INTEGRATION OF MULTIMODAL MULTIMEDIA DATABASE SYSTEM ARCHITECTURE WITH QUERY LEVEL FUSION

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

BY

SAEID SATTARI

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

OCTOBER 2013

Approval of the thesis:

INTEGRATION OF MULTIMODAL MULTIMEDIA DATABASE SYSTEM ARCHITECTURE WITH QUERY LEVEL FUSION

Submitted by **SAEID SATTARI** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering**

Prof. Dr. Adnan Yazıcı
Supervisor, **Computer Engineering Dept., METU**

Examining Committee Members:

Assoc. Prof. Dr. Murat Koyuncu
Information Systems Engineering, Atılım University

Prof. Dr. Adnan Yazıcı
Computer Engineering Dept., METU

Asst. Prof. Dr. Mustafa Sert
Computer Engineering Dept., Başkent University

Assoc. Prof. Dr. Ahmet Cosar
Computer Engineering Dept., METU

Assoc. Prof. Dr. Halit Oğuztüzün
Computer Engineering Dept., METU

Date:

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: SAEID SATTARI

Signature :

ABSTRACT

INTEGRATION OF MULTIMODAL MULTIMEDIA DATABASE SYSTEM ARCHITECTURE WITH QUERY LEVEL FUSION

Sattari, Saeid
M.Sc., Department of Computer Engineering
Supervisor: Prof. Dr. Adnan Yazıcı

October 2013, 78 pages

Multimedia data particularly digital videos that contain various modalities (visual, audio, and text) are complex and time consuming to deal with. Therefore, managing large volume of multimedia data reveals the necessity for efficient methods of modeling, processing, storing and retrieving these data. In this study, we investigate some of the requirements to efficiently deal with multimedia data, especially video data. To satisfy such requirements we aim to integrate specific multimedia database architecture which consists of semantic content extractor, storage, index, query and coordinator modules. In addition, to simplify complicated and time consuming operations on video files some client-server based applications with appropriate graphical user interfaces are implemented to carry out these operations in multi-threaded mode. The proposed architecture also supports different query types including the combination of content as well as concept-based queries which provides users with the ability to perform multimodal query. Furthermore, we introduce a fusion approach at the query level to improve query retrieval performance of the multimedia database system.

Keywords: Multimedia database architecture, integration, multimodal query, query level fusion

ÖZ

SORGU SEVİYESİNDE FÜZYON DESTEKLEYEN MULTİMODAL ÇOKLUORTAM VERİTABANI SISTEMİ MİMARİ ENTEGRASYONU

Sattari, Saeid
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Adnan Yazıcı

Ekim 2013, 78 sayfa

Çokluortam verileri, özellikle bir çok modaliteyi (görsel, işitsel ve metin) içeren dijital videolar, karmaşıktır ve bu verilerin üzerinde yapılan işlemler oldukça zaman alıcıdır. Bu nedenle, büyük hacimli çokluortam veriyi yönetmek, modellemek, işlemek, depolamak ve almak için etkin yöntemlerin geliştirilmesi gereklidir. Bu çalışmada, çokluortam verilerini, özellikle video verilerini etkin bir şekilde yönetme gereksinimleri araştırdık. Bu ihtiyaçları karşılamak için anlamsal içerik çıkarıcı, depolama, dizin, sorgu ve koordinatör modüllerinden oluşan bir multimedya veritabanı mimari entegrasyonunu hedefledik. Buna ek olarak, video dosyalarının karmaşık ve zaman alıcı işlemlerini basitleştirmek için, çoklu iş parçacıklı modda çalışan bazı client-server tabanlı uygulamalar uygun grafik arayüzleri ile geliştirilmiştir. Önerilen mimari aynı zamanda içerik kombinasyonu ve kavram tabanlı sorgular gibi farklı sorgu tiplerini destekleyerek kullancıya çoklu modalitede sorgu yapma imkanı sunar. Ayrıca, multimedya veri tabanı sisteminin sorgu getirme performansını artırmak için sorgu düzeyinde bir füzyon yaklaşımı kullandık.

Anahtar Kelimeler: Multimedya veritaban mimari, entegrasyonu, multimodal sorgusu, sorgu seviyesinde füzyon

To my parents

ACKNOWLEDGEMENTS

First of all, I would like to express my sincerest gratitude and appreciation to my supervisor, Prof. Dr. Adnan Yazıcı for his encouragements and supports throughout this study.

I would also like to thank all the members of Multimedia Database Research Group for their technical guidance and supports.

My final appreciation and thanks goes to the Scientific and Technological Research Council of Turkey (TÜBİTAK) for providing the financial means throughout this study under the project number "EEEAG109E014".

At last but not the least, I am wholeheartedly grateful to my parents who have always encouraged and supported me throughout my life, particularly during my graduate study.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

MPEG	Moving Picture Experts Groups
MFCC	Mel Frequency Cepstrum Coefficients
HMM	Hidden Markov Model
JWS	Java Web Start
JNLP	Java Network Launch Protocol
JNI	Java Native Interface
JNA	Java Native Access
JMF	Java Media Framework
SVM	Support Vector Machine
CCA	Canonical Correlation Analysis
KCCA	Kernel Canonical Correlation Analysis
ICD	Incomplete Cholesky Decomposition
QBE	Query by Example
VSM	Vector Space Model
ZCR	Zero Crossing Rate
CBIR	Content Based Information Retrieval

CHAPTER 1

INTRODUCTION

With an increasing amount of multimedia data production favored by cheap digital devices, such as large capacity and fast accessed media storages, people are exposed to a very large volume of multimedia data in daily life. Storing and searching such a large volume of data in a database becomes a real challenge. Usually most of the users are interested in the semantic contents of videos. Consequently, manual annotation of such a large volume of data in order to prepare them for any content-based search is almost impractical. Thus, dealing with this big amount of digital videos requires automatic methods to extract the semantic contents and efficient ways to store, index and retrieve them.

Due to the nature of multimedia data including multidimensionality, multimodality, complexity in automatic concept extraction and spatiotemporal relations of objects inside video data manual annotation and usually conventional database approaches are not as sufficient as expected for managing multimedia content. Therefore, extracting semantic concepts automatically, considering interaction and relation between objects inside multimedia, as well as efficiently storing and indexing them plus handling multimodality in addition to efficiently retrieving them continue to be one of the most challenging and fast-growing research areas which have attracted many researchers. Hence, having a well-defined architecture and an integrated system that covers most of these requirements is quite vital. In general, such an integrated system should have capabilities to:

- Annotate visual objects, events, audio concepts and named entities in text automatically.
- Store and index semantic concepts and contents efficiently.
- Handle multimodality in storage as well as retrieval phase effectively.
- Have some specialized user interfaces and functionalities that provide various query types such as content-based, concept-based as well as multimodal search support.
- Having an acceptable query retrieval performance for the types of queries supported by the multimedia database system.

It should be mentioned that when multimedia data is the matter of argument, mainly three categories of data stand out:

- Metadata, such as creator, name, date and so on.
- High level semantic concepts that we are exposed to.

- The low-level features such as fundamental frequency, harmonicity, dominant color, region shape and edge histogram [1].

Although managing and retrieving textual content (metadata) are easy and straightforward, users are usually interested in the semantic content of the video.

Audio, text, visual objects and spatiotemporal relations among these entities, called *event*, are considered as the basic components of semantic content of the video data [2]. The process of extracting these entities (annotating semantic information) in the domain of multimedia system research is a challenging [3]. When we use manual annotation methods for multimedia data information extracting it is mostly resource consuming, exhausting and slow therefore, it is expensive, limited and inefficient. Thus, an automatic annotation system is required in real life applications. Since raw multimedia data are collections of pixels and signals, automatic extraction of multimedia data's semantic content is complex and hard [4]. Besides, due to the complexity of multimedia data structure, modeling them is not so straightforward.

In addition, the logical and conceptual design of databases determines and limits the supported data types, indexing and query types. Following these restriction, traditional database management systems support primitive types (number, character, date, etc.) and have some predefined indexing functionality on these types. Hence, semantic multimedia contents which have multi-dimensional information could not be represented by such primitive data types and retrieved by limited indexing mechanisms efficiently.

With this increasing quantity of multimedia data, efficient storage and retrieval of specific semantic concept in large data sets is a basic functional requirement. Since multimedia objects contain huge amount of information and exist in long, nested and hierarchical forms, late responses to the query are inevitable. In addition, previous researches have proven that combining the result of each modal either in data level or query level would result in improvement in query retrieval performance. This evidence requires not only managing each modal individually but also in relation and interaction with each other. Therefore, an acceptable multimedia architecture which is supposed to deal with multimedia data must possess a customized storage system with a fast indexing mechanism for retrieving multimedia contents in order to answer various query types. To achieve a reasonable accuracy, efficiency and usability, such a multimedia database system should consider the mentioned requirements during the architecture design and modules' integration phases.

From query level perspective, multimodal data which originated from the same source tend to be correlated [5]. Since the presence of one modality can help understanding certain semantics of other modalities, different modalities can take a complementary role in retrieving multimedia content. However, in most proposed systems that deal with digital videos, different modalities are treated separately with no combination at a higher level [6]. Due to the fact that each modality may compensate for weakness of the other, we can benefit from relations among modalities. For instance, a video retrieval system that exploits both audio and visual modalities may achieve a better performance in both accuracy and efficiency than a system which exploits either one only [7][8]. As a result, multimodal correlation analysis has attracted a growing attention in multimedia content analysis researches in the recent years [6][9].

Most of the recent methods only focus on the positive correlation among objects of multimedia in the same class while the negative correlation between them is underestimated. From the researcher's point of view in this thesis, both kinds of correlation are important because positive correlation provides the co-occurrence information and negative correlation reflects the exclusive information. For example, in a video shot the "Explosion" concept may come from text as well as visual category. Meanwhile, the related section of audio labeled "Gun" may have strong positive correlation with that shot while some concepts in audio like "Silence" has negative correlation with the same shot. As a result, using both positive and negative correlations would be beneficial. Although combining and propagating correlations along with each modality is proven to improve retrieval performance, the semantic correlations that are propagated among modalities throughout the whole dataset are also beneficial. As a result, the achieved correlations on the objects and concepts can naturally meet the requirements of multimodal retrieval challenge in which, the retrieval results are of the similar semantics and can be of different media types.

1.1 Motivation and Contributions

Although various researches exist about multimedia database architecture, most of them focus just on one aspect of multimedia for instance, working only with limited modalities or using manually annotated concepts. Furthermore, response time for query retrieval is usually neglected in such systems. In addition, most multimedia database related studies usually use limited data source to provide response to queries without exploiting fusion at data level or query level.

The main motivation for this thesis is the need for an integrated multimedia database system architecture that not only is enabled to extract concepts automatically but also is capable to store and index them efficiently as well as possessing an ability to handle different query types. In this study, an integrated multimedia database system that supports automatic and manual annotation of objects, events, audio concepts and texts are presented. In addition, this system provides some functionality of conventional database management systems such as indexing and querying data. From a specific point of view, this integrated multimedia database system supports multimodal content-based and concept-based queries. Compared with the existing systems, the major contributions and advantages of the proposed study are as follow:

1. A multimedia database architecture is proposed that supports multimodal aspect of multimedia data. This architecture allows us to handle different modalities for multimedia data such as visual, audio and text.

2. Full implementation of the proposed architecture is presented by integrating various multimedia modules in which a semantic information extractor, a high-dimensional index structure and an object oriented database are connected by coordinator module to build the desired multimedia database architecture. A complete system that supports automatic semantic concept extraction and enriched with a high-dimensional index structure for enhanced and quick retrieval is developed. The system is gradually developed in a component-oriented approach such that each component can be replaced by alternative modules without affecting others.

3. An ontology for football domain which is based on the relations of semantic objects is used. This ontology is utilized in cooperation with the event extraction module to detect events in this domain. This ontology presented as a proof of concept and any desired ontology can be appended to the proposed system later.

4. A client application for query is developed that supports query by concept, query by content and their combination. It also supports multimodal query with logical operations to join modals.

5. We also develop a different client application that automatically extracts and annotates concepts for supported modalities and allows users to manually manipulate them. Thin client technology is used in order to support the distributed nature of multimedia data processing.

6. We propose a query level fusion that exploits correlation in each modality and among them. We employ CCA and KCCA to catch linear and non-linear relations between modalities which aim at capturing some term to term correlations by looking at co-occurrence information. Then we evaluate them with General Vector Space and Boolean retrieval models which provide improvement in retrieval performance. By combining proposed query level fusion with data fusion introduced in [10], we observe even further improvement in multimodal query retrieval performance.

1.2 Thesis Outline

The rest of the thesis is organized as follows: in Chapter 2, previous works and studies regarding to our work’s topic are explained. Information about technologies and tools that are exploited in this work are provided in Chapter 3. In Chapter 4, proposed architecture and the recommended semantic data model for this architecture are described in details. This chapter also includes steps for online and automatic concept extraction from a given sample video. In Chapter 5, query retrieval models are discussed and supported query types are described. Detail information about query level fusion and results are covered in Chapter 6. The conclusion and recommend future work are provided in last chapter.

CHAPTER 2

RELATED WORK

In this section of the thesis, a brief review of the approaches that were suggested in the past, in order to provide solutions to the multimedia database architecture integration and query level fusion is presented.

With a fast technological improvement in cheap and large capacity storage users need to efficiently search among these mass volume of multimedia data. Hence, multimedia information retrieval necessity is fulfilled when some semantic concepts as well as example multimedia objects can be efficiently searched within multimedia data [11].

The multimedia objects search is facing a well-known problem that is defined as semantic gap [12][13]. The process of querying the multimedia data is complex and depending on not only what information can be retrieved but also how this information can be linked logically to low-level features to reduce the semantic gap efficiently.

A framework for multimedia information retrieval was proposed in [14] that utilizes matrix-based mathematical models for content modeling. In [15] a Framework for querying multimedia data was proposed which was named as "Visual Information Retrieval (VizIR)". Another framework for multimedia information retrieval was proposed in [16] that recommends a uniform solution for structuring of multimedia data as well as supporting automatic, semi-automatic and manual annotation. The mentioned frameworks lack architectural approaches that provide modular methods for integration of multimedia information retrieval architectures. They are also defined at abstract levels and as a result, system development using them is complicated. "Informedia" [17], "Combinformation" [18], "greenstone" [19], "M-Space Browser" [20][21] and "EVIADA" [22] also were proposed as non web-based multimedia information retrieval systems. They mostly provide searching within particular modal of multimedia data [23]. Another multimedia architecture that provides users with interfaces to search multimedia data via fuzzy queries from single modality was proposed in [3].

