SEMANTIC VIDEO MODELING AND RETRIEVAL WITH VISUAL, AUDITORY, TEXTUAL SOURCES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

NURCAN DURAK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2004

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Nurcan DURAK

Signature :

ABSTRACT

SEMANTIC VIDEO MODELING AND RETRIEVAL
WITH
VISUAL, AUDITORY, TEXTUAL SOURCES

Durak, Nurcan
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. Adnan Yazıcı
September 2004, 95pages

The studies on content-based video indexing and retrieval aim at accessing video content from different aspects more efficiently and effectively. Most of the studies have concentrated on the visual component of video content in modeling and retrieving the video content. Beside visual component, much valuable information is also carried in other media components, such as superimposed text, closed captions, audio, and speech that accompany the pictorial component. In this study, semantic content of video is modeled using visual, auditory, and textual components. In the visual domain, visual events, visual objects, and spatial characteristics of visual objects are extracted. In the auditory domain, auditory events and auditory objects are extracted. In textual domain, speech transcripts and visible texts are considered. With our proposed model, users can access video content from different aspects and get desired information more quickly. Beside multimodality, our model is constituted on semantic hierarchies that enable querying the video content at different semantic levels. There are sequence-scene hierarchies in visual domain, background-foreground hierarchies in auditory domain, and subject hierarchies in speech domain. Presented model has been implemented and multimodal content queries, hierarchical queries, fuzzy spatial queries, fuzzy regional queries, fuzzy spatio-temporal queries, and temporal queries have been applied on video content successfully.

Keywords: Multimodal Video Modeling and Retrieval, Audio, Visual, Textual, Spatio-temporality.

ÖZ

GÖRSEL, İŞİTSEL, YAZISAL KAYNAKLARLA
ANLAMSAL VİDEO MODELLENMESİ ve ERİŞİMİ

Durak, Nurcan
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Profesör Dr. Adnan Yazıcı
Eylül 2004, 95sayfa

İçeriğe dayalı video indeksleme ve erişim çalışmaları, videonun içeriğine farklı yönlerden daha etkin ve başarılı bir şekilde ulaşılmasını amaçlar. Çalışmaların çoğu, video içeriğinin modellenmesinde ve erişilmesinde görsel bileşen üzerine yoğunlaşmıştır. Görsel içeriğin yanında diğer medya bileşenleri de değerli bilgiler taşır. Diğer medya bileşenleri video üzerinde ki yazılar, başlıklar, işitsel bilgiler, resimlere eşlik eden konuşmalar olabilir. Bu çalışma videonun anlamsal içeriğini görsel, işitsel ve yazısal bileşenler kullanarak modellemektedir. Görsel alanda, görsel olaylar, görsel nesneler ve görsel nesnelerin uzaysal-zamansal özellikleri çıkartılmıştır. İşitsel alanda, işitsel olaylar ve işitsel nesneler çıkartılmıştır. Yazısal alanda, konuşma metni ve video üzerindeki yazılar göz önüne alınmıştır. Önerdiğimiz modelle, kullanıcılar video içeriğine farklı yönlerden ulaşabilirler ve istedikleri bilgiyi daha çabuk elde ederler. Çok-biçimli yapısının yanında, modelimiz video içeriğini farklı anlamsal katmanlarda sorgulayabilmek için anlamsal hiyerarşiler üzerine kurulmuştur. Görsel alanda sekans-sahne hiyerarşileri, işitsel alanda arkaplan-önplan hiyerarşileri, konuşma alanında da konu hiyerarşileri vardır. Tanıtılan model geliştirilmiş ve çok-biçimli sorgulamalar, hiyerarşik sorgulamalar, bulanık konumsal sorgulamalar, bulanık bölgesel sorgulamalar, bulanık uzaysal-zamansal sorgulamalar, ve zamansal sorgulamalar video içeriği üzerine başarıyla uygulanmıştır..

Anahtar Kelimeler: Çok-biçimli Video Modelleme ve Erişme, İşitsel, Görsel, Yazısal, Uzaysal-Zamansal.

To Selahattin and Makbule Durak,

ACKNOWLEDGMENTS

I would like to thank to Prof. Dr. Adnan Yazıcı for his confidence, positive guidance, suggestions, and corrections. He always motivated and trusted me during my study.

I would like to thank to Computer Engineering Department of Başkent University for letting to my master study.

I would like to thank to my dear friends Oya and Rabia for their moral supports during my study. I am grateful to Muharrem for his confidence, suggestions, questions, and being my side every time. I am thankful to my family for their believing to finish my study and loving me.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ASR	:	**A**utomatic **S**peech **R**ecogntion
AVI	:	**A**udio **V**ideo **I**nterleave
AVIS	:	**A**dvanced **V**ideo **I**nformation **S**ystem
FST	:	**F**rame **S**egment **T**ree
JMF API	:	**J**ava **M**edia **F**ramework **A**pplication **P**rogramming **I**nterface
MBR	:	**M**inimum **B**ounding **R**ectangle
OCR	:	**O**ptic **C**haracter **R**ecognition
OVID	:	**O**bject-Oriented **V**ideo **I**nformation **D**atabase
SQL	:	**S**tructured **Q**uery **L**anguage
VDBS	:	**V**ideo **D**ata **B**ase **S**ystem
VOCR	:	**V**ideo **O**ptic **C**haracter **R**ecognitiom
TV	:	**T**ele**v**ision

CHAPTER 1

INTRODUCTION

With the developments in data capturing, storing, and transferring technologies, video usage has increased in different applications such as digital libraries, distance learning, video conferencing, and so on. Increased video data requires effective and efficient data management. Video data is different from textual data since video has image frames, sound tracks, texts that can be extracted from image frames, spoken words that can be deciphered from the audio track, temporal, and spatial dimensions. Multiple sources of video increase the volume of video data and make it difficult to store, manage, access, reuse, and compose.

Video's semantics are embedded in multiple forms that are usually complimentary of each other. By seeing, hearing, and reading, we can understand the subject of the video. After watching, we usually remember important scenes having distinct events, objects, speech or texts. In other words, we remember the content of the video and want to access these particular scenes without watching the entire video. To model the huge content of the video and to return exact answers to asked questions from video content are the part of the *content based video modeling and retrieval* studies in [1,6,9,17,24]. In content-based video modeling [1,5,8,9,25], video is segmented into meaningfully manageable portions, then these portions are abstracted with key-frames, and finally semantically rich entities and relationships are extracted and indexed. *Indexing* is the process of parsing the video and the associating different segments of the video with events and objects in the given domain. All indexing processes can be manual, segmentation process can be automatic [19, 20, 23, 26, 31,33], and object extraction can be semi-automatic [9, 10, 19].

In the semantic modeling of video content studies, OVID [24] and AVIS [1] are popular models. OVID models objects, whereas AVIS models objects, events and activities (event types). AVIS supports semantic queries in which frame intervals for given semantic entities or semantic entities in the specified time interval can be asked. AVIS also supports conjunctive queries that are a group of same type queries and compound queries involving different relationships of events and objects.

Relationships between semantic entities can be temporal or spatial. Temporal relationships between events clarify event interactions in time order. Temporal information contains either a range of video frames or an ordering relationship between two intervals. A set of temporal relations is well defined in Allen's temporal algebra [4]. Spatial relationships between visual objects provide details of the relative location of objects. Spatial relationships can be directional or topological relations. These relations are defined using Allen's temporal algebra in [18]. Spatio-temporal relationships consider changing object locations and provide to query moving object trajectories.

Köprülü et. al. [17] extend the AVIS model by modeling the spatio-temporal properties of object entities. Spatial relationships between objects can not be described exactly, and objects do not move in strict line direction, considering these uncertainties. Köprülü defines fuzzy spatial relationships between objects, and fuzzy object trajectories and enables fuzzy spatio-temporal queries, fuzzy object trajectory queries and regional queries.

BilVideo in [9] integrates spatio-temporal and semantic queries of video data. Spatio-temporal queries can contain various conditions such as directional, topological, object-appearing, trajectory projection and similarity-based object trajectories. The hierarchical model provides many semantic levels that facilitate understanding of video content. Arslan in [6] proposes a video model in a hierarchy of events, sub-events, and objects of interest. Video consists of events and events consist of sub-events in his work. Objects are modeled at every level in semantic hierarchy.

As mentioned before, video does not only contain visual image frame sequences, but it also contains audio track accompanying frame sequences, speech track, and appearing texts. Considering all information sources in the video, Snoek and Worring [31] define *Multimodality* as expressing content of video using at least two information modalities. The association of visual features with other multimedia features, such as text, speech, and audio, provides another fruitful content.

There are visual, auditory, and textual modalities in video. *Visual modality* contains everything that can be seen in the video document. Visual information provides perceptual properties like color, texture, shape and spatial relationships; semantic properties like objects, roles, and events; impressions, emotions, and meaning with the combination of perceptual features. *Audio modality* contains everything that can be heard in the video document such as speech, music, and environmental sounds. The auditory modality can provide valuable information when analyzing video programs [7, 20, 22,23, 30]. Audio track can be divided into speech, music, sound effect, and background tracks [21]. *Textual modality* contains everything that can be converted into text document in the video document. There are mainly two textual sources: *visible texts* and *speech transcripts*. Visible texts are superimposed text on the screen such as closed captions or natural parts of scenes such as logos, billboard texts, writings on human clothes, etc. Another text source is *speech* that can be transcribed into text [16,19].

A successful skimming approach involves using information from multiple sources, including sound, speech, transcript, and video image analysis. The Informedia [16] project is a good example of multimodal approach, which automatically skims documentary and news videos with textual transcriptions by first abstracting the text using classical text skimming techniques and then looking for the corresponding parts in the video [30]. This method creates a skim video, which represents a short synopsis of the original. The goal was to integrate language and image understanding techniques for video skimming by extracting significant information, such as specific objects, audio keywords, and relevant video structure.

Another multimodal approach to video content can be seen in Mihajlovic and Petkovic [19]. They use multi-modal clues, obtained from three different multimedia components: audio, video, and superimposed text. The audio and video feature extraction subsystems are developed to extract important parameters from multimedia documents. They also performed text detection and recognition to extract some semantic information superimposed on the videos. Petkovic and Jonker [25] model audio primitives beside visual primitives, they define audio events, and compound audio-visual events in their event grammars.

Multimodality is used in automatic segmentation of video that is the base step in video modeling. Video parsing techniques using only visual features tend to segment the clip into too many pieces. By including audio analysis, those segmented shots can be put into bigger semantic units [23,26, 28, 40]. Zhu and Zhou [40] divide video into scenes using audio-visual information and speech transcriptions. Adams et. al. [2] use audio, visual, and text cues for semantic indexing of multimedia. Nam and Tewfik [23] use audio and visual features for determining scenes in violent films. Smith et. al. [30] align speech transcriptions with visual key-frames.

1.1 Contributions of the Thesis

Contributions of this thesis work can shortly be stated as follows:

- The main contribution of this thesis is modeling video content using *auditory, visual, and textual* information altogether. In our model, video content can be accessed with visual events, visual objects, auditory events, auditory objects, visible texts, or speech transcripts combinations. Most of the previous studies model only visual information via visual events and visual objects [1, 9, 17, 24]. Petkovic et. al. [25] use auditory information beside visual information via auditory events. In Informedia project [16], textual information is used beside visual information via speech transcripts and closed captions. There is no video database model considers all modality semantically at the same time, our proposed model combines all data sources and spatio-temporal dimensions in one model. We model appearing events, appearing objects and objects' locations under visual model; audible events and audible objects under auditory model; visible texts under visual model; and speech transcription text under auditory model. We get multimodal video model by combining auditory and visual model.

- Another contribution is modeling visual information, auditory information and speech content *in a semantic hierarchy*. Visual events are modeled in a hierarchy in previous works [3, 6] and visual, auditory, and textual sources are considered in physical hierarchies in [14, 25] but auditory events and speech texts have not been modeled in semantic hierarchies. In [16], spoken documents are grouped under headlines but this study considers only news and documentaries. We propose visual hierarchies, auditory hierarchies, and textual hierarchies semantically. In visual hierarchy structure, visual content is segmented into *sequences* and *scenes* semantic groups by watching video. In auditory hierarchy structure, auditory content is segmented into *background* and *foreground* semantic intervals by listening video. Long speech given one speaker is divided into subgroups according to its subjects in auditory model. The hierarchical representation of visual, auditory and textual information provides both bird's-eye view and detailed view to video content.

- Based on our proposed model, we introduce *multimodal semantic queries* that let to query combinations using visual content, auditory content, and text content. With multimodal semantic queries, users can get almost exact results to their queries. Queries considering only one modality bring more and redundant results. By considering all sources, redundant results are minimized and users can access desired information more quickly. Previous studies worked on either visual semantic queries [1,9,24], audiovisual queries [25], or visual-text queries[16]. To our best knowledge, no one has developed queries considering all data sources' semantics altogether. In our querying structure, visual events, visual objects, and text keywords are visual clues, and they are searched in visual model for getting results. Auditory events, auditory objects, speaker, and speech keywords are auditory clues, and they are searched in auditory model for getting results. If a given query contains both visual and auditory clues, both models are searched separately and their results are combined. We also support *semantic conjunctive queries* containing multiple query sentences. We process each query separately and then combine their results.

- We support *hierarchical queries* that are usable in querying the visual clues at sequence or scene levels and in querying the audio clues at background or foreground levels. In the previous studies using hierarchical structures [3,6,14,25], queries considering hierarchies have not been described properly, we described all algorithms. We also introduce auditory semantic querying.

- We support *temporal queries* that enable to query the temporal relationships between events that can be visual events or auditory events. In [38], temporal queries are implemented considering only visual events. In addition to visual events, we also consider auditory events and visual-auditory events interactions. We use intersection, union, and concatenation operators for retrieving time intervals. We also support *conjunctive temporal queries* containing more than one temporal relation.

Besides contributions, we have implemented previous works' queries [17, 38] with new user interfaces. We support *fuzzy spatial queries* that enable querying the spatial relationships between objects and *fuzzy regional queries* that enable querying the object locations and object trajectories. In computing fuzzy values of the spatial relations and the regional relations, we use fuzzy membership functions that are similar to the ones in Köprülü's study [17]. We have implemented *conjunctive spatial queries* that contain more than one spatial relation. More that conjunctive spatial querying is one of the future works in [17]. We have developed compact Multimodal Video Manager tool in Java. This tool provides annotation and retrieval of video content together.

1.2 Organization of Thesis

The remainder of the thesis is organized as follows. Chapter 2 describes different video components of video data in detail. Related studies on visual modality, auditory modality, textual modality, and multimodality are examined. In Chapter 3, the modeling and querying of video on different aspects is discussed with past studies. In Chapter 4, multimodal video indexing and retrieval model that we propose is introduced. Our model and query processing algorithms are described. In Chapter 5, implemented software tool is described. The video annotation part working procedures, supported query types, and outputs of the queries are showed. Conclusion of our work and future research directions are given in Chapter 6.

CHAPTER 2

VIDEO COMPONENTS

Video is basically a sequence of images relayed at a constant speed, normally 25 to 30 frames per second, with a synchronized audio track. Although visual content is a major source of information in a video, other media components carry valuable information, such as text (overlayed on images, spoken text document), and audio. The content of a video is extracted by combining visual, auditory, and textual modalities. A combined and cooperative of these components would be more effective in characterizing video program for consumer and professional applications. These three video components and different combinations of these modalities are explained in detail in sub sections.

2.1 Visual Modality

Visual modality covers everything that can be seen in the video. The visual data can be acquired as a stream of frames at some lines of resolution per frame and an associated frame rate. The elementary units are the single image frames. Consecutive video frames give a sense of motion in the scene. Visual perception is elementary information while watching video. Visual information provides perceptual properties like color, texture, shape and spatial relationships; semantic properties like objects, roles, and events; impressions, emotions, and meaning with the combination of perceptual features. Visual information includes both 2-D space and time.

Visual object acts as a source of visual data. There are a lot of visual objects, but salient objects are more considerable for viewer. Salient object can be a tiger in a forest, tiger has leading role and more interesting than individual trees or bushes. Visual events are consecutive frame groups, which give semantic meaning. For example in “tiger running” can be visual event. Different events may be extracted from digitized video, either in 2-D space or in 3-D with 2-D space and time. For example, a static logo in a video scene may be considered as a 2-D event, while a moving object may be considered as a 3-D event.

2.2 Auditory Modality

Auditory modality covers everything that can be heard in the video. Audio refers to generic sound signals, which include speech, dialog, sound effects, music and so on. The audio information often reflects directly what is happening in the scenes and distinguishes the actions. Audio information often includes information related to the story. Speech is related to the story and you cannot understand the content well without listening to it. Music changes atmosphere of scene and physiological viewpoint of audience, horrible films do not scare people without sudden sound increases.

The auditory modalities can provide valuable information when analyzing video programs. For example in the sport domain, loud sound bursts such as crowd cheer indicate the highlights of a sporting event. Audio cues such as silence, music, and volume can be used to assist with the video segmentation.

The sound track plays back in parallel with the playback of the video frames. The soundtrack may consist of a large number of individual sound tracks mixed together. Typical types of soundtracks include: speech, music, sound effects, live, mixed. Each of these individual sound tracks can be described using their own domain specific sound events and objects as Hunter et. al. [15]. Moriyama et. al.[21] divide audio component into four tracks, namely speech by actors, background sounds, effect sounds, and BGMs. Shot represents pictorial changes in visual modality. The BGM represents music superimposed on the video. Effect sounds superimposed on video and have no melody such as fight effect. Audio components, which don’t belong to effect sounds or BGMs, are grouped as the background sounds such as pub humming. Figure 2.1 is taken from Moriyama et. al. [21] and shows track structure of in a time interval. Each track spans time interval while related audio event is audible.

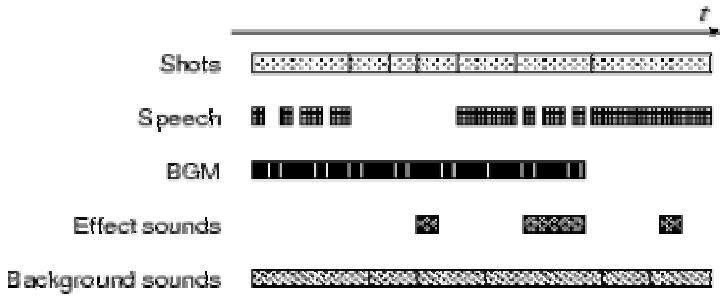


Figure 2.1 Track structure of the drama video [21]

Herrera et. al. [13] propose framework for audio descriptors and define sound object, sound events, and their distinctions. In their proposal, sound object is a sound source and any kind of behavior of that object is an event. *Audience* is a sound object of *applause* sound event. Events develop in time, so all events have a duration property. All events can be considered objects because they can be grasped though an operation of "reduced hearing", from a functional point of view it seems more convenient to separate sources from their behaviors.

For content based audio retrieval, the audio stream has to be segmented and indexed first. Audio scenes are collection of dominant sound sources. According to the study in [33], an audio-scene change is said to occur when the majority of the dominant sources in the sound change. A typical audio scene changes are: ambient music → street sounds → conversation → sounds in a bar.

In the study of Huang et. al. [14], audio events are some distinct acoustic characteristics in continuous audio segments. After audio stream partitioning, each event is described most unique features. Indices to these events can be created, and audio events can be retrieved using these indices. Some audio event examples are single-speaker speech, music, multi-speaker conference speech, or speech over background music.

Audio events are application dependent. Each application contains special audio characteristics. Morony [22] classifies audio events in different TV program categories. There are sport, sitcoms, news, talk shows, films, music programs, and so on. Some sound events are dominant in each program category. For example, talk shows have laughter, applause, and speech sound events; violent films have explosion, gunshot, scream, car crash, and fight sound events.

Beside audio events, Morony [22] surveys ambiances of scenes in TV programs. There are certain ambiences such as traffic noise in busy street scenes; conversation hum in pubs, and restaurants; animal sounds in farm scenes and so on. He also gives hints in labeling sound events. Some single events are slap, single gunshot, single mortar fired. Some events are labeled single event, but they comprise two or three percussive such as clinks, clicks. Several sound events repeating form in continuous time such as clock-tick, computer keyboard, machine-gun, footsteps, and so on. Several repeats of the same event are given a single label such as breath, kiss, meow, sea wave and so on.

The other sources of sound events are film sound clichés in web site [11]. In this page, sound events are categorized under sound objects. Some sound objects are animals, bicycles, bombs, cars, computers, environment, people, guns, and so on. Possible sound events are listed under each sound object. For example, dog barking, cat meowing, wolf howling sound events are listed under animal object; crying, screaming, kissing, foot stepping events are listed under human object.

Moncrieff et. al. [20] study the film audio structure. A rich, complex, and meaningful sound track in film is made of dialogue, music, background sounds and special effects, and expressive silence. High-level scenes require amalgamation of low-level events. The high level sound scene of car chase is characterized by the noise of engines revving, horns, sirens, the skidding of tires, car crashes, and the breaking of glass. Violent sound scenes are characterized by explosions, gun-fire, aural impact of people hitting each other.

2.3 Textual Modality

Text can be thought as a stream of characters. There are mainly two types of textual information in video: visible texts and transcribed speech texts. Texts play important role in illuminating the video content. Especially in news, in documentary videos and in distance learning videos, texts are heart of the video content. For broadcast news videos, text information may come in the format of caption text strings in video frames, as close caption, or transcripts. This textual information can be used high-level semantic content, such as news categorization and story searching. In documentary videos, speech is more dominant while clarifying the subject. In distance learning videos, all stuff can be converted to text from teacher speaking to board content.

In textual information retrieval of video area, the Informedia [16] project has a leading role. This project aims to automatically transcribe, segment, and index the linear video using speech recognition, image understanding, and natural language processing techniques. Video Optic Character Recognition (VOCR) techniques are used for extraction text from video frames and Automatic Speech Recognition (ASR) techniques are used for conversion of speech to text. Their system indexes news broadcasts and documentary programs by keywords that are extracted from speech and closed captions.

In spoken document retrieval systems, generated transcripts are fed into a classical (textual) IR system. Through the use of such transcripts, a number of spoken documents can be retrieved in response to a textual query. In Siegler et. al. [29], Spoken Document Retrieval (SDR) is done using keywords. Steps in SDR are:

1. Keywords are selected from the news article database.
2. A word sequence is produced by transcribing speech to text.
3. Keywords are extracted from the produced word sequence.
4. Spoken documents are retrieved using a group of keywords.

In the study of Mihajlovic et.al [19], visible texts are considered in Formula-1 races videos because of having rich text content. They perform text detection and recognition to extract some semantic information superimposed in the Formula 1 race video. They divide visible text into two classes:

1. *Scene text* appears in a natural part of the actual scene captured by the camera. Examples of scene texts are billboards, text on vehicles, writings on human clothes, etc.
2. *Graphic(superimposed) text* is mechanically added text on video frames in order to supplement the visual and auditory content. It usually brings much more useful information, since it represents additional information for better understanding of the video, and is closely related to video subject. The size and spatial position of the text in the video frame indicate its importance to the viewer. It frequently contains information that is closely correlated with the actual scene. For example, in sport videos, there are large amount of text over the images.

