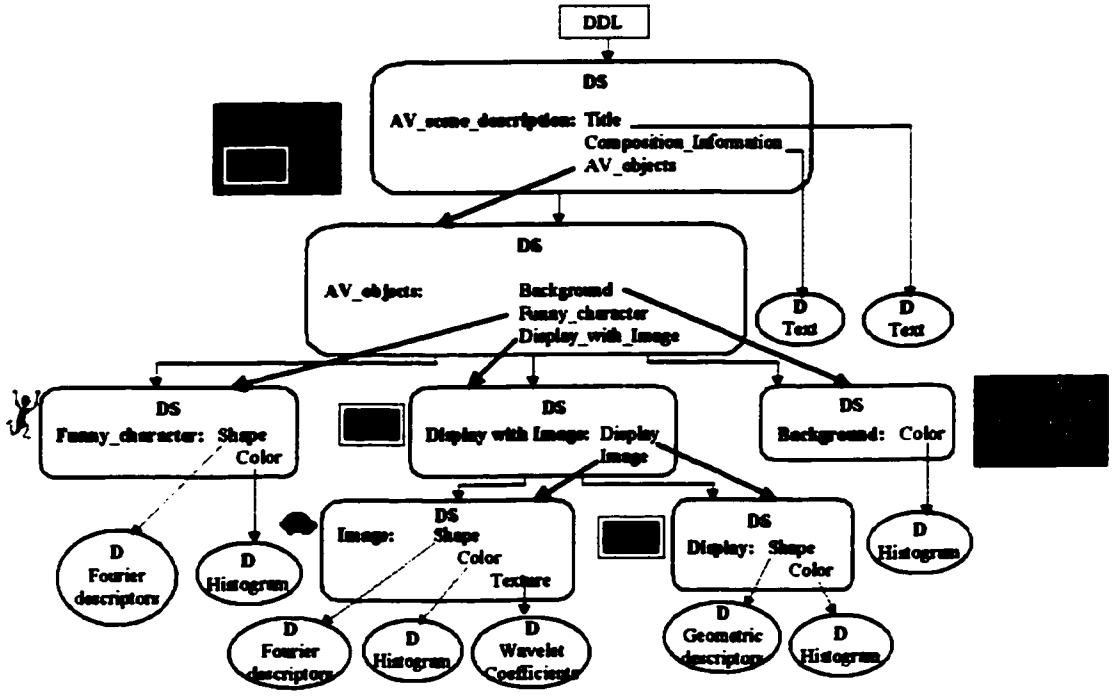



**Figure 4-2.** Example of describing a scene using MPEG-7 description interface.

2- **Analysis:** This process represents a data mining function of the video content. We may need this process to be driven by domain-related knowledge, if available, to improve the results because automatic video summarization process represents a complex task. However, if no domain knowledge is available, general knowledge about interesting events could still be incorporated. Usually, each domain has its own long-term facts and heuristics that are useful to be used. They could be described once and would not changed regularly. This could be regarded as the long-term memory of the system. This knowledge schema could be provided in the form of rules that govern the spatial, temporal and logical spatio/temporal information. This high-level knowledge still has to be mapped into technical feature decisions that are to be recognized with available technical tools. These rules would improve the accuracy and reliability of the entire abstraction task. Therefore, we introduce the following terms to be defined and indexed within the system:

*a) Clues:* They are certain low-level technical features or measurements that collectively could guide us to realize certain events within the video as shown in figure 4-3. As seen in the figure, each event has also a value of the expected minimum time between two consecutive instances of this event. This value could be certain amount of time or an "UNLIMITED" value. Regarding the clues themselves, we could have two types of clues according to available information about the video domain. They are:

*1. General-purpose Clues*: These are general heuristic features, for most domains, that could correspond to important actions within the video contents such as: *sudden* high-energy change of the audio level, text-caption *change*, camera zooming-in operation, shot-repetition, slow motion shots [PAN01] [BAB00].

*2. Domain-specific Clues:* Usually, each domain has its own characteristics and clues that could correspond into certain semantic events. For example in sports: usually a slow-motion video editing signals an important event has occurred. Different audio effects could represent high emotion of the audience or the commentator to certain actions. Distant incremental text caption change measure [LIE00] in certain area on the screen accompanies an increasing score while time advances. Certain domain-oriented clues examples for soccer domain will be represented later. However, we still respect the uncertainties of the parsing tools and their results. Thus while analyzing, only the clues which have the highest confidence multiplied by the system's general confidence in the associated detecting tool, and which exceed a preset threshold are considered for analysis.

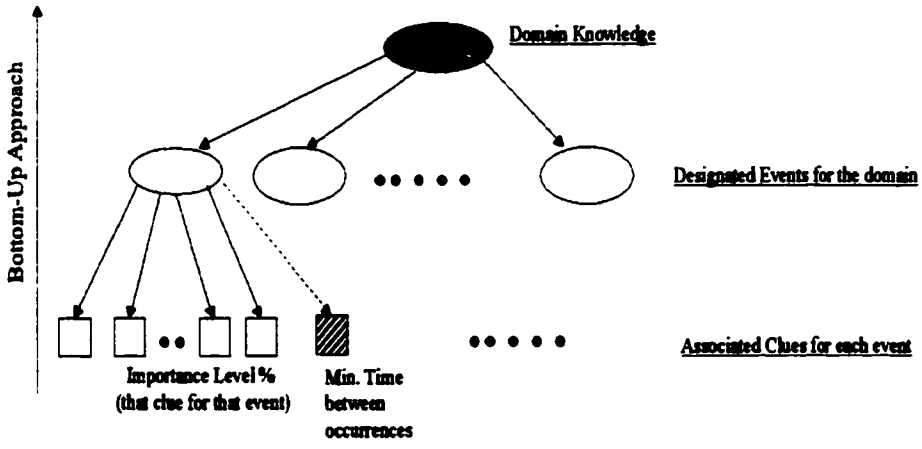



**Figure 4-3.** Domain-Event-Clues Hierarchy.

When a parsing tool recognizes a certain clue, it generates a record in the following standard format:

| | |
|---|---|
| **From:** | hh:mm:ss:oo (or in frame#) |
| **To:** | hh:mm:ss:oo (or in frame#) |
| **Track:** | video, audio, frame, and text caption |
| **Technical Confidence:** | % |
| **Tool:** | object recognition, slow motion detection, text caption recognition, face detection,...etc. |

As stated from the output record structure, we should take into account the confidence of each tool in its own results. For example, most text-caption recognition tools provide the text as an output along with a confidence percentage that the result is correct especially with complex background of the video frames.

*b) Events:* Each domain could be characterized by certain high-level or semantic events that are most likely to occur within its data sources. Thus, the analysis process is mainly used to detect and recognize the events that occur within the video segment according to a given classified list of events that are defined for this domain. It is similar to applying pattern recognition classifier to detect these events from their associated weighted *clue* features. We should take into consideration that some of these events are exclusive. That means there should be one decision for the clues that took place in certain time duration. So, if there are any conflicts, we should resolve it to choose one of the events based upon certain rules and temporal relationships. Thus in a later section, we will describe our technique to resolve the conflicts among the recognized events within certain time window.

*c) Facts:* these are heuristic information, rules and constraints of the domain. These facts will simplify the analysis and semantic reasoning of the parsed data and recognized *events*. It will guide the system to take exclusive decision when there are possibilities of two or more event decisions.
These facts represent ontology of the spatial relationships of objects, the temporal rules of the sequence of events or the logical relationships among different media contents. They could be described using N-

ary operations, as we will describe in section 4-3. Examples of the facts for the soccer game will be described in a next section.

*d) Triggers:* a trigger is a biased-clue, either visual or acoustic that is unique to certain events. Therefore, the discovery of such a trigger could drive the analysis process opportunistically to a fast and more accurate decision within a certain period of time in proximity to the time of this trigger occurrence. Each trigger is associated with a certain confident decoding method such as text caption recognition, slow motion detection. Thus, the performance of this analysis process could be much improved upon the fast search of the known triggers of the domain. For instance, the systems that implement open-vocabulary word spotting algorithms [KNI96] within audio documents could be utilized to search initially for certain important known key words or their equivalents within the video. However again, the confidence and accuracy of the corresponding tool should be higher than certain threshold.

3- **Selection:** In our approach, this step is viewed as both selection and conditioning processes. It uses another type of knowledge, which includes the events semantic value and the interests of the end-user and his expectation of the messages sent to him. Based on the features of each domain, some of its events have relatively more priority than others. Therefore, if we should try to summarize the given data, we could utilize this information to select the events that have higher priorities and/or according to the user preferences given in his profile. We still have to remember that the interests of each user could differ from one to another. Thus, should the user provide his preferences such as his interest in certain domain events and/or subjects and his interface preferences, we take them into consideration within the selection step. Another concern is the output format of the summary. Knowing the current environment's configuration, we should adapt the result to be in best media formatting possible. This could depend upon the run-time available resources, user device's capabilities, and user's permissions. In our prototype, we also take into account the organization's Intranet policies as well. This would lead us to condition the result to be either text, frames audio or video summary of the video source. The level of video conditioning and the negotiations of the different distributed components are handled using our overall prototype described in [HAR99].

## 4.3. Conceptual Dependency Modeling for Video Facts and Events

The domain's schema and facts of the video could be implemented using N-ary relations. N-ary relations [HIR97] are operators that could be used to relate different media segments and objects. They are used originally for document authoring systems for defining spatial or temporal relationships among the document objects. The list of used operations that we selected within our approach is depicted in figure 4-4. To provide a general-purpose schema editor, we could use these operators to describe few known scenarios or rules of the domain's facts and events. We use these operations to relate temporally the expected clues that describe each event as fine-grained depiction and among different events in a coarse-grained level. For example, we could specify that normally the video goal scoring video shots clue occurs **"before"** the clue of text-caption scores data modification to describe a scoring event. In addition, there is a need to represent the relationships between realized high-level event classifications themselves. For example, in soccer, the "**before**" relationships is the only allowed operators among a player substitution event and goal scoring event interchangeably.

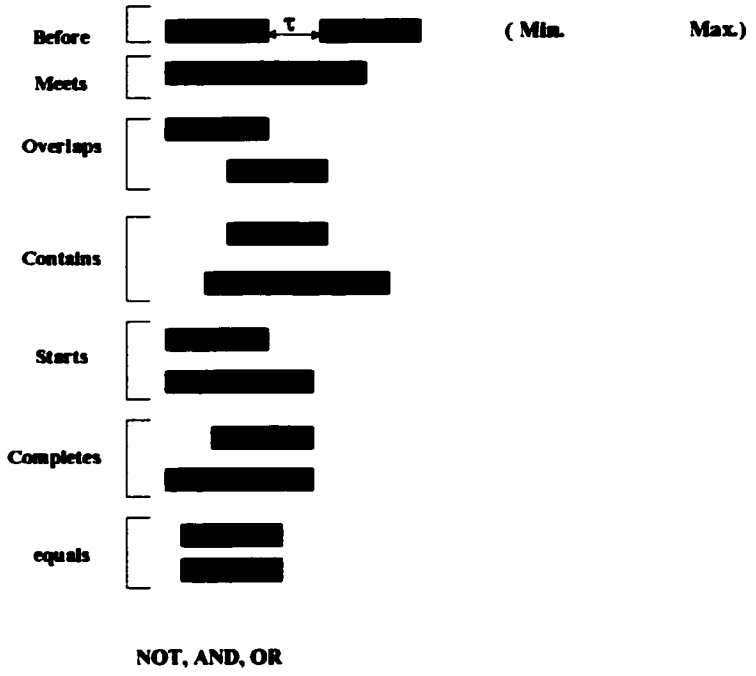



**Figure 4-4.** N-ary operators to describe dependency relationships.

However, we suggest the use of an extended model of the N-ary operators. First, the need for the use of other logical operations is required. Thus, we use "NOT", "AND" and "OR" logical operators along with the N-ary operators to provide more complex expected scenarios.

In addition, we extend the N-ary operators to provide further possible temporal constraints. There could be time limitations among the segments either fine-grained or coarse-grained to correspond to video and domain schema. For example, the different goal scoring events within a soccer game has minimum time of one minute for example. However, the minimum time for basketball games could be one second and so on. Thus, we added the minimum expected time (MIN.) and maximum expected time (MAX.) for the "**before**" relationship as two parameters of the N-ary rules as shown in the figure 4-4.

## 4.4. Example: Media Abstraction for Soccer Domain

Soccer game is a good field for automatic media abstraction. Each game is around 1.5 hour of data. A lot of its content is redundant and non-interesting data. It has a huge population of fans around the world. Many are interested to receive the latest news and summaries of soccer leagues around the world in regular short message summaries. Another interesting criteria of soccer is that there could be a need to accommodate user preferences and interests in a dynamic result manner. Although, we present soccer game as a case study to apply our approach, we could still apply it for other domains such as news, movie, financial segments, live conferences and documentary segments using proper domain schema. As we've shown, we use three modals of data within the video segments. They are the speech phrases and audio effects, video shots-and-objects recognition and text-captions to represent useful clues to recognize the events occurred within a soccer game.

- The Analysis Process:

Now, we present a part of the information schema that we extracted using different examples of soccer games sources (i.e. they represent various video editing). The analysis module for this domain's video

sources uses this schema. This schema, as we defined its structure in a previous section, includes the following:

a) **Events:** represents the expected specific actions within a soccer game. These events are required to be classified through the analysis process first to understand the semantics of these actions. Some of these events are mutually exclusive and some could overlap within a short time period. We could define the following events for soccer games as one possible events list:

-2nd Half Commentary
-1st Half Commentary
-Goals
-Good chances
-Sent-Off
-Player substitutions
-Caution

We associate every event with some clues to describe this event. However, we define a weighting importance level to represent the relevance of each clue for each event. Thus, we actually try to map the high-level semantic events into low-level technical features. These could be specified upon a learning technique, such as UNIMEM learning algorithm [ELO95], from different training record sets. UNIMEM is a symbolic and incremental learning algorithm that uses training examples to infer high level knowledge sources entities (the events) and the associated minimum and sufficient parameters set, which are the clues in this case. It will also associate these knowledge sources with a parameter importance factor according to the successful repetition of these clues for the associated knowledge source, within the training set. The training set could include examples from different video editing sources and could be trained incrementally if new video editing techniques have developed. In addition, UNIMEM has the ability to drop any sudden noisy training data from the schema. The parameters' weights represent the level of importance of this clue for the event and the confidence that the algorithm feels that this clue most probably will lead to recognizing this event. An example of an event-clues-weights record is:

Event: -Goals (i.e. event1)

Clues: -Text Caption score change ($W_{11}$ = 90%)

-Audio Change ($W_{12}$ = 30%)

-Audience shot ($W_{13}$ = 20%)

-Slow motion replay ($W_{14}$ = 70%)

-Goal's lines recognition ($W_{15}$ = 30%)

-Same players' dress colors with hands raised ($W_{16}$ = 60%)

-Audio clues ("score", "goal", "ooh", ...etc) ($W_{17}$ = 40%)

where, $W_{ij}$ is $Clue_j$'s weight for $event_i$

b) **Facts:** which will facilitate the analysis process. Those are rules that govern the expected parsed results and recognized events. Those rules include spatial, temporal and logical relationships. Some of these facts are :

-Second half commentary usually gives quick overall summary.

-First half commentary describes 1st half events.

-Slow motion replay & text caption update happen after the event's occurrence. Thus, we could roll back within a time window and look for decisive clues.

-Text captions change indicates important events.

-Sudden changes of audio energy signal and pitch frequency represent good chance of important events.

-Goal score increases sequentially.

-Goals and good goal chances have separating timing.

-Only one or two commentators usually exist.

-Full game or 1st half "statistics" text-caption shot may occur after or near the end of the 2nd half or 1st half respectively.

-No new goal, good chance, carding, ...etc after the end of each half.

-Only two teams and a field referee.

-"Red Card" means a sent-off.

-"Yellow Card" means a caution.

c) **Triggers:** Some of the clues of every event are decisive clues that have high weighting level. We referred to them as triggers. However, the detection of some triggers doesn't guarantee that the associated event has taken place, as the other tools, which we use collectively may not ensure the occurrence of that event. As we have mentioned, every tool generates the clues it could detect along with its confidence of recognizing that clue.

Examples of these triggers are:

-Text Caption Change.

-Slow motion Shot.

-Certain learnt keyword phrases.

-Red or yellow card object shots.

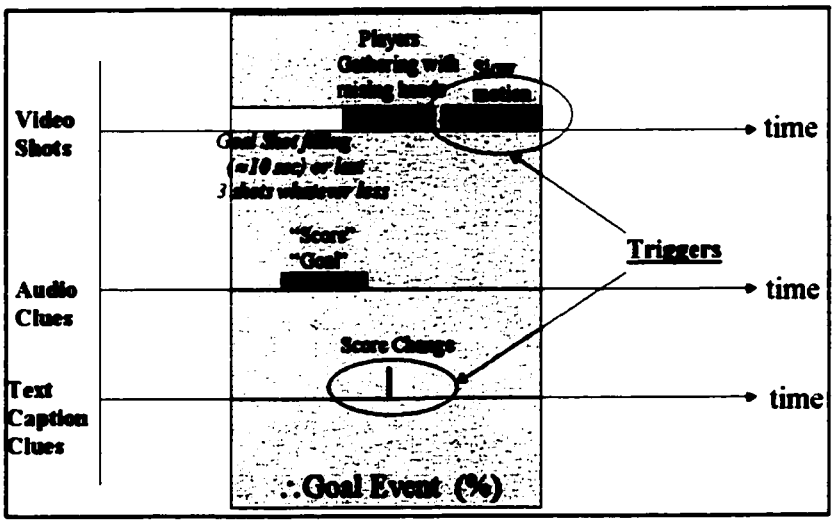



**Figure 4-5.** Goal detection clues through the timeline.

Figure 4-5 represents a typical case that clarifies the use of the clues-triggers to classify the events. As seen, this is actually a goal event example. Usually from the soccer-domain heuristics, a goal detection possibility is triggered by a sudden change in the goal score text-caption sequentially for one team. If this happens, that represents a near-certain chance of a goal. Also, but not as decisively a good chance of

important event is triggered by the occurrence of a slow motion shot "after" the event. However, it is clearly not exclusive for goal events.