Since searching inside one modal of multimedia data can not compensate the requirements of the retrieval of multimedia data [11], searching within multiple modalities or multiple knowledge sources like audio, video and text is almost vital.

Designing the mixture approaches for multimodal retrieval gain great importance lately in developing effective multimedia systems [24]. This issue has created an important challenge for researchers, as pointed out in [25]:

“To deal effectively with multimedia retrieval, one must be able to handle multiple query and document modalities. In video, for example, moving images, speech, music audio and text (closed captions) can all contribute to effective retrieval. Integrating the different modalities in principled ways is a challenge.”

The problem of multimedia modal source combination has been actively investigated in recent years. Westerveld et al. [26] demonstrated how the combination of different modalities can influence the performance of video retrieval. They utilized a model inspired by language modeling approach and a probabilistic approach for image retrieval to rank the video shots. Their final results were obtained by sorting the joint probabilities of all modalities. The video retrieval system proposed by Amir et al. [27] applied a query-dependent combination model that the weights are defined based on user experiences. They also utilized a query-independent linear combination model to merge the text/image retrieval systems where the per-modality weights are chosen to maximize the mean average precision (MAP) score on potential results. Gaughan et al. [28] ranked the video shots based on the summation of feature weights and automatic speech retrieval scores, where the effect of speech retrieval is higher than any other features. Rautiainen et al. [29] used a user-dependant approach to combine the results from text search and visual search. In their system the combinations’ scores are predefined by users when the query is submitted. The QBIC system [30] combines scores from different image retrieval system using linear combination. Gulen also proposed a multiple knowledge sources combination in data level via late fusion technique in [10].

However, until recently most of the multimedia retrieval systems used query independent approaches that combine multiple knowledge sources. This has greatly limited their flexibilities and performance in the retrieval process [31]. Instead, it is more desirable to design a better combination method which can take query information into account without asking information from user. Recently, query class based combination approaches [32][33] were proposed as a practical alternative for the query independent combination which begins with classifying the queries into predefined query classes and then applies the corresponding combination scores for knowledge source combination. Experimental evaluations have demonstrated the effectiveness of these ideas which have been applied in the best-performing systems of well-known datasets [34]. Although the validity of using query-class dependent weights has been confirmed by many later studies [33][35][36][37], defining classes for queries is still a challenging issue. For example, Yuan et al. [36] classified the query space into person and non-person queries in their multimedia retrieval system. To improve the manually defined query classes, Kennedy et al. [37] recently proposed a data-driven learning approach to automatically discover the query-class-dependent weights from training data by means of grouping the queries in a joint semantic space via clustering techniques such as k-means and hierarchical clustering.

A more recent work [38] unified query class categorization and combination weight optimization in a single probabilistic framework by treating query classes as latent variables.

Some of recent studies have also proven that using a latent relation between modalities increased the performance of query retrieval in multimodal systems [39][40][41]. These studies can be classified into two categories [42]: multimodal correlation analysis and cross-media index. Multi-modal correlation analysis [43][44] approach explore statistic correlations between modalities by analyzing their co-occurrence relationship. For instance, after extracting visual and audio features, correlation can be analyzed between their feature matrices to learn their correlations [45] and then apply a hierarchical manifold space to make the correlations more accurate [46]. However, difficulties still exist due to the heterogeneous feature space of visual or audio modalities. Unlike multi-modal correlation analysis methods, cross-media index approach focuses on automatically labeling un-annotated multimedia data using textual models [47][7]. These methods first represent a visual or audio feature cluster with the dictionary index and then construct a linked representation to obtain shots' text (or audio-text) translation results. Despite its success, this approach suffers from several weaknesses. First, representing each local visual or audio feature by a dictionary index can result in severe loss of information. Second, cross-media index actually focuses on the annotation problem and ignores semantics correlation among multi-modal data in query retrieval.

Canonical Correlation Analysis (CCA) is one of the methods of correlating linear relationships between two multidimensional variables. CCA can be seen as using complex labels as a way of guiding feature selection towards the underlying semantics. CCA makes use of two views of the same semantic object to extract the representation of the semantics. The main difference between CCA and the other methods for correlating is that CCA is closely related to mutual information in different sets [48]. Hence CCA can be easily exploited in information retrieval tasks and is our selection in this work.

CCA which was proposed by Hotelling [49] can be seen as the problem of finding basis vectors for two sets of variables such that the correlation between the projections of the variables onto these basis vectors are mutually maximized. In an attempt to increase the flexibility of the feature selection, kernelisation of CCA (KCCA) has been applied to map the data to a higher-dimensional feature space. KCCA has been applied in some preliminary work by Fyfe & Lai [50], Akaho [51] and recently by Vinokourov et al. [52] with improved results. Finally, after investigating previous studies our conclusion leads us to take advantage of CCA and KCCA to analyze the correlation among modals.

CHAPTER 3

BACKGROUND

3.1 MPEG-7

3.1.1 Introduction

Due to Internet's popularity, the last decade has experienced a quick propagation of digital audio-visual information. Though the increasing availability of potentially interesting information has enriched our lives, the overwhelming amount of information also raises fundamental questions and problems:

How fast and easy can desirable information be available? The more interesting material is available, the harder it is to locate. A noticeable indicator of the existing tension between humans and the vast amounts of information lies in the popularity of search engines available on the web. Unfortunately, current solutions let users only search for textual information. Identifying audio-visual information proved to be difficult as no generally recognized description of this material exists. For example it's not possible to efficiently search the web for: a video of the car in accident or shots where a tennis player broke her racket as well as "all video according to given video sample." We can consider similar examples for audio, in which we prepare audio sample and look for similar shots.

It's true that in specific cases, solutions exist. Multimedia databases on the market today let users search for pictures using characteristics like color, texture, and information about the shape of objects in the picture. Furthermore, the question of identifying content is not restricted to database retrieval applications and applies equally to other areas. For instance, we can imagine world with more than 1000 broadcast television channels, which will of course make it harder to select and search a potentially interesting channel. Domains other than search also include image understanding (surveillance, intelligent vision, smart cameras, and so on) or media conversion such as speech to text, picture to speech, visual transcoding, and so on.

In October 1996, the Moving Pictures Expert Group (MPEG) started a new project to provide a solution to the questions described above. The newest member of the MPEG family, called the multimedia content description interface (MPEG-7), extends the limited capabilities of proprietary solutions in identifying content that exists today, notably by including more data types. In other words, MPEG-7 aims to standardize a core set of quantitative measures of audio-visual features, called Descriptors (D), and structures of descriptors and their relationships, called Description Schemes (DS) in MPEG-7. MPEG-7 also standardizes a language - Description Definition Language (DDL) - that specifies Description Schemes to ensure flexibility for wide agreement and durability. We can index and search for audio/visual material that has MPEG-7 data associated with it. This material may include still pictures, graphics, 3D models, audio, speech, video, and information about how these elements combine in a multimedia presentation (for example, scenarios or composition information). We expect the standard core set of MPEG-7 functionality would facilitate those classes of applications that have widespread use and will provide interoperability [53].

3.1.2 Multimedia Content Description

MPEG-7's most important aim is to provide a set of methods and tools for the different classes of multimedia content description. When we mention description classes, we actually mean different possible aspects that a description of audio-visual content might cover. A key concept to remember is that many different methods exist to describe any entity depending on how it will be used. Thus, MPEG7 must accommodate these several ways and make them complementary rather than mutually exclusive. Four fundamental description classes relate to the data or any kind of material to be described which are as follows: Transcriptive, physical, perceptual, and medium-based descriptions that represent largely independent views of the data. On top of these schemes an architectural description resides that provides relationships between large sections of the data and relationships between and within the description(s) below it. The annotative description, a part for human annotation and other sorts of commentary on the data itself, sits on top of all the layers and touches each of them. Most likely, any real-life description for use in MPEG-7 applications would employ only one or two of these classes. We now discuss in details the different possible types of description that may exist.

3.1.3 Scope of MPEG-7

MPEG–7 focuses on the standardization of a common interface for describing multimedia data and representing information about the content, not the content itself. The scope is to define the representation of the features, related to audio/video content. Any application dependent issue is outside of its scope. Therefore, neither feature extraction nor query and retrieval process is in the scope of MPEG7. However, because of some interoperability issues it also specifies extraction process at some degree. To summarize, main goal is to make audiovisual data searchable similar to text.

3.1.4 MPEG–7 Parts and Descriptors

The MPEG-7 Standard consists of the following parts [53]:

- MPEG-7 Systems: the tools needed to prepare MPEG-7 descriptions for efficient ship and storage.

- MPEG-7 Description Definition Language: the language for defining the syntax of the MPEG-7 Description Tools and for defining new Description Schemes.
- MPEG-7 Visual: the Description Tools dealing with (only) Visual descriptions.
- MPEG-7 Audio: the Description Tools dealing with (only) Audio descriptions.
- MPEG-7 Multimedia Description Schemes: the Description Tools dealing with generic features and multimedia descriptions.
- MPEG-7 Reference Software: a software implementation of relevant parts of the MPEG-7 Standard with normative status.
- MPEG-7 Conformance Testing: guidelines and procedures for testing conformance of MPEG-7 implementations.
- MPEG-7 Extraction and use of descriptions: informative material about the extraction and use of some of the Description Tools.
- MPEG-7 Profiles and levels: provides guidelines and standard profiles.
- MPEG-7 Schema Definition: specifies the schema using the Description Definition Language.

MPEG-7 provides tools and structures for describing both visual and audio content. MPEG-7 standards overview documentation [53] gives detailed information about all these descriptors. Although only a few of them are used in the implementation of the proposed architecture, any descriptor can easily be integrated into the system. In the following part, descriptors used in the implementation of prototype application are briefly explained.

- **Color Layout:** Among seven color descriptors, color layout represents the spatial distribution of colors of an image in the frequency domain in a very compact form. This compactness allows it to be used in index structures with small computational costs. Besides, it also provides high-speed image-to-image and sequence-to-sequence matching which requires so many similarity calculations. Since it captures the layout information of colors, this descriptor allows very friendly user interface using hand-written sketch queries. No dependency on image/video format, resolution and bit-depths is advantage of this descriptor. It can be applied to whole image and even to any unconnected parts of an image with arbitrary shapes.

- **Dominant Color:** A small number of representative colors (up to 8) are enough to characterize the color information of an image or a specific region. Such compactness makes this descriptor a good candidate for index structures. Therefore, this descriptor is most suitable for representing color information of objects. To extract a few representative colors, color quantization is used and the percentage of each quantized color is calculated correspondingly. A spatial coherency on the entire descriptor is also defined, and is used in similarity retrieval.

- **Region Shape:** By capturing all the pixel distribution of a shape/region, this descriptor can be used in describing shapes. Not only simple ones but also complex shapes with multiple regions, possibly the ones with holes, can be described. Its small size, fast extraction time and low order of computational complexities for matching ability make this descriptor suitable for shape tracking in images and videos.

- **Edge Histogram:** This descriptor represents the spatial distribution of five types of edges in an image; four directional edges (vertical, horizontal, 45° diagonal, 135° diagonal), and one non-directional edge (isotropic). Since edges play an imprtant role in object detection, this descriptor can be useful for image-to-image matching (by example or by sketch). When it is used in conjunction with other descriptors, such as color and shape descriptors, it may significantly improve the retrieval performance. Due to low computational cost, it is suitable for CBIR or retrieval systems based on textures.

3.1.5 MPEG-7 Reference Software (eXperimental Model)

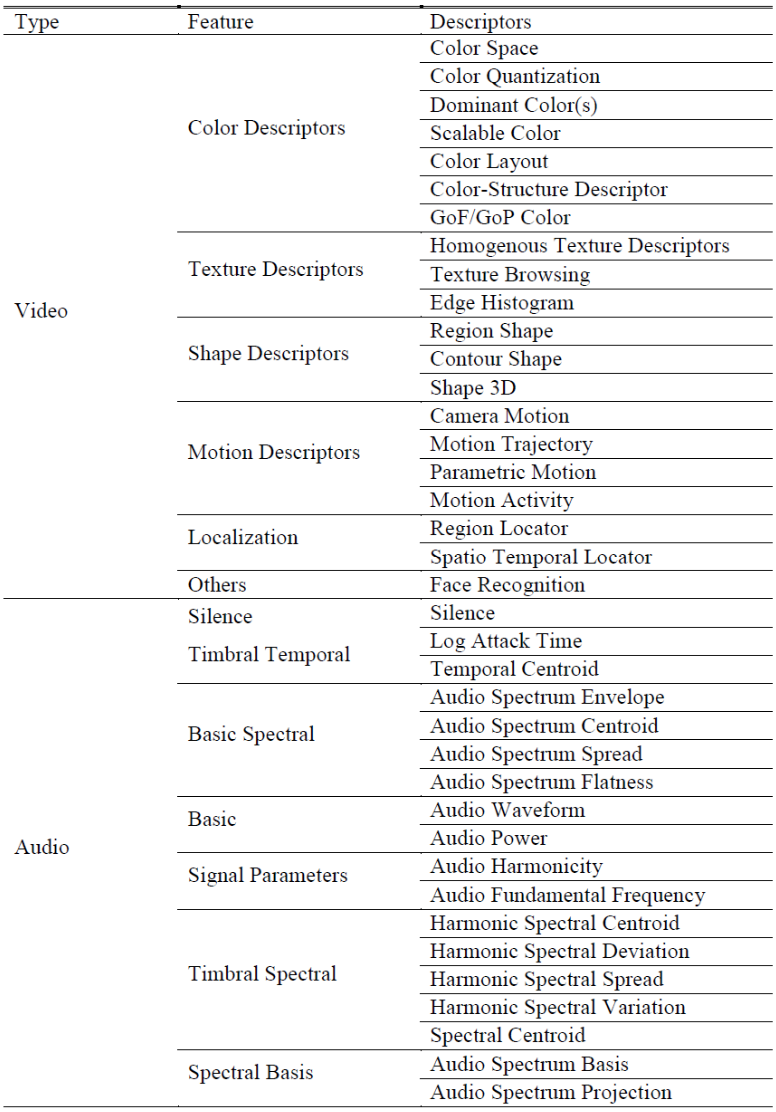
MPEG-7 reference software (eXperimentation Model, shortly XM) is a tool which has ability to extract low level information from video data, using MPEG-7 descriptors. It generates specified MPEG-7 bit streams or DDL streams. Most of the Descriptors and Description Schemes are implemented in XM software. After loading data, the software extracts low-level features and after encoding descriptions it produces a file containing low-level information [53].