Beside visible text, speech transcription texts are used in retrieval of different video applications. There are various works in speech text retrieval, such as [5,7,8,29]. In the study of Ariki [5], the TV news speech is divided into speaker sections at first and then each speaker time interval is indexed with speaker's speech transcriptions. Figure 2.2 shows speakers' time intervals. Another speech segmentation technique is used in Coden et. al.[8]. In that study, the spoken document is segmented into 100 word chunks and time interval of each word is recorded for providing time aligned video browsing. For example, 'When does the announcer shout "goal" in soccer videos?' can be queried. Nearest visual event to the given keyword is shown to user in their works.

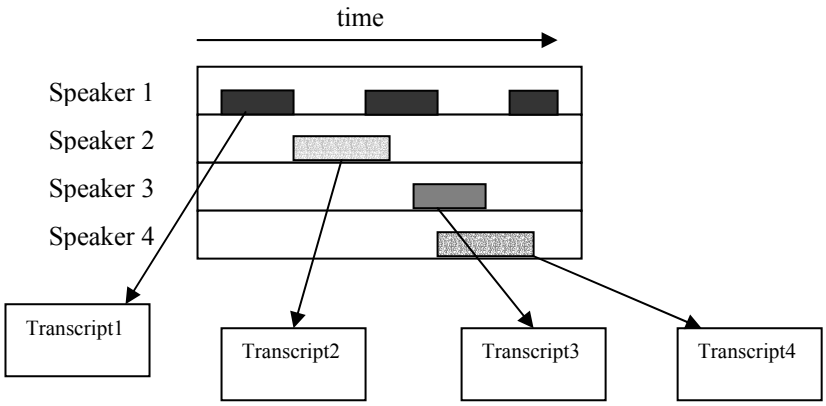


Figure 2.2 Speaker Indexing with transcripts

Another study on the speech retrieval is AudioLogger [7] that transforms audio content of a video into searchable text automatically. By listening to the audio track of a video stream intelligently, the AudioLogger identifies spoken words, speaker names and audio types, and eliminates manual annotation process.

2.4 Multimodality

Snoek and Worring in [31] define multimodality as using at least two information channels in expressing semantic of video. In video semantics: visual effects add descriptive information and expand the viewer's imagination; audio effects add level of meaning and provide sensual and emotional stimuli that increase the range, depth, and intensity of our experience far beyond what can be achieved through visual means alone; overlayed text provides descriptive information about the content and spoken document clarifies subject of video. Figure 2.3 shows combination of different modalities in video.

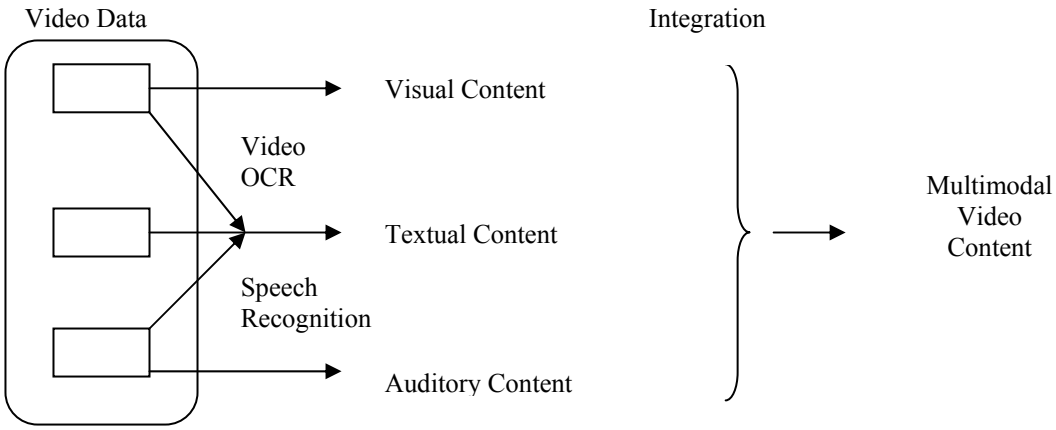


Figure 2.3 Multimodal Video Content

In the synchronization and alignment of the different modalities, all modalities must have a common timeline for providing that the time stamp is used typically. When audio is the main expertise, image frames are converted to (milli) seconds.

Not all of visual, auditory, and textual modalities are dominant in different program categories. Some of them have more density in some programs, and the others are scarce. Some program's modality characteristics are:

- ***Sport videos:*** The visual, auditory, and textual modalities are made to identify events. Some visible texts are the score, player, team information, play time, name of player that received a yellow card and so on. Salient events can be identified with auditory events such as cheering, whistling, speaker shouting, etc.

- ***Talk shows:*** Textual information is scarce in talk shows, visible texts are information about the performing artist and the name of the song on a fixed position. There are auditory events such as applause, singing, laughing, etc.

- ***Feature film, cartoon, sitcom, and soap*** share similar layout and content properties. Feature film, cartoon, sitcom, and soap differ with respect to people appearance, usage of special effects, presence of object and camera motion, and shot rhythm. Generally talking event is dominated. Films sound effects vary with kind of film. For example, violent films have gunshot, fighting, car breaking sound effects. Films contain overlayed text such as location, year information. In sitcoms, auditory events such as applauding, laughing show good jokes. Cartoons are almost free of speech and special effects. The soaps are the poorest in multimodal content. They are based on talking of people in one room.

- ***Documentaries*** can also be characterized by their slow rhythm. Other properties that are typical for this genre are the dominant presence of a voice over narrating about the content in long microphone shots. Textual information is dense in documentaries. Some textual sources are overlayed texts for explanation, graphics, or stressed words. Special effects are seldom used in documentaries.

- ***News*** has well-defined structure, which people talking in front of camera showing little motion. Different news reports and interviews are alternated by anchor persons introducing, and narrating about, the different news topics. News has rich textual information such as speaker speech transcripts, information giving texts, captions, headlines etc. Overlayed text is frequently used on fixed positions for annotation of people, objects, setting, and named events. Similarity of studio setting is also a prominent property of news broadcasts, as is the abrupt nature of transitions between sensor shots.

- ***Commercials*** have a great variety in setting, and share no common structure. Usually lots of object and camera motion, in combination with special effects, such as a loud volume, is used to attract the attention of the viewer. There are visible texts such as brand, cost, product features, etc. There is no station logo in commercials. Different music is used to separate commercials.

In most of the studies, multimodal clues are used in automatic scene detection or object detection. Especially audio and visual modality combination, one of the multimodality example, is used for automatic scene detection in various works. Nam and Tewfik in [23] integrate cues from both the visual and auditory modality symmetrically for characterizing and indexing violent scenes in general TV drama and movies. In [26], Pfeiffer et. al. identify scenes by firstly determining shots using audio features, color features, orientation features and faces appearing in the shots, then merge shots into scenes using all features. In [33], Sundaram et. al. determine audio and visual scene boundaries separately, then fuse the resulted segments with nearest neighbor algorithm.

In [40], Zhu and Zhou segment video into scenes using auditory modality, visual modality, and their interrelations. After segmenting video into scenes, they extract text from key frames using VOCR and speech transcriptions using ASR for classifying video scenes. They also use natural language understanding technique for automatically classify video scenes on the basis of the texts obtained from close caption, video OCR process and speech recognition. Their system is used to content based browsing over key frames and keywords.

Sareceno et. al.[28] use audiovisual information in speaker change detection. They segment audio signals into silence, speech, music, and miscellaneous sounds. Dialogues are detected based on the occurrence of speech and an alternated pattern of visual labels, indicating a change of speaker. The auditory and visual modalities are integrated to detect speech, silence, speaker identities, no face shot, face shot, and talking face shot using knowledge-based rules. At first, talking people are detected by finding faces in the camera shots subsequently a knowledge-based measure is evaluated based on the amount of speech in the shot.

In [30], Smith et. al. create the video skim using spoken text by appending each successive keyword in time dimension. Then important keywords are selected for partitioning spoken documents. Selected keywords are aligned to time intervals and image frame sequences. Figure 2.4 is taken from [30] and shows text alignment to image frames.

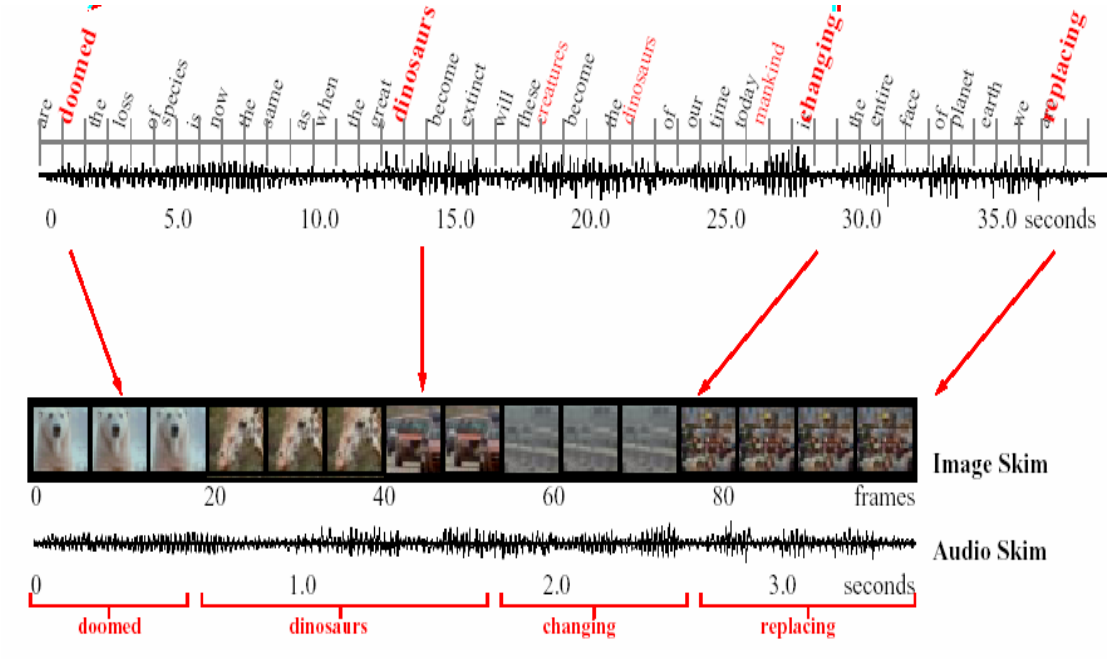


Figure 2.4 Skimming Video Text [30]

Adams and Iyengar in [2] use multiple information in modeling semantic concepts, objects, and events. They use single modalities separately and then fuse these modalities in finding multimodal concepts automatically. Their example is finding rocket launch concept using auditory, visual, and speech information. Visual concepts are rocket object, fire, smoke, sky, outdoor; audio concepts are rocket engine explosion, music, speech, noise, and visual concepts and audio concepts combination gives rocket-launch multi-modal concept. The audio concepts and the visual concepts are detected independently and the event Rocket-launch is a high-level concept that is inferred from the detected concepts in multiple modalities.

Woudstra et. al. [37] model audiovisual information in Admire model and apply it on soccer videos. They represent the different types of media in a uniform way. They characterize information at different levels such as frame, shot, coverage and support different types of relationship such as temporal, spatial and support variety of query mechanisms. They describe shot as continuous audiovisual material. Shots are extracted considering: video shot that a successive collection frames; audio track that shot's audio part; text title that written text on video. Sequence of semantic coherent shots is constituted a scene. They use audio, visual, and text feature in retrieving exciting moments in soccer video.

In the study of Srinivasan et. al. [32], video parsing extracts visual objects and events; audio analysis extracts audio objects, events. After analyzing video components, extracted entities are stored in databases. Shot boundaries of each event and object are represented with time intervals. Their system enables multi-mode queries. “Davidson’s play with loud cheer” is one example of multimode query. For retrieving this query results, each part is processed on related data structures independently, then time intervals of coming results are intersected.

CHAPTER 3

VIDEO MODELING AND RETRIEVAL

People want to browse and to get the multimedia content without considering the low-level features such as color, texture, volume, zero-crossing-rate etc. Video queries based on low-level features were used at the first age of multimedia database. This type of queries is not close to the human perception and human needs. Users want to query video content instead of the raw video data, consequently content-based video indexing and retrieval studies are increased.

For content-based video retrieval, modeling and querying techniques in traditional database require many changes. In modeling, video objects, events, and relationships between objects and events must be represented. Query types in video are also quite different from that in traditional database. Objects, events, objects' locations, events' sequences can be queried. Query processing algorithms and their efficiencies depend on video models. Exact matching of queries is not possible in content-based multimedia databases therefore the results of queries can be retrieved with fuzzy membership values.

3.1 Video Modeling

Content-based video databases require that the source material must be effectively indexed. The indexing approach followed in most projects consists of three consecutive phases: temporal segmentation, abstraction, and content extraction.

First, video must be segmented into short meaningful and manageable portions then these portions must be indexed individually. Segmentation can be done manually, automatically with scene detection algorithms, or semi-automatically with human assistance to automatic scene detection algorithms. In abstraction, indexing is applied on only key frames not all of the video frames. Indexing key frames takes much shorter time than indexing the original video. Content extraction is extracting semantically meaningful objects and events from video. Extracting and indexing semantic content of video sources are very important for efficient and effective video retrieval.

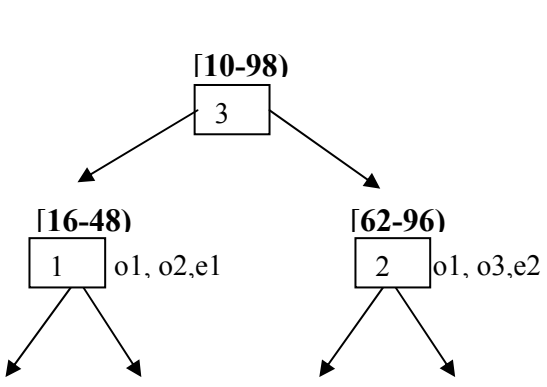
3.1.1 Semantic Modeling

Humans think in terms of events and objects and usually remember different events and objects after watching a video. Events and objects are called high-level concepts and are main components of the semantic video modeling. These high-level concepts are the most important cues in content-based retrieval. Extracting the semantic content is complex due to requiring domain knowledge and human interaction. The simplest way for modeling video content is free text manual annotation, in which video is first divided into segments and then every segment is described with free text.

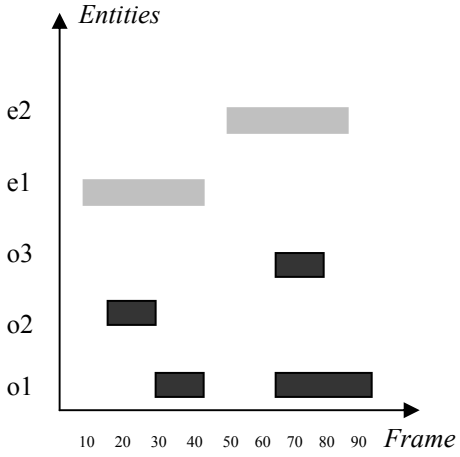
There can be thousands of video objects and events in a single video, but only small sub-set of them is useful and makes sense to watcher. “Hot object” is introduced in the study of Fan et.al [10] for reducing a number of objects to smaller number of interested objects. Video objects should be semantically consistent, representing one real world object, and subject of interest to users or applications. Each video objects in the video has a unique identifier to differentiate from others. Video objects are also described with spatial positions that are changing in time. Some examples of video objects are speaker, notebook, and studio in news video. Video objects can be extracted automatically in a known domain. But object extraction algorithms have been applied to few video applications hence accuracy results are still low. With pre-defined rules, objects can be extracted artificially as in Petkovic et.al [25] and Dönderler et.al. [9].

In OVID [24], video objects correspond to set of video frame sequences. Each video object has a set of attributes and a unique identifier. Video unit can be inherited from another video unit, thus sharing information among video units increase speed of query processing. OVID' s video model does not explicitly support modeling of the video document structure. OVID provides the user with the SQL-based query language VideoSQL that gives the user the ability to retrieve video objects by specifying some attribute values. VideoSQL does not contain language expressions for specifying temporal relations between video objects.

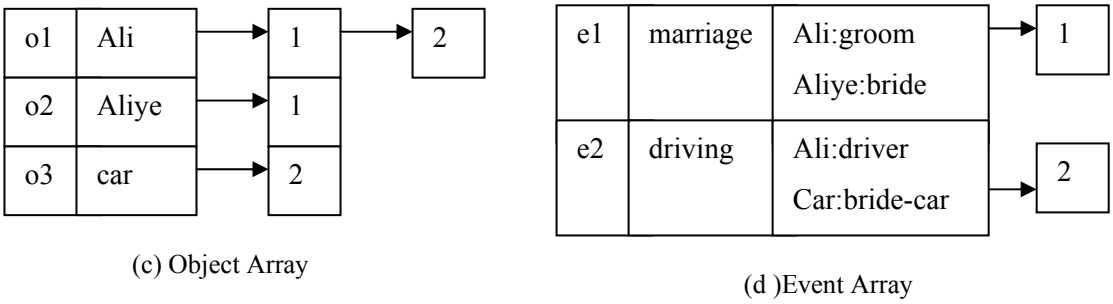
Adali et. al. [1] describe objects, events, activities as event types, roles and players in Advanced Video Information System(AVIS). Objects are real-word entities such as house, car, etc. Activities are type of action such as marriage. Events are instances of activities and consisting of an activity type, roles in the activity, and objects as the actors of roles in the activity. One event example can be "Ali and Aliye get married". Role is object's task in the event such as Ali is groom and Aliye is bride. Players take as input an event and its activity type, as output returns a mapping from the roles of the activity to the entities in the video and to strings. A frame sequence is the set of contiguous frames containing a semantically important data, like an object or an event. An association map specifies which objects or events occur in which video frame sequences. Indexing in AVIS is mainly based on a segment tree, named as frame-segment tree (FST). FST is a binary tree whose nodes are frame intervals. All entities stored in the database are loaded into arrays. Object-Array allows accessing to video segments with object keywords; Activity-Array allows accessing to video segments with activity keywords; Event-Array allows accessing to video segments with event keywords. Figure 3.1 shows FST, Event-Array, Object Array, and association maps.



(a) Frame Segment Tree



(b) Association Map



(c) Object Array

(d)Event Array

Figure 3.1 AVIS Data Structures

3.1.2 Hierarchical Modeling

In hierarchical video structure, complex video units are divided into elementary units recursively. The most often proposed hierarchies have a segment-scene-shot-frame structure. Hierarchical video database provides automatic mapping from high-level concepts to low level features.

Video stream consists of frames, shots, scenes and sequences. Frames are single pictures and the elementary video units. There are 14-25 frames per second, so frame sequences give more meaning than individual frame. Physically related frame sequences generate video **shots.**

Shots are segmented based on low level features and shot boundary algorithms can detect shots automatically. Shots are not sufficient for content-basd video browsing and retrieval because there are too many shots in a long video and shots do not capture the semantic structure of video. Therefore, semantically related and temporally adjoining shots are grouped into **scenes**.

Scenes are segmented on the high-level features logically. The scene boundary detection is more difficult than shot boundary detection. The scenes are usable in the content-based video indexing and retrieval due to their semantic structures. Scenes may be still small for browsing very long video. It might be necessary to combine related scenes into **sequences** or acts. Sequence extraction is also difficult and needs human assistance. Figure 3.2 shows the hierarchical structure of video.

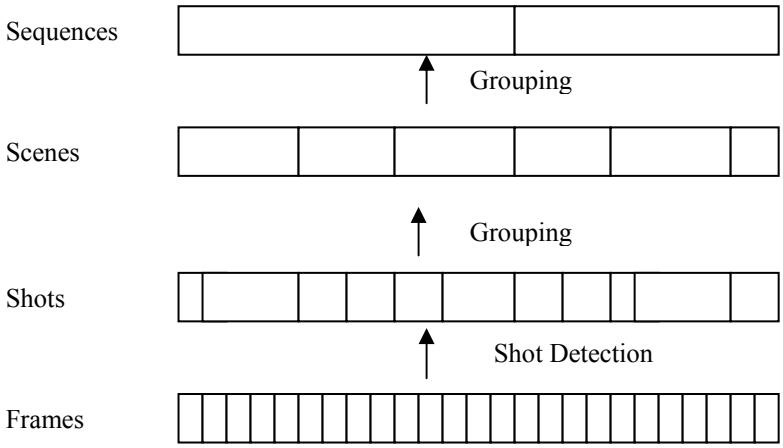


Figure 3.2 A hierarchical representation of video

Tennis video can be given as an example for this structure. Shots are camera motions; scenes are “vole”, “service”, “ace”; and combination of these semantic scenes gives “game” sequence.

In the study of Fan et. al. [10], the hierarchical units are clusters, sub-clusters, sub-regions, shots or objects, frames or video object planes, and regions. Their multi-layer structure provides automatic mapping from features to concepts through a learning-based clustering technique. Their multi-level model also provides a scalable method for retrieving and viewing video contents in database.

In semantically oriented approaches, video elements are organized hierarchically as nodes in a tree. This approach provides user to explore nested relationships between nodes. Li and Özsu [18] use tree structure for video units in Common Video Object Tree model. In [25], Petkovic and Jonker model video data in four layers: the raw data layer that consists of sequence of frames and keeps video attributes; the feature layer that keeps automatically extracted domain-independent features such as shapes, textures, spatio-temporal relations; the object layer consists of entities assigned one or more regions; and the event layer that consists of spatio-temporal combinations of objects. It supports automatic definition of high levels based on low-level features.

As Aigus et.al [3], actions are sub-events that split the event into a number of constituent segments. An action is therefore a more specific description of a part of the event. For example, consider a full-motion video segment depicting a wedding: the corresponding event would be 'wedding,' whereas individual actions would be those such as 'registrar advice' and 'groom kisses bride.' Events frequently consist of many actions and thus often determine the context for their constituent actions. Actions are therefore of shorter duration than the event they belong to, and the duration of an event is thus the union of the constituent actions of that event. Without this semantic aspect, a semantic content-based model does not have the 'full picture' of what is taking place in the media stream. Figure 3.3 shows event and involving actions in each event. Event X involves Action A, Action B, and Action C. Event Y takes after event X and involves Action A, Action B, and Action C.

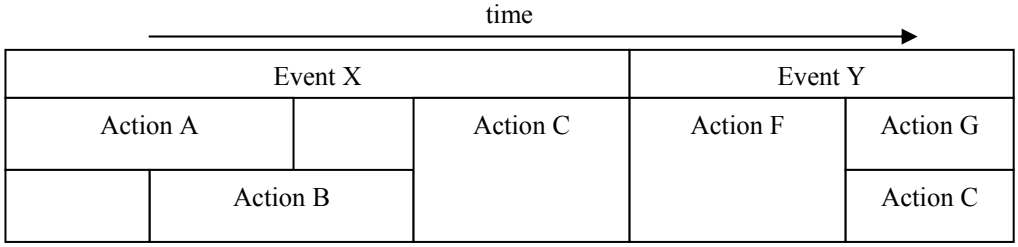


Figure 3.3 Event-Action Hierarchies

Another study that uses semantic hierarchies can be seen in Arslan [6]. Video is modeled in events, sub-events, and object hierarchy. Video semantically consists of events and events consist of sub-events. In each level of the hierarchy objects are extracted. Video is segmented into events and video objects. Events have sub-events and event objects; sub-events have only event objects. Figure 3.4 shows their semantic hierarchy. In hierarchical model retrieval system, query is searched on sequences and children of sequences.