Now, we will describe how to classify the events. After receiving the parsed data from the parsing process, we apply a weighting formula to evaluate the certainties (i.e. an expected possibility measurement) of the different events within the video segment around the time of the realized triggers. This event's decision formula, assuming that the parameters $W_{ij}$ , $C_{jt}$ , $E_L$ and Correlation ($clue_j$ for $event_i$) are mutually independent could be simplified as follows:

$$\text{Certainty}(event_i \text{ at time-period}_t) = \frac{\sum_{j=1}^{N_i} [\, W_{ij} \,.\, C_{jt} \,.\, E_L \,.\, \text{Correlation}(clue_j \text{ for } event_i) \,]}{N_i}$$

$$\text{Where} \quad 0 \leq W_{ij}, C_{jt}, E_L, \text{Correlation}(clue_j \text{ for } event_i) \leq 1$$

Where, $N_i$ = Number of clues for $event_i$

$W_{ij}$ = $Clue_j$'s weight for $event_i$

$C_{jt}$ = Tool's confidence in its result for $clue_j$ at time period $_t$

$E_L$ = System's general confidence in Tool L that recognizes $clue_j$

And the Correlation ($clue_j$ for $event_i$) factor represents the temporal degree of correspondence between the available triggers to the associated and found clues as illustrated in figure 4-6.

The idea is to penalize the clues that are out of the expected range (characterized by minimum and maximum time differences relative to the trigger). Thus, we could formalize this degree of correspondence as a damping function with time using the following formulas:

If (Min. Expected $clue_j$ Time difference <= Time difference ($clue_j$, $trigger_i$) <= Max. Expected $clue_j$ Time difference) Then
  Correlation ($clue_j$ for $event_k$) = 1
Else
  Correlation ($clue_j$ for $event_k$) = 1- | [(Time difference ($clue_j$, $trigger_i$) - Avg. Expected Time difference) / Avg. Expected Time difference] |

***N.B.*** Correlation (the $trigger_i$ for $event_k$) = 1

Then, we choose an event decision if its certainty percentage exceeds the other events' scores AND its score exceeds certain threshold evaluated through the training steps. Otherwise, no-event decision is adopted.

It is clear that $C_{jt}$ values could be improved if we use more accurate tools and also could be increased using different format of information encoding. For example, for such cases such as: using closed captioning with the video in addition to the audio speech track or using certain audio descriptors in MPEG-7 format. It could be increased also if we use MPEG4 video segments that directly have descriptions of the visual objects within the visual shots.

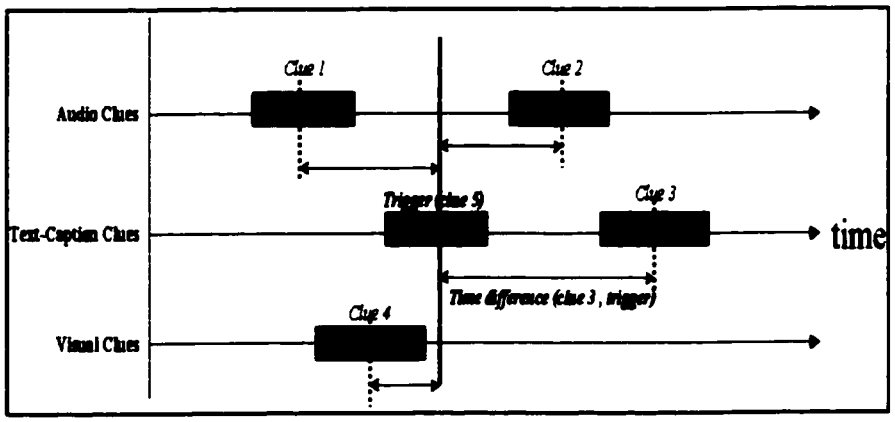



**Figure 4-6.** Temporal clues distribution derived from the parsing process.

- The Selection Process:

After we segment the entire video into different event possibilities and then respect the temporal heuristics and facts of the domain to resolve any conflicts or uncertainties; we select the segments that we include in the abstract. There are two main forces that challenge us to select the contents of the abstract. These forces are the events' importance and the comprehension of the abstract message. For example, from our knowledge about the soccer domain, we see in figure 4-7, there are important events that we need to handle with different priorities. Such as: 2nd half color commentary, 1st half commentary, goals, good chances, ...respectively and removing the commercial segments for instance. However, the user's preferences could override these priorities. Moreover, the media format itself of the summary is another orthogonal issue that could be adapted and conditioned according to the run-time environment's parameters and the nature of the recognized events. For example, using a goal event schema, we might render a goal event as a sequence of consecutive frames instead of a video stream to suit certain user's device or network resources capabilities. Meanwhile, a yellow or red card caution events could be summarized in just one frame. Thus, there could be different list of scenarios of abstraction model that the system would select from dynamically according to the working environment's factors. These factors include the available device capabilities, video domain, connection bandwidth, user preferences and permissions. Thus, there are two issues that we take into consideration in this process:

1- Respecting the user's preferences and the quality of service available.

2- Providing the content's format, either text, frames, audio or video, that is suitable to the user's environment such as his device capabilities or organizational policies.

So within our overall agent-based prototype, as we will describe in chapter 5, the different negotiating agents decide that abstraction level according to the current available QoS, user preferences, permissions, device capabilities...etc.

Important note, which we follow in providing the result to the user, is that we preserve the timing causalities of the events within the result. We time-stamp each event within the abstract with its actual occurrence time.

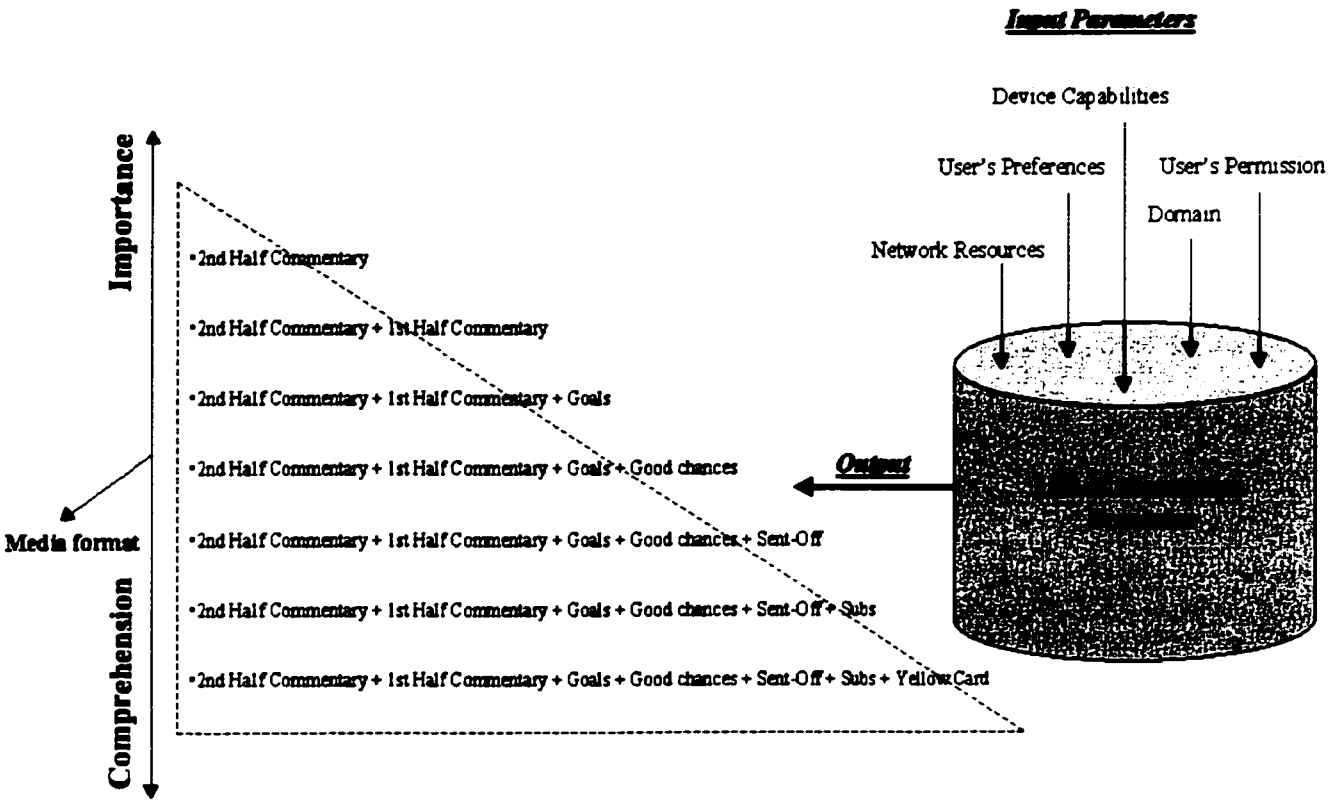



**Figure 4-7.** Possible levels of an abstraction scenario for Soccer Domain based on a priori knowledge and surrounding environment's status.

Regarding the adaptation using media format conversion tools and intelligent services. We could group them as in table 4-1. The table could be used to describe general ways or services and the possibilities of converting different media formats according to different factors such as the available network resources, user preferences, device capabilities, ...etc. In the next chapter, we will describe our effort to present media contents especially contents of video format in different ways to our overall system's clients.

**Table 4-1.** Possible services for Media Adaptation and Conversion.

| | Text | Image | Speech | Audio | Video |
|---|---|---|---|---|---|
| Text | format Conversion | Sentences on Image | Text to Speech Synthesis | Text to Speech Synthesis | Text to Speech Synthesis and Text Box |
| Image | Intelligent Character Recognition (ICR) | format Conversion | ICR/Summarization | ICR/Summarization | Still Video |
| Speech | Speech Recognition | Sentences on Image | format Conversion | format Conversion | Black Video |
| Audio | Speech Recognition/Interpretation | Summarization | Speech Filtering | format Conversion | Black Video |
| Video | Summarization/Interpretation | Summarization | Speech Filtering | Audio Filtering | format Conversion |

## 4.5. Domain's High-Level Event Conflict Resolution

To achieve high-level semantic analysis of video contents, we should consider the possibility of mistakenly realizing conflicting events. These situations could take place because of two major reasons:

1- Noisy decisions: given the independence of the different parsing tools that has no knowledge about other tools results, noise clues could be generated. This could occur as well due to noisy source data of the video editing itself. However, when we use these clues collectively, we could infer the occurrence of mistaken decisions that have been taken.

2- Tools parameters and accuracy: again, in real life, no processing tool is totally perfect to give accurate results for different possible input. Even if the tool has high accuracy, adjusting its parameters and configuration for different domains and possible video editing scenarios could lead to less accurate results. For example, adapting the parameters of video segmentation engine such as the threshold, temporal skip value, ...etc for news segments could be different than for sports and music sources and so on.

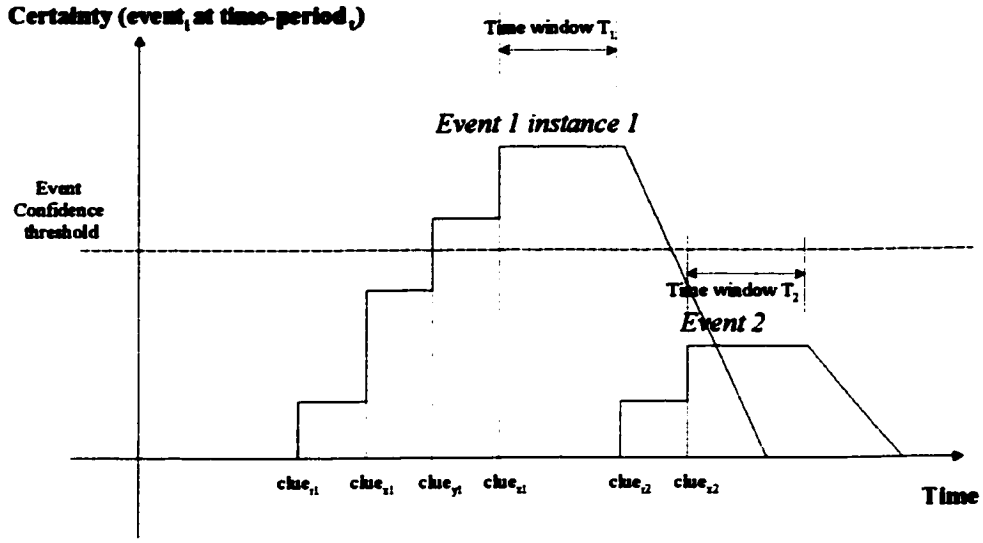



**Figure 4-8.** Thresholding and Time-Window analysis.

High-level video events conflict could be resolved using two paths:

**First** Thresholding and Time Windowing:

using the certainty score measurement equations from section 4-4 and as plotted in figure 4-8, only the event with a score that exceeds the threshold could represent a potential and maximum likelihood to classify its period of time with that event classification. However, there is still a possibility of rather actual *no-event* decision, because of noisy results. In the figure, we realize the use of a time window parameter to reduce the event certainty if its value didn't reach the event's recognition threshold within certain period of time. Each event could have separate value of the time window to recognize the event given that the event's duration is distinct between an event and another. Thus, after this period of time, the event's certainty decreases with time linearly until reaching the zero certainty score and thus the rejection of this event instance. The slope of the decay of the event certainty is a function of the minimum time expected between occurrences of instances of the same event class so that the recognition of a later instance of the same event could still be realized separately.

**Second** High-Level Temporal Causalities:

based on the events received from the previous path. High level knowledge schema built on the extended N-ary schema described in figure 4-4 could be utilized to refine or re-enforce the results and could provide backward resolution to correct previous decisions. This high-level knowledge among the domain's facts and events scenario has two general components:

1) Same-events classification dependencies: These rules are among the instances of the same event classification. For example, a high level constrains fact could state that there is a minimum time delay among two different goal events within a soccer match of one minute.

2) Different-events classification dependencies: These are among the instances of different event instances. For example, a fact rule is acquired or extracted using learning by examples that a minimum delay of one minute is required among goal and player substitution events.

## 4.6. Summary

We have presented our approach of a high-level abstraction (summarization) service. Our open model could be extended to include new sources of information, tools or relationships. Therefore, for example, with the adoption of MPEG-4 and MPEG-7 standards, a summarization service would be an easier and more accurate task. We have defined the notion of triggers that direct and guide our system *opportunistically* to classify the events of the domain during the analysis process. These triggers and domain's schema could be mapped automatically from the human expertise into the technical tool results using training algorithms. Each domain has its own facts, clues, heuristics and interesting events that could be considered within the summary. Each event is mapped as a class that has certain technical clues associated with importance percentages.

The spatial, temporal and logical relationship facts help resolving the conflicts or uncertainties that could arise during the analysis phase. We have provided an event certainty decision formulas that takes into consideration the confidence of the entire system in each tool in addition to the confidence of each tool in its own results at any given time. We still have to take the factor of the relative technical tools' errors

into account, as they are still not as reliable and robust as the human perception till now. We presented soccer games as an example domain to clarify our approach. However, similar reasoning approach could be adapted to different domains such as movies, news, documentaries, ...etc. This work is used as a multimedia service over our agent-based architecture. According to the available resources, user profile preferences and device capabilities, we select the format and level of abstraction.

More related work could be done in the area of media selection and adaptation conversion and optimization. A possible example is to enable the devices that doesn't have Arabic or Chinese language display drivers for instance to still display documents written in those languages through dynamic and automatic media conversion from text to small-size image format that could be displayed on any graphic-enabled device. The message of the video should be adapted to the user's device and be comprehensive at the same time as much as possible. This would include media format conversion and conditioning processes. Thus, a schema about the domain's events themselves including their possible formats of browsing could be stored as a unified XML schema file. Another example, we think that the interesting speech phrases within the video could be heard on a cellular phone or even recognized into a text description for paging services. Thus, this could mean summarizing the video content semi-automatically into short text messages that would be displayed wirelessly on cellular phones or pagers of limited lines of display even if the user were on the road.

# CHAPTER 5

# DESIGN AND IMPLEMENTATION OF VIDEO SERVICES

## 5.1. Introduction

We have implemented the video cut detection, media key framing and summarization sub-systems as a given service over the Internet and we used the binary penetration algorithm, which was described in chapter 3. This service is developed and run on a Pentium II personal computer that has 266 MHz CPU speed, 128 MB RAM and runs over a Windows NT 4.0 operating system. We integrated those subsystems within our agent-based testbed. It is used as a service for authorized users over the World Wide Web. First, we will briefly describe the overall architecture's components, agents and interactions of the different agents. Then, we will describe in detail how we use the media summarization service using this architecture through a complete scenario.