XM software can also be used for distance calculations of similar data. After building a database containing extracted low-level information, the tool has the ability to calculate distance values between each data in the database and the given one. MPEG-7 reference software is used both in concept extraction module, index mechanism and QBE retrieval in this study. While, annotation module utilizes this tool for obtaining low-level features and determining distance values at classification step, index mechanism uses extracted features for building index structure as well as calculating distance of objects.

3.1.6 Audio Features

Audio features are basically some values containing meaningful information extracted from audio signals in order to compare and classify audio data. After the extraction of such information, it is stored in a content description in a compact way. A data descriptor is generally called a feature vector and the process for extracting such feature vectors from audio is called feature extraction. Audio feature extraction is generally based on audio analysis of spectral energy distribution, harmonic ratio or fundamental frequency of the audio signal [54].

Table 3-1 MPEG-7 descriptor list



Type	Feature	Descriptors
Video	Color Descriptors	Color Space
		Color Quantization
		Dominant Color(s)
		Scalable Color
		Color Layout
		Color-Structure Descriptor
		GoF/GoP Color
	Texture Descriptors	Homogenous Texture Descriptors
		Texture Browsing
		Edge Histogram
	Shape Descriptors	Region Shape
		Contour Shape
		Shape 3D
	Motion Descriptors	Camera Motion
		Motion Trajectory
		Parametric Motion
		Motion Activity
	Localization	Region Locator
		Spatio Temporal Locator
	Others	Face Recognition
Audio	Silence	Silence
	Timbral Temporal	Log Attack Time
		Temporal Centroid
	Basic Spectral	Audio Spectrum Envelope
		Audio Spectrum Centroid
		Audio Spectrum Spread
		Audio Spectrum Flatness
	Basic	Audio Waveform
		Audio Power
	Signal Parameters	Audio Harmonicity
		Audio Fundamental Frequency
	Timbral Spectral	Harmonic Spectral Centroid
		Harmonic Spectral Deviation
		Harmonic Spectral Spread
		Harmonic Spectral Variation
		Spectral Centroid
	Spectral Basis	Audio Spectrum Basis
		Audio Spectrum Projection

3.1.7 MPEG-7 Audio Features

MPEG-7 standard is a widely used standard in audio classification area. It provides a large set of audio tools to create descriptions. MPEG-7 standard provides the following main elements [55]:

- Descriptors (D) define semantics and syntax of audio feature vectors.

- Description Schemes (DSc) define the semantics and syntax of the relationships between the components of descriptor.
- Description Definition Language (DLL) defines the syntax of description tools.

The focus of architecture proposed is the Descriptors in which semantic of feature vectors are defined. They are low-level audio descriptors containing temporal and spectral descriptors. These descriptors are classified into basic, basic spectral, single parameter, timbral temporal, timbral spectral and spectral basis descriptors [55] as listed in Figure 3-1.

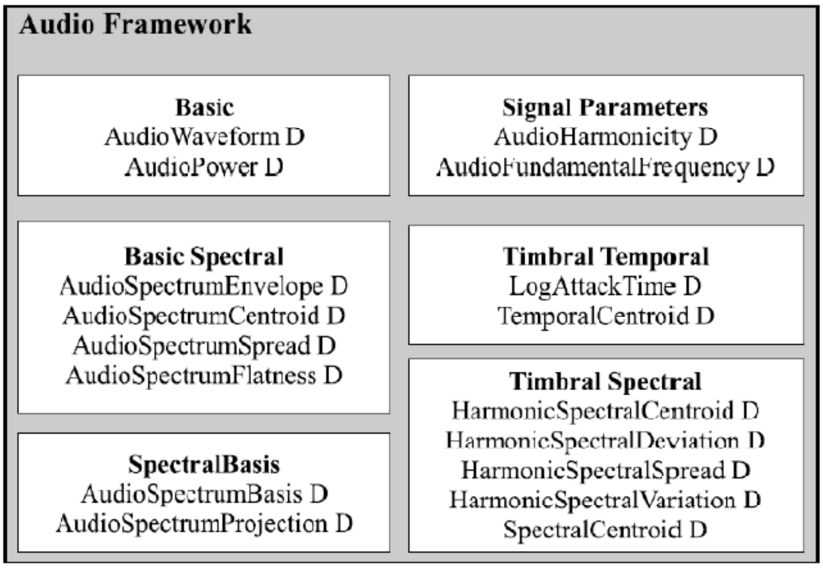


Figure 3-1 Overview of MPEG-7 audio framework

3.2 Object Oriented Database Management Systems

In object oriented (OO) programming paradigm, usually storing and accessing objects are the bottleneck of the system. Furthermore, developing with an OO language, if we use relational database management systems (RDBMS), can result in complicated and difficult-to-maintain code [56].

Most of the time, we have to write object-to-relational mapping code for storing objects in RDBMS (i.e. Spring). Similarly, when an object is to be retrieved from relational database, since objects and their properties are usually stored in a normalized form and distributed in various fields, a group of time consuming retrieve and assemble functions should be executed. Also, when dynamic class structures are used as in agile development environment, for each minor modification we may have to change the schema and alter some queries to handle schema change. The aim of object oriented database management systems (OODBMS) is to handle deficiencies of RDBMSs in object handling approach.

Although some object-relational database systems are offered for handling objects in relational approach recently, they could not reach the compactness and convenient usage of OODMSs. Since objects exist as whole entities in database, storing, retrieving and accessing to them, can be executed with single calls in OODBMSs, even they have compound structures or parent-child hierarchies.

"*Using tables to store objects is like driving your car home and then disassembling it to put it in the garage. It can be assembled again in the morning, but one eventually asks whether this is the most efficient way to park a car.*" [57]

3.2.1 DB4O

DB4O is an open-source object-oriented database, providing a strong integration with object-oriented programming languages, like Java and .Net. As in OODBMSs, it eliminates the translation code which most OO developers should deal with.

It provides high performance, cross platform, simple and easy-to-manage store and access environment. By using DB4O, there is no need to design an additional database schema since the class model becomes the database schema of application [57].

3.3 Large Scale Concept Ontology for Multimedia (LSCOM)

The Large-Scale Concept Ontology for Multimedia (LSCOM) project was a series of workshops held from April 2004 to September 2006 for the purpose of defining a standard and formal vocabulary for the annotation and retrieval of video [58].

3.3.1 Project Description

The LSCOM workshop [59] has developed an expanded multimedia concept lexicon of more than 2000 concepts which slightly over 400 of them have been annotated in 80 hours of video. Concepts related to events, objects, locations, people, and programs have been selected following a multi-step process involving input solicitation, expert critiquing, comparison with related ontologies, and performance evaluation. Participants of the process include representatives from intelligence community users, ontology specialists, and multimedia analytics researchers. In addition, each concept has been assessed according to some criteria, such as utility (usefulness), observability (by humans), and feasibility (by automatic detection). An annotation process was completed in late 2005 by student annotators at Columbia University and CMU over the entire development set of TRECVID 2005 videos. Human subjects judge the presence or absence of each concept in the key frame of each shot, resulting in a total of 61901 labels for each concept.

The first version of the LSCOM annotations consist of keyframe-based labels for 449 visual concepts, out of the 834 initial selected concepts, over the entire TRECVID 2005 development set (61901 shots).

The LSCOM-Lite annotations include 39 high-level features (concepts) which are results from the effort in developing a Large-Scale Concept Ontology for Multimedia (LSCOM). Most of the concepts in LSCOM-Lite overlap with the concepts in LSCOM however, some concepts in LSCOM-Lite are not in LSCOM. The concepts were selected based on semi-automatic mapping of 26377 noun search terms from BBC query logs in late 1998 to WordNet senses, division of semantic concept space into a small number of orthogonal dimensions, and evaluation of 2003 and 2004 TRECVID search topics. The dimensions consist of program category, setting/scene/site, people, object, activity, event, and graphics. A collaborative effort among participants in the TRECVID 2005 benchmark was completed in the summer of 2005 to produce annotations of the 39 concepts over the entire development set of TRECVID 2005 videos. Ten of the LSCOM-Lite concepts have been chosen for evaluation in the TRECVID 2005 high-level feature detection task and 20 LSCOM-Lite concepts were evaluated at TRECVID 2006.

The Revised Event/Activity annotations were conducted on 24 concepts, which contained a temporal component. These concepts were originally annotated in the LSCOM v1.0 release using single keyframes for each shot. Since some concepts require motion, this approach gives unreliable results, so this subset of concepts was re-annotated by having human subjects watch the actual video clips, instead of just viewing single KeyFrame [58].

3.3.2 Use of LSCOM in Larger Research Community

Since its release, LSCOM has begun to be used successfully in visual recognition research: Apart from research done by LSCOM project participants, it has been used by independent research in concept extraction from images, and has served as the basis for a video annotation tool.

We used a subset of the LSCOM concepts to manually annotate a data set for training and testing in this work. The detail process is described in upcoming chapters.

3.4 Java Media Framework (JMF)

The Java Media Framework (JMF) is a Java library that enables audio, video and other time-based media to be added to Java applications and applets. This optional package, which can capture, play, stream, and transcode multiple media formats, extends the Java Platform, Standard Edition (Java SE) and allows development of cross-platform multimedia applications.

An initial, playback-only version of JMF was developed by Sun Microsystems, Silicon Graphics, and Intel, and released as JMF 1.0 in 1997. JMF 2.0, developed by Sun and IBM, came out in 1999 and added capture, streaming, pluggable codecs, and transcoding. JMF is branded as part of Sun's "Desktop" technology of J2SE opposed to the Java server-side and client-side application frameworks. The notable exceptions are Java applets and Java Web Start, which have access to the full JMF in the web browsers or applet viewers underlying JRE.

JMF 2.0 originally shipped with an MP3 decoder and encoder. This was removed in 2002, and a new MP3 playback-only plug-in was posted in 2004. JMF binaries are available under a custom license and the source is available under the SCSL.

The current version ships with four JAR files and shell scripts to launch four JMF-based applications which are:

- **JMStudio:** A simple player GUI.
- **JMFRegistry:** A GUI for managing the JMF "registry," which manages preferences, plug-ins, etc.
- **JMFCustomizer:** Used for creating a JAR file that contains only the classes needed by a specific JMF application and allows developers to ship a smaller application.
- **JMFInit:** Modules and steps to initialize a player.

SJMF (performance pack) is available in an all-Java version and as platform-specific "performance packs" which can contain native-code players for the platform, and/or hooks into a multimedia engine specific to that platform. JMF 2.0 offers performance packs for Linux, Solaris (on SPARC) and Windows.

In January 2011, Tudor Holton, a member of Bentokit Project, released a Debian package for the JMF to alleviate difficulties that had arisen over time when installing the JMF on Debian and Ubuntu GNU/Linux. This package does not contain the JMF, but presents the user with the JMF License, retrieves it from the Oracle website, and then installs it. A similar Debian package installer for the JMF MP3 Plugin was also built in February 2011 [60].

3.4.1 Design Concepts

JMF abstracts the media it works with into *DataSources* (for media being read into JMF) and *DataSinks* (for data being exported out). It does not afford the developer significant access to the particulars of any given format; rather, media is represented as sources (themselves obtained from URL's) that can be read in, played, processed and exported (though not all codecs support processing and transcoding).

A *Manager* class offers static methods that are the primary point-of-contact with JMF for applications. In this work we use JMF to load and play audio and video files.

3.5 Java Web Start Technology (JWS)

Java Web Start is a helper application that gets associated with a Web browser. When a user clicks on a link that points to a special launch file (JNLP file), it causes the browser to launch Java Web Start which then automatically downloads, caches, and runs the given Java-based application. The entire process is typically completed without requiring any user interaction except for the initial single click [61].

From a technology standpoint, Java Web Start has a number of key benefits that make it an attractive platform to use for deploying applications [62]:

- Java Web Start is built exclusively to launch applications written to the Java 2 SE platform. Thus, a single application can be made available on a Web server and then deployed on a wide variety of platforms, including Windows 98/NT/2000/XP/7, Linux, and the Solaris Operating Environment. The Java platform has proven to be a very robust, productive, and expressive development platform, leading to a significant cost savings due to minimized development and testing costs.

- Java Web Start supports multiple revisions of the Java platform. Thus, an application can request a particular version of the platform it requires, such as *J2SE* 7. Several applications can run at the same time on different platform revisions without causing conflicts and Java Web Start can automatically download and install a revision of the platform if an application requests a version that is not installed on the client system.

- Java Web Start allows applications to be launched independently of a Web browser. This can be used for off-line operation of an application, where launching through the browser is often inconvenient or impossible. The application can also be launched through desktop shortcuts, making launching the Web-deployed application similar to launching a native application.

- Java Web Start takes advantage of the inherent security of the Java Platform. Applications are by default run in a protective environment (sandbox) with restricted access to local disk and network resources. It allows the user to safely run applications from sources that are not trusted.

- Applications launched with Java Web Start are cached locally. Thus, an already-downloaded application is launched similar with a traditionally installed application.

3.5.1 Java Network Launch Protocol (JNLP)

The technology underlying Java Web Start is the Java Network Launching Protocol & API (JNLP). This technology was developed via the Java Community Process (JCP). Java Web Start is the reference implementation (RI) for the JNLP specification. The JNLP technology defines, among other things, a standard file format that describes how to launch an application called a JNLP file [63].

The JNLP enables an application to be launched on a client desktop by using resources that are hosted on a remote web server. Java Plug-in software and Java Web Start software are considered JNLP clients because they can launch remotely hosted applets and applications on a client desktop.

Recent improvements in deployment technologies enable us to launch rich Internet applications (RIAs) by using JNLP. Both applets and Java Web Start applications can be launched by using this protocol. RIAs that are launched by using JNLP also have access to JNLP APIs. These JNLP APIs allow the RIAs to access the client desktop with the user's permission.

JNLP is enabled by a RIA's JNLP file. The JNLP file describes the RIA. The JNLP file specifies the name of the main JAR file, the version of Java Runtime Environment software that is required to run the RIA, name and display information, optional packages, runtime parameters, system properties, and so on [64]. In this work, we employ this technology to fully exploit mentioned benefits of the JNLP and JWS.

3.6 Java Native Interface (JNI)

The Java Native Interface is a programming framework that enables Java code running in a Java Virtual Machine (JVM) to call as well as be called by native applications (programs specific to a hardware and operating system platform) and libraries written in other languages such as C, C++ and assembly.

JNI enables us to write native methods to handle situations when an application cannot be written entirely in the Java programming language, e.g. when the standard Java class library does not support the platform-specific features or program library. It is also used to modify an existing application written in another programming language to be accessible to Java applications. Many of the standard library classes depend on JNI to provide functionality to the developer and the user such as file I/O. Including performance and platform sensitive API implementations in the standard library, JNI allows all Java applications to access this functionality in a safe and platform-independent manner [65].