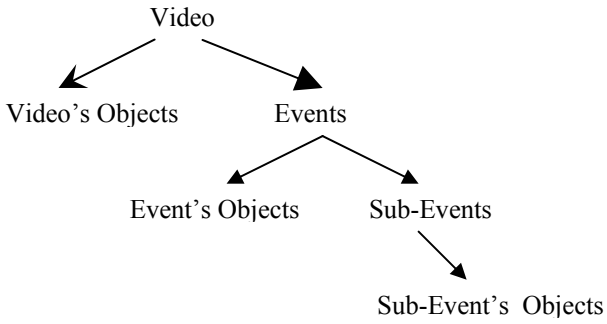


Figure 3.4 Semantic Hierarchy

3.1.3 Spatio-Temporal Modeling

In the semantic content description, high level entities have space and/or time domain. Events span time interval so they are related temporally with other events; objects have spatial features in frames so they are related spatially with other objects; object's position can change in time so object have also time domain.

3.1.3.1 Temporality

Video and audio data types have time dimension. Video elements can be identified by time attributes that are start time, end time, and duration. Video elements span different amounts of time in video. Actions and events take time. Events are extended in time, different events and actions may overlap in time and interact. Temporal relations between time intervals are determined by comparing the start and end points of time intervals. For example, *"a occurs before b"* or *"c happens while d is happening"* are stated by comparing of the start and end points of a, b, c, d events. In [4], Allen defined temporal algebra shown in Table 3.1.

Table 3.1 Temporal Interval Relations between two intervals

Relation	Symbol	Inverse	Meaning
B before C	*b*	*bi*	*BBBB CCCC*
B meets C	*M*	*mi*	*BBBBCCCC*
B overlaps C	*o*	*oi*	*BBBB* *CCCC*
B during C	*d*	*di*	*BBB* *CCCCCCCC*
B starts C	*s*	*si*	*BBBB* *CCCCCCCC*
B finishes C	*f*	*fi*	***BBBB*** *CCCCCCCC*
B equal C	*e*	*e*	*BBBB* *CCCC*

Without temporal relationships, the representation of events becomes unstructured and leads to ambiguity within the model. If we take *'a fight taking place at a party',* we can model two events, party and fight, 'taking place' is just a temporal relation. Without temporal relations, we should define one event, 'fight during party', by omitting that this is in fact two events, 'fight' and 'party,' occurring simultaneously.

While temporality of video is querying, several interval operators are used such as interval union, interval intersection, and interval concatenation in order to compute video intervals as query results. Union, extended union, and intersection operators are defined in Pradhan [27].

- ***Intersection***: This operation takes two intervals and produces intersected interval if max(start(I1), start(I2)) ≤ min(end(I1), end(I2)).

  I1_I2 = I[fs, fe] where fs = max(start(I1), start(I2)) and fe = min(end(I1), end(I2)).
  An example of intersection operation: I1[10, 20] and I2[15, 40] then I1_I2 [15, 20].
  However, for I1[10, 20] and I2[25, 40], I1 _I2 does not produce any interval.

- ***Union:*** This operation takes two intervals containing overlapping or adjacent intervals and produces a single non-overlapping contiguous interval. Union operation works as taking minimum starts of intervals and maximum ends of intervals.

  I1_I2 = I[fs, fe] where fs = min(start(I1), start(I2)) and fe = max(end(I1), end(I2)).
  An example of union operation: I1[10, 20] and I2[15, 40] then I1_I2 [10, 40].

- ***Extended Union:*** This operation can be considered interval concatenation. This operation takes two intervals as input and produces a single contiguous interval. The resulting interval will be contiguous no matter that the input intervals are adjacent, overlapping, non-overlapping and nonadjacent.

  I1_I2 = I[fs, fe] where fs = min(start(I1), start(I2)) and fe = max(end(I1), end(I2)).
  An example of extended union operation 11[10, 20] and I2[45, 60] then I1_I2 [10, 60].

3.1.3.2 Spatiality

Spatial property of an object is the information about the spatial positions of that object on the video frames. In [17], Köprülü et. al. use two-dimensional coordinates to store spatial properties of objects, which are represented by the user-specified rectangles. The spatial property of an object **A** is a tuple (R, I), where, R is a rectangular area which covers all area in which object **A** appears during the time interval I=[ti, tf]. So, R is obviously not the minimum-bounding rectangle (MBR) of **A**. A Minimum Bounding Rectangle is an imaginary rectangle that is accepted to be the minimum rectangle that covers all parts of an object. At any time t in [ti, tf], object **A** may be located at anywhere in R as Köprülü [17]. Since the spatial properties of objects are represented by dynamic rectangles, the spatial relationships between objects are also computed dynamically.

Spatial relationships between two objects can be classified into three categories: topological relations that describe neighborhood, shown in Figure 3.5; directional relations that describe order in space, shown in Table 3.2 with Allen temporal relationships; distance relations that describe space range between objects. In distance relation both the Euclidian distance between two objects and qualitative relations such as *near, far, close* can be used. Li et. al. [18] extended Allen's temporal interval algebra into two-dimensional space in order to define spatial relationships between rectangular areas. They defined two groups of spatial relations: topological and directional. Topological relations are equal, inside, contain, cover, covered by, overlap, touch, and disjoint. Directional relations are *south, north, west, east, northwest*, *northeast, southwest, southeast, left, right, below, and above*.

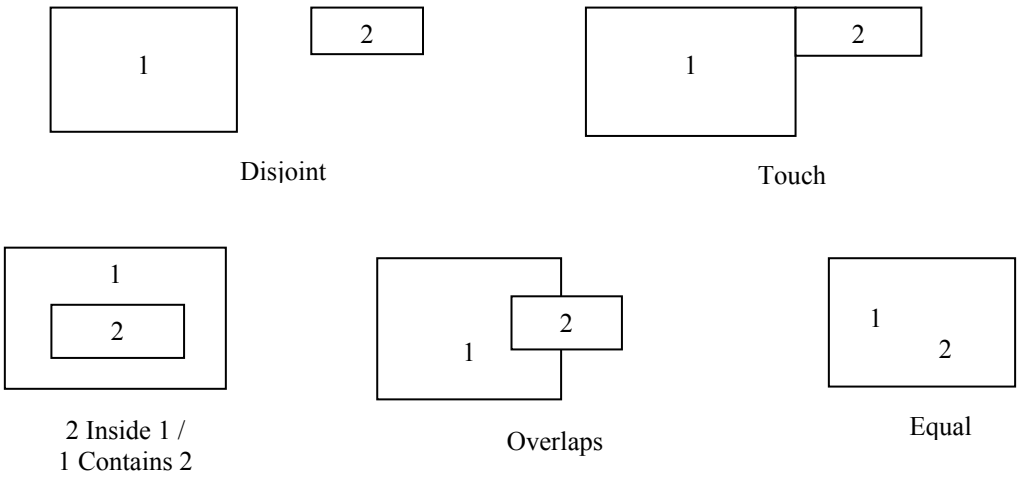


Figure 3.5 Topological Relationships

In [17], Köprülü et. al. use notations *left*, *right*, *top*, and *bottom* instead of *west*, *east*, *north*, and *south*, respectively. This study has covered the following relations between objects: *top*, *bottom*, *right*, *left*, *top-right*, *top-left*, *bottom-right*, *bottom-left*, *overlaps*, *equal*, *inside*, *contain*, *touch*, and *disjoint*. The definitions of these spatial relations, which are constructed by using the study of Li. et. al.[18], can be seen in Table 3.2.

Table 3.2 Spatial Relations between two objects

Relation	Definition
A BOTTOM B	A_x {b, bi, m, mi, o, oi, d, di, s, si, f, fi, e} $B_x \wedge A_y$ {b, m} B_y
A TOP B	A_x {b, bi, m, mi, o, oi, d, di, s, si, f, fi, e} $B_x \wedge A_y$ {bi, mi} B_y
A LEFT B	A_x {b, m} $B_x \wedge A_y$ { b, bi, m, mi, o, oi, d, di, s, si, f, fi, e} B_y
A RIGHT B	A_x {bi, mi} $B_x \wedge A_y$ { b, bi, m, mi, o, oi, d, di, s, si, f, fi, e} B_y
A TOP-LEFT B	(A_x {b,m} $B_x \wedge A_y$ {bi, mi, oi} B_y) $\vee$ (A_x {o} $B_x \wedge A_y$ {bi,mi} B_y)
A TOP-RIGHT B	(A_x {bi, mi} $B_x \wedge A_y$ {bi, mi, oi} B_y) $\vee$ (A_x {oi} $B_x \wedge A_y$ { bi,mi }B_y)
A BOTTOM-LEFT B	(A_x {b,m} $B_x \wedge A_y$ {b,m,o} B_y) $\vee$ (A_x {o} $B_x \wedge A_y$ {b,m} B_y)
A BOTTOM-RIGHT B	(A_x {b,m} $B_x \wedge A_y$ {b,m,o} B_y) $\vee$ (A_x {oi} $B_x \wedge A_y$ {b,m} B_y)
A OVERLAPS B	A_x {d, di, s, si, f, fi, o, oi, e} $B_x \wedge A_y$ { d, di, s, si, f, fi, o, oi, e } B_y
A EQUAL B	A_x {e} $B_x \wedge A_y$ {e} B_y
A INSIDE B	A_x {d} $B_x \wedge A_y$ {d} B_y
A CONTAIN B	A_x {di} $B_x \wedge A_y$ {di} B_y
A TOUCH B	(A_x {m, mi} $B_x \wedge A_y$ {d, di, s, si, f, fi, o, oi, m, mi, e} B_y) $\vee$ (A_x {d, di, s, si, f, fi, o, oi, m, mi, e} $B_x \wedge A_y$ {m, mi } B_y)
A DISJOINT B	A_x {b, bi} $B_x \vee A_y$ {b, bi} B_y

The spatial relationships between objects may not be strictly defined. A certainty level (in fuzzy terms membership value) can be given to the relationships between objects. In [17], Köprülü et. al. does also consider the membership value of any relationship (μ). If rectangular regions representing spatial locations of two objects don't satisfy the conditions stated in Table 3.2, the membership value μ is assumed to be 0. Otherwise, the membership value of the relationship is calculated using the centers of rectangles. The angle between the x-axis and the line between centers of the rectangles is used to calculate the membership value. Table 3.3 gives the relation between the angle and the membership value for each relation. Only relations, *top, left, top-left*, and *top-right* are represented in the table since these relations are inverses of *bottom*, *right*, *bottom-right*, and *bottom-left* relations respectively. As an example, TOP(A, B) is equal to BOTTOM(B, A).

Table 3.3 Relation between the membership value and angle between the centers of rectangles

Relation	Angle	Membership Value
Top	arctan(x/y)	1- (angle/90)
Left	arctan(y/x)	angle/90
Top-Left	arctan(x/y)	1 – (abs(angle-45) / 45)
Top-Right	arctan(y/x)	1 – ((angle – 45) / 45)

In Köprülü's work, the spatio-temporal model is constituted on AVIS model. Object array structure of AVIS is changed in that study. Objects are stored in interval/region pairs in object array. The AVIS's FST node structure is also changed as the tree node containing object/region pairs. Because of object movements during tree node frame interval, each interval is divided into sub-intervals that keep the region of the changing objects' positions. Each sub-interval is represented with a separate tree node. The association map is restructured as an object represented in interval/region pairs along the time interval.

3.13.3 Spatio-Temporality

In spatio-temporal objects, spatial and temporal components of the object cooperate with each other. In [18], Li and Özsu describe moving object that changes position in over time. Figure 3.6 shows spatio-temporal region of an object. Object position is changed with time and each considerable change is shown with reference region.

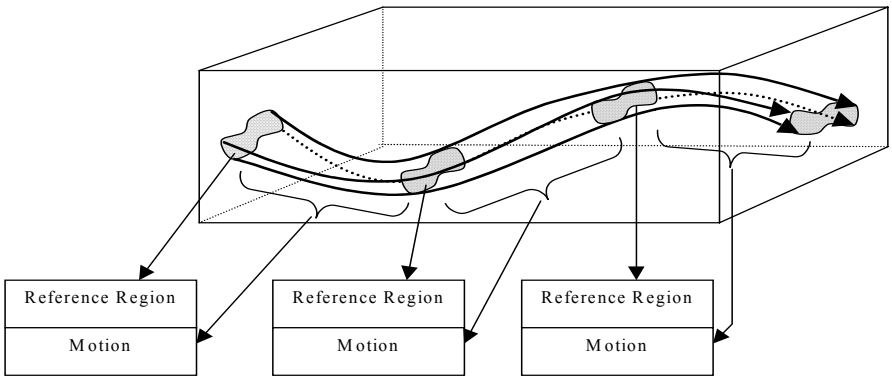


Figure 3.6 A spatio-temporal region

The trajectory is the route of an object over a period of time. Object trajectories are needed for many useful video annotation tasks such as describing activities at city street intersections, sporting events, traffic movements, group of animals, meteorological objects. In Köprülü's work, trajectory is a set of sorted (interval,region) pairs and any region is a neighbor of its predecessor. R1(r1x1,r1y1,r1x2,r1y2) and R2(r2x1,r2y1,r2x2,r2y2) are neighbor if they satisfy at least one of the following conditions:

- $r_1x_1 = r_2x_2 \wedge [r_1y_1, r_1y_2]$ and $[r_2y_1, r_2y_2]$ overlaps,
- $r_1x_2 = r_2x_1 \wedge [r_1y_1, r_1y_2]$ and $[r_2y_1, r_2y_2]$ overlaps,
- $r_1y_1 = r_2y_2 \wedge [r_1x_1, r_1x_2]$ and $[r_2x_1, r_2x_2]$ overlaps, and
- $r_1y_2 = r_2y_1 \wedge [r_1x_1, r_1x_2]$ and $[r_2x_1, r_2x_2]$ overlaps.

where, overlaps is taken from Allen's temporal algebra.

3.1.4 Multimodal Video Indexing

There are various works that consider different video modalities in video indexing and browsing research area. Petkovic and Jonker [25] use audio events in addition to visual events. Audio events can be represented as *primitives* that keep one event; or *compound* that is a composition of visual events and other audio events. For example, *"Long whistle"* is a primitive audio event and *"Loud shouting after long whistle"* is a compound audio event and may show a goal event.

Another study can be seen in [14]. Huang et.al. combine visual, audio, and text media types in indexing and retrieval systems. They partition each media into smaller units based on physical events. These physical events can be indexed effectively. Their model is also one example of a hierarchical video model. A physical layer addresses the media events having time dimensions; a logical layer represents multimedia events intercross of different media, across the event hierarchy with each media, or a combination of both.

Physical event is continuous in time, specified by a single continuous time range (between a beginning and ending time), representing a coherent and meaningful event. A physical event may have counterparts across the various media types. Logical event is a collection of events that are semantically coherent. Logical event is not primitive.

Figure 3.7 shows their layered structure. Three timetables for each media type record time segments of media events. Speech of anchorperson can be audio physical event. They applied their system on news videos. Logical events are stories, news reports and physical events are speech, captions and so on.

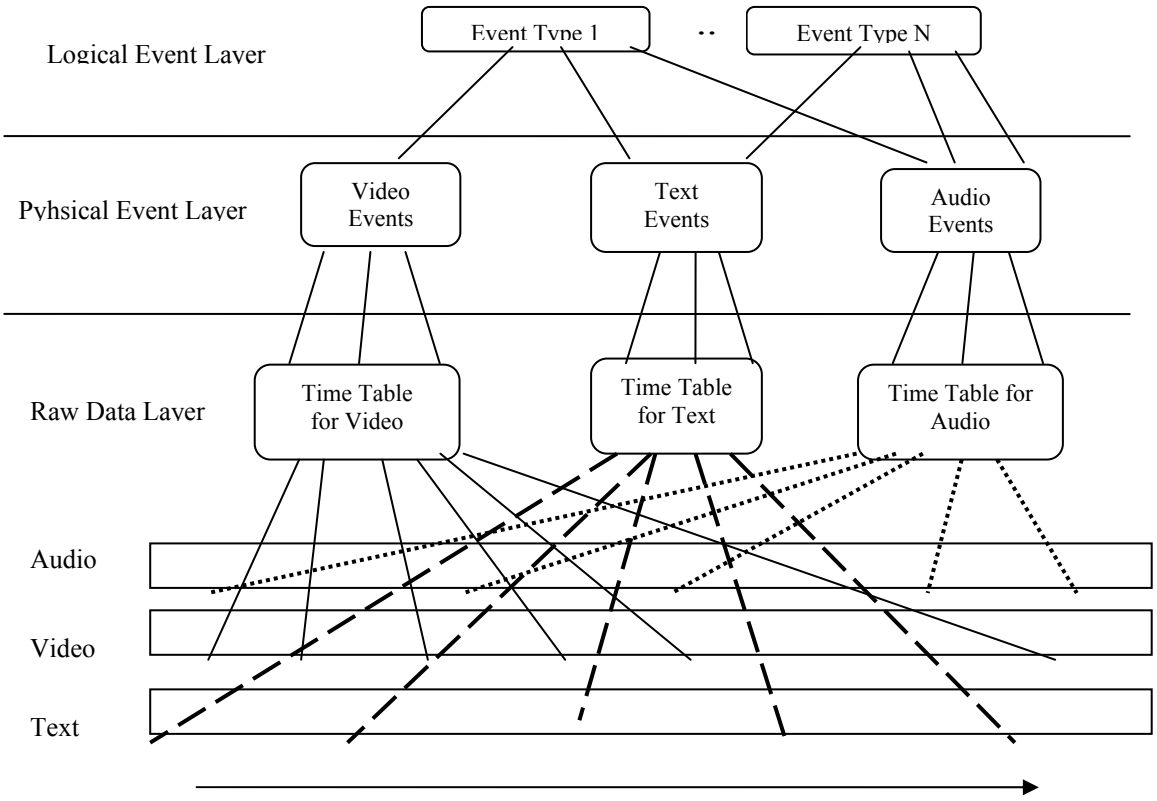


Figure 3.7 Multimodal Video Indexing

In [12], Hampapur uses audio and speech track besides visual information. In this study, speech transcripts are extracted in fixed time intervals, and visual content described in visual event time intervals.

3.2 Video Querying

Video content can be queried at different levels such as raw data level, semantic level, or combination of low and high levels. Users want to be able to retrieve content of video such as objects and events. Users also want to find a particular part of a video that is their interest. Query languages should support to retrieve information based on content.

Queries may be formulated using the standard query language or an extension of SQL such as VideoSQL in [24]. Symbols or icons may be use in the query interface. Visual query interface can list annotated objects and events in pull-down menus, and user can specify queries by browsing the pull-down menus. Some querying interface allows natural language inputs such as Informedia in [16]. In that study, speech recognition and language processing techniques are combined for receiving the result of user's query, which is spoken or typed. The identified terms are searched in terms attached to each clip's key frames. Visual queries, which are based on motion property, are done by sketching of object trajectory pad. System processes these queries and returns video time intervals having pads resemble to drawn pads.

Once a query results are retrieved, user will want to look into all results. Video clips can be shown in thumbnail of returned intervals' key frames. Listing key frames by their chronological order can show to the user what kind of scenes appear in the results. Looking into too many results in key frame structures can be difficult, listing key frames in hierarchical way makes browsing easier.

The SWIM [39] uses hierarchical representation in way selected key frames are indicating time displayed at the top of the screen, the selection of one of these key frames triggers appearing the below set of key frames. The VideoQ in [34], the user can browse the video shots or search video by text. The video shots are catalogued into a subject taxonomy for example sports-> water, sports-> surfing, which the user can easily navigate. Each video shot has also been manually annotated so the user can perform simple text search of keywords. Informedia [16] uses multiple types of information in video querying, and shows the query result as single screen displays player, key frame list, transcripts. Transcript and filmstrip show the current of the player being played, all synchronized together.

3.2.1 Semantic Queries

Users are interested in querying at the semantic level rather than at low level features. The simplest way to achieve basic content-based retrieval is to use keywords/free-text associated with video shots in the annotation process. Annotated video entities and particular shots can be queried in this type of querying.

The semantic queries proposed in AVIS [1] are as follows:

- Queries that take a specification and return a set of frame sequences,
    - *"Find all video frames in which car is seen."*
    - *"Find all video frames in which car accident happens."*
- Queries that take a set of frame sequences and return a set of objects,
    - *"Find all objects that are present in frame sequence [10,100)"*
- Queries that take a set of frame sequences and return a set of events,
    - "*Find all events that are present in frame sequence [40,200)"*
- Conjunctive queries are groups of basic queries,
    - "*Find all frames in which 'dog is chasing' and 'boy is running away'."*
- Compound queries are groups of query that one query output is input of other queries.
    - "*Find all peoples in events Ahmet and Ayse are seen in frame sequence [100,200)."*

3.2.2 Spatial Queries

Spatial queries are related with object spatial positions. Objects positions can be queried in three different ways: spatial relations between two objects can be queried; locations of object can be queried; and object trajectories can be queried. Köprülü et.al. [17] support regional queries, fuzzy spatio-temporal queries, and fuzzy trajectory queries.

- *Regional Queries*: Köprülü et. Al. define two regional queries:
    - Given an object and time interval, regions are asked for appearing object in time interval. An elementary query is *"Find all locations of ball appearing between 2 and 5 minutes"*.

- Given object and region, time intervals are asked. An elementary query is: *"Find the frame intervals in which the goal-post is seen in the rectangular region given by pixel values [20, 60, 100, 120] with a threshold value of 0.8".*

- *(Fuzzy) Spatial relationships queries:* Given two objects and the spatial relationship between them, time intervals satisfying this relationship with specified fuzziness value are asked for. An elementary query is: *"Find the frame intervals in which the plane is appearing in the left of tower with a fuzzy membership value of 0.8".*

- *Trajectory queries:* Trajectory shows the path of object movements. Köprülü et.al. [17] define trajectory queries as given an object, rectangular regions, and membership value, and then time intervals are asked. The degree of the neighbor is computed by 'overlaps' relation between regions. If the neighbor degree is above the given membership value, the region is added to trajectory regions. An example is:
  - "Find trajectories of the ball starting from the region-1 given by pixel values [40, 60, 60, 120] and ending at rectangular region-2 given by pixel values [190, 90, 220, 150] with a threshold value of 0.5."

- *Directional trajectory queries:* Yavuz et.al. [38] define directional trajectories as object trajectories with given directions. In her queries, there are given an object, rectangular region, two spatial relations, and a threshold value. One example is:
  - "Find directional trajectories of the player that start from the left of the rectangular region R given by pixel values [50,60, 50, 120] and end at the bottom of R with a threshold value of 0.7."

3.2.3 Temporal Queries

Temporal query adopts temporal information and events as inputs. Temporal information contains either a range of video frames or an ordering relationship between two intervals. Some temporal queries are: "Find all events occurring between 4^{th} and 12^{th} minutes.", "Find all events after the Mehmet's faul.", "Find video clips in which a scene with a kite flying appears after the one with a child running."

CHAPTER 4

MULTIMODAL VIDEO INDEXING AND QUERYING

In this study, semantic content of video is modeled considering auditory content, visual content and textual content. Based on proposed model, multimodal semantic queries, spatial queries, regional queries, temporal and spatio-temporal queries are handled.