## 5.2. Multimedia Mobile Agents Architecture

This overall system [HAR99] is an integrated research work among the University of Ottawa, Mitel Inc. and NRC in Canada. The system architecture consists of several agents that cooperate in a distributed environment to offer services to nomadic users. We adopt agent-based technology and protocols to build the infrastructure entities of the system. An agent can be defined as a software component that acts autonomously and intelligently on behalf of its user. Thus, it could represent its user (e.g. a human being or even a software process) to take decisions according to some encapsulated ontology. Thus for example, we are able to define a service agent for each service process that we have so that we could make use of this available service process for remote users without re-developing the whole service process. In some cases, these software agents could be mobile and roam among different Agent Execution Environments (AEE) till they accomplish their pre-defined mission. An application example for these mobile agents is to find out the best travelling ticket price over the Internet from certain source till certain destination in certain time. At any time, each agent could have one of different possible states (e.g. idle, running, postponed, migrating, ...etc) during the agent's itinerary. In our architecture, certain agents will communicate and negotiate with one another to allocate a profile of the visiting user at the visited site. This profile includes user's identity, her/his preferences and authorized services, negotiated with the home site of the user. In addition, in order to offer a particular service on the visited site, appropriate agents do some negotiations in a proposal-counter proposal format [HAR99] till they agree upon the availability of the resources associated with that service. This is to determine if the resources are available at the time the visiting user requests the service. In addition, they negotiate the level of quality of the service that could be achieved. Thus, for example, a better quality of service could be supported if the user interface agent accepts to pay more for the service.

Figure 5-1 depicts the agents-based architecture of our prototype on one site. In this architecture, we can identify the following main three layers:

(i)- *Data repository manager layer*:

It maintains information about the users, their device's profiles and their associated preferences. It also manages the resources and list of services available on each site. The device profile attributes could be used to guide us to transcode, condition and/or adapt the service results according to the attributes of the device.

We use Lightweight Directory Access Protocol (LDAP) to implement this distributed data repository. LDAP is an IETF standard that defines a hierarchical directory service, which is distributed and highly protected [REC93].

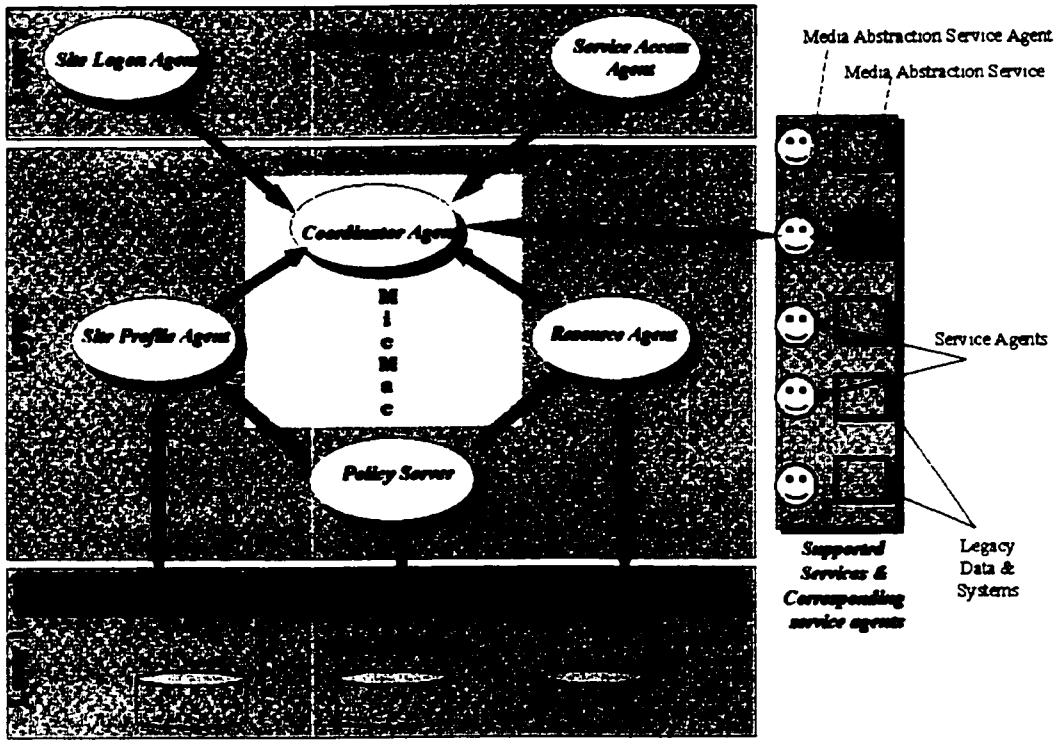



**Figure 5-1.** Site Personal Mobility Architecture. adapted from [HAR99]

(ii)- *Intra-site processing-agent layer* which comprises:

- The **Site Profile Agent,** negotiates the user profile and determines a list of authorized services and preferences that she/he can have at a visiting site.

- The **Policy Server,** maintains certain policies in the form of obligations and authorizations that the authority of each site permits or denies. These policies are represented using XML and visually modeled and edited by a graphical tool [AME01]. Policies regulate and monitor the service and resource utilization, the system configuration, the quality of service for the transfer of multimedia contents and the agents' behaviors. We associate management rules (such as cost and user privileges) with the policies, which are used in the negotiation phase among the agents.

- The **Resource Agent,** negotiates the resources that are required to run each service and configures the environment for the users at the visited site. A negotiation between resource agent and service agent takes place, through the coordinator agent, to make sure which resources are available at the time the service is requested by the user.

- The **Coordinator Agent,** is responsible for managing inter-site and intra-site communications using MicMac [MIT96] protocol. It acts as an administrator for the MicMac server, which behaves as a blackboard, which is used to host the messages of interactions. Its central role is to establish a secure communication among two or more agents, to authenticate agents, and to monitor their activities when exchanging information.

(iii)- *User and service agent layer comprising:*

- The **Site Logon Agent**, interacts with the authentication server to authenticate each user and to create a session key, used for inter-site communication encryption and decryption procedure [ZHE99]. Meanwhile, we don't use encryption and decryption procedure for intra-site

communication assuming that there is trust among the different agents within the same site to reduce the overhead of executing the security mechanism.

- The **Service Access Agent**, displays the authorized services to the authenticated user, according to the device profile capabilities. For each service, there is a corresponding agent (termed Service Agent) that manages the execution environment required by the service.

We use MicMac software system to support agent interactions and communications. It utilizes a shared blackboard mechanism. KQML language expresses the messages exchanged between agents.

Finally, with the limited set of resources that may be available within the end-user devices, service agents enable moving a user request to servers that handle resource intensive services (e.g. media abstraction or media key framing service). After processing the request using the video service process, the corresponding service agent might return back the outcome to the user's end device. The service agent also stores the results. Thus, if the user logged off, the results would be retrieved the next time he logs on.

## 5.3. Profiling the Heterogeneous Device Features

We treat the access devices by their characteristics (hardware, software and general features) rather than their types (whether PDA, PC, cellular phone, ...etc) to build robust and extendable systems that expects to deal with new devices without re-engineering the whole system. The device profile consists of three different sub-profiles, namely hardware, software and general profiles. A general description schema is shown in figure 5-2. Of course, we don't need to encode all the attributes but rather we choose according to the utilized applications. The device profile attributes are used to guide us to transcode, condition and/or adapt the service results automatically to best match the attributes of the device. The device profile could reside within a directory system such as Lightweight Directory Access Protocol (LDAP). Another better alternative, a device profile agent could encapsulate the profile features within the mobile device itself, to represent an identity card of the device as an XML-based profile on the device itself.

This will substantially support dynamic and run-time negotiation in the case of the user's visits to new environments such as ad hoc networks without user's intervention.

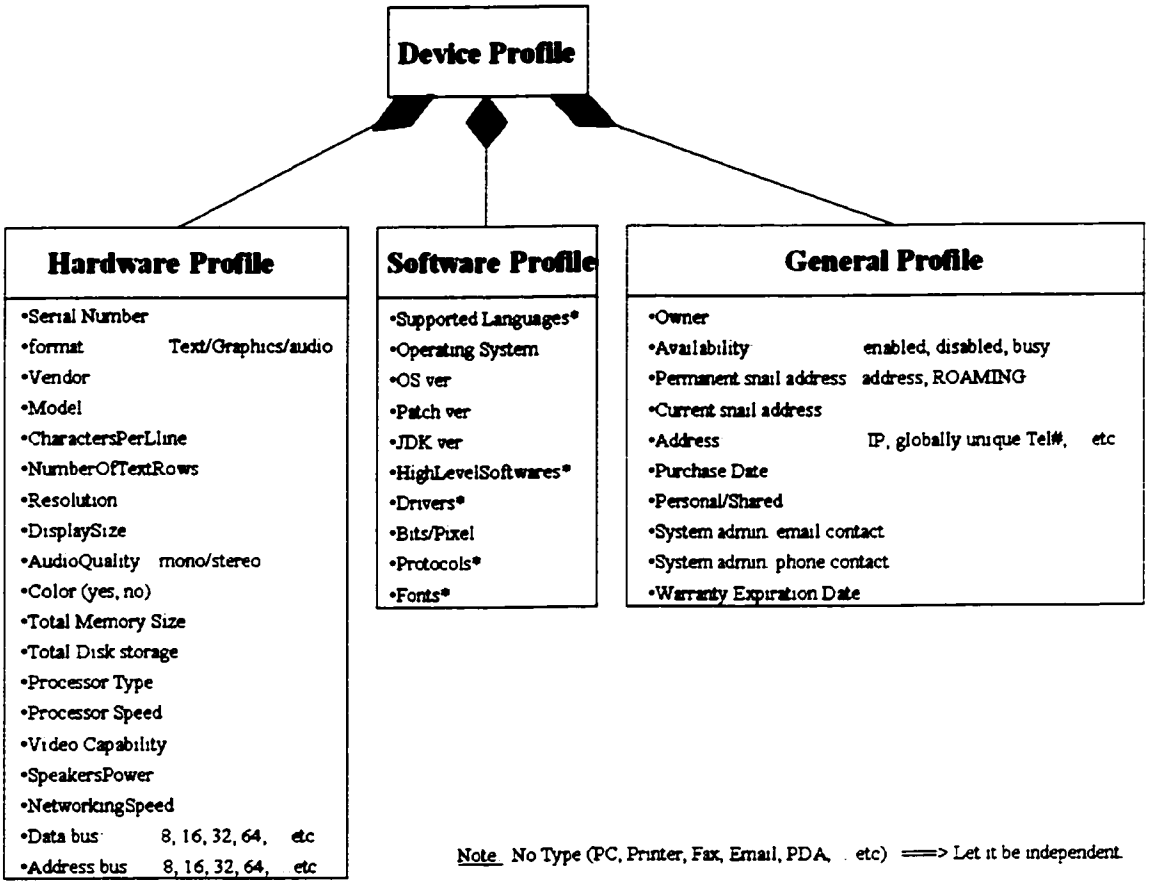



**Figure 5-2.** Device profile description.

Thus, we looked for an XML-based language to specify the data structure description of the device profile that could be suitable for our media summarization service and possible future device features.

We use the Composite Capabilities/Preferences Profile (CC/PP) [W3C99] language for this purpose. CC/PP is a W3C RDF-based framework to describe the capabilities of user agents (web browsers) and user preferences. We use CC/PP as a language to describe the device profile given in figure 5-2. One advantage of CC/PP is the possibility to incorporate inline descriptions or refer to external profiles through their URI addresses. Thus, the use of external links could be used to access default profiles

hosted by the hardware or software vendors in central stores. This could be more useful for wireless connections where the link between the client cell phone for example and the content server/gateway has limited bandwidth while the link between the content server/gateway and the vendor has high speed communication. Another advantage of CC/PP is its suitability for supporting future extensions and incorporating user preferences such as turning sound on/off or preferred language (English, French, ..etc). A third advantage of CC/PP is the use of "defaults" and "modifications" notions. Thus, if the user adds new memory to the device for example, he just added this modification in the modifications part of the hardware. Thus the default profile could be an external link to the vendor web server while the modification is updated by the client application if needed. Thus in new network sessions between the user agent and the content server, only current modifications could be re-transferred.

An example of a CC/PP profile is given in figure 5-3. It is based on RDF graphs. RDF graphs consist of nodes, arcs and leafs. Nodes are resources, arcs are properties and leafs are property values. This RDF graph is based on an example that includes "Default" properties for every major component.

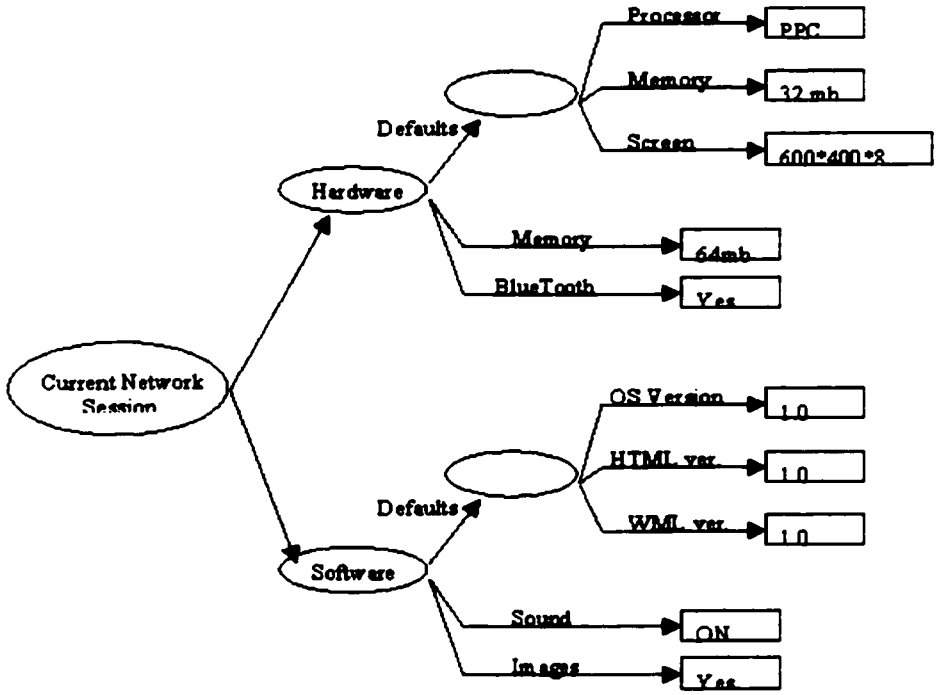



**Figure 5-3.** CC/PP device profile description using RDF graph. from [W3C99]

A typical CC/PP profile implementation for the PDA that we used for testing our service, as we will explain later in section 5.5, is as follows:

```
<?xml version="1.0"?>
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:prf="http://www.w3.org/TR/WD-profile-vocabulary#">
        <rdf:Description about="HardwarePlatform">
        <prf:Defaults
                Vendor="Compaq"
                Model="3870"
                Format="GRAPHICS"
                Resolution="240x320x16"
                Color="YES"
                CPU="StrongArm"
                CPUSpeed="206MHz"
                Keyboard="NO"
                RAM="64MB"
                Rom="32MB"
                Bluetooth="YES"
                Speakers="YES" />
        <prf:Modifications
                WLAN="YES" />
        </rdf:Description>
         <rdf:Description about="SoftwarePlatform">
        <prf:Defaults
                OS="PPC2002"
                HTMLVersion="3.2"
                JavaScriptVersion="1.2"
                WAPVersion="1.0"
                WMLScript="1.0" />
        <prf:Modifications
                Sound="ON"
                Images="ON" />
        </rdf:Description>
        <rdf:Description about="UserPreferences">
        <prf:Defaults
                Language="English"/>
        </rdf:Description>
        <rdf:Description about="General">
```

```
        <prf:Defaults
                Owner="user1"
                Availability="enabled" />
        </rdf:Description>
</rdf:RDF>
```

**Figure 5-4.** CC/PP device profile for a PDA.

## 5.4. Multimedia Service Adaptation

Trying to utilize the subsystem of the media key framing algorithm to generate low-level summary reports, a batch service program is developed. Figure 5-5 represents the described subsystem of media key framing within chapter 3 as a black box and its interface to the corresponding service agent. The process accepts an input request file, which is submitted from the service agent. This input mission file contains the parameters of the applied request and it generates a result report that contains the detected key frames along with the confidence percentage of each key frame. Then the service agent utilizes the results and reformats it, using XML as we will show later, before passing this report to the user.