The JNI framework lets a native method use Java objects in the same way that Java code uses these objects. A native method can create Java objects and then inspect and use these objects to perform its tasks. A native method can also inspect and use objects created by Java application code.

JNI is sometimes referred to as the "escape hatch" for Java developers because it enables them to add functionality to their Java application that the standard Java APIs cannot otherwise provide. It can be used to interface with code written in other languages, such as C and C++. It is also used for time-critical calculations or operations like solving complicated mathematical equations, because native code may be faster than JVM code [66]. We utilize JNI to access XM Software in order to extract MPEG-7 low-level features.

3.7 Java Native Access (JNA)

Java Native Access provides Java programs easy access to native shared libraries without using the Java Native Interface. JNA's design aims to provide native access in a natural way with a minimum of effort [65].

The JNA library uses a small native library called foreign function interface library (libffi) to dynamically invoke native code. The JNA library uses native functions allowing code to load a library by name and retrieve a pointer to a function within that library, and uses libffi library to invoke it, all without static bindings, header files, or any compile phase. The developers use Java interfaces to describe functions and structures in the target native library. This makes it quite easy to take advantage of native platform features without incurring the high development overhead of configuring and building JNI code [67].

JNA is built and tested on Mac OS, AIX, Microsoft Windows, Solaris, FreeBSD, OpenBSD, Linux, X, Windows Mobile, and Android. It is also possible to recompile the native build configurations to make it work on most other platforms that run Java. For segmenting key-frames we utilize JNA to exploit previously developed libraries of JSEG.

3.8 Servlet Technology

A Servlet is a Java programming language class used to extend the capabilities of a server. Although Servlets can respond to any types of requests, they are commonly used to extend the applications hosted by web servers, so they can be thought of as Java Applets that run on servers instead of in web browsers. These kinds of Servlets are the Java counterpart to non-Java dynamic Web content technologies such as PHP and ASP.NET.

A Servlet is a Java-based server-side web technology. Technically speaking, a Servlet is a Java class in Java EE that conforms to the Java Servlet API, a protocol by which a Java class may respond to requests. Servlets could in principle communicate over any client–server protocol. A software developer may use a servlet to add dynamic content to a web server using the Java platform. The generated content is commonly HTML, but may be other data such as XML. Servlets can maintain states in session variables across many server transactions by using HTTP cookies, or URL rewriting.

To deploy and run a Servlet, a web container must be used. A web container (also known as a Servlet container) is essentially the component of a web server that interacts with the Servlets. The web container is responsible for managing the lifecycle of Servlets, mapping an URL to a particular Servlet and ensuring that the URL requester has the correct access rights [68].

The Servlet API, contained in the Java package hierarchy *javax.servlet* defines the expected interactions of the web container and a servlet.

A Servlet is an object that receives a request and generates a response based on that request. The basic Servlet package defines Java objects to represent Servlet requests and responses, as well as objects to reflect the Servlet's configuration parameters and execution environment.

The package *javax.servlet.http* defines HTTP-specific subclasses of the generic servlet elements including session management objects that track multiple requests and responses between the web server and a client. Servlets may be packaged in a WAR file as a web application.

Servlets can be generated automatically from Java Server Pages (JSP) by the *Java Server Pages* compiler. The difference between Servlets and JSP is that Servlets typically embed HTML inside Java code, while JSPs embed Java code in HTML. While the direct usage of Servlets to generate HTML has become rare, the higher level MVC (Model View Controller) web framework in Java EE (JSF) still explicitly uses the Servlet technology for the low level request/response handling via the *FacesServlet*. A somewhat older usage is to use Servlets in conjunction with JSPs in a pattern called "Model 2", which is a flavor of the model–view–controller pattern.

The Servlet specification was created by Sun Microsystems, with version 1.0 finalized in June 1997. Starting with version 2.3, the Servlet specification was developed under the Java Community Process. JSR 53 defined both the Servlet 2.3 and *JavaServer* Page 1.2 specifications. JSR 154 specifies the Servlet 2.4 and 2.5 specifications. As of March 26, 2010, the current version of the Servlet specification is 3.0.

The advantages of using Servlets are their fast performance and ease of use combined with more power over traditional CGI (Common Gateway Interface). Traditional CGI scripts written in Java have a number of disadvantages when it comes to performance [69]:

- When an HTTP request is made, a new process is created for each call of the CGI script. This overhead of process creation can be very system-intensive, especially when the script does relatively fast operations. Thus, process creation will take more time than CGI script execution. Java Servlets solve this, as a Servlet is not a separate process. Each request to be handled by a Servlet is handled by a separate Java thread within the web server process, omitting separate process forking by the HTTP daemon.

- Simultaneous CGI request causes the CGI script to be copied and loaded into memory as many times as there are requests. However, with Servlets, there are the same amounts of threads as requests, but there will only be one copy of the Servlet class created in memory that stays there also between requests.

- Only a single instance answers all requests concurrently. This reduces memory usage and makes the management of persistent data easy.

- A Servlet can be run by a Servlet container in a restrictive environment, called a sandbox. This is similar to an applet that runs in the sandbox of the web browser. This makes a restrictive use of potentially harmful Servlets possible.

3.8.1 Life Cycle of a Servlet

During initialization stage of the Servlet life cycle, the web container initializes the Servlet instance by calling the *init()* method, passing an object implementing the interface. This configuration object allows the Servlet to access *name-value* initialization parameters from the web application [69].

After initialization, the Servlet can service client requests. Each request is serviced in its own separate thread. The web container calls the *service()* method of the Servlet for every request. The *service()* method determines the kind of request being made and dispatches it to an appropriate method to handle the request. The developer of the Servlet must provide an implementation for these methods. If a request is made for a method that is not implemented by the Servlet, the method of the parent class is called, typically resulting in an error being returned to the requester.

Finally, the web container calls the *destroy()* method that takes the Servlet out of service. The *destroy()* method, like *init()*, is called only once in the lifecycle of a Servlet.

Therefore, three methods are central to the life cycle of a Servlet. These are *init()*, *service()*, and *destroy()*. They are implemented by every Servlet and are invoked at specific times by the server. The following is a typical user scenario of these methods [69].

1. Assume that a user requests to visit an URL.
   - The browser then generates an HTTP request for this URL.
   - This request is then sent to the appropriate server.

2. The HTTP request is received by the web server and forwarded to the Servlet container.

- The Servlet container maps this request to a particular Servlet.
- The Servlet is dynamically retrieved and loaded into the address space of the Servlet container.

3. The Servlet container invokes the *init()* method of the Servlet.
    - This method is invoked only when the Servlet is first loaded into memory.
    - It is possible to pass initialization parameters to the Servlet so that it may configure itself.
4. The Servlet container invokes the *service()* method of the Servlet.
    - This method is called to process the HTTP request.
    - It is possible for the Servlet to read data that has been provided in the HTTP request.
    - The Servlet may also formulate an HTTP response for the client.
5. The Servlet remains in the Servlet container's address space and is available to process any other HTTP requests received from clients.
    - The *service()* method is called for each HTTP request.
6. The Servlet container may, at some point, decide to unload the Servlet from its memory.
    - The algorithms by which this decision is made are specific to each Servlet container.
7. The Servlet container calls the Servlet's *destroy()* method to relinquish any resources such as file handles that are allocated for the Servlet; important data may be saved to a persistent store.
8. The memory allocated for the Servlet and its objects can then be garbage collected.

3.9 Servlet Container

Servlet container (also known as a Web container) is the component of a web server that interacts with Java Servlets. A web container is responsible for managing the lifecycle of Servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights.

A web container implements the web component contract of the Java EE architecture, specifying a runtime environment for web components that includes security, concurrency, lifecycle management, transaction, deployment, and other services. A web container provides the same services as a JSP container as well as a federated view of the Java EE platform APIs.

3.9.1 Apache Tomcat

Apache Tomcat is an open source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the *JavaServer Pages* (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run [70]. Tomcat is an application server that provides software applications with services such as security, data services, transaction support, load balancing, and management of large distributed systems [70] (Figure 3-2).

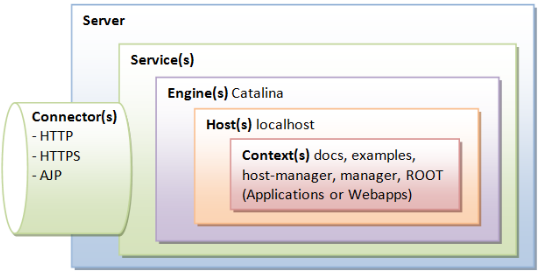


Figure 3-2 Tomcat architecture

Tomcat is not the same as the Apache web server, which is a C implementation of an HTTP web server; these two web servers are not bundled together, although they are frequently used together as part of a server application stack [71]. Apache Tomcat includes tools for configuration and management, but can also be configured by editing XML configuration files. In this work we employ servlet technology along with Apache as a servlet container.

3.10 Serialization

In the context of data storage and transmission, serialization is the process of translating relational database modeling data structure or object state into a format that can be stored in a file or memory buffer or transmitted across a network connection link and deserialized later in the same or another computer environment. When the resulting series of bits is reread according to the serialization format, it can be used to create a semantically identical clone of the original object. For many complex objects such as those that make extensive use of references, this process is not straightforward. Serialization of object-oriented objects does not include any of their associated methods with which they were previously tightly linked in source environment [72].

This process of serializing an object is also called deflating or marshalling an object. The opposite operation, extracting a data structure from a series of bytes, is deserialization (which is also called inflating or unmarshalling).

The Xerox Network Systems Courier technology in the early 1980s influenced the first widely adopted standard. Sun Microsystems published the External Data Representation (XDR) in 1987.

In the late 1990s, a push to provide an alternative to the standard serialization protocols started and XML was used to produce a human readable text-based encoding. Such an encoding can be useful for persistent objects that may be read and understood by humans, or communicated to other systems regardless of programming language. It has the disadvantage of losing the more compact, byte-stream-based encoding, but larger storage and transmission capacities made file size less of a concern than in the early days of computing. Binary XML has been proposed as a compromise which is not readable by plain-text editors, but is more compact than regular XML. In the 2000s, XML is often used for asynchronous transfer of structured data between client and server in Ajax web applications.

JSON is a more lightweight plain-text alternative to XML which is also commonly used for client-server communication in web applications. JSON is based on JavaScript syntax, but is supported in other programming languages as well.

Another alternative, YAML, is effectively a superset of JSON and includes features that make it more powerful for serialization, more human friendly and potentially more compact. These features include a notion of tagging data types, support for non-hierarchical data structures, the option to structure data with indentation, and multiple forms of scalar data quoting.

Java provides automatic serialization which requires that the object be marked by implementing the *java.io.Serializable* interface. Implementing the interface marks the class as "okay to serialize," and Java then handles serialization internally. There is no serialization methods defined on the *Serializable* interface, but a serializable class can optionally define methods with certain special names and signatures that if defined, will be called as part of the serialization/deserialization process. The language also allows the developer to override the serialization process more thoroughly by implementing another interface, the *Externalizable* interface, which includes two special methods that are used to save and restore the object's state [68].

The standard encoding method uses a simple translation of the fields into a byte stream. Primitives as well as non-transient, non-static referenced objects are encoded into the stream. Each object that is referenced by the serialized object and not marked as transient must also be serialized; and if any object in the complete graph of non-transient object references is not serializable, then serialization will fail. The developer can influence this behavior by marking objects as transient, or by redefining the serialization for an object so that some portion of the reference graph is truncated and not serialized.

When Java objects use serialization to save state in files, or as blobs in databases, or transferred over network the potential arises that the version of a class reading the data is different than the version that wrote the data.

Versioning raises some fundamental questions about the identity of a class, including what constitutes a compatible change. A compatible change is a change that does not affect the contract between the class and its callers [73]. In this study we exploit Java serializing for data transfer over network and versioning for potential problem in version mismatch.

3.11 Image Segmentation

In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels). The goal of segmentation is to simplify and change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristics.

3.11.1 JSEG

The essential idea of JSEG is to separate the segmentation process into two independently processed stages, color quantization and spatial segmentation. In the first stage, colors in the image are quantized to several representing classes that can be used to differentiate in the image. This quantization is performed in the color space alone without considering the spatial distributions. Afterwards, image pixel colors are replaced by their corresponding color class labels, thus forming a class-map of the image. Applying the criterion to local windows in the class-map, results in the "J-image" in which high and low values correspond to possible boundaries and interiors of color-texture regions. A region growing method is then used to segment the image based on the multi-scale J-images [74].

We compare Normalized Cut Image segmentation and JSEG and finally decide to use JSEG. The reasons are quick processing time and independency to have initial value for number of segments beforehand. A sample and its segmented image are presented in Figure 3-3.

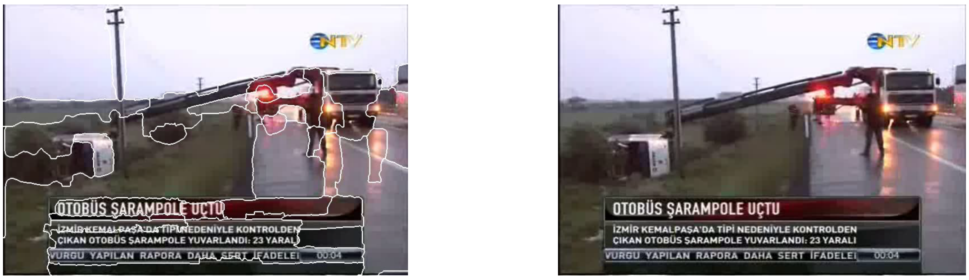

OTOBÜS ŞARAMPOLE UÇTU
İZMİR KEMALPAŞA'DA TİPİ NEDENİYLE KONTROLDEN
ÇIKAN OTOBÜS ŞARAMPOLE YUVARLANDI: 23 YARALI
VURGU YAPILAN RAPORA DAHA SERT İFADELEI
00:04

Figure 3-3 JSEG segmentation

3.12 Ontology

In computer and information science, ontology is defined as a formal representation of the knowledge by a set of concepts and the relationships between these concepts within a domain. It is used to reason about the properties of that domain, and even sometimes, it is utilized for describing the domain. Ontologies are used in artificial intelligence, semantic web, software engineering, biomedical informatics, and information architecture as a form of knowledge representation about the world or some part of it [6].

The body of formally represented knowledge is based on conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them. A conceptualization which called abstract, simplified view of the world that needs a formal representation for some purpose. Since it provides a shared vocabulary, which can be used to model a domain, ontology can be defined as formal specification of a shared conceptualization. It includes machine-interpretable definitions of basic concepts in the domain and relations among them. In other words, it defines what the data is and it's relation to everything else.