Visual content consists of image frame sequences so it has space and time dimensions and can be segmented into either frame sequences or time sequences. However, auditory content has only time dimension and must be segmented into time sequences. For modeling auditory information and visual information in one model, both modalities must be segmented by using either time sequence or frame sequence. In AVIS model, video is segmented into frame sequences that are groups of following frames giving a semantic meaning. But we preferred segmenting video into time sequences for aligning both modalities.

Visual segments and auditory segments do not fit each other semantic interval boundaries. While visual semantics are changing, auditory content can give still the same meaning or vice versa. Auditory content contains speech documents whereas visual content contains readable texts. Because of all these differences, we modeled auditory content and visual content separately. But in video content querying, we used both visual model and auditory model.

4.1 Visual Model

Visual content of video is modeled by watching video without listening. Visual modeling consists of segmentation video into semantically coherent time intervals and extraction visual events, visual objects, and objects' locations for each interval.

4.1.1 Visual Segmentation

Visual content is segmented into time intervals manually by considering visual happenings. In our model, visual content is segmented into time intervals in two steps:

- Video is segmented into ***sequences*** which are big time intervals having own semantic integrity. This partition can be based on locations, times, events, concepts and so on. Some sequence labels are party, street, war, night, wedding, and so on. Sequence may involve more than one visual event and visual object. Sequences' time intervals do not overlap with each other.

- Each sequence can be partitioned into atomic sub-intervals considering that *each sub-interval must be related with one event*. These sub-intervals are called ***scenes***. Scenes are children of sequences and scene time interval is bounded with its parent sequence's time interval boundaries. Figure 4.1 shows an example of a sequence and multiple scenes. In wedding sequence, there are "signing the book", "groom kissing bride", "staff giving family book" scenes. Scene may involve visual objects having spatial features. Scenes' time intervals can overlap with other scenes' time intervals in the sequence.

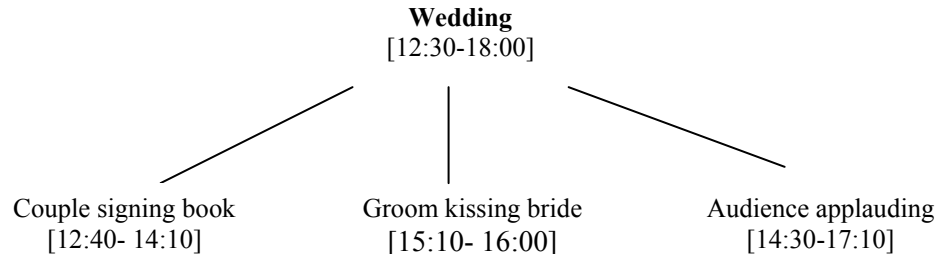


Figure 4.1 Sequence and Scenes

4.1.2 Visual Semantics

Visual semantic entities are visual events, visual objects, and their spatial-temporal relations. Visual semantics are extracted for each visual time interval and stored in related data structures. Each time interval is labeled with one visual event and zero or more visual objects.

4.1.2.1 Visual Events

Visual events are visible happenings that may involve visual objects. Visual event spans time interval. There is no necessity of each visual event having objects. Visual events are extracted at both sequence level and scene level.

We label a sequence meaning as a visual event, even sequence does not mean a happening. For example, street is a location, but we consider it is a visual event while labeling the sequence interval. We categorized sequence events in *"Visual-Sequence"* type. The objects of sequence event can be entered moreover all objects of its subsequences can be considered the objects of sequence event.

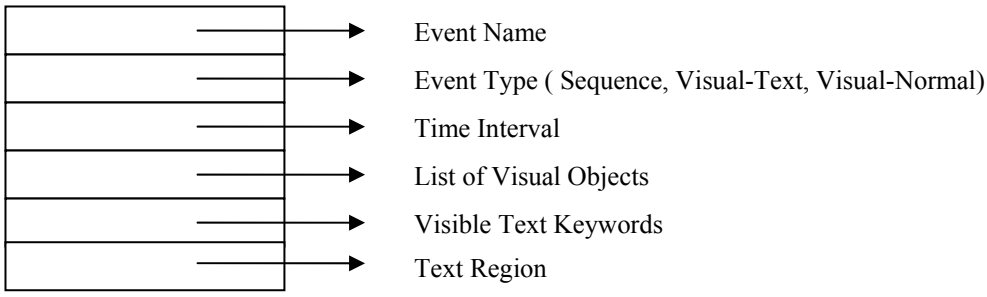


Figure 4.2 Visual Event Node

In scene level, there are text appearing events and normal events. For distinguishing normal events from text events, we give *"Visual-Normal"* or *"Visual-Text"* types for each visual event. Events in "Visual-Text" type do not contain objects, but contain text content and text's locations. In "Visual-Text" events, keywords of the text and the location of the text are kept. All visual events in Visual-Text type are named with "Text"+Number that number starts from 1 and is increased automatically for each new text.

Location of the text generally static in superimposed text (i.e. logo, caption), but can be moving in graphic texts (i.e. billboards, score-boards). Location of the text is taken with minimum bounding rectangles and kept in MovingRegion data structure shown in Figure 4.4. There can be more than one visible text in the video. They differ from each other with their content and their time intervals. All visible events except "text appearing" are in "Visual-Normal" type in scene level. Events in "Visual-Normal" type can contain visual objects.

Figure 4.2 shows visual event node structure. Event-Name, Event-Type, and Time-Interval must be filled, but other fields depend on event types. After extracting an event, we put it in visual event array whose elements are visual event nodes. Event names in the visual event array need not be unique, the same event can be seen with other objects in other times, each event must be stored with own time interval separately.

4.1.2.2 Visual Objects

Visual objects are real world enitities. A lot of objects are seen in video, but we consider only salient objects in our model. One visual object can be seen in different visual events and in different visual time intervals. Visual objects can be extracted at both sequence and scene levels. Figure 4.3 shows the node structure of a visual object. Beside having events and time intervals, visual objects have spatial properties. Visual objects' positions are extracted at scene level. The position of an object can change during time interval. While sequence intervals span long time, scene intervals span much shorter time. To handle changing object positions in sequence interval is impossible, so we prefer handling moving objects position at scene level. During a scene interval, the object position is taken in every considerable object movements.

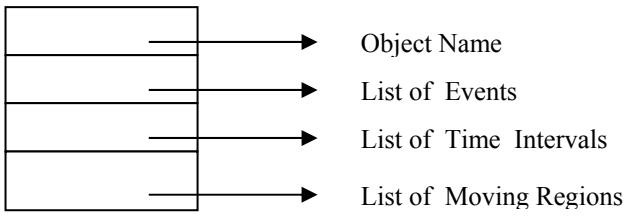


Figure 4.3 Visual Object Node Structure

We kept changing object positions and their taken times in MovingRegion data structure shown in Figure 4.4. The positions of the object in one frame are taken using minimum bounding rectangular that covers object's top-left point and bottom-right point. These two points are stored in Location data structure shown in Figure 4.5.

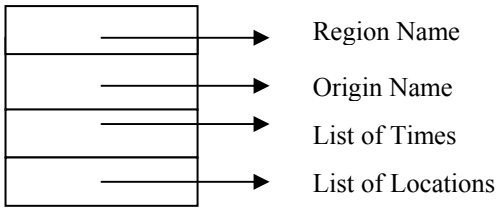


Figure 4.4 Moving Region Node

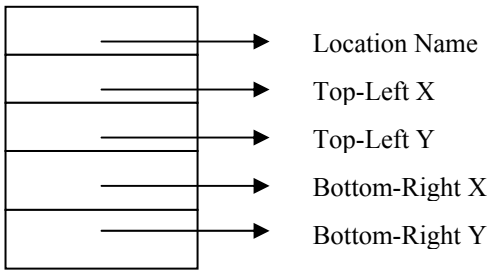


Figure 4.5 Location Node

After annotating the visual object, we put it into visual object array whose elements are visual object nodes. There is no duplicated visual object name in visual object array.

4.1.2.3 Visual Time Intervals

Visual content of a video is segmented into time intervals as described in Chapter 4.1.1. These intervals construct a multi-way tree structure whose elements are visual time interval nodes shown in Figure 4.6. Time interval's boundaries are between start time and end time. Each time interval is associated with one event, zero or more objects. Children keep visual time intervals for storing scene intervals under the sequence interval. If visual time interval is related with scene event, then there is no child. If visual time interval is sequence type, then there are children of this interval.

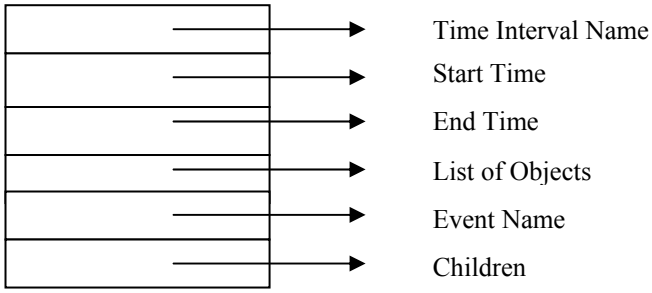


Figure 4.6 Visual Time Interval Node

4.1.2.4 An Example of Visual Content Model

In Figure 4.7, there is a video taking 12:30 minutes. There are two sequences: party and street. The "Party" sequence has "Text appearing", "Clara sitting", and "John eating" scenes; the "Street" sequence has "Text appearing" and "Clara Driving" scenes. We can see that scene time intervals are overlapping with each other in party and street sequence.

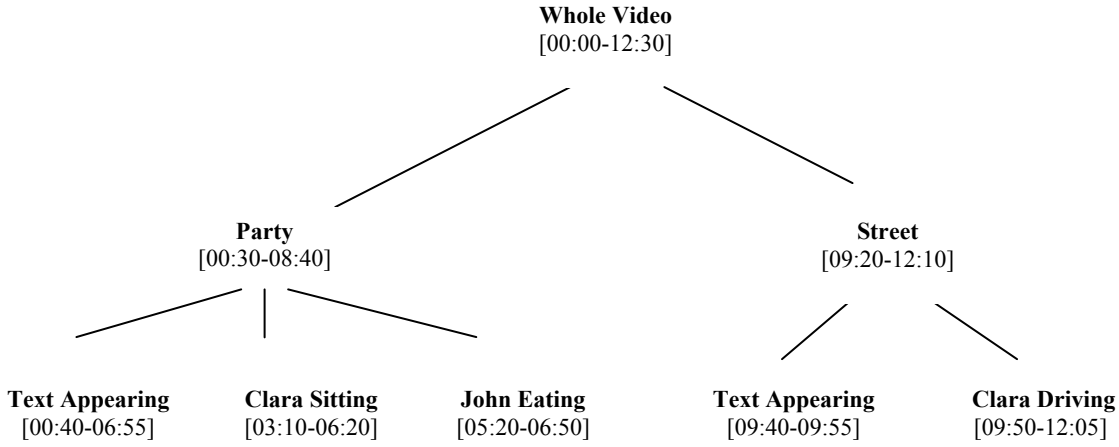


Figure 4.7 Sample Visual Segment Tree

Figure 4.8 shows time interval node structures in this tree. T0 is the root of the tree and has two children. T1 and T2 are sequences of the whole video and have children. The rest of nodes are scenes of the sequences and do not have children.

T0	T1	T2	T3	T4	T5	T6	T7
00:00	00:30	09:20	00:40	03:10	05:20	09:40	09:50
12:30	08:40	12:10	06:55	06:20	06:50	09:55	12:05
				Clara	John		Clara
	Party	Street	Text1	Sitting	Eating	Text2	Driving
T1,T2	T3,T4,T5	T6, T7					

Figure 4.8 Visual Time Intervals

Figure 4.9 shows visual objects, region and location structures. All moving region are not given, only R4 and first location of R4 are given. In our approach, Clara's positions in T4 time interval are kept in R4 region and Clara's position between 03:25-04:45 is kept in L1.

Clara	John
Sitting, driving	Eating
T4, T7	T5
R4, R7	R5

R4
03:25 04:45 05:25 05:50
L1, L2, L3, L4

L1
12
24
240
124

Figure 4.9 Visual Objects and their regions

Figure 4.10 shows visual events. Text1 and Text2 have text keywords and text moving regions.

Party	Street	Text1	Sitting	Eating	Text2	Driving
Seqence	Sequence	Text	Visual	Visual	Text	Visual
T1	T2	T3	T4	T5	T6	T7
			Clara	John		Clara
		Beer Tuborg			California 2000	
		R3			R6	

Figure 4.10 Visual Event Nodes

4.2 Auditory Model

Auditory modeling consists of dividing video into semantically related time intervals by listening video and relating semantic entities of audio content with time intervals.

4.2.1 Auditory Segmentation

Auditory content of video is segmented coherent time intervals having distinct audio characteristics. Auditory content segmentation is done in two steps manually.

- Firstly, auditory content is segmented into ***background intervals*** considering background sounds. Background sounds give general feeling of the ambiance of time intervals. Some examples are music, bar ambience, street sounds and so on. Background intervals do not overlap with each other.

- Each background interval is divided into ***foreground intervals***, which are more distinguishable audio happenings. Their time intervals are bounded with theirs background time interval boundaries. Foreground intervals in a background interval can overlap with each other intervals. For example, Ali can speak during Aliye singing in pub ambiance. Figure 4.11 shows a sample relationship between background interval and foreground intervals. There are three foreground intervals in pub humming background interval.

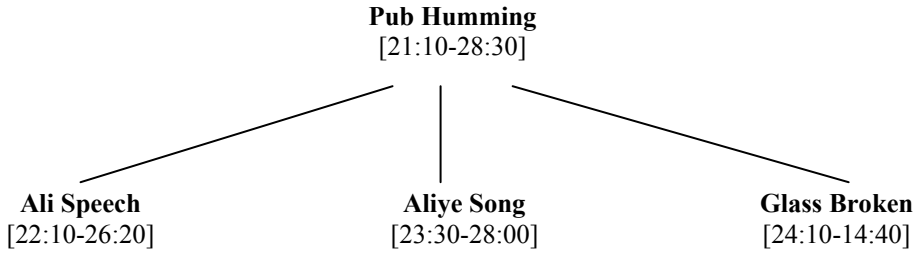


Figure 4.11 Background Interval and Foreground Intervals

4.2.2 Auditory Semantics

Auditory semantics of video are auditory events, auditory objects, and their temporal relations. After segmentation of video into background intervals and foreground intervals, auditory semantics are associated with these time intervals.

4.2.2.1 Auditory Objects

Auditory objects are sources of sound. Instruments of music, speaker of speech can be auditory objects. There can be a lot of sound sources in video, but salient objects are considered in this study. Sometimes finding the sound source can be difficult. For example, a lot of sound is mixed in pub ambiance, and separating from each other can be difficult. For this kind of sound, we did not label the objects. Generally objects are not determined in background intervals, but objects are more distinct in foreground intervals. Figure 4.12 shows auditory object node structure. One auditory object can be heard in different time intervals and in different events.

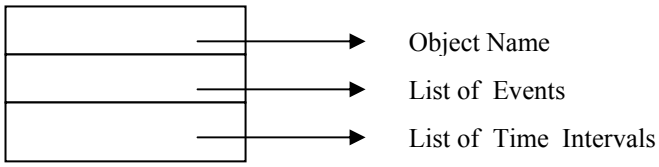


Figure 4.12 Auditory Object Node Structure

After annotated an auditory object, we put it into auditory object array whose elements are auditory object nodes. There is no duplicated auditory object name in auditory object array.

4.2.2.2 Auditory Events

Audible happenings in the auditory content are considered auditory events. Auditory events span time duration so they are related with time intervals. We labeled background interval's meaning that gives a feeling of ambiance as auditory event. ***"Auditory-Ambiance"*** type is given to events related with background intervals.

Foreground interval contains more distinct events taking short time such as speech, gun shot, bomb explosion, and so on. Speech event is different from other events because it contains speech text. So we separate speech events from other normal auditory events with ***"Auditory-Normal"*** and ***"Auditory-Speech"*** types. In events with "Auditory-Speech" type, keywords of speech text are associated with event. Auditory-Normal events are all aural happenings except speech at foreground level. Figure 4.13 shows auditory event node structure. Some auditory events have not auditory objects, for example silence event has not auditory object. Speech keywords are taken only events in speech type.

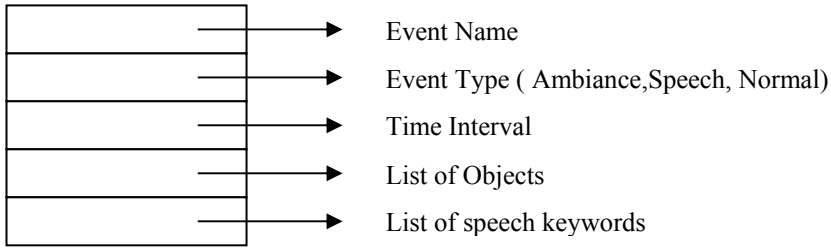


Figure 4.13 Auditory Event Node Structure

Speech events can be labeled in both foreground interval level and background interval level. If the speech takes short time, then speech is labeled at foreground level as shown in Figure 4.14. If there is a long speech given by one speaker, we label speech event at background level, then we divide this speech into sub-sections according to its subjects. The keywords are extracted at the foreground level, and the speaker of the speech is annotated at the background level. Figure 4.15 shows an example of long speech given by one speaker. All speech events in Auditory-Speech type are named with "speech"+Number that number starts from 1 and is increased automatically by one for new speech.

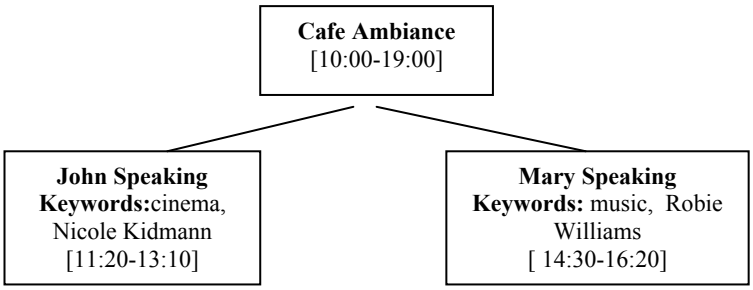


Figure 4.14 Short Speech Annotation at Foreground Level

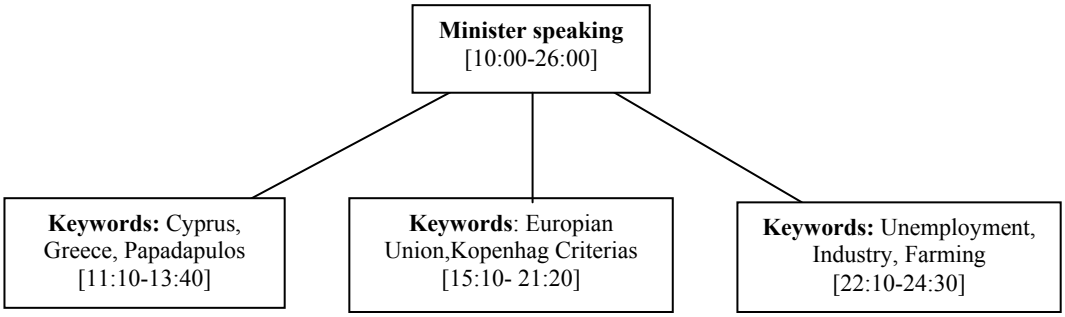


Figure 4.15 Long Speech Annotation both Background and Foreground Levels

Each auditory interval is related with one auditory event. After annotating an auditory event, we put it into the auditory event array. Same auditory event can be heard in different time intervals, so event name is not unique in auditory event array. Events having the same name differ from each other with distinct time intervals.

4.2.2.3 Auditory Time Intervals

Auditory content of video is segmented into time intervals as described in Chapter 4.2.1. These intervals construct a multi-way tree structure whose elements are auditory time interval nodes shown in Figure 4.16. Each time interval is associated with one event, zero or more objects. Time interval is bounded with event's starting time and event's ending time. Children keep auditory interval nodes for storing foreground intervals under background intervals.

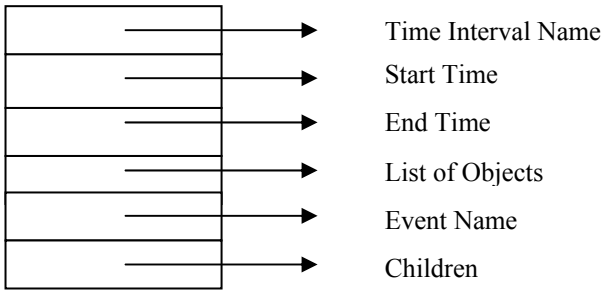


Figure 4.16 Auditory Time Interval Node

4.2.2.4 An Example of Auditory Content Model

Figure 4.17 shows a sample auditory content tree. Background levels of whole video are "Party Ambiance" and "Street Ambiance". Background intervals do not overlap with each other. There are three foreground intervals under "Party Ambiance" and two foreground intervals under "Street Ambiance". Barking and breaking; speaking and singing foreground intervals have common time intervals.

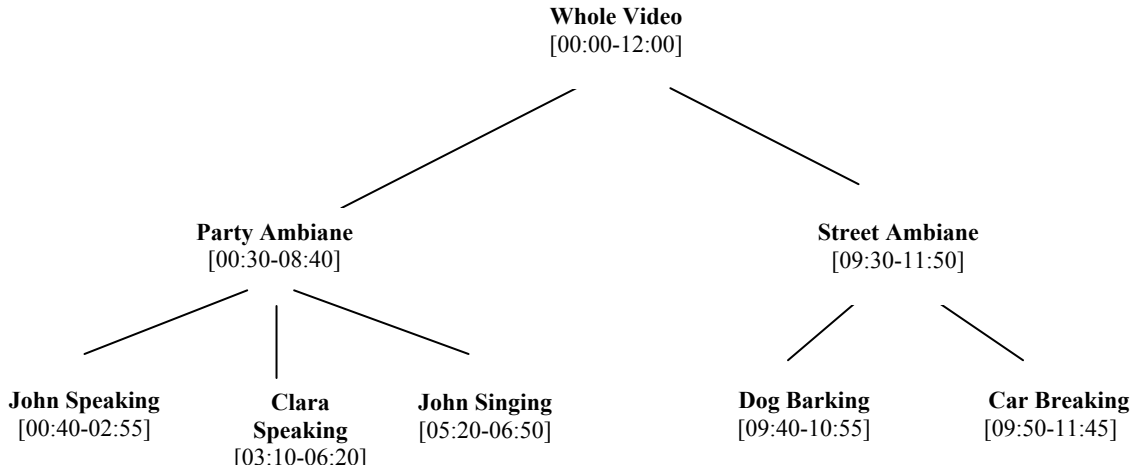


Figure 4. 17 Sample Audio Segment Tree

Figure 4.18 shows time intervals of the given tree. T0 is the root of the tree and has two sequence intervals. T1and T2 intervals are background intervals and have children. The others are foreground intervals and do not have children.

T0	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**
00:00	00:30	09:30	00:40	03:10	05:20	09:40	09:50
12:00	08:40	11:50	02:55	06:20	06:50	10:55	11:45
			John	Clara	John	Dog	Car
	Party Ambiance	Street ambiance	Speech1	Speech2	Singing	Barking	Breaking
T1,T2	T3,T4,T5	T6,T7					

Figure 4.18 Auditory Time Intervals

Figure 4.19 shows auditory objects and Figure 4.20 shows auditory events. Events in "Auditory-Speech" type have speech keywords. Objects are not extracted in the ambiances.