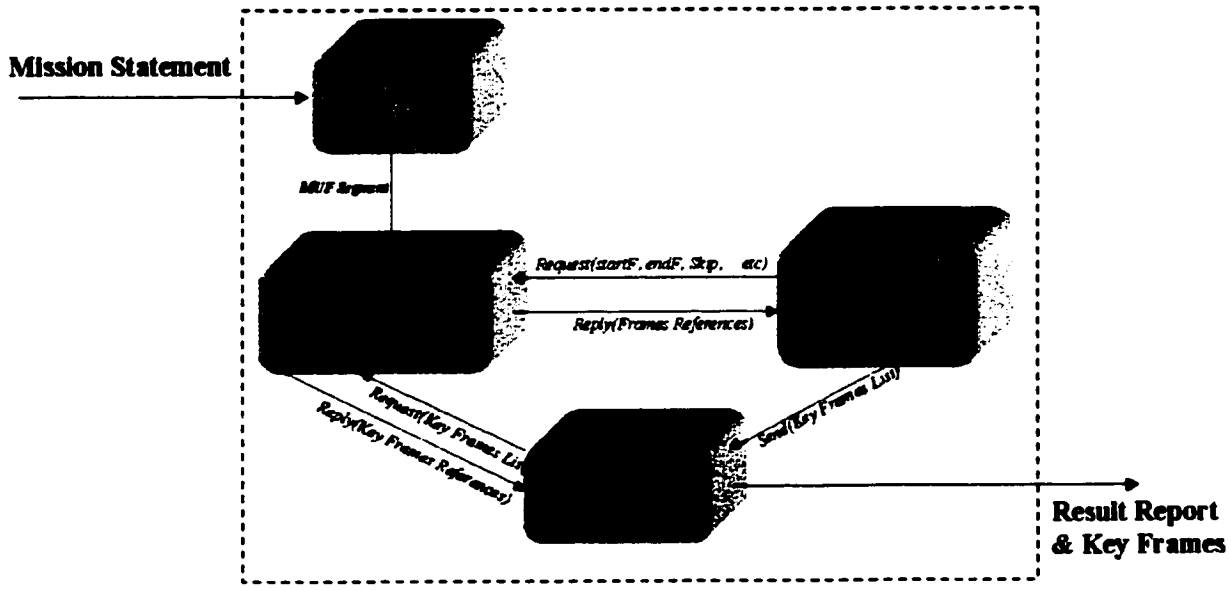



**Figure 5-5.** The use of the media key framing as a batch process.

Figure 5-6 depicts an example of input file prepared by the service agent before submitting this mission request to the service process. We notice that there are only two mandatory parameters while the other ones are optional. These mandatory parameters are the reference number of the request and the video part itself. The other optional parameters contain some other input parameters about the request. They include as well some parameters that could affect the behavior of the media key framing algorithm itself such as the threshold for cut detection, the temporal and spatial skips used, start and end of the requested segment of the media file, ...etc. However, these optional parameters have default values that are included in the algorithm itself, which could be overridden by the input mission request if necessary.

| | | |
|---|---|---|
| *ReferenceNumber:* | 624 | **(Mandatory)** |
| *InputMediaFile:* | video122.avi | **(Mandatory)** |
| *UserID(Email):* | xyz@sol.genie.uottawa.ca | (Optional) |
| *iDate(DD/MM/YYYY):* | 28/9/1998 | (Optional) |
| *iTime(HH:MM:SS?M):* | 4:25:00PM | (Optional) |
| *Start(Frames):* | 1/4 | (Optional) |
| *End(Frames):* | 3/4 | (Optional) |
| *TemporalSkip:* | 5 | (Optional) |
| *SpatialSkip:* | 5 | (Optional) |
| *Threshold:* | 25 | (Optional) |
| *Operation:* | KeyFraming | (Optional) |
| *KeyframingMethod:* | 6MSB_Blocks | (Optional) |
| *Performance_Method:* | Binary_Penetration | (Optional) |
| *Number_of_Blocks:* | 9 | (Optional) |
| *WorkingDirectory:* | c:\users\mahmed\vb5\for_integration\outputs | (Optional) |
| *FramesFormat:* | JPG | (Optional) |
| *Color_Grey:* | Color | (Optional) |
| *ColorQuality(of_255):* | 30 | (Optional) |
| *InitialFormSize:* | Normal | (Optional) |

**Figure 5-6.** An example of input request files written by the service agent.

Upon a successful completion of the key framing process, the system generates a result report. An example of such a report is depicted in figure 5-7. The result report includes as well the algorithm's configuration parameters, which lead to the associated result. Of course, references to the indices of the

key frames themselves are written within the report along with the algorithm's confidence of realizing these frames as true key frames. Also, the verified format of the video is stated along with the total size of the key frames. In addition, the processing time is written as a way of possible billing mechanism for the user's request.

Meanwhile, the system could detect and report different error possibilities. These error descriptions along with their assigned codes are depicted in figure 5-8.

| | |
|---|---|
| Start_Processing_Date: | 07/10/1998 |
| Start_Processing_Time: | 04:21:56 PM |
| *Status:* | Processing |
| *Reference_Number:* | 624 |
| *Input_Configuration_File:* | c:\users\mahmed\vb5\for_integration\624_InputConfig.txt |
| *UserID(Email)* | xyz@sol.genie.uottawa.ca |
| *iDate(DD/MM/YYYY):* | 28/9/1998 |
| *iTime(HH:MM:SS?M):* | 4:25:00PM |
| *Operation:* | KeyFraming |
| *KeyFraming_Method:* | 6_MSB_Blocks |
| Performance_Method: | Binary_Penetration |
| *Number_of_Blocks:* | 9 |
| *Input_Media_File:* | video122.avi |
| *Frames_Format:* | JPG |
| *MediABS Version:* | 1.0 |
| *Start_Frame:* | *1/4* |
| *End_Frame:* | 3/4 |
| *Temporal_Skip:* | 5 |
| *Spatial_Skip:* | 5 |
| *Used_Threshold:* | 25% |
| *Color_Grey:* | Color |
| *Color_Quality(of_255):* | 30 |
| *InitialFormSize* | Normal |
| | |
| *Input_Media_File_Format:* | AVI OK ... |
| *ApplicationFileName:* | c:\users\mahmed\VB5\for_Integration\BatchRun.exe |
| *ResultDirectoryPath:* | c:\users\mahmed\vb5\for_integration\outputs\624_result\ |
| *Analysis_Frames_Width(Pixels):* | 100 |
| *Analysis_Frames_Height(Pixels):* | 100 |
| *Video_Length(Frames):* | 295 |
| *Bits/Pixel:* | 24 |

| | | | |
|---|---|---|---|
| *Video_Size(Bytes):* | 2,310,656 | | |
| *Video_Width(Pixels):* | 160 | | |
| *Video_Height(Pixels):* | 120 | | |
| | *1st Frame* | *2nd Frame* | *Difference(%)* |
| KF# 1 | *73* | 74 | 60.16% |
| KF# 2 | 120 | 121 | 53.54% |
| KF# 3 | 185 | 186 | 57.01% |
| KF# 4 | 221 | 222 | 49.61% |
| Total_Processed_Frames: | 28 | | |
| Processed_PixelsPerFrame: | 891 | | |
| *Number_of_KeyFrames:* | 4 | | |
| *Processing_Time(Seconds):* | 8 | | |
| *Finished Ok...* | | | |
| *at_Date(DD/MM/YYYY):* | 07/10/1998 | | |
| *at_Time(HH:MM:SS ?M):* | 04:22:12 PM | | |

**Figure 5-7.** An example of result report files written for the service agent.

| | |
|---|---|
| •Code 0: | OK |
| •Code 1: | Input Media File " & InputMediaFile & " Doesn't Exist! |
| •Code 2: | Not Allowed Media Format! |
| •Code 3: | Start Frame Number Should be LESS Than End Frame Number |
| •Code 4: | Start Frame Number Should be EQUAL OR GREATER THAN 1 |
| •Code 5: | End Frame Number Should be EQUAL OR LESS THAN Video Length (" & Video.Length & ") Frames |
| •Code 6: | You must give a Reference Number for this request! |
| •Code 7: | Unsupported Frames Format |
| •Code 8: | Input Configuration File " & InputConfigurationFile & " Doesn't Exist! |
| •Code 9: | Media Server Stopped Suddenly! |

**Figure 5-8.** List of possible errors that could be detected.

Figure 5-9 shows the finite state machine diagram of using the media key framing process as a multimedia service of the Internet. As shown, the service is initially in a "starting" state. When the

service agent decides to process some request, it calls the service process after writing an input request file. Thus, the service process moves into a "checking" state. In that state, the video service tries to ensure that the request is valid or not. If the request is invalid due to different reasons as shown in the list of possible errors in figure 5-8, the service process generates an error report to the service agent, which then reports it back to the user accordingly. However, if the request is valid, the service process enters the "processing" state and the process instantiates the key framing algorithm. This "processing" state itself consists of two sub-states. The first sub-state is "extracting" video frames separated by a temporal skip value assigned in the input request file or using the default value if not assigned explicitly. The second sub-state is "analyzing" these extracted frames using a media key framing algorithm to detect the key frames for the required media segment. Finally, the process generates the report of execution in a result report file with error code 0, which represents a successful processed request. The service agent uses this report to generate a custom report in an XML language form. At the end, the process returns back to the "starting" state and waits for a new request.

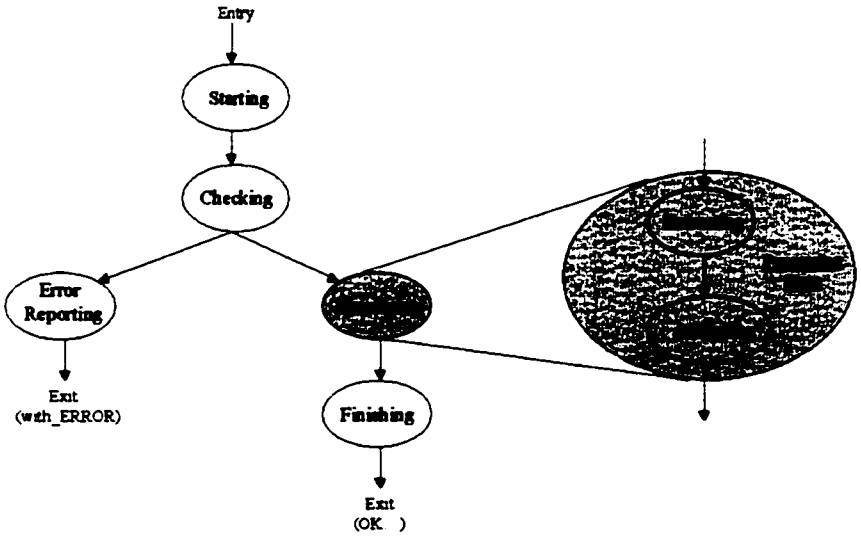



**Figure 5-9.** FSM diagram of media key framing process.

A runtime snapshot of the execution of the media key framing system as a batch process within the media server is depicted in figure 5-10.

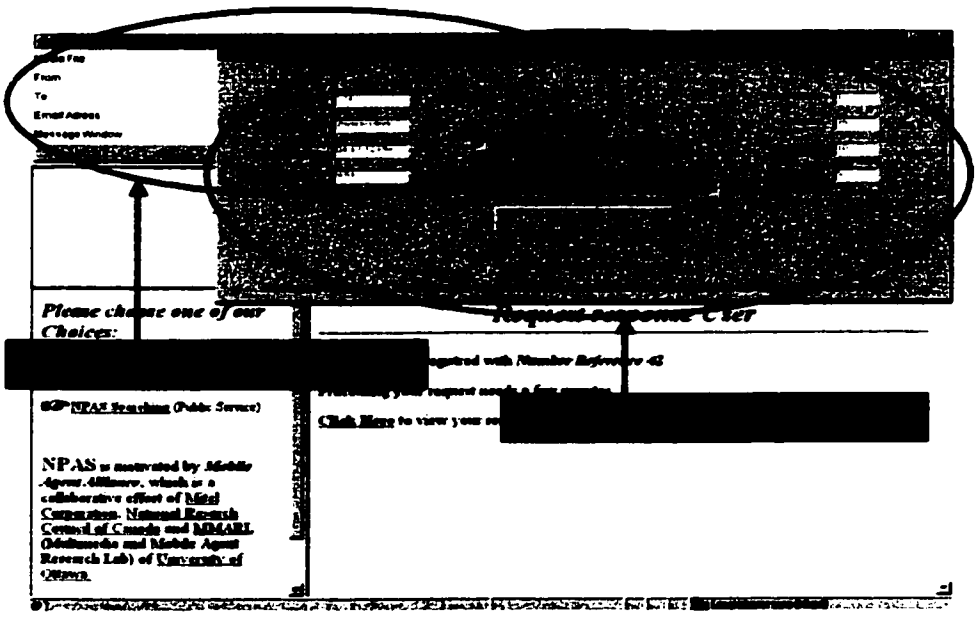


**Figure 5-10.** A snapshot of the use of media key framing service as a batch process on the media server.

## 5.5. User Interface of Multimedia Service

In this section, we will provide a complete scenario, along with some screen shots, which are built using small footprint Java applets (just around 10 KB), for providing the key framing function as a service for authorized users over the World Wide Web. In our overall prototype, we try to provide similar environments to the users of the system regardless of his location. He could access the same available services from any place he is currently in. To achieve that, the user is first authenticated as shown in figure 5-11. He provides his user ID and password to the system. He is authenticated within his current visiting environment by communicating to his home environment. This communications are handled through secure communication messages [ZHE99].

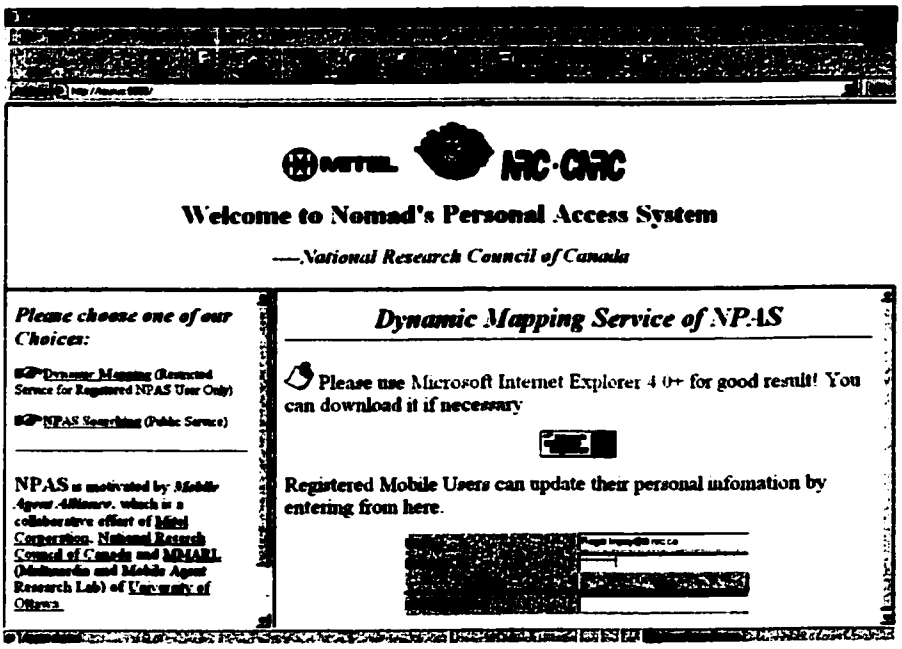



**Figure 5-11.** User Authentication Interface.

After authentication, the list of authorized services of this particular user become available for him in his current location. A sample list of these authorized services such as media key framing and email forwarding is shown in figure 5-12.

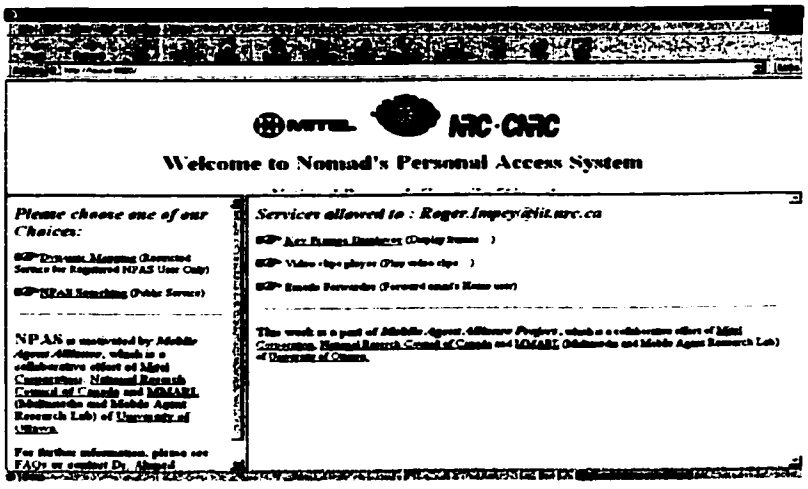



**Figure 5-12.** Authorized services list for the current user.

Hence, one of these available authorized services in this example is the key framing service. Thus, the user could still utilize this service in his current visiting environment. For example as depicted in figure 5-13, the user requested to analyze certain portion of a video segment to receive only the key frames of this part to him.

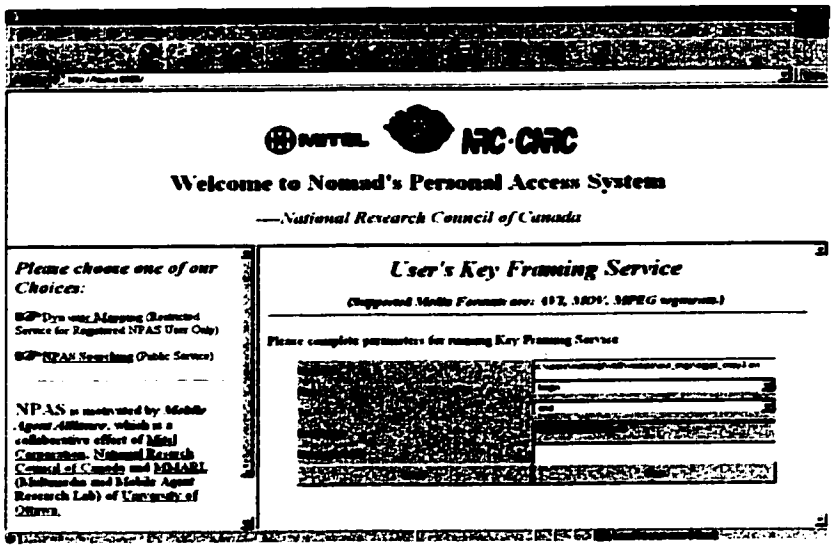



**Figure 5-13.** User Interface of Video Service Request.

Generally, the user could select a portion of the video file he/she is interested in. In addition, we handle different video formats such as AVI, MOV and MPEG file formats seamlessly. The summarization service is structured upon modular components so that we could adopt new video analysis algorithms or handling new video formats with minimum changes. As an example of overall scenario conclusion, figure 5-14 and figure 5-15 present the output user interface to the end-user. The figures present an adapted result that corresponds to different circumstances.

The service could render a normal video streaming result to the user providing the availability of communication resources such as the link's bandwidth, CPU power, ...etc. Also, the system provides the complete stream only if the user has the device capability to browse video contents. Otherwise, in the case of lack of resources or less device capabilities, the distributed architecture could choose after a negotiation process to automatically furnish only few key frames of the selected video segment to the

user through some negotiations between the resources management and device handling modules. The system supplies this summary in two defined quality levels (note: the QoS management is performed through the other system's infrastructure components and agents and described in details within other researcher's thesis [SAL02]). The first quality option is to use color and high quality JPG image files. The second possibility is to provide only a gray scale version of these key frames with lower JPG quality. We select the key frame as the $3/4^{th}$ frame of selected recognized shots. We select this frame to represent the focus of the corresponding shot. In addition, we left $1/4^{th}$ of the shot because of the possibility of having a gradual transition video editing between two consecutive shots. The result report, delivered to the user, shows the processing time to extract these key frames along with the size of the original video segment and the total key frames size for possible corresponding billing procedures.