We may develop ontology:

- To make users agree on the meaning of terms,
- To analyze domain and make sure the domain assumptions are explicit,
- To enable reuse of the domain knowledge and separate it from the operational knowledge.

In summary, ontology describes the logical structure of a domain as well as its concepts and the relations. Therefore, they should be constructed by a domain expert to guarantee consistency and accuracy. In this study, ontology is used to define the events in specific domain by analyzing visual objects and their relation.

3.13 Canonical Correlation Analysis (CCA)

Canonical Correlation Analysis which was proposed by Hotelling [49] can be seen as the problem of finding basis vectors for two sets of variables such that the correlation between the projections of the variables onto these basis vectors is mutually maximized.

Correlation analysis is dependent on the coordinate system in which the variables are described so, even if there is a very strong linear relationship between two sets of multidimensional variables - depending on the coordinate system used - this relationship might not be detectable as a correlation. Canonical correlation analysis seeks a pair of linear transformations - one for each of the sets of variables - such that when the set of variables are transformed, the corresponding coordinates are maximally correlated (Figure 3-4).

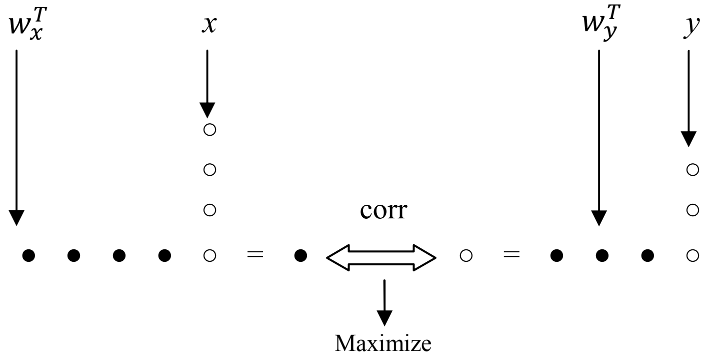


Figure 3-4 CCA

3.14 Kernel Canonical Correlation Analysis (KCCA)

Even if strong linear relationship between variables may exist, some hidden relationships might not be visible as a correlation. CCA may not extract useful descriptors of the data because of its linearity [75].

Kernel CCA offers an alternative solution by first projecting the data into a higher dimensional feature space:

$\emptyset : \mathrm{x} = (\mathrm{x}_1, \dots, \mathrm{x}_n) \rightarrow \emptyset(\mathrm{x}) = (\emptyset_1(\mathrm{x}), \dots, \emptyset_N(\mathrm{x}))\ (n < N)$

This is done before performing CCA in the new feature space. Kernels are methods of implicitly mapping data into a higher dimensional feature space using the methods known as the kernel trick.

A kernel is a function K, such that for all $x, z \epsilon X \rightarrow K(x, z) =< \emptyset(x). \emptyset(z) >$ $w$here $\emptyset$ is a mapping from X to a feature space F. Kernels offer a great deal of flexibility, as they can be generated from other kernels. In the kernel, the data only appears through entries in the

Gramian matrix, therefore this approach gives a further advantage as the number of tunable parameters and updating time does not depend on the number of attributes being used.

CHAPTER 4

PROPOSED ARCHITECTURE

In the previous chapter we describe about technologies and frameworks that we utilize in the proposed architecture. In this chapter we describe proposed architecture's modules briefly. Then we elaborate each module in details. Finally, the process of inserting a new video and concept extraction for each modality is described.

4.1 Ground Truth Data

In order to have a unified data set for training and testing for all modality and compare the results of automatic extracted results with a ground truth data, we manually annotated a set of multimedia files. Some detail information about them are describing as follows.

4.1.1 Dataset Preparation

We downloaded some news videos from NTV [76] news archive then categorized them into 5 categories:

- Accident
- Military
- Natural Disaster
- Sport
- Politics

Then shot boundaries, visual objects, audio concepts and subtitle texts were annotated on the all 76 video clips. The concepts list we used for annotation is subset of LSCOM list. The following information is brief notes about manually annotated data.

At first step, for each video splitting it to meaningful shots was done. After that for each shot Keyframes were extracted. Afterward, on each frame visible objects' regions were annotated with related concepts. Furthermore, for each audio segment a suitable audio concept is assigned. At the last step, the subtitle for all shot is manually extracted and converted to the named entities and some words were selected as the important words to be used in fusion. Then all data were stored in database. The quantitative information about videos is as follows:

- Min duration: 00:14

- Max duration: 15:05
- Min number of shots: 4
- Max number of shots: 152
- Audio sample rate: 44.1 kHz
- Bit depth: 16 bits
- Channel: Mono channel
- Average video duration: 3-5 min
- Total videos duration: 76*4 = ~ 5 hours

We use this dataset for training and testing throughout this work.

4.2 Overall Implementation Details

There are three main components in the proposed architecture; Semantic Concepts Extraction, Coordinator and Storage named as METUMMDS (Figure 4-1).

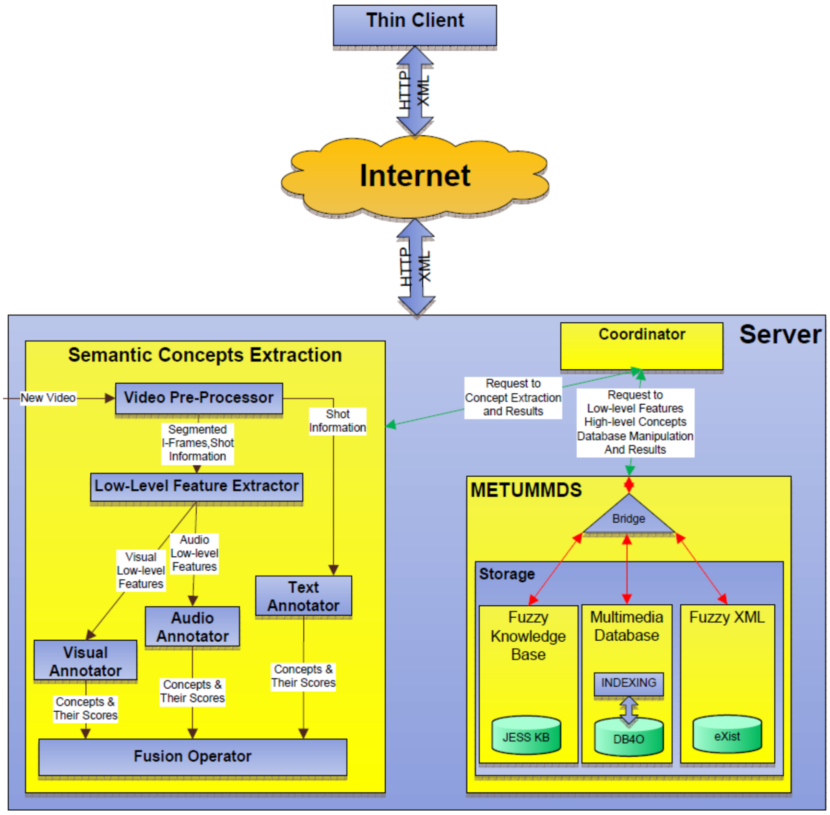


Figure 4-1 Proposed architecture

Semantic Concepts Extraction module which in turn consists of different sub-modules is responsible for semantic concept extraction. Inserting new video to the system triggers this module. After some transcoding this video is passed to preprocessing sub-module. At this module shot boundary detection and important frame (iFrame) extraction are performed on each video. Afterward, for each modality low-level features and concepts are extracted. Each modality has a separate sub-module to accomplish semantic concept extraction.

At audio annotator module, by means of HMM and SVM classifiers, audio segments are annotated. By utilizing SVM classifier visual object annotation performed in prepared module which annotates regions for potential objects in each iFrame automatically. The subtitles of each video are passed to text annotator module. The extracted named entities along with information from other modalities are transferred to the fusion module. At fusion module new scores are calculated for the semantic concepts. Finally, all concepts are ready to be stored in object oriented database and queried by appropriate GUIs.

The extracted information can be stored in database by means of the coordinator. This module is an interface between storage system and other modules. Set of servlets are implemented to prepare utilities like: hiding the access complexities from other modules, get information from client and process them, manage interaction between other modules and so on. The storage system is responsible for storing and retrieving of data. In proposed architecture we use object oriented database with embedded built-in B^+ tree indexing structure. Furthermore, another multidimensional indexing system is implemented that both indexing are used to answer queries and fetch results as quick as possible. The client applications which are developed in multi-threaded mode are prepared which can be used to online video processing, automatic semantic extraction and querying purpose.

4.3 Data Model

Data model is one of the core concepts in multimedia database design. Functionalities of systems are determined according to the prepared data model. Most of existing data models are not integrated enough to support some required functionalities of multimedia materials such as storing, indexing, retrieving and supporting multimodality. So, a data model for multimedia database is presented in this work (Figure 4-2).

As stated before, multimedia data contains huge amount of information that exist in complex and compound structures. Being in various forms and the diversity of semantic contents make it difficult to model multimedia data. In this study, multimedia data are categorized as combination of: *visual* modal, *audio* modal, *textual* modal and fused concepts. Since their structures differ from each other, they should be analyzed separately.

In conceptual data models, entities and relations are completely defined. Therefore, in such a model that is related to multimedia data, semantic entities, objects and the relations of entities should be defined clearly. Besides the hierarchical structure of these entities, since some multimedia data types contain time-specific components, temporal segmentation of such data also should be considered in data modeling.

In the proposed architecture, we follow a well-known temporal segmentation for visual data that has time information. The smallest temporal segments, shots, are defined as the minimal group of adjacent frames stating a continuous action and having images from the same area. Therefore, shots contain some common low-level features. Video objects are composed of these shots and audio segments are aligned with each shot according to the time overlapping.

Temporal segmentation is also important for event definition in videos. Although objects can be extracted from a single frame or image, events which contain temporal information in a group of continuous frames and relations between objects, require more than one frame to be annotated. In many studies events are bounded with a single or a few contiguous shots.

For key-frames, spatial segmentation can be applied for partitioning images into smaller parts in space dimension. If a meaningful spatial segmentation is applied using some low-level features, as in study [2], some regions may directly be mapped into objects easily. JSEG is an example of the spatial segmentation that we utilize in this thesis. For audio concepts, only temporal segmentation is considered since concepts in audio modal lack the spatial relation as defined in visual data. In many studies, a conceptual segmentation and classification is applied for partitioning audio materials. So, audio objects are divided and classified into some predefined taxonomy.

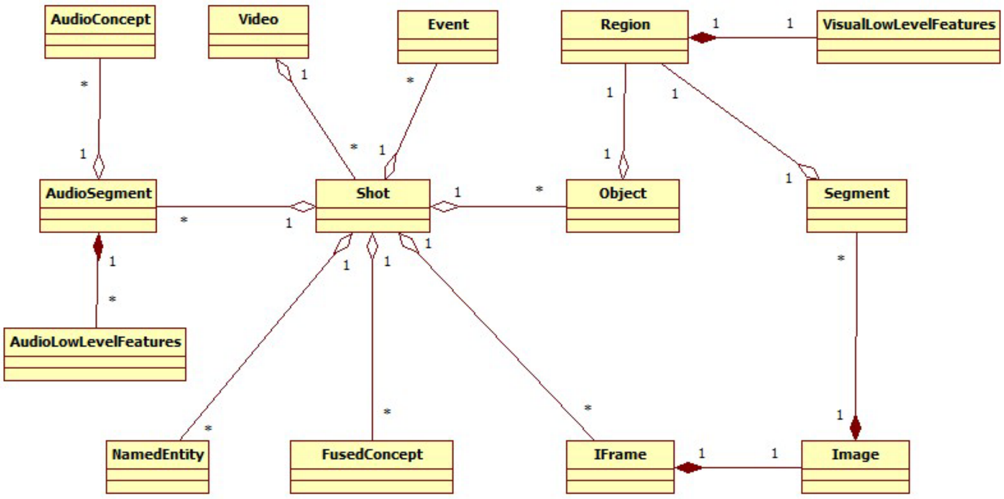


Figure 4-2 Data model

A structural partitioning approach for textual materials is *paragraph-sentence-word* segmentation. Since time and spatial information is bound into textual materials, this type of division can be considered as temporal or spatial segmentation. Objects and events are extracted from textual materials using *words* or *word-groups*. These *word-groups* are called named entities. We have 7 groups for word which are PERSON, LOCATION, ORGANIZATION, TIME, DATE, MONEY and PERCENT. All of these partitioning approaches, temporal and spatial segmentation, are useful for easy modeling and associating semantic contents with the physical portions of multimedia data. Along with named entities, we also exploit key-words to use in data and query fusion.

4.4 Detail View of Architecture

In this section we describe each module and sub modules in details (Figure 4-3).

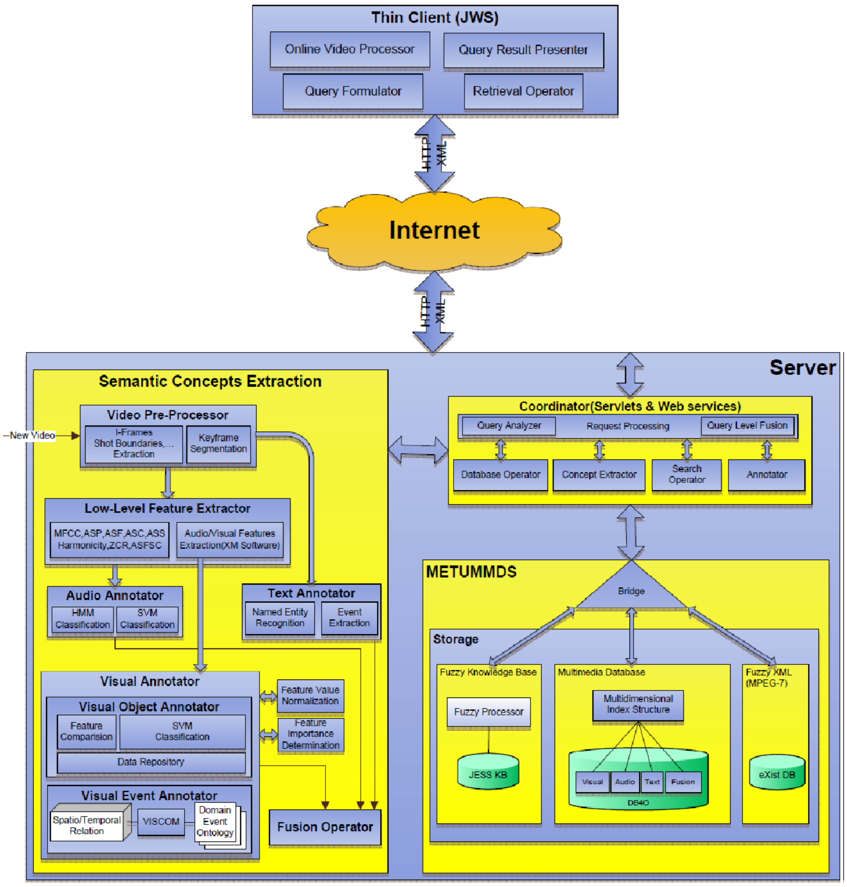


Figure 4-3 Architecture in detail view

Two versions of client are implemented to support proposed architecture. One application that supports online video processing in which concepts extracted automatically and users are allowed to manipulate concepts. The other application provides users with query system. In second application, users can perform various query types that are supported by our proposed architecture.