John	Clara	Dog	Car
Speech1, Singing	Speech2	Barking	Breaking
T3, T5	T4	T6	T7

Figure 4.19 Auditory Objects

Party Ambiance	Street Ambiance	Speech1	Speech2	Singing	Barking	Breaking
Ambiance	Ambiance	Speech	Speech	Normal	Normal	Normal
T1	T2	T3	T4	T5	T6	T7
		John	Clara	John	Dog	Car
		Music, Hiphop	Cinema, Brad Pitt			

Figure 4.20 Auditory Events

4.3 Supported Queries

The proposed model in this thesis supports all semantic queries in the AVIS model and fuzzy spatial and spatio-temporal queries in Köprülü [17]. Beside them, video content can be queried on visual content and auditory content. Temporal relationships between auditory events and visual events can also be queried. Spatial relationships between object and spatio-temporal characteristics of an object can be queried. Video content can be queried hierarchically. Conjunctive queries of the semantic queries, the spatial queries, and the temporal queries are also supported.

4.3.1 Auditory Queries

Queries in this type contain auditory clues, so auditory model is searched for retrieving the query results. Some auditory query types containing auditory clues are:

- *Given interval, find auditory events:* For these queries, auditory time intervals intersecting with the given interval are found and their events are listed. An example query for this type can be "Find all auditory events between 3^{th} and 9^{th} minutes".

- *Given interval, find auditory objects:* For these queries, auditory time intervals intersecting with the given interval are found and their objects are listed. An example query for this type can be "Find all auditory objects between 4^{th} and 6^{th} minutes".

- *Given auditory object, find intervals:* For these queries, the given auditory object is found in the auditory object array and its time intervals are listed. An example query for this type can be "Find all time intervals where ***gun*** is hearable".

- *Given auditory object, find events*: For these queries, the given auditory object is found in the auditory object array and its events are listed. An example query for this type can be "Find all auditory events where ***dog*** is hearable".

- *Given auditory event, find objects:* For these queries, auditory events labeled with the given auditory event are found in the auditory event array and their objects are listed. One auditory event can be labeled for different time intervals and can contain different objects in each interval, so all intervals related with the given event are searched. An example query for this type can be "Find all auditory objects in ***music*** event".

- *Given speech keyword, find intervals:* For these queries, firstly, auditory events in "Auditory-Speech" type are found, then the given keywords are searched in events' speech-list. If the given keyword is in the speech list of the event, event's time interval is listed. An example query for this type can be "Find all intervals where ***"Hakan Şükür***" keyword is being spoken".

- *Given speech keywords and speaker, find intervals:* For these queries, auditory events with "Auditory-Speech" type and with the given speaker auditory object are found. Then the given keywords searched in events' speech-list. If there is the given keyword is in the speech list of the event, event's time interval is listed. An example query for this type can be "Find all intervals where ***Erman Toroğlu*** is speaking about ***Hagi***".

Conjunctive auditory queries contain more than one auditory query, and each query is processed separately and their results are combined. Some conjunctive query types are:

- *Given more than one auditory event, find objects:* For these queries, each given event is found from the auditory event array then their objects are listed without duplicates. An example query for this type can be "Find all auditory objects in ***fighting*** and ***cheering*** events".

- *Given more than one auditory object, find events:* For these queries, auditory events having all given objects are listed without duplicates*:* An example query for this type can be "Find all auditory events having ***gun*** and ***car*** objects".

- *Given more than one auditory object or event, find intervals:* For these queries, each query chunk is processed separately then their results are listed by intersecting time intervals. An example query for this type can be "Find all intervals having ***referee whistle*** and ***audience cheering***".

- *Given speech keywords and auditory semantic entities, find intervals:* For these queries, each query chunk is processed separately then their results are listed by intersecting time intervals. An example query for this type can be "Find all intervals where ***penalty keyword is being spoken*** and ***audience is cheering***".

4.3.2 Visual Queries

In this type of queries contain visual clues, so visual model is searched for retrieving query results.

Some visual query types containing visual clues are:

- *Given interval, find visual events:* For these queries, visual time intervals intersecting with the given interval are found and their events are listed without duplicates. An example query for this type can be "Find all visual events between 2^{th} and 12^{th} minutes".

- *Given interval, find visual objects:* For these queries, visual time intervals intersecting with the given interval are found and their objects are listed without duplicates. An example query for this type can be "Find all visual objects between 4^{th} and 8^{th} minutes".

- *Given visual object, find intervals:* For these queries, visual object is found from the visual object array and its time intervals are listed. An example query for this type can be "Find all intervals in which ***goal-post*** is appearing".

- *Given visual object, find events*: For these queries, the given visual object is found from the visual object array and its events are listed without duplicates. An example query for this type can be "Find all visual events having ***plane*** object".

- *Given visual event, find visual objects:* For these queries, visual events labeled with the given event are found from the visual event array, and their objects are listed without duplicates. An example query for this type can be "Find all visual objects in ***crash*** event".

- *Given visual event, find intervals:* For these queries, visual events labeled with the given event are found from the visual event array and events' time intervals are listed. An example query for this type can be "Find all intervals in which ***penalty is given***".

- *Given visible text keywords, find intervals:* For these queries, visual events in *Visual-Text* type are found, and their text-lists are searched by the given keyword. If there is the given keyword in event's text-list, event's time interval is listed. An example query for this type can be "Find all intervals in which ***yellow card text is appearing***".

Conjunctive visual queries contain more than one visual query, and each query is processed separately and their results are combined. Some conjunctive query types are:

- *Given more than one visual event, find objects:* For these queries, each event is found from the visual event array then their objects are listed without duplicates. An example query for this type can be "Find all visual objects in ***driving*** and ***accident*** events".

- *Given more than one visual object, find events* For these queries, visual events having all given objects are listed without duplicates. An example query for this type can be "Find all events having ***plane*** and ***tower*** objects".

- *Given more than one visual object or event, find intervals:* For these queries, each query chunk is processed separately then their results are listed by time intervals intersection. An example query for this type can be "Find all intervals having ***referee waving the offside flag*** with ***İlhan scoring a goal***".

- *Given visible text keywords and visual semantic entities, find intervals:* For these queries, each query chunk is processed separately then their results are listed by intersecting time intervals. An example query for this type can be "Find all intervals in which ***Score text appearing*** and ***footballers hugging***".

4.3.3 Multimodal Queries

Multimodal querying is to query the video content using multiple data sources. Video content can be queried using visual content, text content, auditory content and speech content. With multimodal queries, user can search the video content from different aspects. The combination of different modalities provides to get absolute time intervals or absolute semantic entities to user's queries. According to clues in the query, query keywords are searched in the auditory model and the visual model, and then query results are combined with other query results.

Some multimodal query types are:

- *Given visual and auditory semantic entities, find intervals:* For these queries, each query chunk is processed in appropriate model separately, and then query results are shown by intersecting other results' time intervals. An example query for this type can be "Find all intervals that contain ***İlhan falling visual event*** and ***Referee whistling audio event***". In this query there are two query sentences: falling is visual type and whistling is auditory type. So we look first visual model for getting visual event intervals, secondly auditory model for getting auditory event intervals. If there are common intervals between two events result lists, these common intervals are shown to user.

- *Given speech keywords and visual semantic entities, find intervals:* An example query for this type can be "Find all intervals that contain ***'Hakan falling'*** and ***'Hıncal Uluç speaking about penalty and red card'*** ". In this query, there are two query sentences: falling is visual clue and speaking is auditory clue. So we look firstly visual model for getting visual event intervals and secondly look auditory model for getting auditory event intervals. Then we take intersection of these intervals.

- *Given visible text keywords and auditory semantic queries, find intervals:* An example query for this type can be "Find all time intervals that ***'America keyword is appearing'*** and ***'plane explosion is being heard'* ".** In this query, America keyword appearing is text-typed visual event and plane explosion is normal-typed auditory event. For retrieving first sentence's results, visual model is searched. For retrieving second sentence's results, auditory model is searched then results of these two sentences are intersected.

- *Given visible text keywords and speech text keywords, find intervals:* An example query for this type can be "Find all time intervals that ***'Gol keyword is appearing'*** and ***'Speaker is speaking about Zidane'*** ". In this query, there are text-typed visual event and speech-typed auditory event. For retrieving first sentence's results, visual model is searched. For retrieving second sentence's results, auditory model is searched then results of these two sentences are intersected.

- *Given interval, find all objects:* For these queries, auditory time intervals intersecting with the given interval are found and their objects are listed without duplicates. Then visual time intervals intersecting with the given interval are found. After that, visual objects are listed with auditory objects without duplicates. An example query for this type can be "Find all objects between 3th and 9th minutes".

- *Given interval, find all events:* For these queries, auditory time intervals intersecting with the given interval are found and their events are listed without duplicates. Then visual time intervals intersecting with the given interval are found. After that, visual events are listed with auditory events without duplicates. An example query for this type can be "Find all events between 4th and 12th minutes".

4.3.4 Hierarchical Queries

In this type of queries, the hierarchical structure of video content is queried. We can search events with Visual-Sequnce or Auditory-Ambience types by their sub-events or sub-objects.

- *Given visual event, find all events:* We find the given event from visual event array. If the given event type is Visual-Sequence, then its time interval's children are taken, and all children's events are listed. An example query for this type can be "Find all events in ***Party*** event".

- *Given visual event, find all objects:* We find the given event from visual event array. If the given event type is Visual-Sequence, then its time interval's children are taken, and all children's objects are listed. An example query for this type can be "Find all objects in ***Wedding*** event".

- *Given auditory event, find all events:* We find the given event from auditory event array. If the given event type is Auditory-Ambiance, then its time interval's children are taken, and all children's events are listed. An example query for this type can be "Find all events in ***Street-Ambiance*** event".

- *Given auditory event, find all objects:* We find the given event from auditory event array. If the given event type is Auditory-Ambiance, then its time intervals children are taken, and all children's objects are listed. An example query for this type can be "Find all objects in ***Music*** event".

4.3.5 Spatial Queries

Spatial relationships between two objects are queried with a membership values. Spatial query examples are:

- *Given two objects, spatial relationship, and fuzziness value; find all intervals:* Time intervals, in which two objects appear at the same time, are found. Then fuzzy spatial relations' formulas, defined in Köprülü's work [17], are applied to objects' locations. An example query for this type can be "Find all intervals in which ***plane*** is left of ***tower*** with 0.6 certainty level".

- *Conjunctive spatial queries:* In these queries, multiple spatial queries are given and time intervals satisfying all conditions are asked. Each spatial query is processed separately then their results are intersected. An example query for this type can be "Find all intervals in which ***'plane is left of tower with 0.6 certainty level'*** and ***'plane is right of tank with 0.5 certainty level***' ".

4.3.6 Regional Queries

- *Given object or text and time interval, find all locations:* Object or text locations, in the given interval, are asked. An example query for this type can be "Find all locations, in which ***score-board*** is seen between 4^{th} and 9^{th} minutes".

- *Given object-text, region, and fuzzy value; find all intervals:* Time intervals, in which object locations are matching with the given rectangular and with the given a fuzzy value, are asked. An example query for this type can be "Find all intervals ***Efes Beer*** text is seen the region given by pixel values [20, 60, 100, 120] with a threshold value of 0.8".

4.3.7 Spatio-Temporal Queries

Object trajectories are queried in this query types. Whether object follows path between two rectangles is queried.

- *Given one object, two regions, and fuzzy value, find all trajectories:* The processing algorithm of this query is similar with Köprülü's work[17]. An example query for this type is "Find all trajectories of the ***plane*** starting from the location L1 given by pixel values [50, 60, 50, 120] and ending at the location L2 given by pixel values [80, 90, 80, 150] with a threshold value of 0.5".

4.3.8 Temporal Queries

Temporal relationships between auditory events, between visual events, between auditory event and visual event are queried in this type of queries.

- *Given one visual or auditory event and temporal relationship, find all events:* Given event's time intervals are found and they are compared with the others time intervals by the given temporal relationship. An example query for this type can be "Find all events during ***Party-Ambiance***". This query retrieves all auditory and visual events during the Party-Ambiance event.

- *Given two events and temporal relationship, find all time intervals:* Time intervals of each event are found and they are compared with other events' time intervals by the given temporal relationship. An example query for this type is: "Find time intervals in which ***Ali and Hakan are fighting before Hakan has died***" ". Different operations are applied on different relations. If the temporal relation is "overlaps", then intersection operator is applied, else if it is "before", then concatenation operator is applied, else it is one of them of "meets", "starts", "finishes", "during", then union operator is applied.

- *Conjunctive temporal queries:* There are multiple temporal relations in query sentence. Result of each query is intersected, unioned, or concatenated according to the given temporal relationships meaning.

CHAPTER 5

IMPLEMENTATION

Multimodal video annotation and retrieval software is coded in Java using Java Media Framework Application Programming Interface (JMF API). JMF API handles streaming media, in which data is received and processed in particular time intervals. Audio and video can be example of streaming media. JMF API package gives video handling capabilities such as pausing the stream, restarting the stream, positioning the stream at any time, disabling sound (muting), enabling sound, and so on. JMF API is operating system dependent and supports different video formats and capabilities in different operating systems. We developed the program in Windows platform. Our program supports AVI (.avi) video files decoded with Cinepak codec.

In implementation phase, firstly data structures of the proposed model were designed and implemented as Java classes. Section 5.1 explains data structures and their relations in detail. After constituting data structures, user interface of the software was designed. Section 5.2 describes user interface design. On the basis of data structures and user interface design phases, video annotation part and video retrieval part were implemented. Section 5.3 explains the video annotation part and Section 5.4 explains video retrieval part with program screen shots.

5.1 Data Structures

Required data structures and data relationships for multimodal video annotation and retrieval software are defined and represented as a class diagram shown in Figure 5.1. Model classes are implemented according to this class diagram. The types of attributes are Time that represents JMF API Time class; Vector that represents linked-list structure in the java.util package; others types are primitive types.

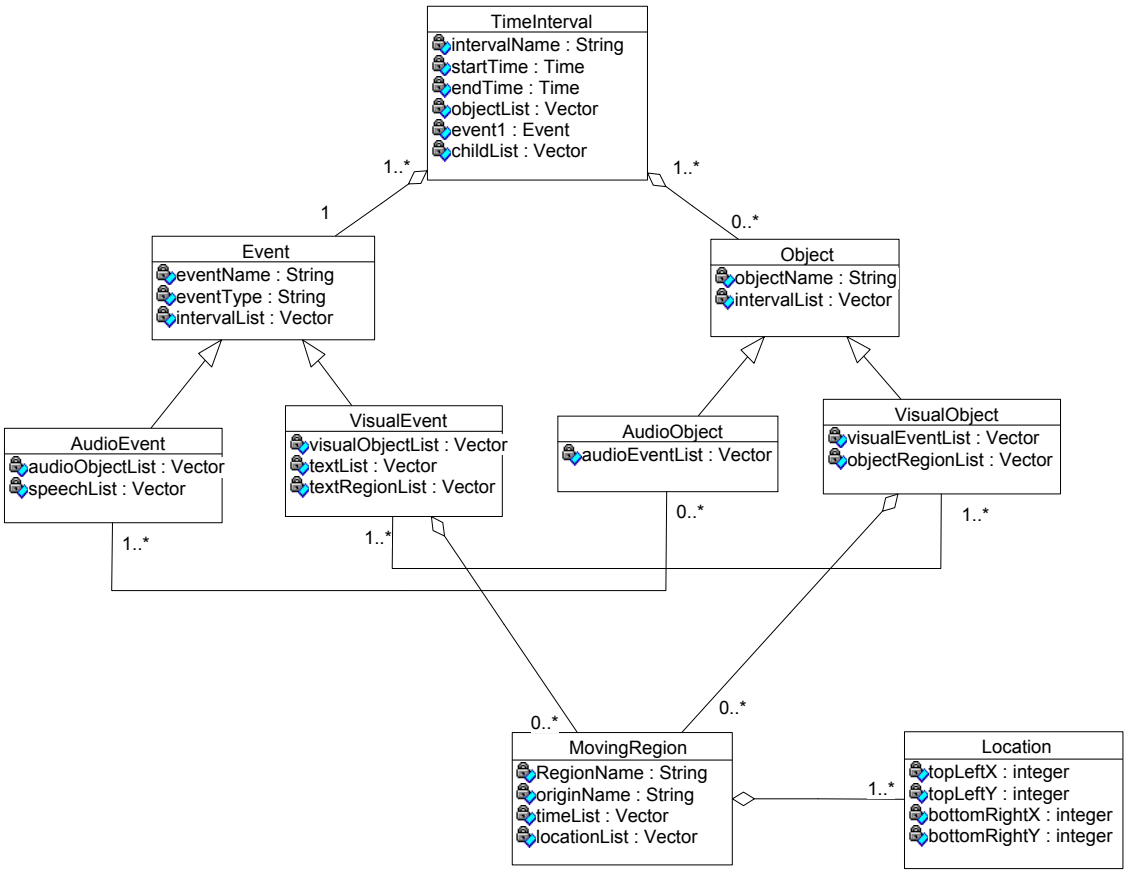


Figure 5.1 Class diagram of Multimodal Video Model

The specifications of the classes are:

- The **Time-Interval** class keeps start-time and end-time of the interval; objects and events in the interval; and child-list of the interval. The child-list keeps the sub-intervals of the interval. The child-list contains elements in Time-Interval type hence it enables constructing multi-way tree structure.

- The **Event** is the super-class of Audio-Event and Visual-Event. The audio event contains audio objects and speech content, whereas the visual event contains visual objects, text content and text regions. Events contain zero or more objects.

- The **Object** is the super-class of Audio-Object and Visual-Object. The audio object contains audio event list that keeps audio events containing this audio object. The visual object contains visual event list that keeps visual events containing this visual object, and region list that keeps this object's MovingRegions in every time interval. Visual object is represented with one MovingRegion for one time interval. Every object is related at least one event in own modality.

- The **MovingRegion** is a group of object or text locations in one time interval. The location of object can change during the interval, so location and time are kept for each different object location in the interval. MovingRegion keeps also origin-name for storing the region belonging to which object or text.

All of the data structures are stored in Vectors. There are two time-interval Vectors, one of them is for visual segment tree, and the other is for auditory segment tree. The Audio-Object Vector keeps all audio objects in the video. The Visual-Object Vector keeps all visual objects in the video. The visual-event Vector keeps all visual events in the video and the audio-event Vector keeps all audio events in the video. The region vector keeps object's regions and text regions.

5.2 User Interface

We combined video annotation and video retrieval part in one tool. First, a video is opened, and then data annotation and video querying are done over opened video. The tool desktop containing opened video is shown in Figure 5.2. The menu-bar of the desktop contains video handling menu, video data annotation menu, and video querying menu. VDBS menu handles "Opening Video", "Closing Video", "Saving extracted video information into files", "Loading extracted video information into program from files". Annotation menu handles annotating "Visual Sequence", "Visual Scene", "Auditory Background Interval", and "Auditory Foreground Interval". Querying menu handles "Content-based Querying", "Regional Querying", "Spatial Querying", and "Temporal Querying".

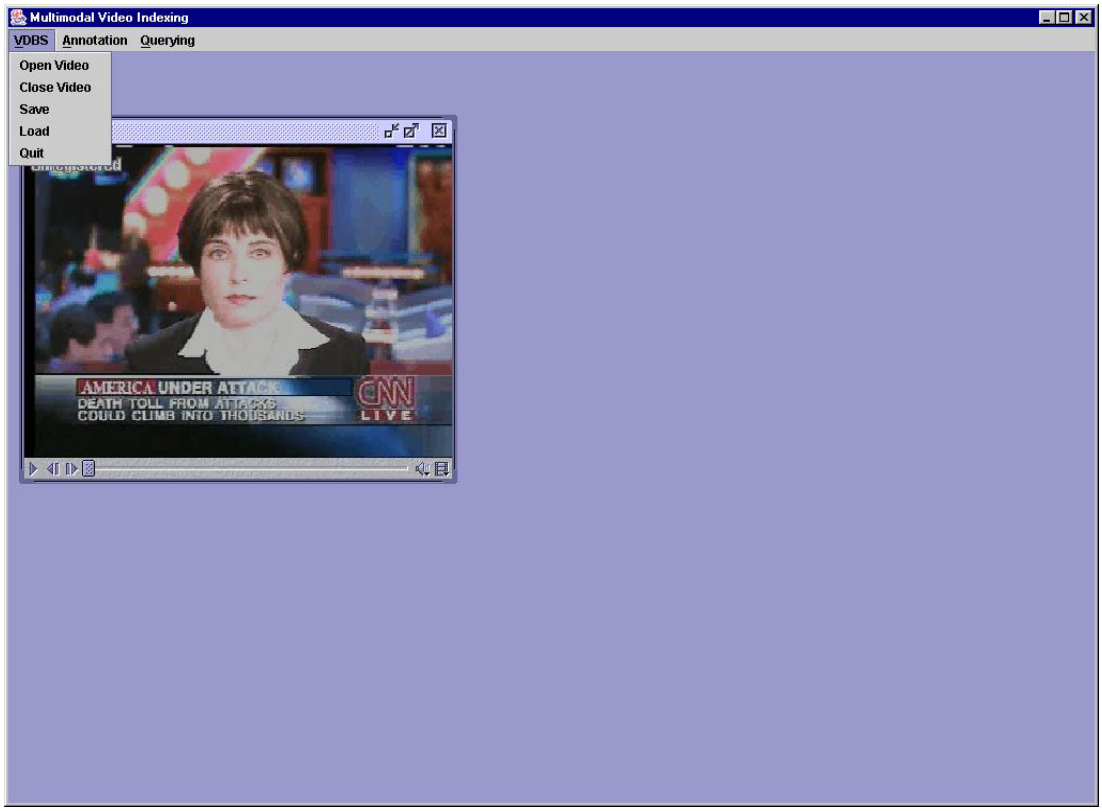


Figure 5.2 Multimodal Video Database Manager Desktop

5.3Data Annotation

All information can be extracted manually via graphical user interfaces. We extracted visual and auditory information separately, and inserted them into correspondingly visual data structures and auditory data structures. We kept visual data in visual time interval tree, visual event array, visual object array, region array; auditory data in audio time interval tree, audio event array, and audio object array. After extracting data, we wrote this information in files with 'Save' menu item under VDBS menu. There are separate files for keeping all data structures.

5.3.1 Visual Data Annotation

In visual data annotation, sound is not important so video is muted automatically. First of all, we determine video sequences and relate each sequence with visual event and visual objects. After extracting sequence, we divide each sequence into scenes and relate each scene with visual event and visual object. Object locations, text content, and text locations are extracted in scene annotation part.

5.3.1.1 Visual Sequence Annotation

In visual sequence part, beginning time and ending time of the sequence are determined. This time interval is related with one visual semantic event that is a group of small events. The objects of this event can be extracted via this interface, also sequence sub-events' objects are considered as sequence's objects. User interface of visual sequence extraction is shown in Figure 5.3. With *'Set Event'* button, extracted event is inserted into visual event array and sequence's event list. With *'Add Object'* button, extracted object is inserted extracted event's object list and extracted sequence's object list. With *'Add Sequence'* button, extracted sequence is inserted into visual time interval tree, and extracted objects are inserted into visual object array. After adding this sequence, user interface will be ready for extracting another sequence.