For example, for the same given request with different environment cases (resources and device capabilities), in figure 5-14(a), the system provides the whole stream of about 11.5 MB of video content. However, in the case of medium quality level due to cases of lack of resources and/or device capabilities, as in figure 5-14(b), only 24 KB in total size of color and good quality key frames is transferred over the network without losing much of the content. It could be seen that opportunistically text-caption summary frames and especially in the $2^{nd}$ half, resulted from the video editing of the game, are included in the summary results. Moreover, in the lowest quality level adopted by the system, figure 5-14(c) represents a total size of just about 6.3 KB (of gray scale images and with lower image quality) of key information for the same request.

Another example where we used general-purpose knowledge is shown in figure 5-15. In this case, a video documentary about different tourist destinations in Egypt, that has no text-caption or audio content, is retrieved. Thus in this case, we utilize only the scene change feature using the binary penetration algorithm. Here, similarly for the same given request with different environment cases (resources and device capabilities), in figure 5-15(a), the system provides the whole stream of about 2.3 MB of video content. However, in the case of medium quality level in figure 5-15(b), only 18 KB in total size of color and good quality key frames is transferred over the network. In the lowest quality level

adopted by the system, figure 5-15(c) represents a total size of just 6.3 KB (of gray scale images and with lower image quality) of key information for the same request.

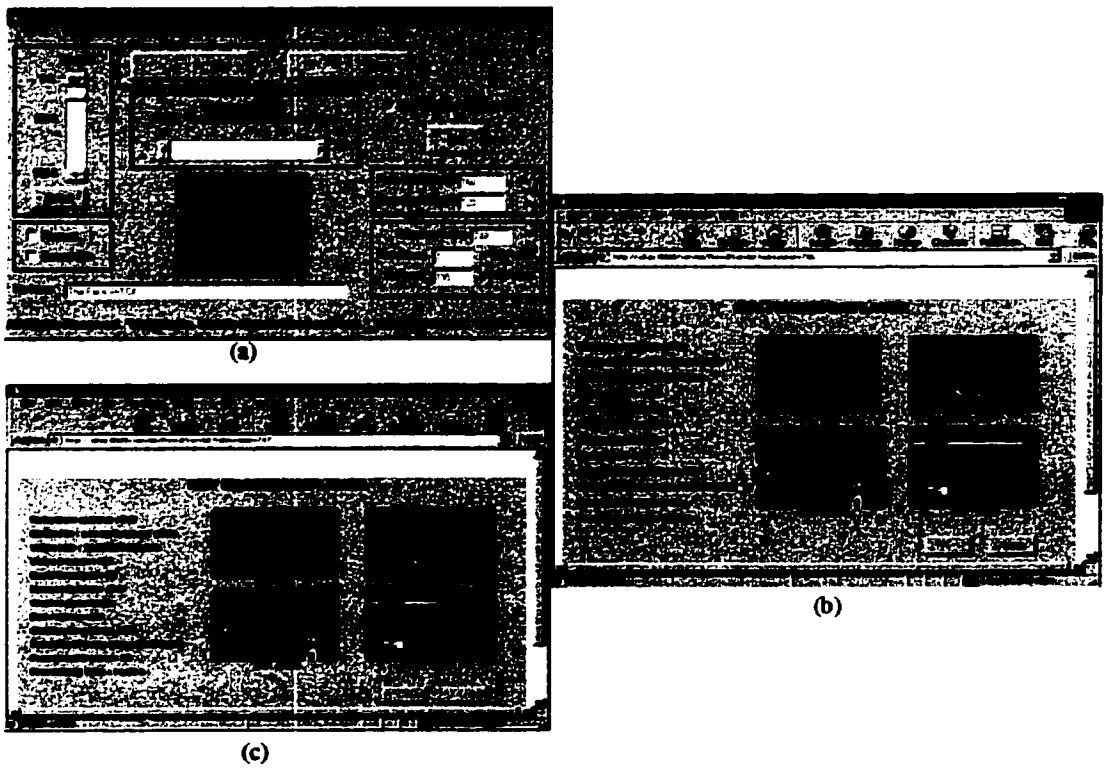



**Figure 5-14.** The user interface for browsing soccer game with text captioning and opportunistic 2nd half summary.

(a) In case of highest quality level: Video Streaming;

(b) In case of medium quality level: Color and High Resolution Key Frames;

(c) In case of lowest quality level: Gray and Low Resolution Key Frames.

Thus, we could see in the cases of figures 5-14(b), 5-14(c), 5-15(b) and 5-15(c) that low network resources is required to send the alternative message. In addition any device such as most web-enabled PDAs and handheld devices with limited capabilities of presentation using as text and graphics requirements could already interact with our system and access the service's small foot print reports of video analysis and summarization using low bandwidth wireless connections. Our distributed prototype

would include information about the device's profiles of each authorized user. Thus, we would condition the results according to the device specifications: either hardware such as the screen's dimensions, resolution and color depth or software such as the browser application, version and available media formats for possible media conversion and transcoding post-processes.

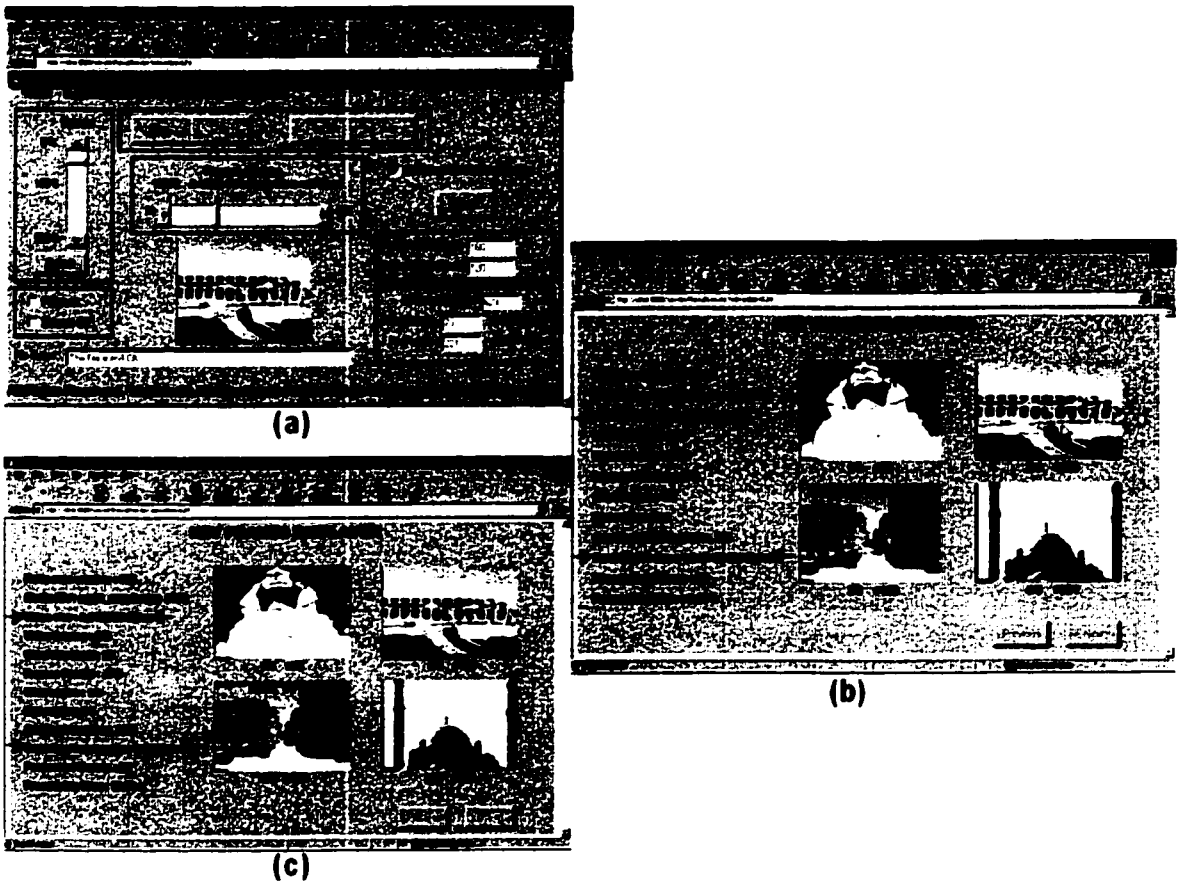



**Figure 5-15.** Samples of the user interface for browse tourist destinations in Egypt.

(a) In case of highest quality level: Video Streaming;

(b) In case of medium quality level: Color and High Resolution Key Frames;

(c) In case of lowest quality level: Gray and Low Resolution Key Frames.

We didn't implement the video service on PDA within our overall architecture due to the unavailability of this device while evaluating the system. However, we investigated separately the possibility to use our

video indexing service on IPAQ 3870 PDA as an example device. We simulated its browsing capabilities of our service results without the authentication and negotiation steps. As seen in figure 5-16 and the later description of the H/W and S/W features, the PDA could load and run the Java clasees and display the text and images of our video service results very efficiently. The reasons are that the service generates very small text and images result reports and that we utilize very small footprint Java classes for the user interface. These Java classes could be executed on the Java Virtual Machine software installed on this PDA. However, we could still utilize the service through other recent technologies as well rather than using Java technology. Such technologies deliver normal but dynamic HTML pages created on the fly using server processing before submitting the result to the client. Some of these tools are Active Server Pages (ASP), eXtensible Server Pages (XSP), Hypertext Preprocessor (PHP), Cold Fusion, or even through E-Mail if necessary or preferred.


**Figure 5-16.** Simulation of the video indexing and summarization service on a PDA.

The main features of Compaq IPAQ 3870 are:

1- 206 MHz Intel StrongARM SA-1110 32-bit RISC Processor.

2- 64 MB SDRAM, 32-MB Flash ROM Memory.

3- 16-bit color touch-sensitive reflective thin film transistor (TFT) liquid crystal display (LCD) Viewable image size – 2.26 in wide x 3.02 in tall (5.7 cm wide x 7.7 cm tall).

4- Powered by Microsoft Windows Pocket PC 2002 Operating system and Microsoft ActiveSync version 3.5

5- Compaq Wireless LAN Card.

6- PC Card Expansion Pack.

7- Integrated Bluetooth version 1.1

8- TCP/IP stack.

9- DHCP (Dynamic Host Configuration Protocol) to dynamically assign IP address, default gateway and domain name servers for the PDA.

10- Pocket Internet Explorer.

11- JeodeRuntime JVM from Insignia Solutions® which is a fully certified implementation of Sun's PersonalJava 1.2 specification.

Figure 5-17 shows the adopted report structure of our video indexing service. Each report has the shown four main components. First, we will provide the high—level description of these components. Then, a detailed XML result file will be explained later.

The components of the service report are:

1- **Request Component:**
   It has the details of the submitted request. It should uniquely describe a certain request. This is guaranteed through a unique serial number that is incremented sequentially each time a new request is submitted to the system.

2- **User Component:**
   The details of the user of the system who submitted the request are stored within this part. This could be useful for accounting and billing reasons. It could be useful as well for future access to this report and other submitted queries by this user after authentication.

3- **Service Component:**

This part describes the properties of the selected video service for the mentioned request. The system elects the appropriate service according to accepted level of service for this request after the proposal counter proposal procedure among our system agents that are explained at the start of this chapter.

4- **Result Component:**

The results themselves of the video service reside within this component. The detailed description of this component will differ according to the elected service described within the service component. This component should preserve some indication about the processing time or power that have been undergone to process this request for billing and accounting reasons.

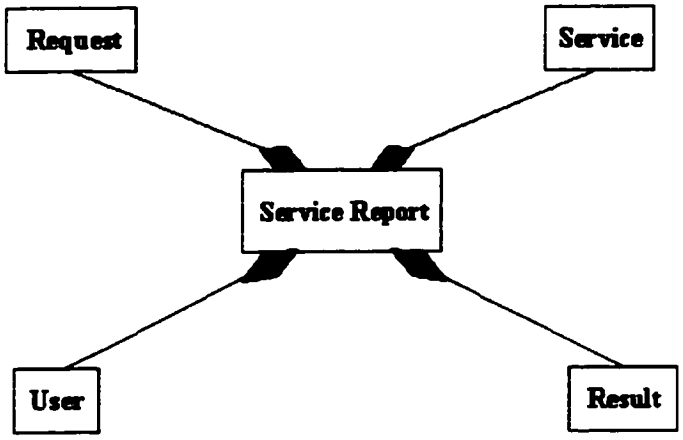



**Figure 5-17.** Adopted video indexing service structure.

Figure 5-18 shows a result record sample in a well-formed and validated XML format generated by the service agent for the corresponding output results, which are shown in figure 5-15(b). We chose XML as a result format language because first XML is extensible while the tags used to markup HTML documents and the structure of HTML documents are predefined. The author of HTML documents can only use tags that are defined in the HTML standard. In contrast, XML allows the author to define his own tags and his own document structure. Secondly, It is strongly and widely believed that XML will be as important to the future of the Web as HTML has been to the foundation of the Web. XML is regarded as the future for all data transmission and data manipulation over the Web especially among heterogeneous environments and devices. The XML result file is well formed and validated against the associated DTD structure, presented later, using a third-party software tool called XMLwriter [XML02].

```
(1)  <?xml version="1.0" ?>
(2)  <!DOCTYPE report SYSTEM "KF_report.dtd">
(3)  <report>
(4)      <user>
(5)          <id>xyz@sol.genie.uottawa.ca</id>
(6)          <title>PhD student</title>
(7)          <vlocation>Mitel</vlocation>
(8)          <hlocation>UoOttawa</hlocation>
(9)      </user>

(10)     <service>
(11)         <name>MediABS</name>
(12)         <location>UoOttawa</location>
(13)         <parameter>
(14)             <key>cost</key>
(15)             <value>15</value>
(16)         </parameter>
(17)         <parameter>
(18)             <key>quality-of-service</key>
(19)             <value>colorKeyFrames</value>
(20)         </parameter>
(21)     </service>
(22)     <request>
(23)         <reference>624</reference>
(24)         <input>video122.avi</input>
(25)         <startframe>1/4</startframe>
(26)         <endframe>3/4</endframe>
(27)         <operation>key framing</operation>
(28)         <submitdate>17 Aug 1999</submitdate>
(29)         <submittime>14:26:47 ET</submittime>
(30)     </request>
(31)     <result>
(32)         <processing>ok</processing>
(33)         <time>8.19 s</time>
(34)         <initial_size>2.310 Mb</initial_size>
(35)         <totalframes>295</totalframes>
(36)         <processedframes>28</processedframes>
(37)         <frames>
(38)             <number>4</number>
(39)             <fvalue>73</fvalue>
(40)             <fvalue>120</fvalue>
(41)             <fvalue>185</fvalue>
(42)             <fvalue>221</fvalue>
```

```
(43)                        <total_size>18.428 kb</total_size>
(44)                        <format>JPG</format>
(45)                        <dimension>160x120</dimension>
(46)                    </frames>
(47)            <dlocation>
(48)                        <protocol>http</protocol>
(49)                        <host>altair.genie.uottawa.ca</host>
(50)                        <path>\mediabs\outputs\624_result\</path>
(51)            </dlocation>
(52)            </result>
(53)    </report>
```

**Figure 5-18.** Example of a result report for the low-level summarization service in XML format.

The XML result file contains four main parts namely: user, service, request and result as illustrated in figure 5-18.

The elements used within the *user* part are:

Line 5: the identity that uniquely describes the user who submitted the request.

Line 6: title or affiliation of the user.

Line 7: visiting site or server of the user where he submitted the request.

Line 8: original home site or server of the user.

The elements used within the *service* part are:

Line 11: name of the service.

Line 12: location of the main service.

Line 14-15: cost attribute name and value of the request.

Line 18-20: the accepted level of service of the request.

The elements used within the *request* part are:

Line 23: the request unique reference number.

Line 24: the name of the media file that has been processed.

Line 25-26: the start and end points of the media segment within the video.

Line 27: the name of the operation or function used to process this request.

Line28-29: the date and time of the request.

The elements used within the *result* part are:

Line 32: the end status of the request.

Line 33: the CPU processing time of serving the request.

Line 34: the initial size of the media file.

Line 35: total number of video segment frames.

Line 36: number of processed frames.

Line 38: count of key frames for this request.

Line 39-42: key frames numbers or references within the media file.

Line 44: format of the result key frames.

Line 45: width and height of the key frames in pixel units.

Line 48: the protocol that could be used to retrieve and browse this report.

Line 49: the server address of the XML and frames files.

Line 50: the path of the XML and frames files within the server given in line 49.

In addition, we defined a Document Type Definition (i.e. DTD) description file associated with the XML result document. The main purpose of a DTD description is to define and validate the structure of an XML document when an XML parser parses the XML document. It defines the XML document structure with a list of approved elements. A DTD can be declared inline within the XML document itself, or as an external reference file.

In summary, XML provides an application-independent way to share information. With a DTD, independent groups of people or organizations can agree to use a common certain DTD for interchanging and understanding the meaning of the shared data. Thus, any application could use a standard DTD file to verify that data that you receive from the outside world is valid. In addition, you can also use a DTD to verify your own data before delivering it to the user or another organization.