The client applications can be run via JNLP which were described before. All tasks are managed to run in multi-threaded environment so that, independent tasks do not block the application during their execution.

The coordinator module is responsible for providing the interactions between all parts. To support operation proposed in our architecture, client applications exploit various servlets which are implemented in coordinator module. Detail view of architecture is depicted in (Figure 4-3). The next sections elaborate each boxes of the proposed architecture.

4.4.1 Automatic Shot Detection

As stated before, a shot is a sequence of frames captured by a single camera in a single continuous action, and shot boundaries are transitions between shots. Shot boundary detection is the first step of automatic annotation process.

The algorithm known as Edge Change Ratio is a widely used shot boundary detection algorithm that is robust to different parameter values, which is important for variable video data. To perform shot boundary detection with Edge Change Ratio, frames are captured from video file and frame edges are extracted with “Canny Edge Detection” algorithm using OpenCV library [77].

Automatic shot detection module which runs in a separate thread, takes video file that introduced by user and calculates Edge Change Ratio for all frames then determines edge boundaries with these values. Afterward, automatic shot detection module stores frame numbers of start and end frames of each shot. There may be gaps between shots because of gradual changes; so last frame of shot S_N and first frame of S_{N+1} may not be consecutive. Core module gets these frame numbers and sends them to Video Frames Module. Video Frames Module saves shot’s start and end frames and I-frames between these frames. These frames are considered as keyframes of a shot. Each shot’s start frame number, end frame number and keyframe numbers are saved to a text file. For each video file, core module checks this text file and loads shot boundary information if exists. [77].

4.4.2 Audio Concept Extraction

Audio features and classifiers are two main concepts in audio classification research. Audio features are vectors or scalars containing several descriptive measures of an audio stream whereas classifiers are statistical or linear models representing specifically the intended classes. Audio features are utilized to create these models, which is called model training. In order to create and verify these models, audio dataset is divided into train and test sets. Train set is used for model training and test set is used for verification. In the proposed architecture's audio module, HMM and SVM are used as classifiers which operate on combinations of MPEG-7, MFCC and ZCR as audio features [54].

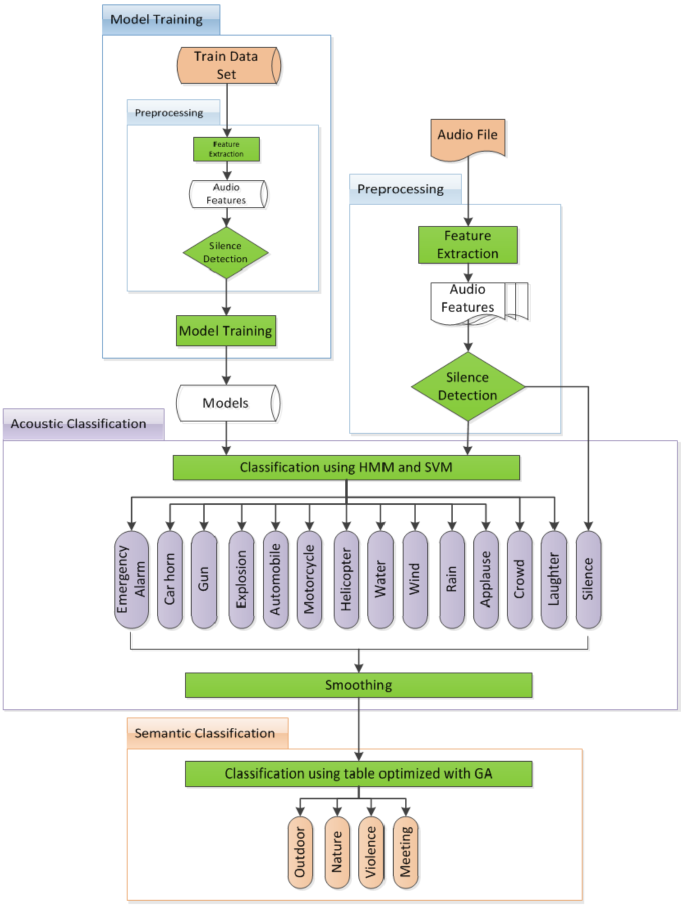


Figure 4-4 Audio annotation blocks

Proposed audio classification system in [54] consists of four main blocks (Figure 4-4): preprocessing, model training, acoustic and semantic classification. In preprocessing block, a given audio clip (file) is divided into one-second segments which will be a sequence of consequent segments. Audio features are extracted from each segment and system labels the silence and non-silence segments of this segment sequence. In order to build up the decision mechanism, the models are created from the train dataset after a preprocessing step within the model training block. The best representative feature and classifier combination is utilized for model training. In acoustic classification block, non-silence segments of the given segment sequence are classified into emergency alarm, car horn, gun-shot, explosion, automobile, motorcycle, helicopter, wind, water, rain, applause, crowd and laughter classes.

In the content of audio classification module, these classes were named as acoustic classes and this process is called acoustic classification which refers to the classification using trained models with extracted audio features. In order to avoid the classification errors, given segment sequences are passed through the smoothing process. Acoustically labeled segment sequence is the input for semantic classification. In semantic classification block, this sequence is classified into higher level semantic classes, namely outdoor, nature, violence and meeting [54].

4.4.3 Visual Object Extraction

A Genetic Algorithm based (GA) object extraction methodology is presented in [2]. At that study, the object extraction process is handled as a categorization problem and the proposed GA based classifier is utilized for classification of candidate objects in image/frame segments. Mentioned work has full support for MPEG-7 standard and uses some MPEG-7 descriptors to classify objects using image segments and to define objects with the "Best Representative and Discriminative Feature (BRDF)" model. A revised version of that study which uses SVM instead of GA is utilized in order extract the visual semantic information in our proposed architecture.

Prior to feature extraction and classification steps, JSEG segmentation library is employed in segmentation module. By applying mentioned algorithm, extracted key frames of video are partitioned spatially into meaningful granularities. Low-level features of these segments are extracted using MPEG Reference Software (eXperimental Model, XM) in feature extraction module and a developed SVM based object classifier is employed in classification module for making decisions about possible objects. The classification is performed iteratively hence, in the each iteration some neighboring segments are combined and the new segment is added to the classification queue. In training phase, to find the best representative object instances for categories, instead of using the average feature values of training samples, a random set of feature values is stored, then for each training sample, the relevance of sample object to object categories is calculated and SVM classifier is used to find the best ranked representative set. This representative set is used in calculating similarity distances of query to the shots' object categories [2].

4.4.4 Event Annotator Module

In [78], a general purpose ontology based model called "VISCOM (VIdeo Semantic COntent Model)" was introduced. In the mentioned model, object definitions, spatiotemporal relations in event and concept definitions are defined.

Various relation types are defined to describe spatial-temporal relations between ontology classes in order to construct domain ontologies. Besides, domain ontologies are enriched with rule definitions to lower spatial relation computations and to be able to define complex events more efficiently. After determining objects, events, and concept individuals, spatial relations between objects and temporal relations between events are decided and individual classes for each relation are defined.

Finally, similarity and role definitions were included and creating domain ontology is completed by adding a number of domain specific rules. A sample ontology definition for *Goal* event is given in Figure 4-6.

4.4.5 Text Annotator Module

Usually, text tags play an important role in information retrieval. In proposed architecture we use a named entity recognizer proposed in [79] to tag named entities in videos' subtitles. The following paragraphs that describe the module are selected from the original work [79].

It is widely known that named entities in different genres of text show considerable diversity. For instance, political news' texts usually come with the names of countries, political parties, governmental institutions, and politicians. On the other hand, frequent named entities in financial texts are usually company names, the names of the heads of these companies as well as that of the governmental.

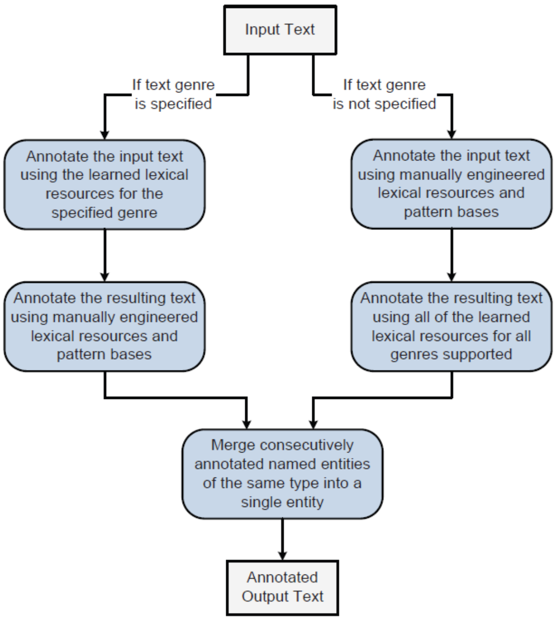

Input Text
If text genre is specified
If text genre is not specified
Annotate the input text using the learned lexical resources for the specified genre
Annotate the input text using manually engineered lexical resources and pattern bases
Annotate the resulting text using manually engineered lexical resources and pattern bases
Annotate the resulting text using all of the learned lexical resources for all genres supported
Merge consecutively annotated named entities of the same type into a single entity
Annotated Output Text

Figure 4-5 Execution flow of the used named entity recognizer

Usually, locations names seem to demonstrate the least variety in texts of different genres. Furthermore, a named entity of a certain type in a specific text genre may be a named entity of another type in another text genre. Above mentioned diversity of the named entities in texts of different genres, results in the observation that the rule based named entity recognizers manually engineered for specific genres usually require manual revisions –which are usually time-consuming and labor-intensive– to adapt the recognizers to other genres of text. But, rule based systems usually achieve higher success rates for the specific text domain that they are engineered for. On the other hand, learning systems are preferable in cases where a considerable amount of training data is available since they do not require human intervention and hence they are easily extensible to other application domains. The ultimate hybrid named entity recognizer which is utilized in proposed architecture has the ability to enrich its lexical resources with those that it learns from annotated texts. The overall schema of this module is presented in Figure 4-5.

With an intention to combine the advantages of text modality and exploit the existing rule based named entity recognizer, we exploit the proposed text annotation developed in [79].

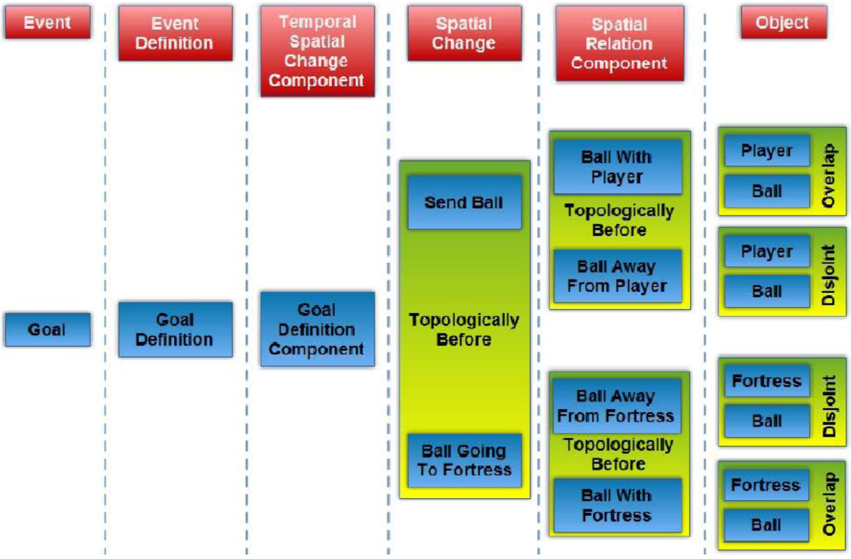


Figure 4-6 Ontology definition for goal event

4.4.6 Data Fusion Module

The data fusion system that we applied in our architecture was proposed in [10]. It is designed to work in cooperation with other modalities such as visual, auditory, textual content analyzers, to carry out the semantic video analysis task and consequently extract semantic information ready to be stored in a database for further retrieval tasks. Since separate information fusion system is intended for integrating the semantic information obtained from independent modalities and due to the existing studies showing that a late fusion scheme performs better than an early one, a late fusion approach which is performed at the score level, is chosen for fusion system proposed in [10]. Score-level fusion provides more information among other late fusion schemes.

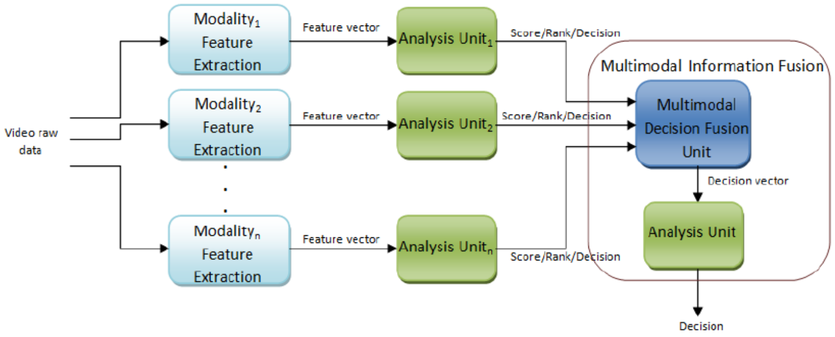


Figure 4-7 A General scheme for late fusion

Before briefly explaining the fusion method and the motivation behind it, let’s mention the nature of the inputs or the outputs of unimodal content analyzers of the system. Different modalities may intrinsically contain relevant but different types of information. For instance, in a football, while the visual content contains objects like ball, referee, field, player, etc., the audio content includes sounds like commentator’s speech, applause, and whistle. Apart from these, the text may contain the names of the teams, the time information, or events like corner, goal, etc.

Some of this information can be extracted from more than one modality such as goal. But, most of the time, the extracted information is not same between the modalities. However, generally most of them are related (e.g. the relationship between cheering concept extracted from the auditory modality and the goal event extracted from textual or visual modalities). Regarding these issues, the fusion problem and the purpose of the fusion system are the integrating of the observations belonging to the same class and utilizing the interactions and relations between the observations of different classes captured from independent modalities [10].