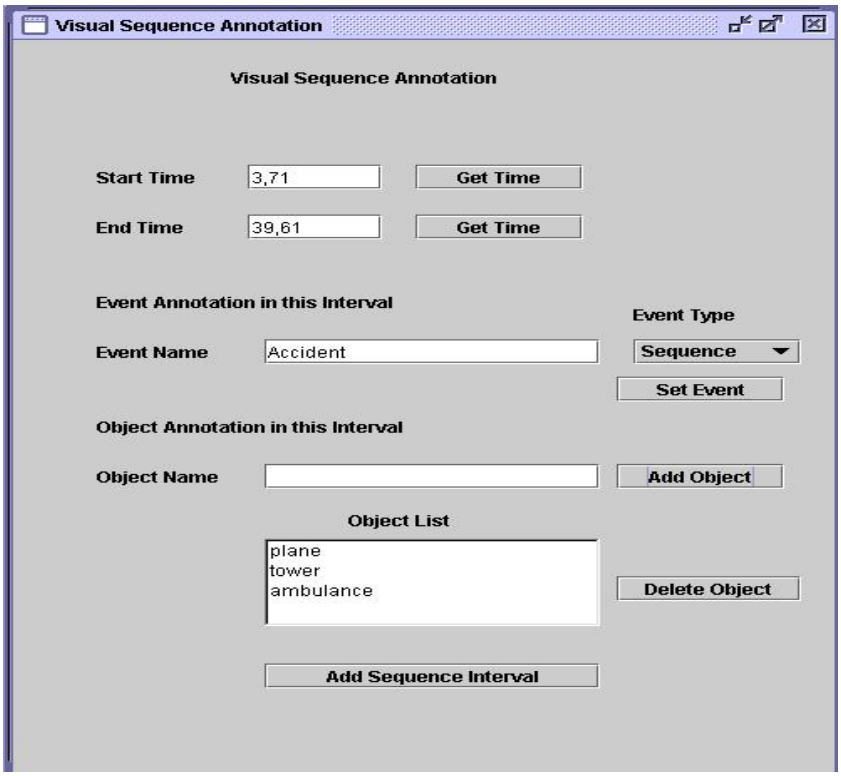


Figure 5.3 Visual Sequence Annotation Interface

5.3.1.2 Visual Scene Annotation

After annotation of all video sequences, scene annotation is done for each sequence. In the scene annotation, each sequence is divided into scenes separately. For each scene, starting time and ending time are annotated and event name of this scene is entered. Then event type is determined, there are two types of event at scene level: text and visual (normal) types. After entered name and determined type, event is inserted into visual event array. Event is set as scene event by pressing *'Set Event'* button.

If event type is visual, then objects and objects' locations can be annotated. While grabbing object locations during interval, object location should be kept for every considerable changes in object's location. By selecting an object from object list, selected object is activated in the location frame. By dragging mouse around object, rectangular is drawn; left-top, right-bottom positions, and current time are automatically shown in related text-boxes. With *'Add Location'* button, location is inserted into location list of the object's MovingRegion and time is inserted into time list of the object's MovingRegion. Visual event annotation is shown in Figure 5.4.

Figure 5.4 Normal-Visual Event Extraction Interface

If event type is text, text content is entered partly or entirely with *'Add Text'* button, and each text is inserted into event's text list. The text location can also be extracted with this interface. For text location, first, related text is selected from text list, then a rectangle that contains whole text is drawn. By pressing *'Add Location'* button, location is inserted into location list of the event'a MovingRegion, and time is inserted into time list of the event's MovingRegion. A snapshot of text event extraction is given in Figure 5.5.

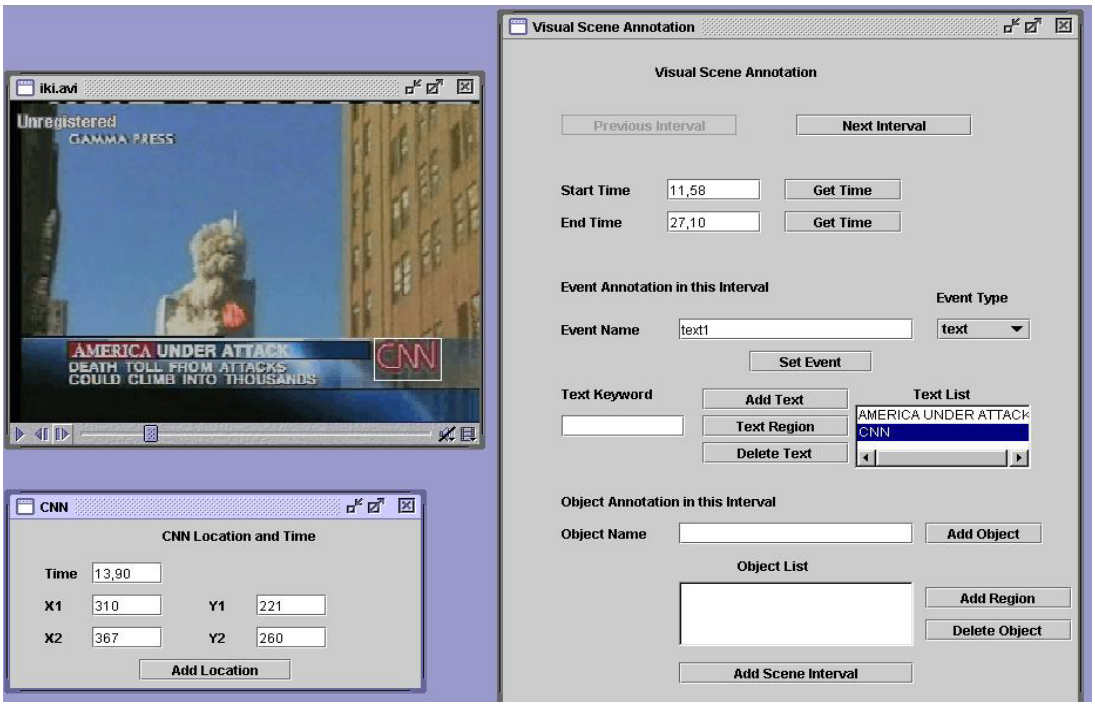


Figure 5.5Text Event Extraction Interface

5.3.2 Auditory Data Annotation

Audio data is extracted by listening sound track without video track. This extraction is done in two phases. First, sound track is divided into big chunks at audio background annotation phase, and then these big chunks are partitioned into small pieces in foreground annotation phase.

5.3.2.1 Audio Background Annotation

First, divide audio content into big time intervals considering background sounds. The audio background event indicates the interval's ambiance. User interface of the background annotation is shown in Figure 5.6.

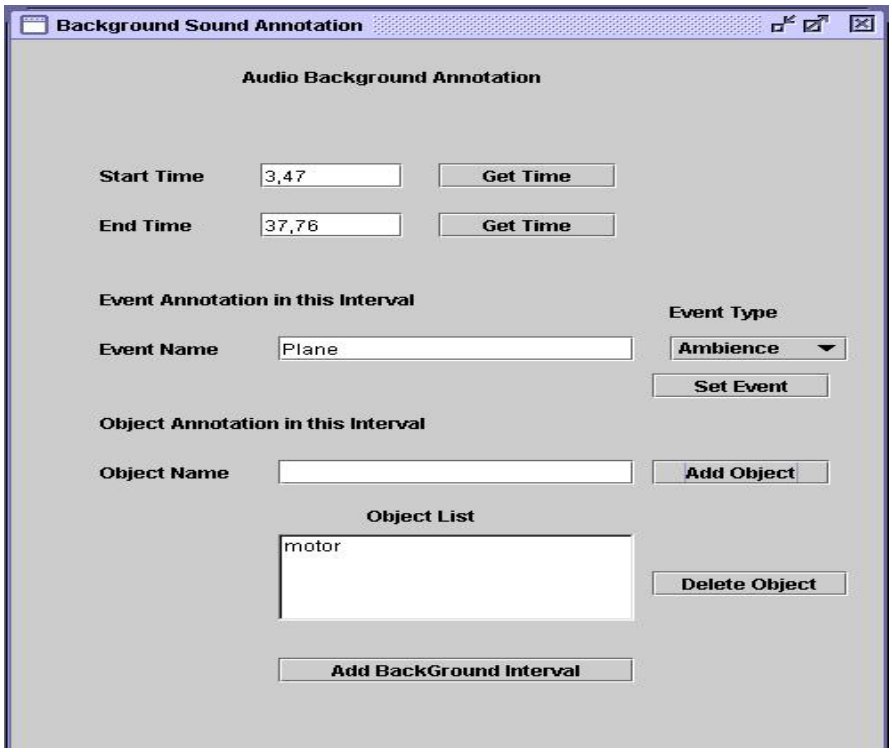


Figure 5.6 Audio Background Interval Annotation Interface

The starting and ending time of the background interval are got. By pressing *'Set Event'* button, audio event is entered into audio event array and background interval's event. If objects are extracted, they are inserted into current audio event's object list, and current interval's object list. With *'Add Background Interval'* button, extracted interval is inserted into the audio time interval tree, and extracted objects are inserted into audio object array. After adding this background interval, consecutive intervals can be added via this interface.

5.3.2.2 Audio Foreground Annotation

In each background interval, foreground sound intervals are extracted. Foreground sound interval is related with one audio event. Audio event type can be speech or audio (normal) at foreground level. In speech events, speech content is annotated and inserted into event speech list with *'Add Speech'* button. Speaker of speech is annotated as audio object, and inserted into event's object list and audio object array. Speech event extraction is shown in Figure 5.7. *'Add Foreground Interval'* button inserts the interval into child-list of related background interval. All sound events in the background interval are also annotated in this interface.

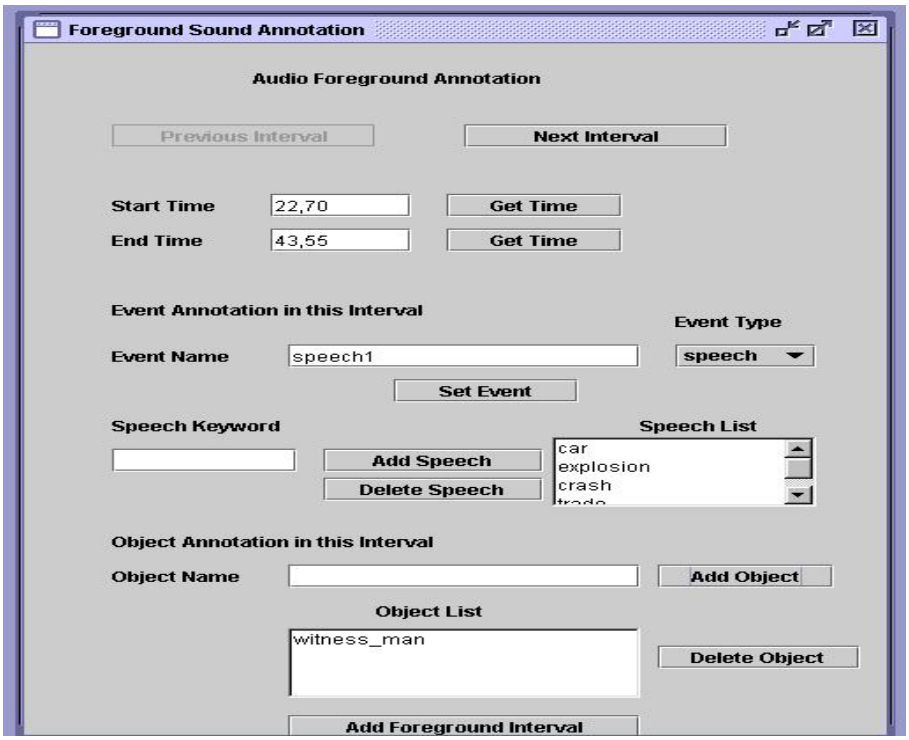


Figure 5.7 Speech Event Extraction Interface

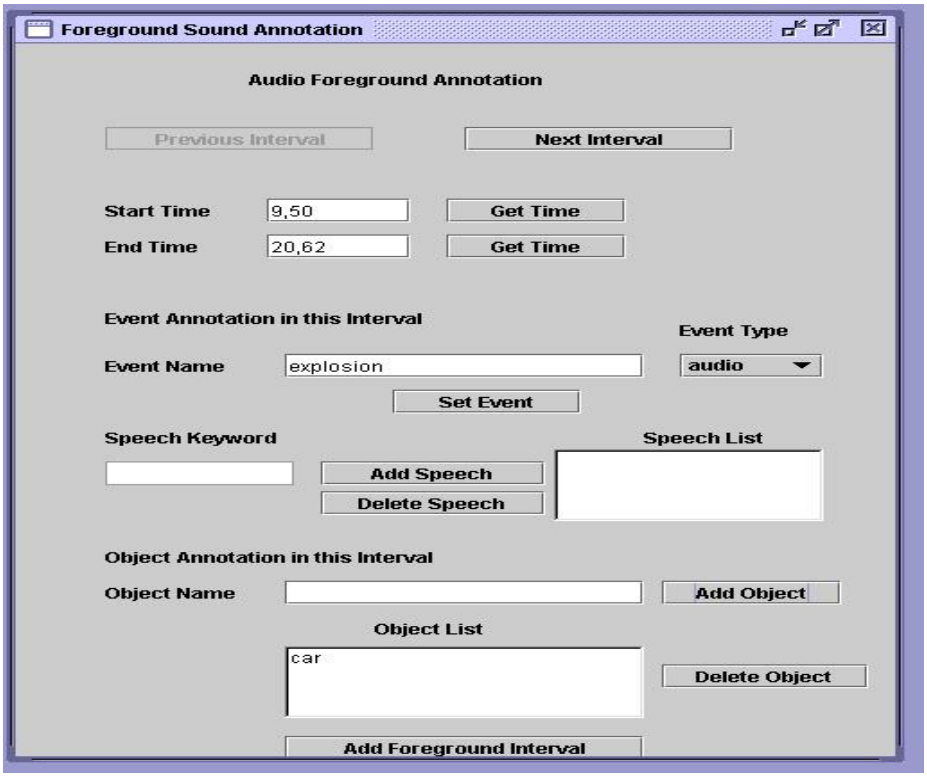


Figure 5.8 Normal-Audio Event Extraction Interface

In audio (normal) events, event is inserted into scene interval's event list and audio event array. Extracted audio objects are inserted into audio object array and event's object array. Foreground Annotation Interface is shown in Figure 5.8.

5.4Query Processing

Video content querying is done for opened video. Before video querying, opened video content must be extracted with the data annotation part. Extracted data was kept in files and these files are loaded into program data structures with *'Load'* menu item under VDBS menu. After loading data structures, video can be queried. There are mainly four query types:

- *Content based querying:* Time intervals, events, objects, and their relationships are queried in this type.
- *Regional querying:* Object's and text's locations and object trajectories can be queried.
- *Spatial querying:* Spatial relationships between objects are queried.
- *Temporal querying:* Temporal relationships between events can be queried.

5.4.1 Content based Querying

Content based querying provides us to reach content of video from various aspects. Visual, textual, and auditory information can be queried with different combinations. In User Interface of this type query, there are two frames: one of them is for specifying query conditions and the other is for showing results of specified query. In Query Creation frame, visual query and audio query specifications are selected from different panel. Visual query panel contains visual events, visual objects, and text content entities, whereas audio query panel contains audio events, audio objects, and speech content entities. While events and objects are selected from combo boxes, speech and text content are typed in text boxes. With *'Add Visual Query'* and *'Add Audio Query'* buttons, query sentence are inserted in entire query array. That means, more than one query sentence can be specified. User can bound the result's time interval by specifying starting and ending time of the interval. *'Create Query'* button shows whole query. *'Execute Query'* button processes query and shows query results in the Query Result Frame. *'Clear Query'* button empties all query related data structures and query text area. In Query Result Frame, there are a list for showing results of the query and video for positioning the selected result over video. Figure 5.9 shows Query Result Frame, and Figure 5.10 shows Query Specification Frame.

There are mainly three types in content based querying:

- ***Find All Intervals***: Intervals in which given objects are seen or heard; or given events occurrences are found.
- ***Find All Objects***: Objects in given interval or in given event are found.
- ***Find All Events***: Events having given objects or in given interval are found.

5.4.1.1 Find All Intervals

In this type of query; object, event, speech content, or text content can be given, time intervals satisfying the given conditions are asked. All modalities can be specified under this query keyword, so both auditory data structures and visual data structures are scanned for getting results. If there is one query sentence, all time intervals satisfying this query conditions are listed. If there exist more than one query sentence, each query results are intersected with other queries' results. Visual object, visual event, and text specifications are processed onto visual data structures. Audio event, audio object, and speech specifications are processed onto auditory data structures. If there are auditory query sentences and visual query sentences, audio queries and visual queries are processed separately and results are intersected. The user interface of query creation is shown in Figure 5.10. In this example, intervals containing following items altogether are found: "tower" visual object, "AMERICA" text, and "explosion" audio event. The results of this query are shown in Figure 5.9. By selecting an interval from result list, video is set to start time of the selected interval and the selected interval is played with play button.

Figure 5.9 The Query Result Interface

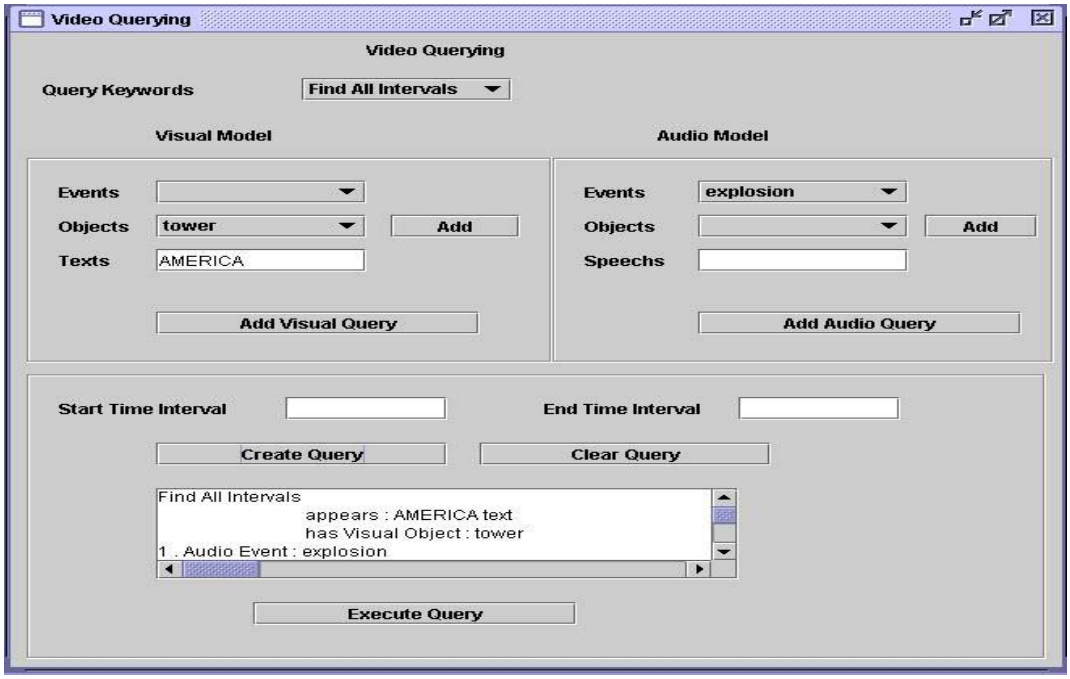


Figure 5.10 "Find all Intervals" Query Creation Interface

Some visual query examples and their algorithms:

- "Find all intervals containing visual object ***plane.***"
    - In this query, time intervals, in which visual object appears, are found. Algorithm 1 shows the processing of the queries in this type.

      Algorithm 1:

      1. For each sequence time interval in visual time interval tree,
         a. If there exists given object in interval's object list, add this interval into result list.
         b. Repeat **a** for each scene interval under current sequence interval.

- "Find all intervals in which occur visual event ***shout.***"
    - In this query, time intervals, in which visual event occurs, are found. Algorithm 2 shows the processing of the queries in this type.

      Algorithm 2:

      1. For each sequence time interval in visual time interval tree,
         a. If there exists given event in interval's event list, add this interval into result list.
         b. Repeat **a** for each scene interval under current sequence interval.

- "Find all intervals in which occur visual event ***crash*** has ***plane*** and ***tower*** visual objects**.**"
    - In this query, time intervals, in which visual event having visual objects occur, are found. Algorithm 3 shows the processing of the queries in this type.

      Algorithm 3:

      1. Find time intervals by Algorithm2.
      2. For each interval in the results of Algorithm2,
         a. If interval contains all given objects in query, add this interval into result list.

- "Find all intervals appearing ***Gool*** text."
    - In this query, time intervals, in which given text appears, are found. Algorithm 4 shows the processing of queries in this type.

      Algorithm 4:

      1. For all events in visual event array,
         a. If the type of event is "text", then

            If there exists given text in event's text list, add event's interval into result list.

- "Find all intervals containing visual event ***scoring a goal*** has visual object ***goalkeeper*** and ***ball,*** and ***Score*** text appears."
    - In this query, time intervals having visual event with visual objects and text are found. This is an example of the conjunctive query. In conjunctive queries, we use existing algorithms for query processing. For this query, Algorithm 3 and Algorithm 4 are used and their results are intersected by Algorithm 5.

      Algorithm 5:

      1. For each element in first query's results,
         a. Compare first query result's time interval with all time intervals in second query results,

            If there is common time between first time interval and second time interval, add this common time into result list.

- "Find all intervals occurring 'visual event ***crash*** has visual object ***tower'*** and 'visual event ***burn*** has visual object ***plane'***."
    - In this query, there are two query sentences. This is an example of another conjunctive query. Again each query results are intersected according to Algorithm 5. The results of above query are time intervals, in which crash event having tower overlaps with burn event having plane.

Some audio query examples and their algorithms:

- "Find all intervals containing audio object ***audience.***"
    - In this query, time intervals, in which audio object is heard, are found. The processing of this query resembles Algorithm 1. In this time, repeat all steps for audio time intervals.

- "Find all intervals occurring audio event ***explosion***."
    - In this query, time intervals, in which audio event occurs, are found. The processing of this query resembles Algorithm 2. In this time, repeat all steps for audio time interval tree.

- "Find all intervals occurring audio event ***explosion*** has ***bomb*** and ***plane*** audio objects."
    - In this query, time intervals, in which audio event with audio objects occurs, are found. The processing of this query resembles Algorithm 3. In this time, repeat all steps for audio time interval tree and audio event array.

- "Find all intervals having ***Bush and Blair*** spoken words."
    - In this query, time intervals, in which given keywords were spoken, are found. Algorithm 6 shows the processing of the queries in this type.

      Algorithm 6:

      1. For all events in audio event array,
         a. If the type of event is "speech", then

            If there exists given keyword in event's speech list, add event's interval into the result list.

- "Find all intervals occurring 'audio event ***whistle*** has audio object referee' and 'audio object speaker says ***Hakan and Arif***'."
    - In this query, there are two query sentences. Both query sentences are processed by above algorithms then results are intersected according to Algorithm 5. The results of above query are time intervals in which referee's whistle overlaps with speech event containing "Hakan and Arif" keywords.