In our system, the external DTD file, KF_report.dtd, associated with the provided XML result file is shown in figure 5-19.

```
<?xml version="1.0"?>
<!ELEMENT report (user,service,request,result)>

<!ELEMENT user (id,title,vlocation,hlocation)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT vlocation (#PCDATA)>
<!ELEMENT hlocation (#PCDATA)>

<!ELEMENT service (name,location,parameter+)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT location (#PCDATA)>
<!ELEMENT parameter (key,value)>
<!ELEMENT key (#PCDATA)>
<!ELEMENT value (#PCDATA)>

<!ELEMENT request
(reference,input,startframe,endframe,operation,submitdate, submittime)>
<!ELEMENT reference (#PCDATA)>
<!ELEMENT input (#PCDATA)>
<!ELEMENT startframe (#PCDATA)>
<!ELEMENT endframe (#PCDATA)>
<!ELEMENT operation (#PCDATA)>
<!ELEMENT submitdate (#PCDATA)>
<!ELEMENT submittime (#PCDATA)>

<!ELEMENT result
(processing,time,initial_size,totalframes,processedframes,frames,dlocation)>
<!ELEMENT processing (#PCDATA)>
<!ELEMENT time (#PCDATA)>
<!ELEMENT initial_size (#PCDATA)>
<!ELEMENT totalframes (#PCDATA)>
<!ELEMENT processedframes (#PCDATA)>
<!ELEMENT frames (number,fvalue*,total_size,format,dimension)>
<!ELEMENT number (#PCDATA)>
<!ELEMENT fvalue (#PCDATA)>
<!ELEMENT total_size (#PCDATA)>
<!ELEMENT format (#PCDATA)>
<!ELEMENT dimension (#PCDATA)>
```

```
<!ELEMENT dlocation (protocol,host,path)>
<!ELEMENT protocol (#PCDATA)>
<!ELEMENT host (#PCDATA)>
<!ELEMENT path (#PCDATA)>
```

**Figure 5-19.** The DTD file used to validate the XML low-level summarization result report.

The snapshot in figure 5-20 presents a tree-like stored XML result of a certain request browsed using Internet Explorer browser. This result XML file could be utilized for further offline use, not necessarily at run time. This could be useful for instance for extending Multimedia database management systems, Tele-learning applications, Entertainment, ...etc.

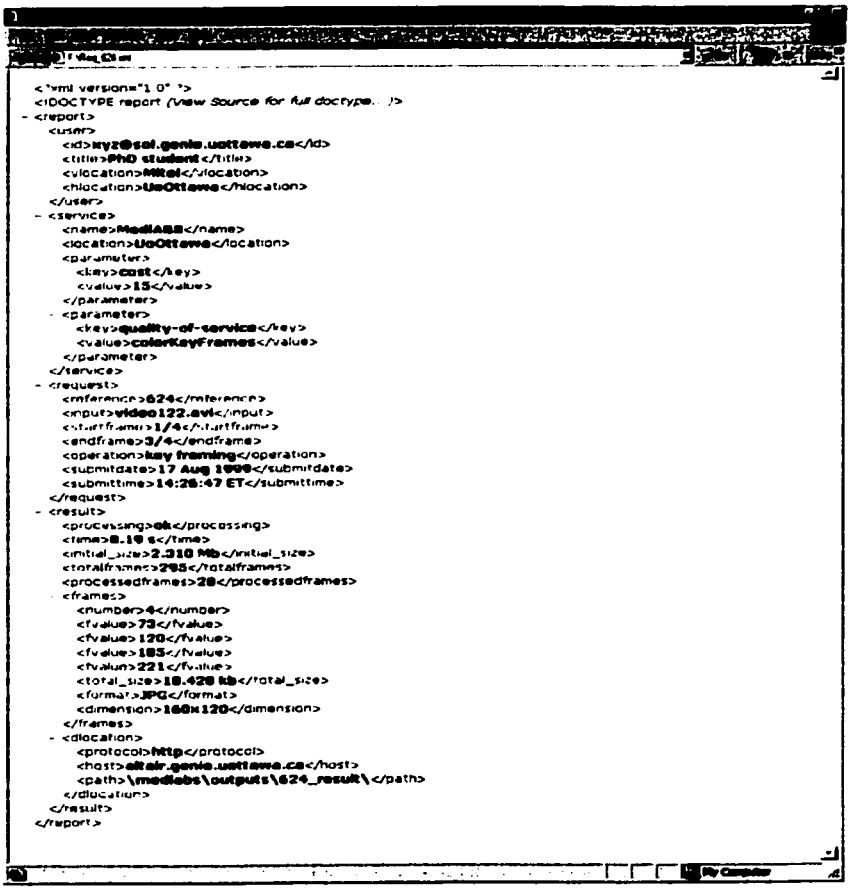



**Figure 5-20.** Snapshot of stored well-formed and validated XML result record for a video service request.

## 5.6. Summary

In this chapter, we presented how we use our video indexing and summarization services within our project. We gave a brief description of the underlying distributed agent-based infrastructure that is built for nomadic computing. We introduced the incorporated agents along with the interaction protocol among these agents. As we expect the use of heterogeneous end-user devices, we keep the properties of user devices within a profile for the inter-agent negotiation to furnish the optimum presentation for this user. We explained how we use our video services over this infrastructure and the different data structures we designed to accomplish these video services over the Internet. We presented various scenarios of utilizing the service over different resource availability and device capabilities. This detailed environment state is acquired upon negotiations among the architecture's agents.

# CHAPTER 6

# TESTING AND EVALUATION

## 6.1. Introduction

Various testing experiments have been performed. First, regarding the media key framing and video segmentation, we developed a software system on a Pentium II PC that has 266 MHz CPU and 64 MB RAM. We conducted various experiments using different file formats (AVI, MOV and MPEG) under various testing conditions, such as dark shots and normal camera effects. Additionally, the testing used same testing conditions for the different segmentation algorithms such as using color, 24 bits/pixel and same compression quality of the extracted frames.

## 6.2. Initial Three Algorithms Comparison

Initially, we tested three cut detection algorithms to compare their behavior with the different file formats and the different testing conditions. These algorithms are:

### a) The Hue Histogram Difference Algorithm:

To utilize this algorithm, the system first converts each compared pixel of the frames, taking the defined spatial skip parameter into consideration, from RGB color space into HVC color space. The Hue [HIR97] component histogram for each frame is evaluated. Then, the system uses this histogram difference of the hue component to discriminate between every two consecutive frames, taking the defined temporal skip parameter into consideration. The following formula is used to evaluate the hue histogram difference:

$$\text{Hue Histogram Difference} = \tfrac{1}{2} \cdot \Sigma \mid H_2(i) - H_1(i) \mid \; / \; \Sigma H_1(i) \qquad \text{for } i = 1 \text{ To } N$$

Where, $H_1(i)$ is the Hue Histogram for frame M,

$H_2(i)$ is the Hue Histogram for frame (M + Temporal Skip),

N is the possible Hue values

Then, if this hue difference exceeds some defined threshold, the two frames are said to represent a camera cut operation that separates two video shots.

### b) The 6 Most Significant RGB bits Intensity Difference Algorithm

In this algorithm, the system makes use of the 24 bit RGB color space components of each compared pixels (each RGB component has 8 bits representation). However, to speed the performance considerably, the system exploits only the 2 most significant bits [ZHA93] of each component (using a masking operation). This actually means that we define only 64 ranges of color degrees for the entire

RGB color space. Then, similar steps as in the previous algorithm are used to detect camera cut operations.

### c) The 6 Most significant RGB bits with the use of Blocks Intensity Difference

As described in chapter 3 using this algorithm, we try to utilize the locality of the color distribution within the video frames. We have to handle the cases of different consecutive shots with similar color global histogram information. Hence, there is a good possibility to misinterpret existing true cuts within the video segments. That could take place especially in the cases of using different camera angles (shots) within the same scene, which means a high probability of similar background within these shots. Thus in this algorithm, we segment the frames first into exclusive blocks and compare the corresponding block difference. Then, we take the average difference to represent the overall difference between the two frames.

In addition, for the modified algorithm (i.e. the 6 Most significant RGB bits with the use of Blocks Intensity Difference), we repeat the testing with a different number of blocks in each case to study the sensitivity effect of this parameter on the efficiency of the algorithm.

To evaluate the accuracy of the algorithms, we used two well-known accuracy measures from the information retrieval society. They are the Recall and Precision measures [SAL83]. They are defined as:

$$\text{Recall} = \frac{\text{Correct Cuts}}{\text{Correct Cuts} + \text{Missed Cuts}} \quad , \quad \text{Precision} = \frac{\text{Correct Cuts}}{\text{Correct Cuts} + \text{False Cuts}}$$

The tests make use of initial temporal skip and spatial skip values of 5 and 5 respectively. This clearly resulted in quicker processing time performance than using all the redundant data of visual track. The summary of the results, using seven different videos from different domains such as music, news, movies and documentaries, is shown in table 6-1.

**Table 6-1.** Results summary of using seven different format files.

| | Correct | False | Missed | PRECISION | RECALL |
|---|---|---|---|---|---|
| Hue Difference Percentage | 40 | 10 | 26 | 0.8 | 0.606061 |
| 6 MSBs Intensity Difference | 57 | 4 | 9 | 0.93442623 | 0.863636 |
| 6 MSBs, 4 BLOCKS Intensity Difference | 61 | 3 | 5 | 0.953125 | 0.924242 |
| 6 MSBs, 9 BLOCKS Intensity Difference | 62 | 4 | 4 | 0.939393939 | 0.939394 |
| 6 MSBs, 25 BLOCKS Intensity Difference | 64 | 4 | 2 | 0.941176471 | 0.969697 |
| 6 MSBs, 64 BLOCKS Intensity Difference | 63 | 4 | 3 | 0.940298507 | 0.954545 |

The results show a better accuracy for the 6 Most Significant RGB bits with the use of Blocks Intensity Difference algorithm over the other two algorithms for detecting camera cut operations. The correct cut detection has increased significantly versus the other two techniques.

The recall measure is improved because of the significant decrease of the missed true cuts. This occurs because of the use of disjoint blocks as shown previously in figure 3-3. Meanwhile comparing to the Hue Histogram Difference algorithm, the precision measure is improved mainly because of the increase in the number of detected correct cuts in addition to reducing the number of obtained false cuts.

Another issue is the number of the partitioning blocks. The results show the consistent behavior of the algorithm with respect to the number of blocks. Nevertheless, the use of 25 number of blocks (5 Horizontally * 5 Vertically) has shown slightly better overall results relative to the other numbers. The argument behind that is that we should not increase the number of blocks very much. That is first because it will result in slow performance of the cut detection process. In addition the algorithm will tend to simulate Pixel-Pair wise algorithms and hence increase the probability of detecting false camera cuts and will result in decreasing the efficiency in the case of quick camera or large object movement. At the same time, the number of blocks should not be a small number to avoid the global distribution problem of missing true visual cuts which we mentioned and shown previously in figure 3-2.

## 6.3. Cut Detection using Binary Penetration vs. Blind Approaches

As described in chapter 3, we find that there is a place of improvement of processing performance within the area of video processing and analysis algorithms. Thus, we introduced our approach of using binary penetration to achieve higher performance without sacrificing the accuracy results. We designed and developed our binary penetration algorithm. Then in order to study and evaluate the new algorithm, we put it into the test against the algorithm that uses the six most significant RGB bits with the use of Blocks Intensity Difference. As realized from the previous section, this algorithm already gave the best accuracy results using 25 blocks partitioning. The system also evaluates the total processing time in each experiment as an indication of the timing performance of the algorithms. The testing makes use of initial spatial skip values of five pixels.

General-purpose video files available in the public domain were used to provide the evaluation results. These files contain camera effects such as dissolving and fading in addition to normal shots. The results, in figure 6-2, show that although the accuracy is not 100%, it is still very effective. Table 6-2 shows the overall characteristics of the testing video files. Figure 6-3 shows frame samples from the test videos.

**Table 6-2.** Tested Video Files Description.

| Video File | Length (frames) | Format | Style | Description |
|---|---|---|---|---|
| egypt_copy3.avi | 302 | MS-AVI | Documentary | Documentary file about tourist destinations in Egypt, doesn't include sound |
| bigmist.mov | 390 | Apple-MOV | Song | Part of a song, includes sound |
| offswitch.mpeg | 568 | MPEG-1 | Film | Short film trailer. Includes sound |

Figure 6-1 shows the summary results of the processing time performance, of the two approaches with different values of the temporal skip parameter as an indication of the sensitivity analysis against this parameter. However, we should not forget the effectiveness of comparing the two algorithms in the presence of the timing performance measurements. Thus, figure 6-2 shows the recall and precision measures in each case. In both figures, we refer to the original approach by "Blind" and the new one by

"Binary Penetration". In the "Blind" approach, we use the six most significant RGB bits with the use of Blocks Intensity Difference algorithm and using a temporal skip parameter. When the difference of the two separated frames exceed the threshold, we analyze all the frames between the two separated frames. However in the "Binary Penetration" approach, we do the further analysis in a binary penetration fashion as we described in chapter 3.

As shown from figure 6-1, the generalized binary penetration algorithm has shown better processing time requirements in all cases. There is more obvious improvement when you use a larger temporal skip. The reason for this is that the original algorithm tests all the included frames when the difference between the two ends exceeds the threshold. This means that if the temporal skip is high, two side effects take place. First, the number of extracted and analyzed frames is larger. Second, the probability of obtaining false cut decisions at the first try is greater, due to any large object movement or camera operation even during the same camera shot. However, in the binary penetration algorithm, even this probability remains the same at the first level, the system could recognize this false cut operation directly in the second processing level or later without extracting and analyzing all the frames within the region's two ends. This represents a two-fold advantage. One is the need to extract less number of frames, thus relaxing the memory and storage requirements. Second, the speed of detecting a true cut is faster by average percentage ($T / \log_2 T$), where T is the temporal skip selected. Meanwhile, a false cut decision could be taken in as few as two levels.

Another inherent advantage is the use of already constructed frame histograms from upper levels. That means that we could use the histograms of the higher level (n) into the next lower level (n+1) testing. There are actually two histograms used from each higher level into the next lower level. For example, in figure 3-4, level 2 uses the histograms of frames {m and (m+k) / 2} from level 1 and so level 2 evaluates only the histogram of frame {(m+k) / 4} to make the decision at this level, and so on. As would be expected, this re-use of histograms saves a lot of computations and thus processing time.

At the same time as we mentioned, we didn't forget the effectiveness of the algorithm while evaluating the processing time performance. Figure 6-2 shows that we evaluated the recall and precision of all the

test experiments done in figure 6-1. We realize that the binary penetration algorithm still shows promising accuracy in all cases. However, we see that a temporal skip of ten frames gives the best results and good processing time performance for the given test set. The argument behind this is that in the binary penetration algorithm, the value of the temporal skip is more sensitive. Of course, it is better that the value of this temporal skip parameter be as high as possible to improve the performance. However, if this value is too high, another problem arises because, in the original binary penetration algorithm, there is the possibility of finding only one camera cut at maximum within a temporal skip number of frames. Therefore, if there is more than one camera cut within the temporal skip, only one will be detected and the others will be missed. Alternatively, we will need to use the generalized algorithm that was described in chapter 3, which will relatively reduce the performance in such situations. That means that this temporal skip value should be moderate and preferably related to the domain of the video information. For example, a news clip could have higher temporal skip value than musical or commercial clips.