The purpose of the chosen system is to fuse the observations with the purpose of increasing detection accuracy of the concepts and obtaining new information by exploiting the relations of the concepts that is not retrieved from each modality. Before deciding on the fusion method, several aspects and purposes of the expected system are evaluated in the original work [10]. First of all, it aims to build a system as generic as possible; it is not domain-dependent because it enables expansion with new domain knowledge and concept definitions.

Due to these explanations, the system is decided not to be predicated on custom-defined rules. Besides, the inputs of the fusion system mostly will be different prediction scores of different classes, so the methods, e.g. AND, OR, MIN, MAX aggregations, which focuses on merging the decisions belonging to the same class is not sufficient enough. Therefore, SVM, one of the most successful classification methods, is chosen for the fusion strategy. Besides, SVM is observed to be the most used and proved to be very effective in the studies which follow a multimodal approach for semantic concept detection and mostly used fusion methods for semantic concept detection task. Even the proposed fusion system is established on an existing supervised learning method (SVM); it can be viewed as a naive approach when it is analyzed all in all. The prominent features of the fusion system we utilize in proposed architecture are the ability to detect a new concept, performing a Relief based feature selection procedure to select important concept scores, utilizing concept interactions, and using the appropriate evaluation metric in performing the cross-validation. The proposed fusion system brings a different approach in multimodal information fusion. In Figure 4-8, the overall architecture of the semantic video system for fusion is illustrated.

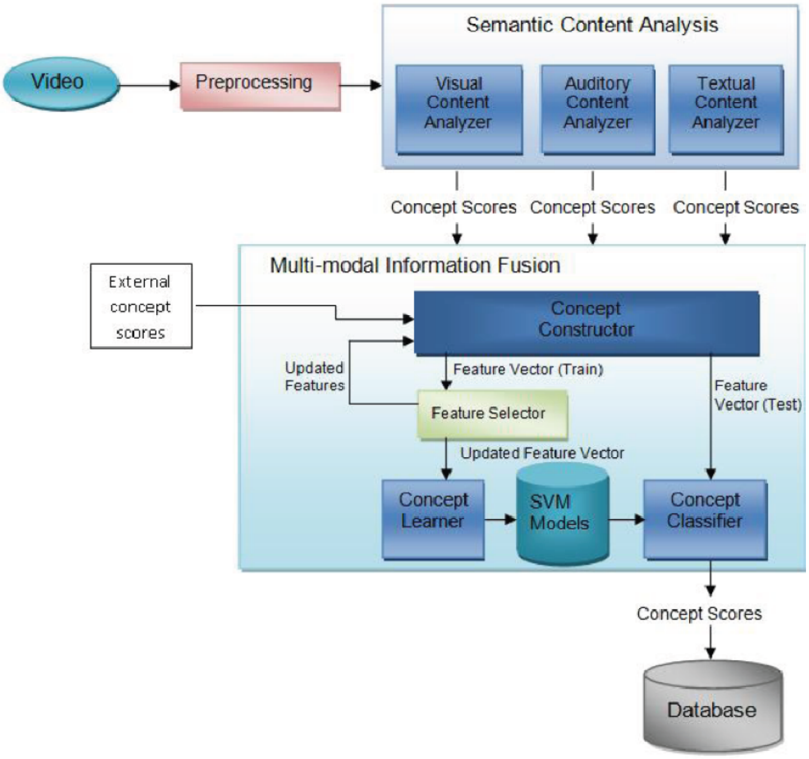


Figure 4-8 A General architecture of semantic video analysis system

The focus of the selected fusion module is the multimodal information fusion part of the architecture which mainly consists of four modules; concept construction module, feature selection module, concept learner and finally the concept classifier. Briefly, the concept constructor simply processes the concept definitions to form the concept instance and performs a temporal alignment between the modalities according to the shot boundaries. Finally, for each concept it constructs the training or test data according to the purpose.

Feature selection module calculates the scores of all features according to the training data and rejects the features having the weight values below the specific threshold. Note that the features of the fusion system refer to the concept scores obtained from single modality based systems. It then gives the updated training data to concept constructor, which updates the concept feature information and weights.

After the training data is transferred to SVM format, it is passed into the concept learner. Then the concept learner constructs the concept model after some series of processes. In the testing phase, the test data is formed according to the scores obtained from several modalities and then they are given to the concept classifier to create a new concept or generate a new integrated concept score.

4.5 Online Video Concept Extraction

The proposed system has the ability to extract concepts from new video online. First we describe it briefly and then sift through the consecutive steps in details. This part of application is developed as a thin client and the implemented graphical user interfaces are mostly used for data entry and result presenting. Almost all of process-intensive tasks are done on the server side. Furthermore, it is multi-threaded application and each autonomous task can be run and communicate with server independently, which would not cause application to block and wait for other tasks.

The online processing starts with determining a video clip from local disk. This video uploaded to server and after transcoding, it is ready to concept extraction processes. Next step is extracting shot boundaries. At this level, shot boundaries and iFrames are extracted then, each iFrame is segmented into meaningful regions. These regions are sent to annotation application in which some concepts along with scores are assigned to mentioned regions. Following the shot boundary extraction, the audio concept extraction performed on audio file and probabilistic concepts are assigned to segments the audio. Afterward, using some ontology along with visual objects detected in second steps, some events are extracted by probing spatial-temporal relations between objects. The next step is subtitle analyzing in which manually extracted subtitles are processed in named entity recognition subsystem and meaningful words are extracted from subtitle. The last step is fusing of visual, audio, events and text results. The whole data are stored in database at server and can be queried via query application. The sequence diagram for these tasks is presented in Figure 4-9.

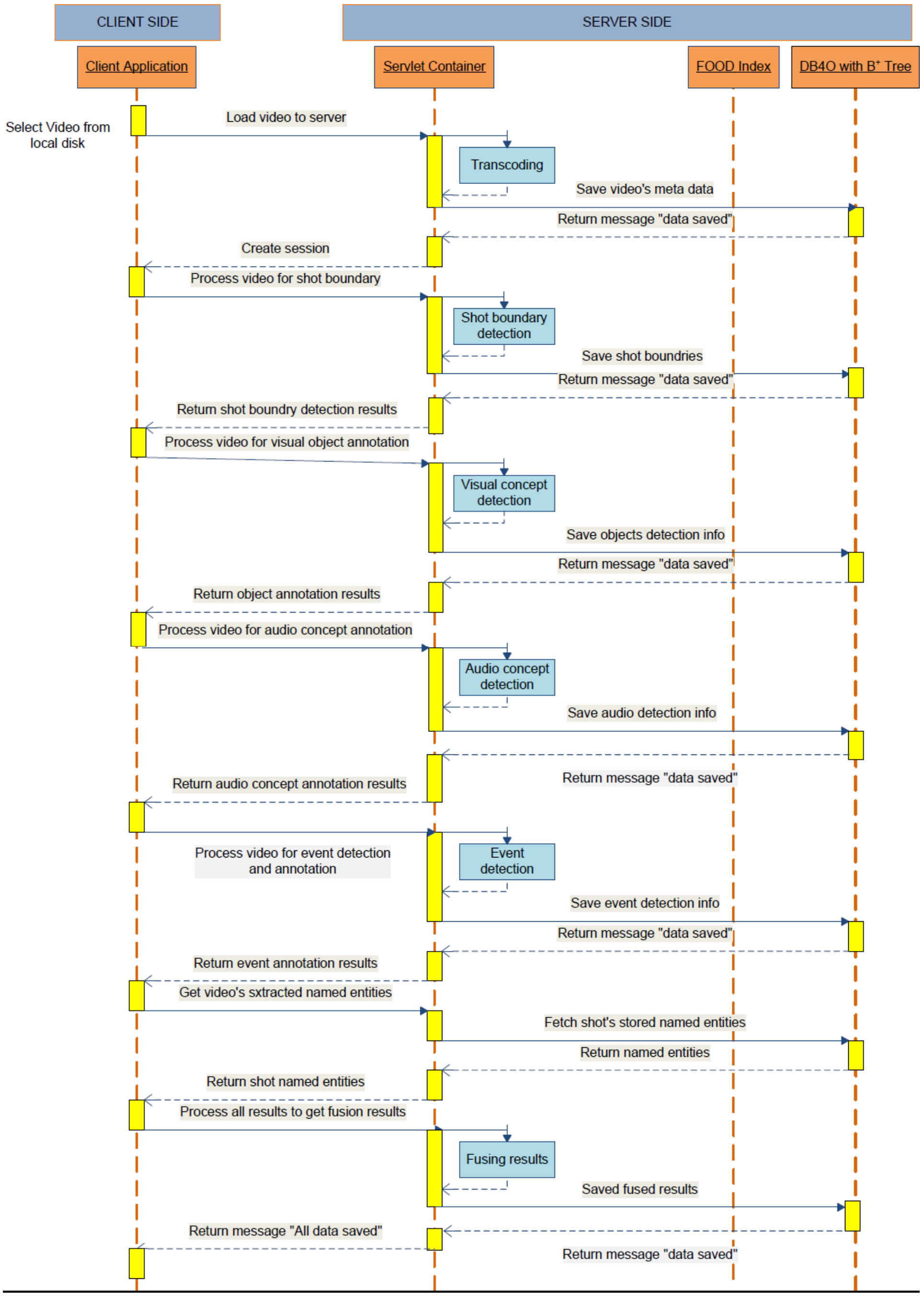
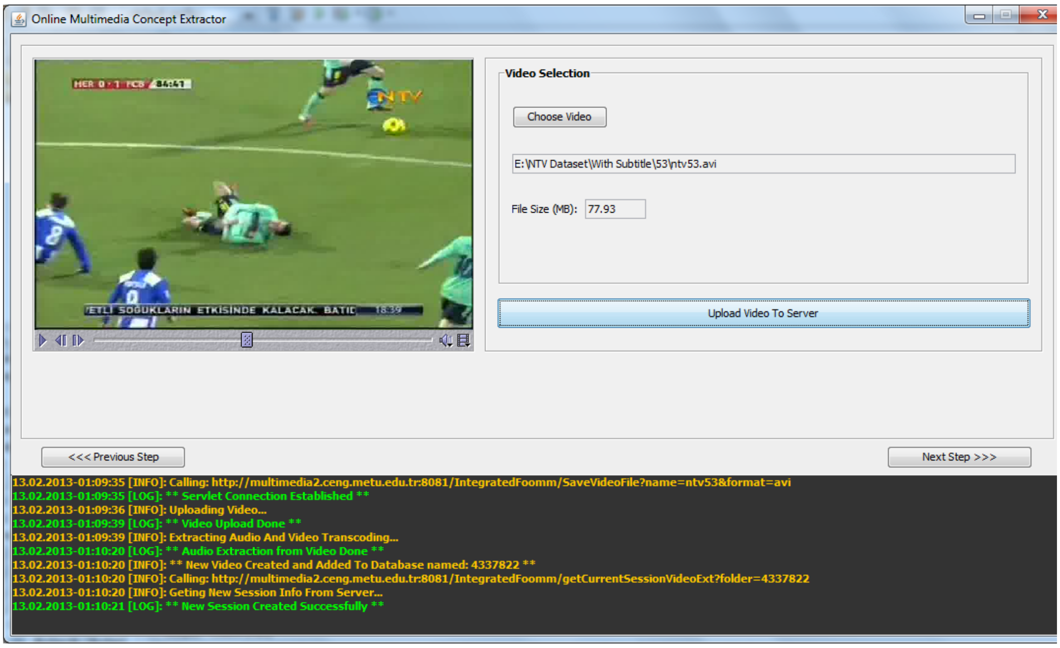

CLIENT SIDE
SERVER SIDE
Client Application
Servlet Container
FOOD Index
DB4O with B^+ Tree
Select Video from local disk
Load video to server
Transcoding
Save video's meta data
Return message "data saved"
Create session
Process video for shot boundary
Shot boundary detection
Save shot boundries
Return message "data saved"
Return shot boundry detection results
Process video for visual object annotation
Visual concept detection
Save objects detection info
Return message "data saved"
Return object annotation results
Process video for audio concept annotation
Audio concept detection
Save audio detection info
Return message "data saved"
Return audio concept annotation results
Process video for event detection and annotation
Event detection
Save event detection info
Return message "data saved"
Return event annotation results
Get video's sxtracted named entities
Fetch shot's stored named entities
Return named entities
Return shot named entities
Process all results to get fusion results
Fusing results
Saved fused results
Return message "All data saved"
Return message "data saved"

Figure 4-9 Sequence diagram for online concept extraction

4.5.1 Video Uploading

In this form (Figure 4-10), user selects a video file from a local disk that intends to start concept extraction. Then by calling appropriate servlet on server, the video is transferred from client to server. At server, audio is separated from original file and after conversion to specific format which is suitable for audio concept detection; it is stored at server along with original video clip. A replica of the original video clip is created and transcoded to low resolution video in order to be transferred over the internet with short delay.