When audio and visual query sentence are combined in one query, that query is called multimodal video query. The processing of multimodal video queries resembles conjunctive queries. That means, each sentence is processed according to related algorithm, and then results are intersected according to Algorithm 5. Some multimodal query examples and their processing are shown below.

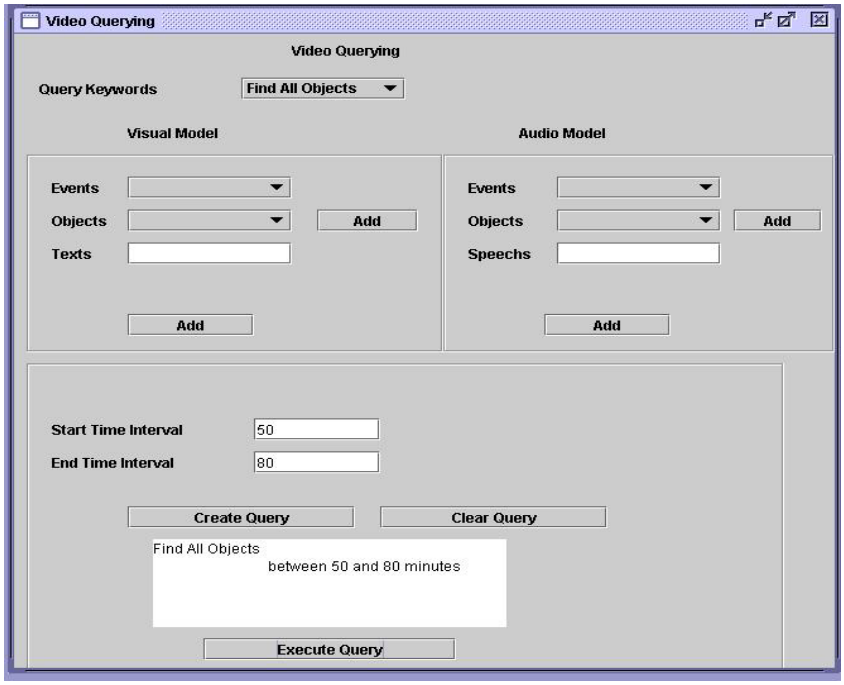
- "Find all intervals containing visual event ***scoring a goal*** has ***Hakan Şükür*** and spoken words ***fault*** and ***referee***."
    - This is an example of the least complex multimodal query. There are one visual query sentence and one auditory query sentence.
        - "Find all intervals containing visual event ***scoring a goal*** has ***Hakan Şükür.***" This is a visual query sentence and processed as Algorithm 3.
        - "Find all intervals containing spoken words ***fault*** and ***referee***." This is an auditory query sentence and processed as Algorithm3 using audio data structures.

- "Find all intervals in which ' visual event ***burn*** has visual object ***plane ';*** '***Crash*** text appears'; 'audio event ***explosion*** has audio object ***motor ';*** and '***Time and Date*** keywords are spoken."
    - This is an example of the most complex multimodal query. There are 'audio event having audio object', 'visual event having visual object', 'text event', and 'speech event'. There are four query sentences.
        - "Find all intervals containing visual event ***burn*** has visual object ***plane***." This visual query sentence is processed as Algorithm 3.
        - "Find all intervals appearing ***Crash*** text." This visual query sentence is processed as Algorithm 4.
        - "Find all intervals containing audio event ***explosion*** has audio object ***motor***." This auditory query sentence is processed as Algorithm 3 using audio data structures.
        - "Find all intervals, in which ***Time and Date*** keywords are spoken." This auditory query sentence is processed as Algorithm 6.

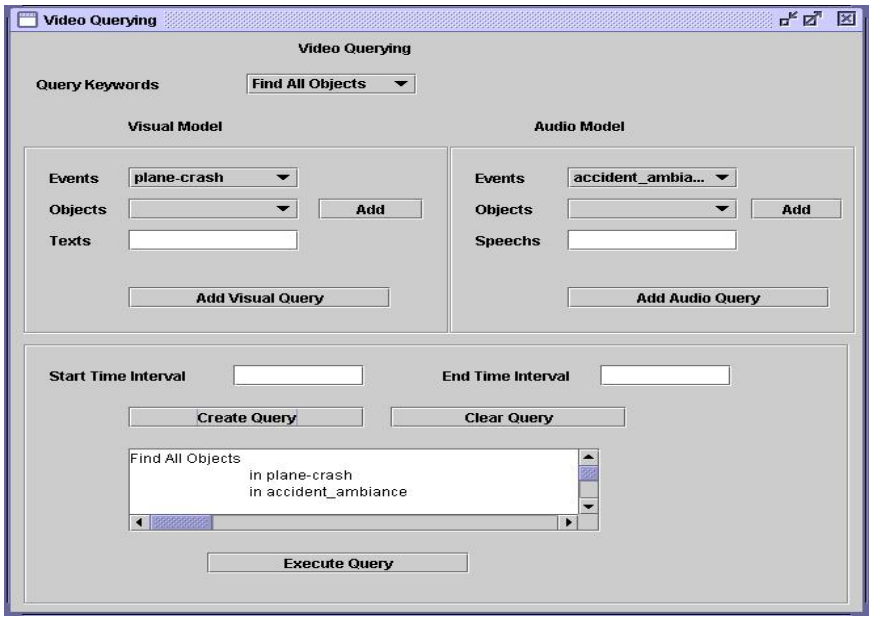
5.4.1.2 Find All Objects

In this type of query, objects in given interval or in given event are found. For getting objects in an interval, starting and ending time of the interval should be entered. For getting objects in an event, visual event or audio event should be selected from the combo boxes. The *'Create Query'* button shows whole query; the 'Execute Query' button processes query and shows query results in the Query Result Frame. Figure 5.12 shows two different query specification interfaces. In Figure 5.12.a, interval is given and objects in this interval are queried. In Figure 5.12.b, events are given and objects are in these events are queried. Figure 5.11 shows results of this type of queries. The results of the query are listed.

Figure 5.11 "Find all Objects" Query Result Interface



(a) A query example of given interval



(b) A query example of given events

Figure 5.12 "Find all Objects" Query Specification Interfaces

Some query examples in "Find All Objects" type and their algorithms:

- "Find all objects between ***03.10*** and ***06.20*** minutes."
    - In this query, the time interval is specified and all objects in this interval are found. Algorithm 7 shows the processing of queries in this type.

      Algorithm 7:

      1. For each sequence time interval in visual time interval tree,
         a. If sequence time interval intersects with given interval, then add sequence time interval's object list into result list.
         b. For each scene time interval in current sequence time interval,
            i. If scene time interval intersects with given interval, then add scene time interval's object list into result list.
      2. For each time interval in audio time interval tree
         a. If background time interval intersects with given interval, then add background time interval's object list into result list.

b. For each foreground time interval in current background time interval
    i. If foreground time interval intersects with given interval, then add foreground time interval's object list into result list.

- "Find all objects in visual event ***crash***."
    - In this query, visual event is given and objects in this event are found. Algorithm 8 shows the processing of queries in this type.

      Algorithm 8:

      1. Add visual event's object list into result list.
      2. For each interval in event's interval list,
         If interval has child intervals, then
             For each scene interval in sequence interval, add scene interval's object list into result list.

- "Find all objects in audio event ***explosion***."
    - In this query, audio event is given and objects in this event are found. Algorithm 9 shows the processing of queries in this type.

      Algorithm 9:

      1. Add audio event's object list into result list.
      2. For each interval in event's interval list,
         If interval is background interval,
             For each foreground interval in background interval, add foreground interval's object list into result list.

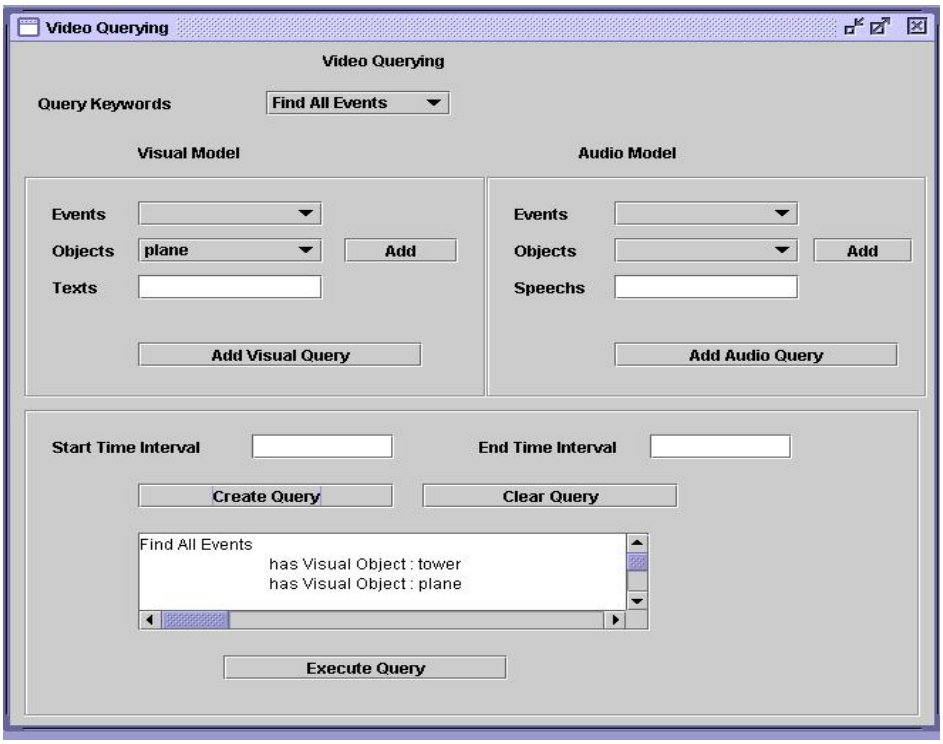
- "Find all objects in visual event ***crash*** and in audio event ***explosion***."
    - This is an example of conjunctive query. For this type of query, each query sentence is processed according to Algorithm 8 and Algorithm 9, then results of queries are combined as Algorithm 10.

Algorithm 10:

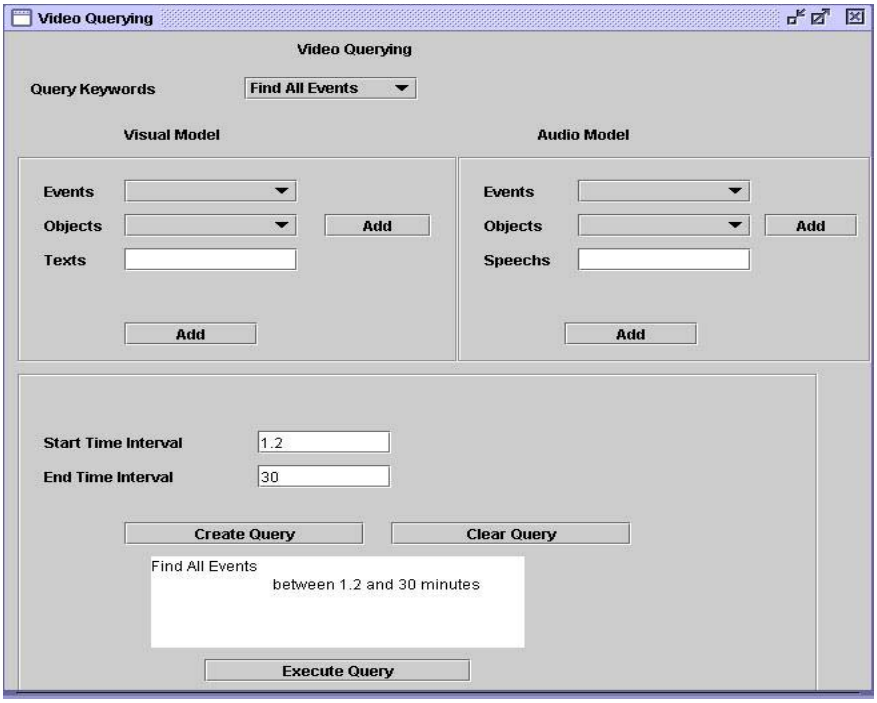
1. For each query results,
    a. For each element in query result list,
        If result list element is not in total-result list, then add this element into the total-result list. Otherwise continue with next element in result list.

5.4.1.3 Find All Events

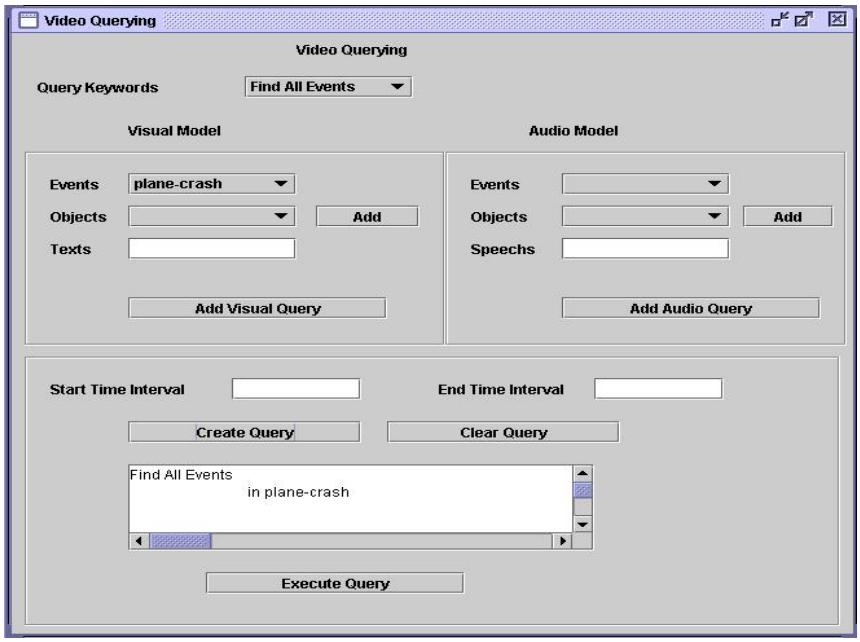
In this type of query, events in given interval, in given event, containing specified objects are found. For getting events in specific time interval, starting and ending time should be entered in user interface. For getting sub-events of the event, event should be selected and added into query. To get events containing some objects, objects should be selected and added into query. *'Create Query'* button shows entire query, *'Execute Query'* button processes the query and shows the results in the Query Result Frame. A selected event can be played with play button. Figure 5.13 shows the user interfaces with specified query conditions. In Figure 5.13.a, objects are given and events containing these objects are found. In Figure 5.13.b, time interval is specified and events in this interval are found. In Figure 5.13.c, event is given and sub-events in this event are found.



(a) A query example of given objects



(b) A query example of given interval



(c) A query example of given event

Figure 5.13 "Find all Events" Query Condition Specification Interfaces

The Figure 5.14 shows the Query Result Frame. All events satisfying conditions are listed, and with selection of an event, video is set to start time of the event, and with play button selected event can be played.

Figure 5.14 "Find all Events" Query Result Interface

Some query examples in "Find All Events" type and their algorithms:

- "Find all events between ***1.2*** and ***30.8*** seconds."
    - In this query, the time interval is specified and all events in this interval are found. Algorithm 11 shows the processing of the queries in this type.

      Algorithm 11:

      1. For each sequence time interval in visual time interval tree,
         a. If sequence time interval intersects with given interval, then add sequence time interval's event list into the result list.
         b. For each scene time interval in current sequence time interval,
            If scene time interval intersects with given interval, then add scene time interval's event list into the result list.
      2. For each time interval in audio time interval tree,
         a. If background time interval intersects with given interval, then add background time interval's event list into the result list.
         b. For each foreground time interval in current background time interval,
            If foreground time interval intersects with given interval, then add foreground time interval's event list into the result list.

- "Find all events containing visual object ***plane*** and visual object ***tower***."
    - In this query, objects are given and events containing these objects are found. Algorithm 12 shows the processing of this type queries.

      Algorithm 12:

      1. For each event in the visual event array,
         If visual event contains all given visual objects, add event into result list.
      2. For each event in auditory event array,
         If audio event contains all given audio objects, add event into result list.

- "Find all events in visual event ***party*** ."
    - In this query, event is given and sub-events under this event are found. Algorithm 13 shows the processing of the queries in this type.

      Algorithm 13:

      1. If given event type is "Sequence" or "Ambience", then interval's children events are inserted into the result list. Otherwise, result list has not element.

5.4.2 Regional Querying

In regional queries, visual object locations and text locations are queried. There are three types in regional querying: Locations of an object or text are queried; Time intervals of an object in specified location are queried; Trajectory of an object between specified two locations are queried. There is fuzziness capability for second and third query type.

In regional query interface, object and text can be selected from combo boxes. There is a threshold combo box containing values between 0.1 and 1.0. There are white panel for drawing rectangles; result panel for showing results as list format; and video panel for showing result over the video. White panel is used for entering input location for some queries and showing results for some queries.

Some regional queries and their algorithms:

- "Find all Intervals has visual object ***tower*** in given location with a ***certainty level 0.6***"
    - In this query, firstly, visual object is selected from visual object combo box, then location is drawn in white panel, and lastly membership value that the certainty level of object being in given location is specified between 0.1 and 1.0 value. Algorithm 14 shows the steps in finding fuzzy membership value between two rectangles. Algorithm 15 shows the steps in processing of the queries in this type.

      Algorithm 14:

      1. Find the rectangle having smaller area.
      2. Find intersected area of these rectangles.
      3. Find membership value with Formula 5.1.

$$\mu = \frac{\text{intersected\_area(Rectangle1, Rectangle2)}}{\text{minimum\_area(Rectangle1, Rectangle2)}}$$

Formula 5.1 Rectangle Matching

Algorithm 15:

1. Get whole regions of the object.
2. For each region,
    a. For each location of the region,
        i. Find membership value between object location and drawn rectangular.
        ii. If membership value is equal or greater than the specified membership value, add time interval between this location's time and the next location's time into result list.

Figure 5.15 shows the interface of this query. Before creating query, a rectangle should be drawn in the white panel, visual object and membership value should be selected. The results of the query are listed. With selection of a time interval from result list, video is set to start time of the selected interval. Play button plays the selected interval.

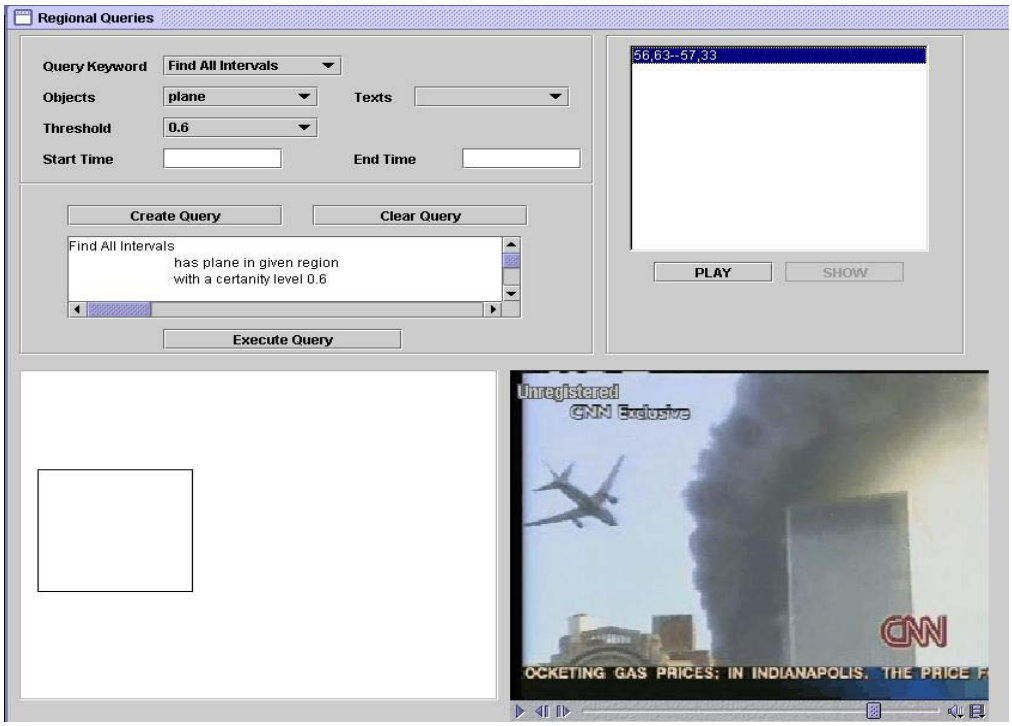


Figure 5.15 "Find All Intervals" Regional Query Interface

- "Find all locations of visual object ***plane***."
    - In this type of query, object or text is specified from combo boxes, and all locations of this object or text are queried. Algorithm 16 shows the processing of queries in this type.

      Algorithm 16:

      1. If object locations are queried, add object locations into result list.
      2. If text locations are queried, add text locations into result list.

- "Find all locations of visual object ***tank*** between ***1.9*** and ***2.6*** minutes."
    - In this type of query, object's locations or text's locations in the specified interval are queried. Algorithm 17 shows the processing of this type queries.

      Algorithm 17:

      1. If object locations are queried, get the object's interval vector. If text locations are queried, get the text event time interval.
      2. For each time interval of object or text,

         If there is intersection between specified time interval and current interval, add all locations in current interval's region vector into result list.

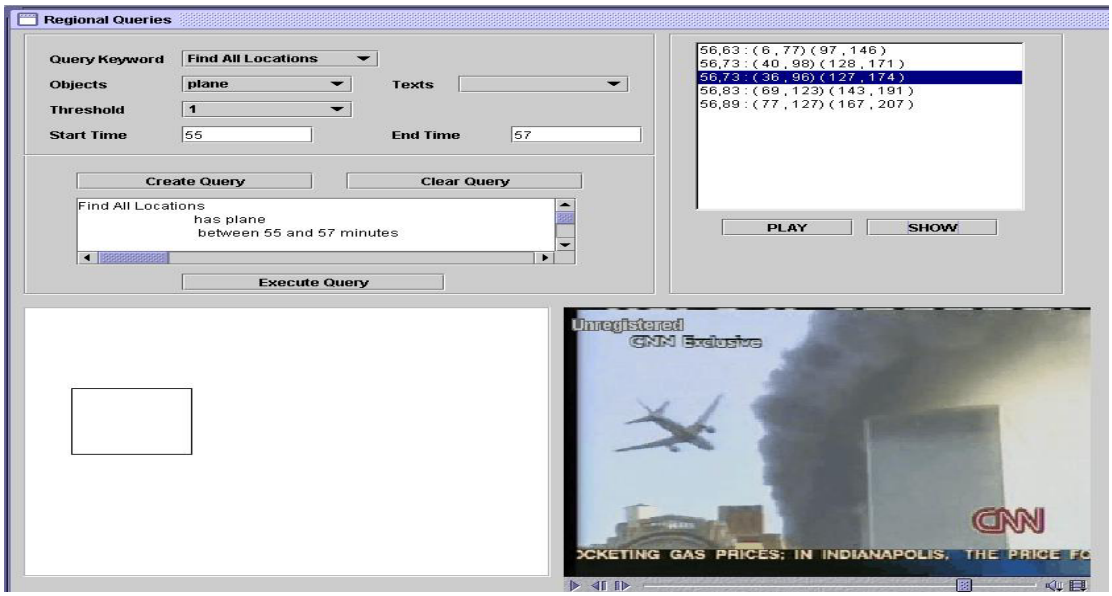


Figure 5.16 "Find All Locations" typed Regional Query Interface

Figure 5.16 shows a sample interface of queries in this type. Locations are listed with time and position information. With selection of a position from result list, video is set to location time. The show button shows the selected location in white panel.