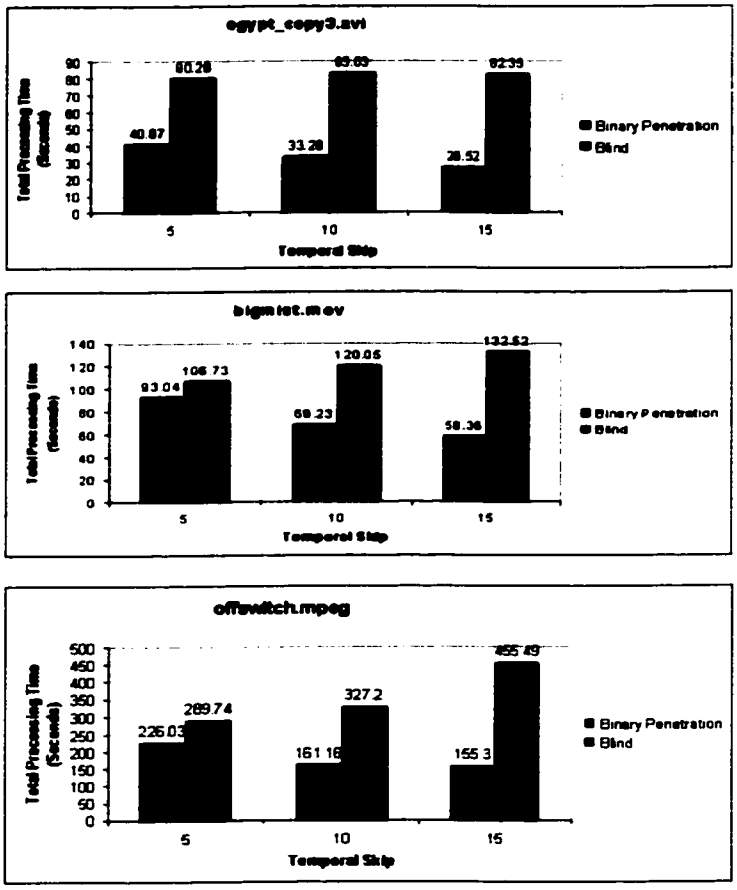



**Figure 6-1.** Performance comparative results.

*egypt_copy3.avi*

| Blind, 5 | Binary_Penetration, 5 | Blind, 10 | Binary_Penetration, 10 | Blind, 15 | Binary_Penetration, 15 |
|---|---|---|---|---|---|
| RECALL = 100 % | RECALL = 100 % | RECALL = 100 % | | RECALL = 100 % | RECALL = 100 % |
| PRECISION = 100% | PRECISION = 100% | PRECISION = 100% | | PRECISION = 100% | PRECISION = 100% |

*bigmist.mov*

| Blind, 5 | Binary_Penetration, 5 | Blind, 10 | Binary_Penetration, 10 | Blind, 15 | Binary_Penetration, 15 |
|---|---|---|---|---|---|
| RECALL = 85.71 % | RECALL = 85.71 % | RECALL = 85.71 % | | RECALL = 71.43% | RECALL = 71.43% |
| PRECISION = 100% | PRECISION = 100% | PRECISION = 100% | | PRECISION = 100% | PRECISION = 100% |

*offswitch.mpeg*

| Blind, 5 | Binary_Penetration, 5 | Blind, 10 | Binary_Penetration, 10 | Blind, 15 | Binary_Penetration, 15 |
|---|---|---|---|---|---|
| RECALL = 75 % | RECALL = 87.5 % | RECALL = 87.5 % | | RECALL = 87.5 % | RECALL = 75 % |
| PRECISION = 66.7% | PRECISION = 77.78% | PRECISION = 77.78% | | PRECISION = 63.64% | PRECISION = 75% |

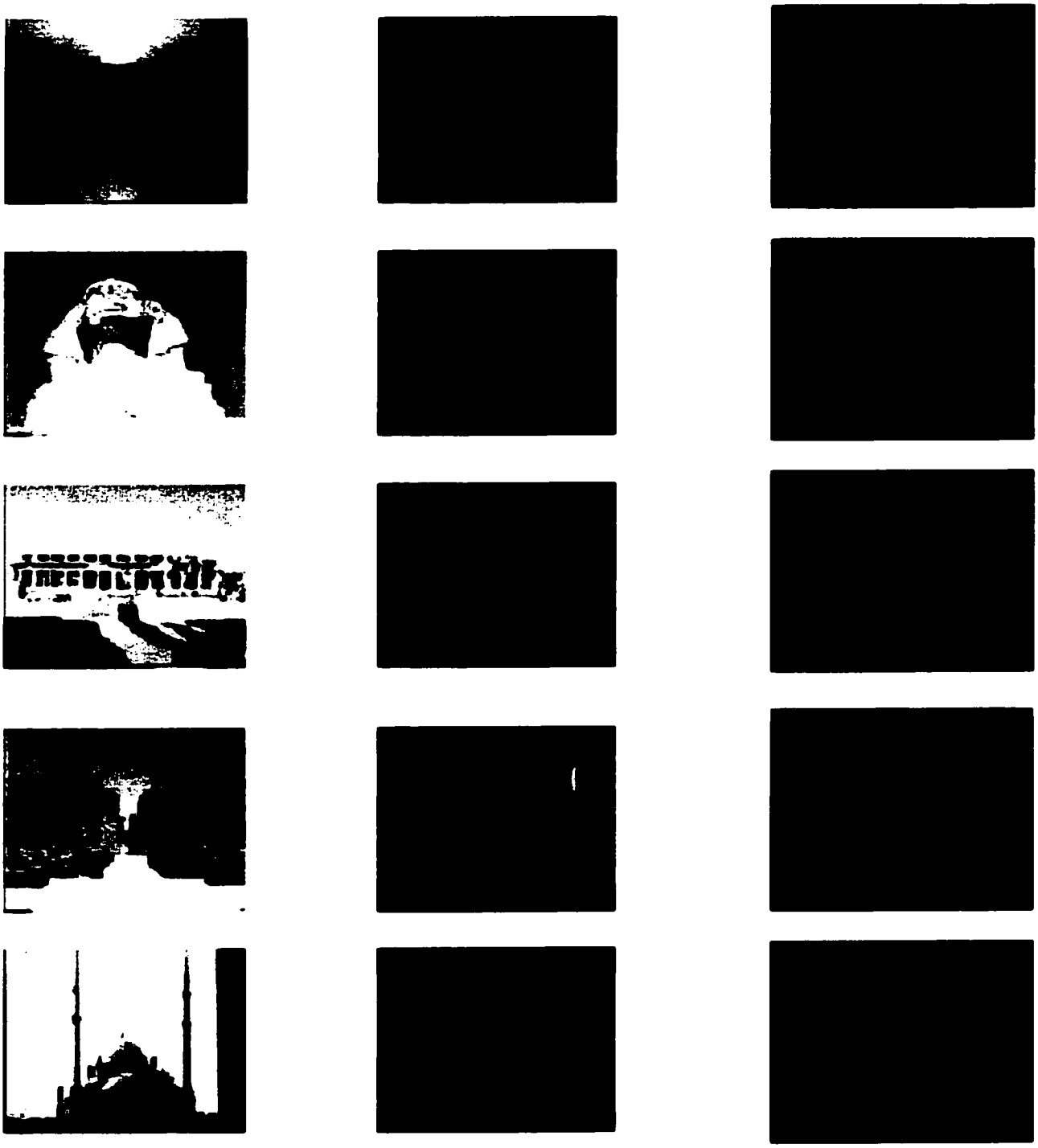
**Figure 6-2.** Recall and Precision comparative results.

Another important note is that the accuracy results of the third file as shown in figure 6-2, especially the precision measurement, are slightly less than the other files. The reason is that the use of the "six most significant RGB bits" generally reduces the sensitivity of the analysis algorithms in the case of dark video shots. As shown in figure 6-3, this file contains more dark shots than the others. However, the accuracy results are still considered reasonably acceptable and useful.

## 6.4. Histogram Threshold Sensitivity Analysis

Another important issue studied is the selection of the threshold value as a reference to measure the level of frame change. We performed some tests to discover an optimum value of the threshold percentage to be used to compare the different algorithms. We utilized the recall and precision measures to infer an effective threshold value. Intuitively, if the threshold tends to zero percent, the precision indicator will also tend to zero per cent. But the recall indicator will tend to 100% because the number of missed true video cuts will tend to zero. The analogy is reversed if the threshold level tends to 100%.

| | | | | | | |
|---|---|---|---|---|---|---|
| i.e. | if threshold → 0%, | Then, | precision → 0% | and | recall → 100% |
| and | if threshold → 100%, | Then, | precision → 100% | and | recall → 0% |



a) egypt_copy3.avi b) bigmist.mov c) offswtich.mpeg

**Figure 6-3.** Example Snapshots from Test Videos.

We used the video files to measure the sensitivity of these accuracy measures against the value of the threshold and thus to select an effective threshold value accordingly. As seen in figures 6-4 and 6-5, the effective threshold value should range between 20% and 30%. Therefore in our implementations, we used the value of 25% for the histogram threshold parameter.

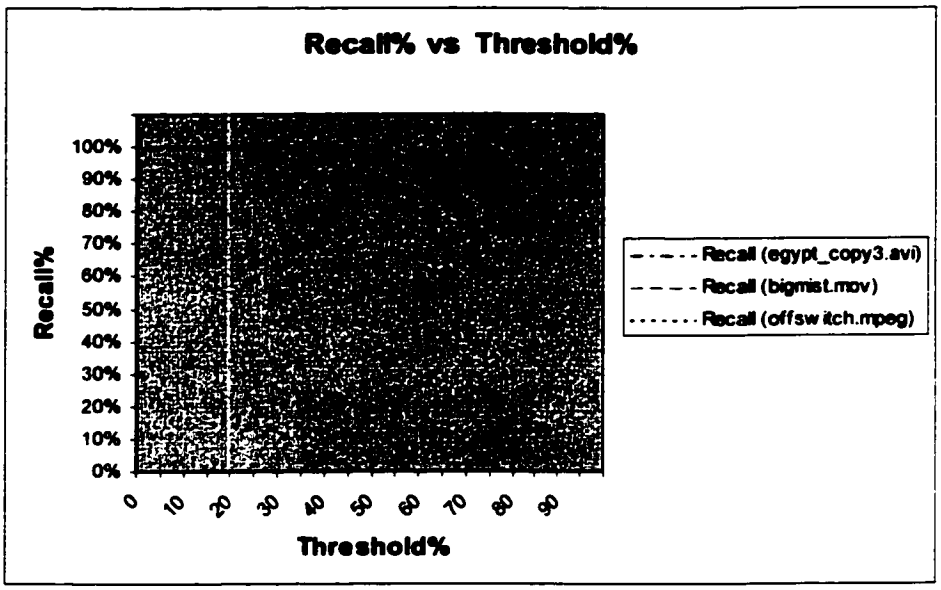



**Figure 6-4.** Sensitivity Analysis of Recall vs. Threshold Value Selection.

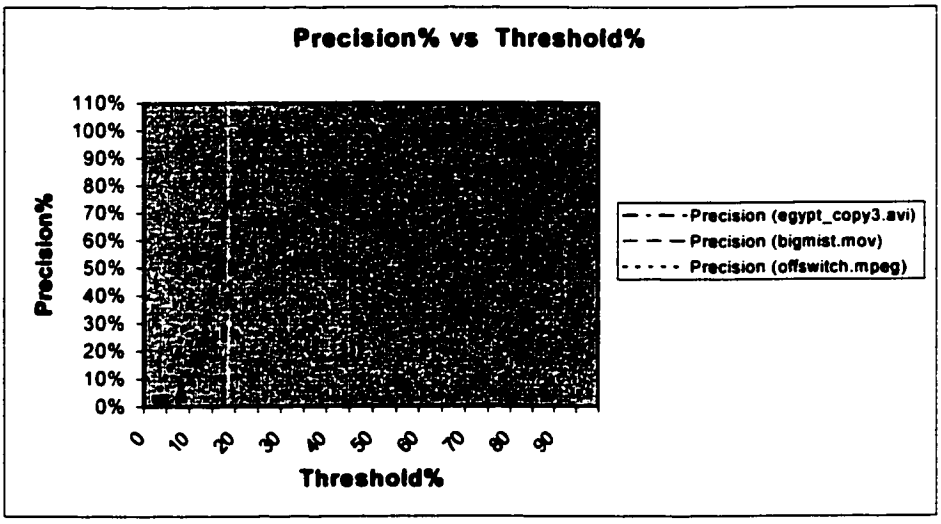



**Figure 6-5.** Sensitivity Analysis of Precision vs. Threshold Value Selection.

# CHAPTER 7

# CONCLUSIONS AND FUTURE DIRECTIONS

## 7.1. Conclusions

Current problems in accommodating multi-media content in telecommunications environments vary from the lack of resource availability, limited device capabilities, inadequate time for the user to browse video contents and organizations with restrictive policies for the use of its infrastructure resources. With the present boom in the use of mobile and wireless devices, we have to look for new methods to convey information.

In this thesis, we began by describing the original algorithm developed to segment the video stream and

detect key frames. These key frames can be used to index the segment and to partition the video into its original shots. We partitioned the selected frames into exclusive blocks as a solution to the problem of missing true cuts. The system uses spatial and temporal skips to enhance the performance of the analysis. However, to compensate for the use of the temporal skip, the system re-analyzes all the frames in the area if the first test shows the potential for a cut.

We moved on to explain the updated algorithm using the binary penetration mechanism. The algorithm uses the heuristics resulting from the temporal dependency of the visual information. The new approach resulted in performance superior to the original algorithm, both in processing time, and in memory and storage requirements. Tests of the analysis show no loss of accuracy.

We presented the system developed to allow seamless browsing of multi-format media and the detection of camera cuts. We described the architecture used in the system, and its modules. We provided the functions and services of each module. In addition, we explained the flow of data and information in the system.

We also described the prototype developed for our agent-based project for nomadic computing. We explained the components of the prototype and the interactions between these components. We then demonstrated the use of the video indexing and summarization system as a service built on that architecture.

Our approach to video semantic analysis uses a general abstraction model. We introduce the notion of triggers that direct and guide our system to recognize the events of the domain. The triggers and the features of the domain can be mapped into the technical tool using human expertise. Each domain has its own facts, interesting events, and heuristics to consider. Each event is mapped by its distinct technical clues. The use of generic knowledge to recognize expected important events in the video stream remains as an option.

Spatial, temporal and logical relationships help to detect, and then resolve, the conflicts or uncertainties

that can arise during the analysis phase. We provided event decision formulas that take into consideration the confidence of the entire system in each tool in addition to the confidence of each tool in its own results. The procedure takes into account the effect of technical tool errors because the tools are still not as efficient as human perception. Different levels of abstraction can be defined by human expertise or machine training. We chose a soccer game as a case study. Similar reasoning can be adapted to other domains such as movies, news, and documentaries.

We operated the work over our agent architecture. According to available resources, user profile preferences and device capabilities, the system selects the format and level of the abstraction content. We positioned our service within the overall infrastructure, and we provided scenarios illustrating its use in different conditions over the Internet.

## 7.2. Limitations of this thesis

Though many of the topics related to multimedia communications have been discussed, we know that some issues remain to be addressed.

1. Our key framing algorithm was not designed for the detection of gradual transitions operations. However, we believe that our approach could be used orthogonal to both types of camera operations, producing faster results that would remain highly efficient.

2. Our soccer domain example needed a limited set of processing tools. However, we think that other sport domains are similar. With the emphasis on a highly semantic segmentation of events, we feel that the approach also can be applied to other domains such as news and documentaries, even general subject matter. Further, more profound psychological studies could be conducted to relate the user's comprehension of the content to different levels of summarization and formats.

3. More studies are needed to measure the reliability and scalability of our architecture and service.

4. For mobile use of multimedia content, research could be conducted on the best user interface for each type of mobile device. This study should not just focus on the conversion of and adaptation to the media formats, but also on the logical layout of multimedia documents.

## 7.3. Future Directions

The areas of intelligent multimedia and mobile communications are very interesting and promising. Many companies and research organizations are currently working in these fields, especially on the development of third generation wireless communications and IPv6. We describe below the directions in which we believe that our work could be extended.

1. We believe that multimedia services could become native functionalities in media capturing and storage systems. Semantic services, for example, would not just be external or optional, especially for MPEG-7 and MP3's ID3 and ID3v2 contents in multimedia databases and media production systems. This will guarantee faster, more accurate and consistent user access, query and navigation.

2. Our services could become video web services given the increase in such services and peer-to-peer technologies. This would provide different function granularities from a number of vendors using well-defined, high-level interface descriptions and cooperation semantics. Other growing technologies could also be incorporated. Grid computing [FOS02], for example, seeks to provide transparent sharing of data across high-speed networks, with services such as math libraries, visualization, and DSP tools, and with resources such as memory, secondary storage, CPU power and available cycles. Other possibilities also include resource discovery, intelligent job management and forecasting, scheduling, single sign-on authentication, delegation of credentials and the like, irrespective of any dispersed network domains.

3. Our approach could be used with initiatives just getting under way that seek to provide location-based services for mobile users. One example is the MPEG-7 working document on mobile requirements and applications [ISO01] from the Moving Picture Experts Group (MPEG). They are

currently working on enabling mobile access to MPEG-7 information using GPS, context-aware and location-dependent technologies. A user would be able to watch the trailers of current movies wirelessly from the nearest cinema while he is walking or driving his car. They also intend to use both push and pull mechanisms for information access.

# References

[AHM99a] Ahmed M., Abu-Hakima S. and Karmouch A., "Key Frame Extraction and Indexing for Multimedia Databases," *Visual Interface 99 Conference (VI'99)*, Québec Canada, May 19-21 1999, pp. 506-511.

[AHM99b] Ahmed M. and Karmouch A., "Improving Video Processing Performance using Temporal Reasoning," *SPIE-Applications of Digital Image Processing XXII*, Denver CO USA, Vol. 3808, July 20-23 1999, pp. 645-656.

[AHM99c] Ahmed M. and Karmouch A., "New Architecture for Multi-Format Video Browsing and Cut Detection," *Proc. of IEEE CCECE'99 Conference*, Edmonton Canada, May 9-12 1999, pp. 821-826.

[AHM99d] Ahmed M. and Karmouch A., "Video Segmentation using an Opportunistic Approach," *MultiMedia Modeling 99*, Ottawa Canada, October 4-6 1999, pp. 389-405.

[HAR00] Harroud H., Ahmed M. and Karmouch A., "An Agent-based Service Provisioning System for Mobile Users," *International Symposium on Image/Video Communications over Fixed and Mobile Networks (ISIVC'2000)*, Rabat Morocco, April 17 -20 2000, pp. 203-212.

[AHM00] Ahmed M. and Karmouch A., "Model-Based Video Summarization for Mobile Users," *MultiMedia Modeling 2000*, Nagano Japan, November 13-15 2000, pp. 287-311.

[HAR01a] Harroud H., Ahmed M. and Karmouch A., "Agent-based Personalized Services for Mobile Users over a VPN," *IEEE 3rd International Conference on Enterprise Information Systems (ICEIS'2001)*, Setúbal Portugal, July 7-10 2001, pp. 1110-1117.

[AHM01] Ahmed M. and Karmouch A., "Uncertainties and Event Conflict Resolution within Video Analysis Service," *MultiMedia Modeling 2001*, Amsterdam Netherlands, November 5-7 2001, pp. 19-38.

[HAR01b] Harroud H., Ahmed M. and Karmouch A., "Agent-based Personalized Services in a Mobile Computing Environment," *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'01)*, Victoria Canada, August 26-28, 2001, pp. 728-731.

[HAR02] Harroud H., Ahmed M. and Karmouch A., "Agent-based Personalized Services for Mobile Users over a VPN," *selected from ICEIS'2001 to represent a chapter in the book: Enterprise Information Systems III, published by Kluwer Academic Publishers, Dordrecht Hardbound*, April 2002, pp. 312-319, ISBN 1-4020-0563-6.

[AHM02] Ahmed M. and Karmouch A., "Video Indexing Using a High-Performance and Low-Computation Color-Based Opportunistic Technique," *Journal of SPIE Optical Engineering*, Volume 41, Issue 2, Feb. 2002, pp. 505-517.