- "Find all trajectories in which ***tower*** moves from Location1 to Location2 with ***a certainty level 0.4.***"
    - In this type of query, firstly, object is selected from object combo box, then Location1 and Location2 are drawn in white panel, and lastly membership value is specified. Algorithm 18 shows the processing of the trajectory queries. A trajectory of the object is kept in a trajectory class that keeps the locations and times of the object between Location1 and Location2.

      Algorithm 18:

      1. For each region of the object,
          a. For each location in the region,
              i. If location is intersected with Location1 with specified membership, create a trajectory object and add location and time into the trajectory object.
              ii. Else if location is intersected with Location2 with specified membership and trajectory object has location, add location and time into trajectory object and add this trajectory object into result list. Then create a new trajectory object.
              iii. Else
                  1. If trajectory object location list is not empty
                      a. If location is adjacent to previous location of current trajectory object, add location and time into the trajectory object. Else clear all locations and times from current trajectory object.

Figure 5.17 shows the interface of queries in this type. Location1 and Location2 are drawn in white panel, object and membership value are selected from combo boxes then query is created with *'Create Query'* button. By selecting a trajectory from result list, 'Show' button and 'Play' button are activated and video is set the first location of trajectory which object is seen in Location1. With *'Play'* button, the time interval between object located in Location1 and object located in Location2 is played. With 'Show' button, all object locations between Location1 and Location2 are shown.

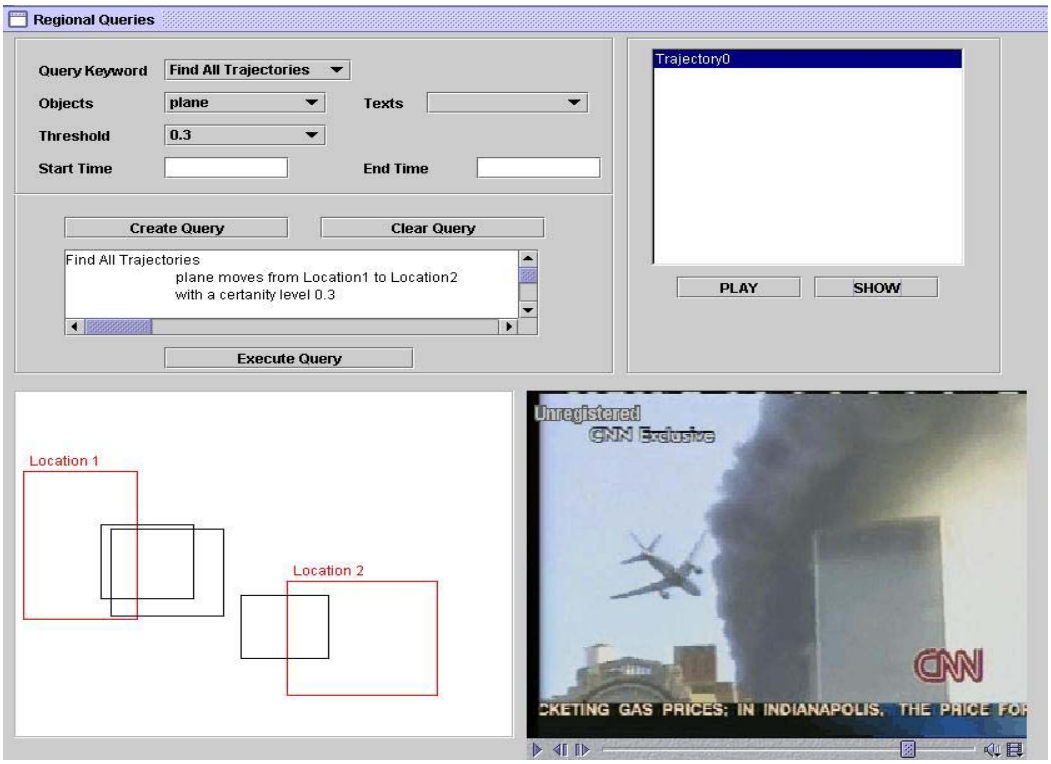


Figure 5.17 "Find All Trajectories" typed Regional Query Interface

5.4.3 Fuzzy Spatial Querying

In this type of query, spatial relationships between two objects are queried. More than one query sentence can be queried together. Spatial relation between two objects can be directional such as "left", "right", "bottom", "top", "top-left", "top-right", "bottom-left", "bottom-right"; or topological such as "overlaps", "equal", "inside", "contain", "touch", "disjoint" relations. Directional relations can have uncertainty, so we use certainty levels between 0.1 and 1.0. Some spatial queries and their algorithms:

- "Find all intervals ***plane*** is top-left of ***tower*** with a certainty level 0.6."
    - In this query, two visual objects are given, "top-left" spatial relation is selected, and membership value is specified. Algorithm 19 is used for processing of the queries in this type.

Algorithm 19:

1. Take time intervals that both two objects are seen. If two objects appear at the same time, then locations of objects in this time are added into separate location-lists, and this time is added into time-list.
2. For each time that objects are appearing together,
   a. Compare first object location with second object location according to given spatial relation and threshold value. If relation is satisfied with given threshold ratio, add time interval between this time and next time into result list.

- "Find all intervals visual object ***plane*** is right of visual object ***tower*** with a certainty level 0.8 ***and*** visual object ***plane*** is left of visual object ***tank*** with a certainty level 0.6."
  - In this query, there are two query sentences. This is an example of conjunctive query. Each sentence is queried according to Algorithm19. The results of the query sentences are intersected according to Algorithm 5.

The Figure 5.18 shows the user interface of spatial query creating. After selecting two visual objects, spatial relation, and membership value, *'Add'* button adds this spatial query into entire query. The *'Create Query'* button shows entire query, and 'Execute Query' button processes entire query and shows the results of the entire query in the Query Result Interface.

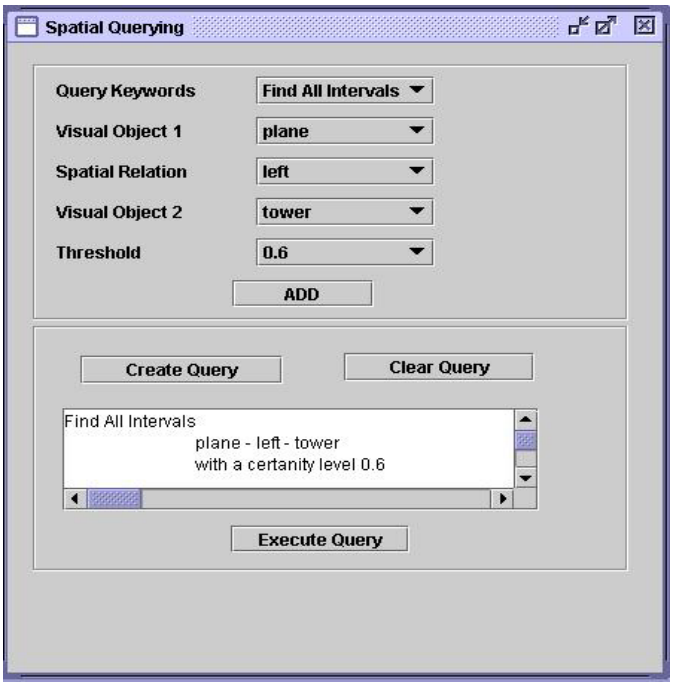


Figure 5.18 Spatial Query Specification Interface

The Figure 5.19 shows the results of the spatial query. By selecting a time interval from list, video is set to start time of the interval. The *'Play'* button plays the selected interval.

Figure 5.19 Spatial Query Result Interface

5.4.4 Temporal Querying

In temporal querying, two events are compared with each other by their time sequences. Conjunctive queries can also be applied on temporal relations. Events can be chosen both auditory type events and visual type events. Temporal relationships are "before", "meets", "overlaps", "during", "starts", "finishes", "equal". Some temporal queries and their algorithms:

- "Find all intervals overlapping 'visual event ***decline*** has visual object ***plane'*** and 'audio event explosion has audio object motor'."
    - In this query, there are two events and one temporal relation, and time intervals satisfying this temporal relation are asked. Algorithm 20 is used for queries in this type.

      Algorithm 20:

      1. Take events' intervals.
      2. For each interval of first event,
         a. Compare with each interval of second event in given temporal relation. If given temporal relation is satisfied between these two intervals, then union, intersection, or concatenation operators are applied according to temporal relation.
            i. If temporal relation is *before*, add first event's interval into result list.
            ii. If temporal relation is *overlaps*, take intersection of intervals and add these intersected time intervals into result list.
            iii. If temporal relation is *meets*, take union of intervals and add these time intervals into result list.
            iv. If temporal relation is *during*, add first event's interval into result list.
            v. If temporal relation is *starts*, add second's event interval into result list.
            vi. If temporal relation is *finishes*, add second's event interval into result list.
            vii. If temporal relation is *equal*, add first's event interval into result list.

- "Find all events before ***10.2*** minutes."
    - In this query, time is given; temporal relation is specified; and all events satisfying given relation are asked. Algorithm 21 is used for processing of the queries in this type.

      Algorithm 21:

      1. For all event in visual event list and audio event list,
          a. For each interval of event,

             Compare interval with given time in specified temporal relation. If temporal relation between time and time interval is satisfied, add this event into result list and add this interval into time result list.

- "Find all events during visual event ***party***."
    - In this query, visual or audio event is given; temporal relation is specified; and events satisfying given temporal relation are asked. Algorithm 22 is used for processing of the queries in this type.

      Algorithm 22:

      1. Take intervals of given event,
      2. For each event in audio event list and visual event list,
          a. Take intervals of current event.
          b. For each interval of current event,

             Compare interval with given event intervals according to specified temporal relation. If temporal relation is satisfied between two intervals, then add current event into result list.

- "Find all events before ***10:20*** minutes and overlaps visual event ***decline***."
    - In this query, there are two query sentences. This is an example of conjunctive query. Each query sentences are processed as related algorithms and results are intersected. Event intersection is done by getting common events from result lists.

Figure 5.20 shows the user interface of temporal query specification. Events are selected from visual events or audio events combo boxes. Temporal relation is selected, and time can be typed into text-box. After specifying conditions, 'Add' button inserts this query sentence into entire query. More than one query sentence can be added with add button.

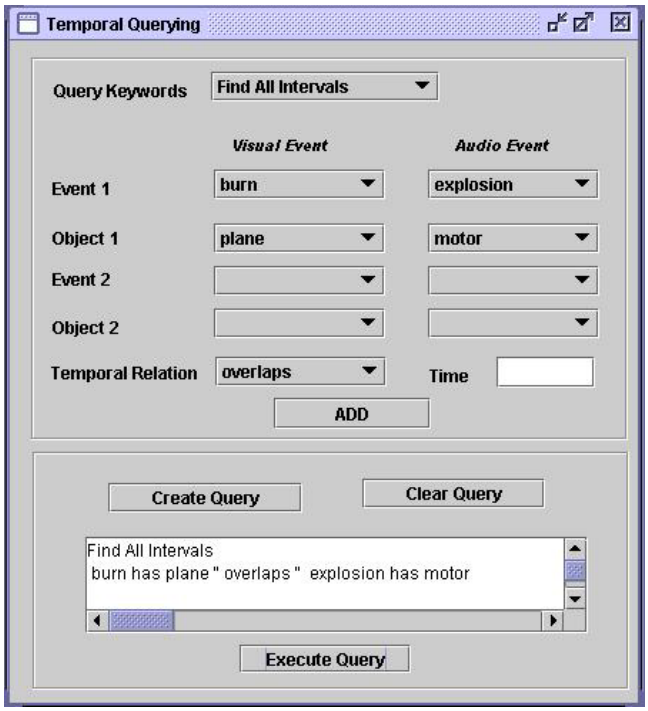


Figure 5.20 Temporal Query Specification User Interface

Figure 5.21 shows the user interface of the temporal query. A selected time interval or event can be played with play button.

Figure 5.21 Temporal Query Result User Interface

CHAPTER 6

CONCLUSION AND FUTURE WORK

The content of video is composed of visual, auditory, and textual sources. Visual effects add descriptive information and expand the viewer's imagination; audio effects add level of meaning and provide sensual and emotional stimuli; superimposed text brings some additional information and spoken document helps to viewers better understand the video content. Spatial and temporal properties are also important part of video content.

In this thesis, we have considered all data sources with spatial and temporal dimensions. We have proposed a semantic video model that combines auditory content, visual content, and textual content. By combining all data sources of video in one model, we provide richer and more complete video content. We introduced general video model covering different applications, such as news, sports, films, talk-shows, and so on.

We modeled visual, auditory, and speech contents in hierarchical structures. The hierarchical structure in both auditory model and visual model provides querying video content from more general abstract groups to specific single events. Speech hierarchies provide to reach specified keywords at almost exact place in speech duration. The modeling auditory content and speech content with semantic levels is as useful as the modeling visual content in levels.

Our model supports querying the video content with combination of different modalities. Visual content and auditory content can be queried with events or objects, while speech content and text content can be queried with important keywords. Based on spatial properties, we handled querying the spatial relations between two objects with fuzzy values; trajectory of an object from start to end regions with fuzzy values; and regional querying of an object. We support temporal queries that investigate temporal relations between events. We also support conjunctive queries in content-based querying, spatial querying, and temporal querying. In conjunctive queries, multiple query sentences are processed separately and then their results are combined by intersection, union or concatenation operators. By combining spatial and temporal features with multimodality, we extend the content-based video models and rich existence query types.

We developed a software prototype, which has annotation and querying capabilities. We applied object-oriented principles on proposed model and implemented our software in Java. Java developed JMF API for multimedia applications, but it does not support all video formats. Consecutively, we could not support all video formats in our software. In our software, queries are processed over one video data structures. One extension of our program can be querying multiple videos together using highly developed databases such as Oracle, SQL Server. Our software does not support browsing and updating the annotated data, software can be improved as supporting these capabilities. Another suggestion is that results can be shown in one interface as thumbnails of time intervals' key-frames for easy browsing in the case of query results are time intervals. Our model is designed by considering all domains. It can be utilized for different domains considering their own dominant data source. In our model, we assigned speech keywords to foreground interval duration. Speech keywords can be aligned to real spoken times for exact query matching.

We annotated data manually in our software because we concentrated on modeling and querying. But manually data annotation is tedious, subjective and time consuming. Automatically video segmentation and semi-automatically object extraction techniques help annotation processes. However automatic segmentation and data extraction studies are still state of art, and there is a big gap between semantic and syntactic parts of video. In our study, video can segmented into semantic units by combining of speech content, readable text content, visual and auditory content together. For reaching this aim, VOCR, ASR, image processing, signal processing, machine learning, and artificial intelligence techniques should be cooperated.

That different indexer can give different label to same object or same event in annotation phase is the other problem. Salient events and salient objects can be standardized in highly using applications to reduce ambiguity. Another suggestion is query processor supporting human's daily questions that can be developed using natural language techniques.

REFERENCES

[1] Adalı S., Candan K. S., Chen S., Erol K., Subrahmanian VS., "The Advanced Video Information System: Data structures and Query Processing", *Multimedia Systems*, vol. 4, pp. 172-186, 1996.

[2] Adams W.H., Iyengar G., Lin C-Y, Naphade M.R., Neti C., Nock H.J., Smith J.R., "Semantic Indexing of Multimedia Content Using Visual, Audio and Text Cues", *Eurasip Journal on Applied Signal Processing Vol 2003, No 2,* pp. 170-185, 2003.

[3] Agius H.W., Angelides M.C., "Modeling Content for Semantic-Level Querying of Multimedia", *Multimedia Tools and Applications* 15(1), pp.5-37, 2001.

[4] Allen J. F., "Maintaining Knowledge about Temporal Intervals", *Communications of ACM*, 26 (11), pp. 832-843, 1983.

[5] Ariki Y., "Organization and Retrieval of Continuous Media", *ACM Multimedia Workshops 2000,* pp.221-226, 2000.

[6] Arslan U., "A Semantic Data Model and Query Language for Video Databases", *M. Sc. Thesis, Department of Computer Engineering, Bilkent University, Ankara, Turkey*, January 2002.

[7] Audio Logger, Web Site: www.virage.com/products/audiologger.html, July 2004.

[8] Coden A., Haas N., Mack R., "Multi-Search of Video Segments Indexed by Time-Aligned Annotations of Video Content", Internal working paper in IBM Research Center, 1999.

[9] Dönderler Mehmet Emin, "Data Modeling and Querying for Video Databases", *P.Hd. Thesis, Department of Computer Engineering, Bilkent University, Ankara, Turkey*, July 2002.

[10] Fan J., Yau D. K., Hacid M.S., Elmagarmid A., "Model-based Semantic Object Extraction for Content-based Video Representation and Indexing", *Proc. SPIE Vol. 4315, Storage and Retrieval for Media Databases*, pp. 523-535, 2001.

[11] Film Sound Cliches, Web Site: http://www.filmsound.org/cliche/ , July 2004.

[12] Hampapur A, "Semantic Video Indexing: Approach and Issues", *SIGMOD Record, Vol. 28, .no.1,* pp. 32-38, March 1999.

[13] Herrera P., Serra X., Peeters G., *"* A Proposal for The Description of Audio in The Context of MPEG-7*",Proceedings of First International Workshop on Content-Based Multimedia Indexing. Tolousse, France*,1999.

[14] Huang Q., Puri A., Liu Z., **"**Multimedia Search and Retrieval: New Concepts, System Implementation, and Application", *IEEE Transactions On Circuits And Systems For Video Technology, Vol. 10, No. 5*, pp. 679-692, August 2000.

[15] Hunter J., Iannella R., "The Application of Metadata Standards to Video Indexing", S*econd European Conference on Research and Advanced Technology for Digital Libraries ECDL'98*, Crete, Greece, 1998.

[16] Informedia-II Digital Video Library, Web Site: http://www.informedia.cs.cmu.edu/, 15 July 2004.

[17] Köprülü M., Cicekli N.K., Yazici A., "Spatio-temporal Querying In Video Databases", *Inf. Sci. 160(1-4),* pp. 131-152, 2004.

[18] Li J.Z., Özsu M.T., Szafron D., "Modeling of Moving Objects in a Video Database", *Proceedings of IEEE International Conference on Multimedia Computing and Systems, Ottawa, Canada*, pp. 336-343, 1997.

[19] Mihajlovic V., Petkovic M., "Automatic Annotation of Formula 1 Races for Content-Based Video Retrieval", *Technical report no. TR-CTIT-01-41, Centre for Telematics and Information Technology, University of Twente,* 2001.

[20] Moncrieff S., Dorai C., and Venkatesh S., "Detecting Indexical Signs in Film Audio for Scene Interpretation", *International Conference on Multimedia and Exposition, IEEE International Conference on Multimedia & Expo 2001 Tokyo, Japan,* pp. 1192-1195, 2001.

[21] Moriyama T., Sakauchi M., "Video Summarisation based on the Psychological Content in the track structure", *ACM Multimedia Workshops 2000,* pp. 191-194, 2000.

[22] Morony M., "Audio Processing to Indexing and Retrieval of Arbitrary TV Video", *Technical report in Dublin City University,* 1998.

[23] Nam J., Alghoniemy M., Tewfik A.H., ``Audio-Visual Content-based Violent Scene Characterization', *In IEEE International Conference on Image Processing, volume 1,* pp. 353--357, 1998.

[24] Oomoto E., Tanaka K., "OVID: Design and Implementations of a Video-Object Database System", *IEEE Trans. On Knowledge and Data Engineering, 5,4,* pp. 629-643, August 1993.

[25] Petkovic M., Jonker W., "A Framework for Video Modeling", *Eighteenth IASTED International Conference Applied Informatics, Innsbruck, Austria,* February 2000.

[26] Pfeiffer S., Lienhart R., and Effelsberg W., "Scene Determination Based on Video and Audio Features", *Multimedia Tools and Applications,15*, pp.59-81, 2001.

[27] Pradhan S., Tajima K., Tanaka K., "A Query Model to Synthesize Answer Intervals from Indexed Video Units", *IEEE Trans. on Knowledge and Data Eng. Vol.13, No.5,* pp. 824-838, Sept./Oct. 2001.

[28] Saraceno C., Leonardi R., "Identification of Story Units in Audio-visual Sequences by Joint Audio and Video Processing", *In IEEE International Conference on Image Processing, Chicago, USA,* pp. 363-367, 1998.

[29] Siegler M. A., Witbrock M. J., Slattery S. T., Seymore K., Jones R. E., Hauptmann A., "Experiments in Spoken Document Retrieval at CMU", *Proceedings of TREC-6, The Sixth Text Retrieval Conference*, 1997.

[30] Smith M., Kanade T**., "**Video Skimming for Quick Browsing Based on Audio and Image Characterization", *Carnegie Mellon technical report CMU-CS-97-111*, February 1997.

[31] Snoek C.G.M., Worring M., "Multimodal Video Indexing: A Review of the State-of-the-art", *Multimedia Tools and Applications*, 2004 (in press).

[32] Srinivasan U., Lindley C., Simpson-Young W.G, "A Multi-model Framework for Video Information Systems", *Database Semantics - Semantic Issues in Multimedia Systems, Kluwer Academic Publishers,* pp 85-108, 1999,

[33] Sundaram H., Chang S., "Video Scene Segmentation using Audio and Video Features*", 1st IEEE International Conf. on Multimedia and Expo. (ICME-2000) NY*, Aug. 2000.

[34] VideoQ, Web Site: http://www.ctr.columbia.edu/VideoQ, July 2004.

[35] Wactlar, H., Witbrock, M., Hauptmann, A., "Informedia News-On-Demand: Using Speech Recognition to Create a Digital Video Library", *CMU Technical Report CMU-CS,* pp.98-109, 1998.

[36] Weiss R., Duda A., Gifford D., "Composition and Search with a Video Algebra", *IEEE Multimedia, Vol. 2, No. 1,* pp. 140-151, 1995.

[37] Woudstra A., Velthausz D. D., Poot H. J. G., Moelaert El-Hadidy F., Jonker, Maurice A., Houtsma W., Heller R. G., Heemskerk J. N. H., "Modeling and Retrieving Audiovisual Information: A Soccer Video Retrieval System", *Multimedia Information Systems,* pp. 161-173, 1998.

[38] Yavuz Ö., "A Video Database Management System Based On Mpeg-7 Standard", *M. Sc. Thesis, Department of Computer Engineering, Middle East Technical University, Ankara, Turkey*, September 2002.

[39] Zhang, H. J., Low, C. Y., Smoliar, S. W., Wu, J. H., "Video Parsing, Retrieval and Browsing: An Integrated and Content-Based Solution", *Proc. ACM Multimedia* '95*, ACM Press*, pp. 15 - 24, 1995.

[40] Zhu Y., Zhou D., **"**Video Browsing and Retrieval Based on Multimodal Integration", *Proceedings. IEEE/WIC International Conference on*, pp. 650-653, 2003.