[AME01] Amer M., Karmouch A., Gray T. and Mankovskii S., "An Agent Model for the Resolution of Feature Conflicts in Telephony," *Journal of network and systems management*, Vol. 8, No. 3, Sept. 2000.

[BAB00] Babaguchi N., Kawai Y., Yasugi Y. and Kitahashi T., "Linking Live and Replay Scenes in Broadcasted Sports Video," *Proceedings of ACM Multimedia 2000 Workshop on Multimedia Information Retrieval (MIR2000)*, Los Angeles - California, October 30 - November 4 2000, pp. 205-208.

[BEY98] Beyerlein P., Aubert X., Haeb-Umbach R., Klakow D., Ulrich M., Wendemuth A. and Wilcox P., "Automatic Transcription of English Broadcast News," *DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne VA USA, Feb 8-11, 1998.

[BOL99] Boll S., Wolfgang K. and Wanden J., "A cross-Media Adaptation Strategy for Multimedia Presentation," *Proceedings of the ACM Multimedia'99*, Orlando FL USA, Oct. 30 – Nov. 5 1999, pp. 37-46.

[BOR96] Boreczky J. and Rowe L., "A Comparison of Video Shot Boundary Detection Techniques," *Journal of Electronic Imaging*, Vol. 5, No. 2, 1996, pp. 122-128.

[BRA94] Brandenburg K., Stoll G. et al., "ISO-MPEG-1 Audio: A Generic Standard for Coding of High Quality Digital Audio," *Journal of the Audio Engineering Society*, Vol. 42, No. 10, Oct. 1994, pp. 780–792.

[BUL98] Bulterman D., "User-Centered Abstractions for Adaptive Hypermedia Presentations," *Proc. ACM Multimedia'98*, Bristol UK, Sept. 12-16 1998, pp. 247-256.

[CCI90] CCITT Recommendation H.261: "Video Codec for Audio Visual Services at $p$ * 64 kbits/s," COM XV-R 37-E, 1990.

[CHR97] Christel M., Winkler D. and Taylor C., "Multimedia Abstractions for a Digital Video Library," *2nd ACM International Conference on Digital Libraries*, Philadelphia PA, July 1997, pp. 21-29.

[CHR98] Christel M., Smith M., Taylor C. and Winkler D., "Evolving Video Skims into Useful Multimedia Abstractions," *Proceedings of the CHI '98 Conference on Human Factors in Computing Systems*, Los Angeles CA, April 1998, pp. 171 - 178.

[COR98] "Corel WordPerfect Suite 8 – Legal Edition with Dragon NaturallySpeaking", *White paper*, May 1998. http://www.corel.com/products/wordperfect/legal8/product_overview.htm

[DAW99] Dawood A., and Ghanbari M. "Content-Based MPEG Video Traffic Modeling," *IEEE Transactions on Multimedia*, Vol. 1, No. 1, Mar. 1999, pp. 77-87.

[DUN97] Dunn D., Weldon T. and Higgins W., "Extracting Halftones from Printed Documents using Texture Analysis," *Optical Engineering*, Vol. 36, No. 4, April 1997, pp. 1044-1052.

[ELO95] Elofson G., "Intelligent Agents Extend Knowledge-Based Systems Feasibility," *IBM Systems Journal*, Vol. 34, No. 1, 1995, pp. 78-95.

[FIS95] Fischer S., Lienhart R. and Effelsberg W., "Automatic Recognition of Film Genres," *ACM Multimedia 95*, San Francisco CA, Nov. 1995, pp. 295-304.

[FOS02] I. Foster I., Kesselman C., Nick J. and Tuecke S., "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Jan. 2002.
http://www.globus.org/research/papers/ogsa.pdf

[Gal91] Gall D. J. Le, "MPEG: A video compression standard for Multimedia applications," *Communications of the ACM*, Vol. 34, No. 4, April 1991, pp. 47-58.

[GAR96] Gargi U. and Kasturi R., "An Evaluation of Color Histogram Based Methods in Video Indexing," *Proceedings of the first international workshop on Image databases and multi-media search (IDB-MMS '96)*, Amsterdam-The Netherlands, August 22-23 1996, pp. 75-82.

[GAR98] Gargi U, Kasturi R. and Antani S., "Performance Characterization and Comparison of Video Indexing Algorithms," *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, June 1998, pp. 559-565.

[GAR00] Gargi U., Kasturi R. and Strayer S., "Performance Characterization of Video-Shot-Change Detection Methods," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 10, No. 1, February 2000, pp. 1-13.

[GOM96] Gómez H., Martínez B., Ballester R., Núñez V. and García F., "Audiovisual content encoding technologies for a new generation of services," *Special Issue on On-Line Services, Journal of Comunicaciones de Telefónica I+D*, Issue 14, December 1996.

[GUS88] Gustafson, J. L.. "Reevaluating Amdahl's Law," *Journal of Communications of the ACM*, Vol. 31, No. 5, May 1988, pp. 532-533.

[HAR85] Haralick R. M. and Shapiro L. H., "Image Segmentation techniques," *Journal of Computer Vision, Graphics and Image Processing*, Academic Press, 1985, pp. 100-132.

[HAR99] Harroud H., Karmouch A., Gray T. and Mankovski S., "An Agent-based Architecture for Inter-Sites Personal Mobility Management System," *Mobile Agents for Telecommunication Applications Workshop, MATA '99*, Ottawa Canada, Oct. 6-8 1999, pp. 345-357.

[HIR97] Hirzalla N. B., "Media Processing and Retrieval Model for Multimedia Documents," *PhD Thesis*, Ottawa University, Jan. 1997.

[HUA97] Huang J., Kumar S., Mitra M., Zhu W. and Zabih R., "Image Indexing using Color Correlograms," *IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, June 1997, pp. 762-768.

[HUA99] Huang J., Kumar S., Mitra M., Zhu W. and Zabih R., "Spatial Color Indexing and Applications," *International Journal of Computer Vision*, Vol. 35, No. 3, Dec. 1999, pp. 245-268.

[ITU95] ITU-T SG 15 WP 15/1, Draft Recommendation H.263 "Video coding for low bitrate communications", Document LBC-95-251, Oct. 1995.

[ISO94] ISO/IEC 13818-3, "Information Technology: Generic coding of Moving pictures and associated audio - Audio Part," International Standard, 1994.

[ISO97] ISO/IEC 13818-7, "MPEG-2 advanced audio coding, AAC," International Standard, 1997.

[ISO98] ISO/IEC JTCI/SC29/WG11, "MPEG-4 Overview - (Dublin Version)," N2323, July 1998.

[ISO00a] ISO/IEC JTCI/SC29/WG11, "MPEG-4 Overview- (V.16 – La BauleVersion)," N3747, October 2000.

[ISO00b] ISO/IEC JTCI/SC29/WG11, "Overview of the MPEG-7 Standard (version 4.0)," N3752, October 2000.

[ISO01] ISO/IEC JTCI/SC29/WG11, "Study of MPEG-7 Mobile Requirements & Applications (ver.2)," Nxxxx, Singapore, 2001.

[KAN97] Kanade T., Satoh S. and Nakamura Y., "Accessing Video Contents: Cooperative Approach between Image and Natural Language Processing," Proc. of ISDL'97, 1997, pp. 143-150.

[KAN99] Kang H. B., "Key Frame Selection using Region Information and its Temporal Variations," *Internet and Multimedia Systems and Applications, IMSA '99*, Oct. 18-21, 1999, pp. 33-37.

[KAS96] Kasturi R., Strayer S., Gargi U. and Antani S., "An Evaluation of Color Histogram Based Methods in Video Indexing," *Technical Report CSE-96-053*, Dept. of Computer Science and Engineering, The Pennsylvania State University, 1996.

[KAT98] Katsaggelos A., Kondi L., Meier F., Ostermann J. and Schuster G., "MPEG-4 and rate-distortion-based shape coding techniques," *Proc. IEEE*, Vol. 86, 1998, pp. 1029-1051.

[KNI96] Knill K. and Young S., "Fast implementations of Viterbi-based word-spotting," *In Proc. of ICASSP'96*, Atlanta USA, May 1996, pp. 520-523.

[LAW99] Lawrence S., Ziou D. and Wang S., "Motion Insensitive Detection of Cuts and Gradual Transitions in Digital Video," *MultiMedia Modeling 99*, Ottawa Canada, Oct. 4-6 1999, pp. 407-420.

[LEE95] Lee J. C. and IP D. M., "A Robust Approach for Camera Detection in Color Video Sequence," *Technical Report HKUST-CS95-14*, April 1995.

[LIE97] Lienhart R., Pfeiffer S. and Effelsberg W., "Video Abstracting," *Communications of the ACM*, Vol. 40, No. 12, Dec. 1997, pp. 55-62.

[LIE99] Lienhart R., "Abstracting Home Video Automatically," *Seventh ACM International Multimedia Conference*, Orlando FL USA, Oct. 30 - Nov. 5 1999, pp. 37-40.

[LIE00] Lienhart R., Effelsberg W. and Jain R., "VisualGREP: A Systematic Method to Compare and Retrieve Video Sequences," *Multimedia Tools and Applications*, Vol. 10, Issue 1, 2000, pp. 47-72.

[JIA98] Jiang H., Helal A., Elmagarmid A. and Joshi A., "Scene Change Detection Techniques for Video Database Systems," *ACM Multimedia Systems*, Vol. 6, No. 3, May 1998, pp. 186-195.

[MIT96] Mitel Corporation, "Mitel-CATA (The MicMac software Testbed)," *Technical Report*, CITI/Adaptive Information Systems, 1996.

[NAK97] Nakamura Y. and Kanade T., "Semantic Analysis for Video Contents Extraction- Spotting by Association in News Video," *ACM MultiMedia 97*, Seattle USA, Nov. 8-14 1997, pp. 393-401.

[NAM99] Nam J. and Tewfik A., "Event-Driven Video Abstraction and Visualization," *MultiMedia Modeling 99*, Ottawa Canada, Oct. 4-6 1999, pp. 193-213.

[OTS93] Otsuji K. and Tonomura Y., "Projection Detecting Filter for Video Cut Detection," *ACM Multimedia 93*, CA USA, June 1993, pp. 251-257.

[PAN01] Pan H., Beek P. and Sezan M., "Detection of slow-motion replay segments in sports video for highlights generation," *will be presented in IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP'01)*, Salt Lake City, Utah, May 7-11, 2001.

[PAS96a] Pass G. and Zabih R., "Histogram Refinement for Content-Based Image Retrieval," *In IEEE Workshop on Applications of Computer Vision*, Dec. 1996, pp. 96-102.

[PAS96b] Pass G., Zabih R. and Miller J., "Comparing Images Using Color Coherence Vectors," *ACM Multimedia 96*, Boston MA, USA, 1996.

[PAS99] Pass G. and Zabih R., "Comparing Images Using Joint Histograms," *ACM Journal of Multimedia Systems*, Vol. 7, No. 3, May 1999, pp. 234-240.

[PFE96] Pfeiffer S., Lienhart R., Fischer S. and Effelsberg W., "Abstracting Digital Movies Automatically," *Journal of Visual Communication and Image Representation*, Vol. 7, No. 4, pp.345-353, Dec. 1996, pp. 345-353.

[PLA97] Placeway P., Chen S., Eskenazi M., et al., "The 1996 Hub-4 Sphinx-3 System," *Proc. of DARPA Spoken Systems Technology Workshop*, Morgan Kaufmann Publishers, Chantilly Virginia, Feb. 1997, pp. 85-89.
http://www.cs.cmu.edu/afs/cs.cmu.edu/user/pwp/web/papers/h496_system/H496CMU.HTM

[PUR93] Puri A., "Video coding using the MPEG-2 compression standard," *Proc. Of SPIE Intl. Conf. Visual Communications and Image Processing (VCIP'93)*, Cambridge, MA, 1993, vol. 2094, pp. 1701-1713.

[REC93] Recommendation X.500 ITU-T. "Information technology -Open Systems Interconnection- The Directory: Overview of Concepts, Models, and Services," 1993.

[ROU99] Rousseau F., Antonio García-Macías J., Valdeni de Lima J. and Duda A., "User Adaptable Multimedia Presentations for the WWW," *Proc. Eight International World-Wide Web Conference*, Toronto Canada, May 1999, pp. 1273-1290.

[SAL83] Salton G. and McGill M., "Introduction to Modern Information Retrieval," *McGraw-Hill*, 1983.

[SAL02] Sallabi F., "End-to-End Quality of Service Support for Multimedia Applications in the Internet," *PhD Thesis*, Ottawa University, 2002.

[SAT99] Sato T., Kanade T., Hughes E., Smith M. and Satoh S., "Video OCR: indexing digital news libraries by recognition of superimposed captions," *ACM Journal of Multimedia Systems*, Vol. 7, No. 5, Sep. 1999, pp. 385-395.

[SET98] Sethi I., Coman I., Day B., et al. "Color-WISE: A System for Image Similarity Retrieval Using Color," *Proc. of SPIE, Storage and Retrieval for Image and Video Databases VI*, Vol. 3312, Jan. 1998, pp. 140-149.

[SIK97] Sikora T., "MPEG-4 very low bit rate video," *Proc. of the IEEE International Symposium on Circuits and Systems ISCAS '97*, Hong Kong, June 6-9 1997.

[SMI97] Smith M. and Kanade T., "Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques," *Computer Vision and Pattern Recognition Conference*, San Juan- Puerto Rico, June 1997, pp. 775-781.

[SWA91] Swain M. and Ballard D., "Color Indexing," *International Journal of Computer Vision*, Vol. 7, No. 1, 1991, pp. 11-32.

[SWE95] Swets D., Punch B. and Weng J., "Genetic Algorithms for Object Recognition in a Complex Scene", *Proceedings of International Conference on Image Processing (ICIP'95)*, Washington D.C. USA, October 1995, pp. 595-598.

[TAS98] Taskiran C. and Delp E., "Video Scene Change Detection Using the Generalized Trace," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP98)*, Seattle Washington, May 12-15 1998.

[UCH99] Uchihashi S., Foote J., Girgensohn A. and Boreczky J., "Video Manga: Generating Semantically Meaningful Video Summaries," *Seventh ACM International Multimedia Conference*, Orlando FL USA, Oct. 30 - Nov. 1999.

[W3C99] W3C Note, "Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiation," 27 July 1999.

[WAC00] Wactlar H., Hauptmann A., Christel M., Houghton R. and Olligschlaeger A., "Complementary Video and Audio Analysis for Broadcast News Archives," *Communications of the ACM*, Vol. 32, No. 2, February 2000, pp. 42-47.

[WEL96] Weldon T., Higgins W. and Dunn D., "Efficient Gabor Filter Design for Texture Segmentation." *Pattern Recognition*, Vol. 29, No. 12, Dec. 1996, pp. 2005-2015.

[WEL94] Weldon T., Higgins W. and Dunn D., "Efficient Gabor Design using Rician Output Statistics," *in IEEE Int. Symposium. Circuits, Systems*, London England, Vol. 3, 30 May-2 June 1994, pp. 25-28.

[WOL96] Wolf W., "Key Frame Selection by Motion Analysis," *Proc. Of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta USA, Vol. 2, May 1996, pp. 1228-1231.

[WU97] Wu V., Manmatha R. and Riseman E., "Finding Text in Images," *In the Proceedings of the 2nd ACM International Conference on Digital Libraries (DL'97)*, Philadelphia USA, July 23-26 1997, pp. 3-12.

[XIO97] Xiong W., Lee J. and Ma R., "Automatic Video Data Structuring through Shot Partitioning and Key Frame Computing," *Machine Vision and Applications*, Vol. 10, Issue 2, 1997, pp. 51-65.

[XML02] XMLwriter version 1.21, website, http://www.xmlwriter.net/index.shtml

[YEU96] Yeung M., Yeo B. and Liu B. "Extracting Story Units from Long Programs for Video Browsing and Navigation," *International Conference on MultiMedia Computing and Systems*, Hiroshima Japan, June 17-21 1996, pp. 296-305.

[YOW95] Yow D., Yeo B., Yeung M. and Liu B. "Analysis and Presentation of Soccer Highlights from Digital Video," *Second Asian Conference on Computer Vision (ACCV)*, Singapore, Vol. 2, Dec. 1995, pp. 499-503.

[ZAB95] Zabih R., Miller J. and Mai K., " A Feature-Based Algorithm for Detecting and Classifying Scene Breaks," *Proc. Of ACM Multimedia*, San Francisco CA USA, 1995, pp. 189-200.

[ZAB99] Zabih R., Miller J. and Mai K., "Feature-Based Algorithms for Detecting and Classifying Production Effects," *ACM Journal of Multimedia Systems*, Vol. 7, No. 2, March 1999, pp. 119-128.

[ZHA93] Zhang H., Kankanhalli A. and Smoliar S. "Automatic Partitioning of Full-Motion Video," *Multimedia Systems*, Vol. 1, No. 1, Apr. 1993, pp. 10-28.

[ZHA98] Zhang T. and Kuo J., "Content-Based Classification and Retrieval of Audio" *SPIE's 43rd Annual Meeting - Conference on Advanced Signal Processing Algorithms, Architectures, and Implementations VIII*, San Diego, SPIE Vol. 3461, July 1998, pp. 432-443.

[ZHA99] Zhang T. and Kuo C., "Heuristic Approach for Generic Audio Data Segmentation and Annotation," *Seventh ACM International Multimedia Conference*, Orlando FL USA, October 30-November 5, 1999, pp. 67-76.

[ZHE99] Zheng C., Karmouch A., Gray T., Mankovski S. and Impey R., "Ensuring Secure Communication for a Distributed Mobile Computing System based on MicMac," *Mobile Agents for Telecommunication Applications Workshop, MATA '99*, Ottawa Canada, Oct. 6-8 1999, pp. 375-391.