E:\NTV Dataset\With Subtitle\53\ntv53.avi

File Size (MB): 77.93

```
13.02.2013-01:09:35 [INFO]: Calling: http://multimedia2.ceng.metu.edu.tr:8081/IntegratedFoomm/SaveVideoFile?name=ntv53&format=avi
13.02.2013-01:09:35 [LOG]: ** Servlet Connection Established **
13.02.2013-01:09:36 [INFO]: Uploading Video...
13.02.2013-01:09:39 [LOG]: ** Video Upload Done **
13.02.2013-01:09:39 [INFO]: Extracting Audio And Video Transcoding...
13.02.2013-01:10:20 [LOG]: ** Audio Extraction from Video Done **
13.02.2013-01:10:20 [INFO]: ** New Video Created and Added To Database named: 4337822 **
13.02.2013-01:10:20 [INFO]: Calling: http://multimedia2.ceng.metu.edu.tr:8081/IntegratedFoomm/getCurrentSessionVideoExt?folder=4337822
13.02.2013-01:10:20 [INFO]: Geting New Session Info From Server...
13.02.2013-01:10:21 [LOG]: ** New Session Created Successfully **
```

Figure 4-10 User interface for video uploading

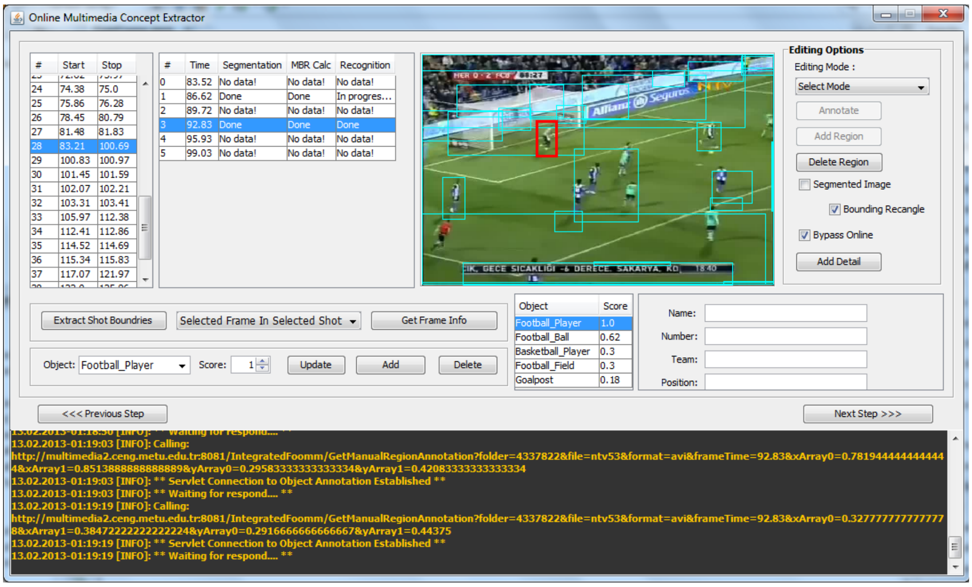
4.5.2 Visual Processing

4.5.2.1 Shot Boundary Extraction

Uploaded video clip is ready for being processed. By pressing appropriate button, a servlet call is invoked and at server side the shot boundary extraction module starts accomplishing its task. As described before, this module partitions video clip into segments that in each shot, camera movement is minimum and logically it is supposed that each shot describes a unique scene and related concepts collection. For each shot, after finishing shot partition, some frames are extracted which we call them iFrames (important frames). These frames normally are extracted from shot's frames on specific predefined time intervals. Selecting this interval is highly dependent on event detection which uses consequent frames and spatial-temporal relations of the objects inside them to deduce some events. In this application we use 2 frames per each second of video. On client application, clicking on each frame causes the frame to be transferred from server to client and cached in client side until the application being closed. A snapshot of this form is shown in Figure 4-11.

4.5.2.2 Automatic Object Annotation

Previous step provides client with extracted iFrames. Now each frame should be segmented to meaningful regions and for each region, automatic annotation should be invoked. User can select how the automatic annotation should be performed. Options range from single frame to all frames in video. By invoking automatic annotation for each frame, new thread is created and started. This thread invokes appropriate servlet in server by providing it all information about the selected frame. At server side, the frame is segmented by JSEG algorithm which was described in details before. Consequently, for each segment a Minimum Bounding Rectangle (MBR) is calculated and for each MBR automatic annotation is called. The annotation module produces a list of probabilistic name-value combination for each region. Top 5 most probable annotations are sent to the client as a result (Figure 4-11).



#	Start	Stop
24	74.38	75.0
25	75.86	76.28
26	78.45	80.79
27	81.48	81.83
28	83.21	100.69
29	100.83	100.97
30	101.45	101.59
31	102.07	102.21
32	103.31	103.41
33	105.97	112.38
34	112.41	112.86
35	114.52	114.69
36	115.34	115.83
37	117.07	121.97

#	Time	Segmentation	MBR Calc	Recognition
0	83.52	No data!	No data!	No data!
1	86.62	Done	Done	In progres...
2	89.72	No data!	No data!	No data!
3	92.83	Done	Done	Done
4	95.93	No data!	No data!	No data!
5	99.03	No data!	No data!	No data!

Editing Mode : Select Mode

□ Segmented Image

☑ Bounding Recangle

☑ Bypass Online

Selected Frame In Selected Shot

Object: Football_Player Score: 1

Object	Score
Football_Player	1.0
Football_Ball	0.62
Basketball_Player	0.3
Football_Field	0.3
Goalpost	0. 18

Name:

Number:

Team:

Position:

```
13.02.2013-01:19:03 [INFO]: Calling:
http://multimedia2.ceng.metu.edu.tr:8081/IntegratedFoomm/GetManualRegionAnnotation?folder=4337822&file=ntv53&format=avi&frameTime=92.83&xArray0=0.7819444444444444
4&xArray1=0.8513888888888889&yArray0=0.29583333333333334&yArray1=0.42083333333333334
13.02.2013-01:19:03 [INFO]: ** Servlet Connection to Object Annotation Established **
13.02.2013-01:19:03 [INFO]: ** Waiting for respond.... **
13.02.2013-01:19:19 [INFO]: Calling:
http://multimedia2.ceng.metu.edu.tr:8081/IntegratedFoomm/GetManualRegionAnnotation?folder=4337822&file=ntv53&format=avi&frameTime=92.83&xArray0=0.3277777777777777
8&xArray1=0.38472222222222224&yArray0=0.2916666666666667&yArray1=0.44375
13.02.2013-01:19:19 [INFO]: ** Servlet Connection to Object Annotation Established **
13.02.2013-01:19:19 [INFO]: ** Waiting for respond.... **
```

Figure 4-11 User interface for visual processing

4.5.2.3 Manual Region Annotation

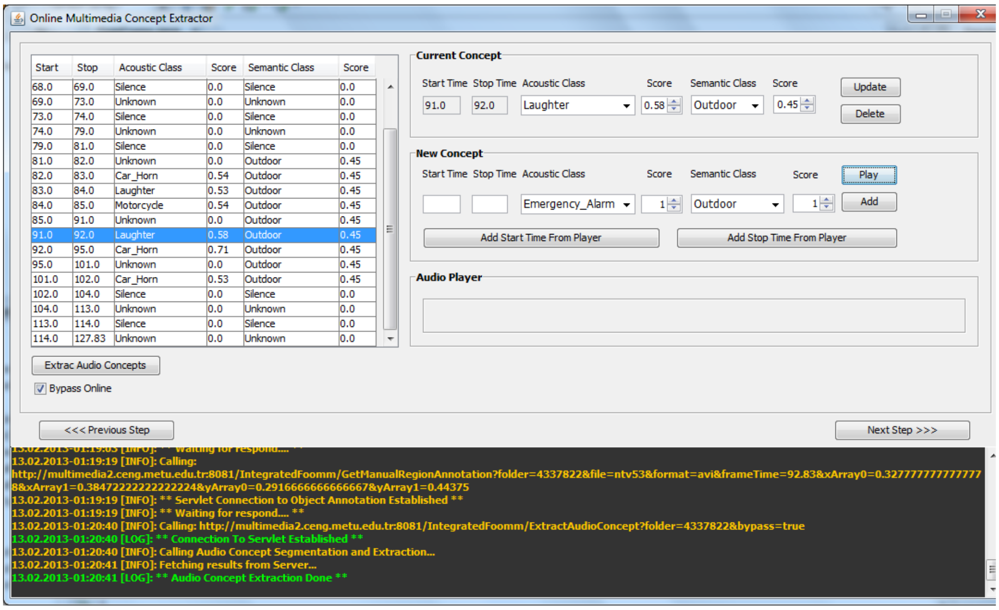
At client side, a new region can be added to frame or previously extracted region can be removed. For any region, all scores and concepts can be manipulated manually by user. For the first top concepts, extra information can be added. For instance if a first top annotation is *Football_Player*, information like *Name, Team Name, comments* and etc. can be added to the object. After accomplishing these steps, all frames in shots contain some regions with related annotation which were created automatically or manually [Figure 4-11].

4.5.3 Audio Concept Annotation

This step is responsible for performing audio annotation and manually manipulating them. Here, pressing appropriate button will trigger servlet at server side to start the audio classification phase. The return value of this process is a list of audio segments that include start and stop time for each segment as well as assigned semantic and acoustic class and relevant score. By using related GUIs, user can add, remove and update these concepts while playing audio on client side (Figure 4-12).

4.5.4 Event Detection

At this step some domain specific events are extracted. By defining ontology for a domain and analyzing spatial-temporal relations between detected objects, some events can be extracted. At this phase for each detected shot and all iFrames inside it, the relations between extracted objects in consecutive frames are spotted by using domain event ontology. In this architecture we just use event ontology for football domain as a proof of concept but any other ontology can be manually defined and appended to this system. In football domain, events such as *Goal, Pass, shoot,* etc. can be detected. By using appropriate GUIs user can manually add new events and delete or update extracted events (Figure 4-13).



Online Multimedia Concept Extractor

Start	Stop	Acoustic Class	Score	Semantic Class	Score
68.0	69.0	Silence	0.0	Silence	0.0
69.0	73.0	Unknown	0.0	Unknown	0.0
73.0	74.0	Silence	0.0	Silence	0.0
74.0	79.0	Unknown	0.0	Unknown	0.0
79.0	81.0	Silence	0.0	Silence	0.0
81.0	82.0	Unknown	0.0	Outdoor	0.45
82.0	83.0	Car_Horn	0.54	Outdoor	0.45
83.0	84.0	Laughter	0.53	Outdoor	0.45
84.0	85.0	Motorcycle	0.54	Outdoor	0.45
85.0	91.0	Unknown	0.0	Outdoor	0.45
91.0	92.0	Laughter	0.58	Outdoor	0.45
92.0	95.0	Car_Horn	0.71	Outdoor	0.45
95.0	101.0	Unknown	0.0	Outdoor	0.45
101.0	102.0	Car_Horn	0.53	Outdoor	0.45
102.0	104.0	Silence	0.0	Silence	0.0
104.0	113.0	Unknown	0.0	Unknown	0.0
113.0	114.0	Silence	0.0	Silence	0.0
114.0	127.83	Unknown	0.0	Unknown	0.0

Current Concept

Start Time	Stop Time	Acoustic Class	Score	Semantic Class	Score
91.0	92.0	Laughter	0.58	Outdoor	0.45

New Concept

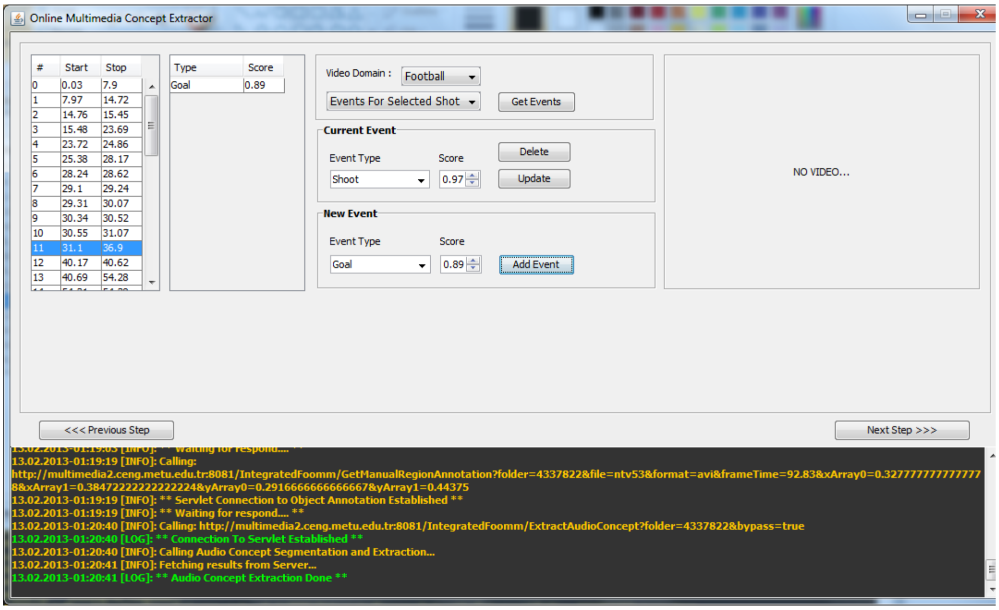
Start Time	Stop Time	Acoustic Class	Score	Semantic Class	Score
		Emergency_Alarm	1	Outdoor	1

Audio Player

☑ Bypass Online

```
13.02.2013-01:19:03 [INFO]: ** Waiting for respond.... **
13.02.2013-01:19:19 [INFO]: Calling:
http://multimedia2.ceng.metu.edu.tr:8081/IntegratedFoomm/GetManualRegionAnnotation?folder=4337822&file=ntv53&format=avi&frameTime=92.83&xArray0=0.327777777777777
8&xArray1=0.3847222222222224&yArray0=0.2916666666666667&yArray1=0.44375
13.02.2013-01:19:19 [INFO]: ** Servlet Connection to Object Annotation Established **
13.02.2013-01:19:19 [INFO]: ** Waiting for respond.... **
13.02.2013-01:20:40 [INFO]: Calling: http://multimedia2.ceng.metu.edu.tr:8081/IntegratedFoomm/ExtractAudioConcept?folder=4337822&bypass=true
13.02.2013-01:20:40 [LOG]: ** Connection To Servlet Established **
13.02.2013-01:20:40 [INFO]: Calling Audio Concept Segmentation and Extraction...
13.02.2013-01:20:41 [INFO]: Fetching results from Server...
13.02.2013-01:20:41 [LOG]: ** Audio Concept Extraction Done **
```

Figure 4-12 User interface for audio extraction



#	Start	Stop
0	0.03	7.9
1	7.97	14.72
2	14.76	15.45
3	15.48	23.69
4	23.72	24.86
5	25.38	28.17
6	28.24	28.62
7	29.1	29.24
8	29.31	30.07
9	30.34	30.52
10	30.55	31.07
11	31.1	36.9
12	40.17	40.62
13	40.69	54.28

Type	Score
Goal	0.89

Video Domain : Football

Events For Selected Shot

Current Event

Event Type: Shoot   Score: 0.97

New Event

Event Type: Goal   Score: 0.89

```
13.02.2013-01:19:03 [INFO]: ** Waiting for respond.... **
13.02.2013-01:19:19 [INFO]: Calling:
http://multimedia2.ceng.metu.edu.tr:8081/IntegratedFoomm/GetManualRegionAnnotation?folder=4337822&file=ntv53&format=avi&frameTime=92.83&xArray0=0.327777777777777
8&xArray1=0.3847222222222224&yArray0=0.2916666666666667&yArray1=0.44375
13.02.2013-01:19:19 [INFO]: ** Servlet Connection to Object Annotation Established **
13.02.2013-01:19:19 [INFO]: ** Waiting for respond.... **
13.02.2013-01:20:40 [INFO]: Calling: http://multimedia2.ceng.metu.edu.tr:8081/IntegratedFoomm/ExtractAudioConcept?folder=4337822&bypass=true
13.02.2013-01:20:40 [LOG]: ** Connection To Servlet Established **
13.02.2013-01:20:40 [INFO]: Calling Audio Concept Segmentation and Extraction...
13.02.2013-01:20:41 [INFO]: Fetching results from Server...
13.02.2013-01:20:41 [LOG]: ** Audio Concept Extraction Done **
```

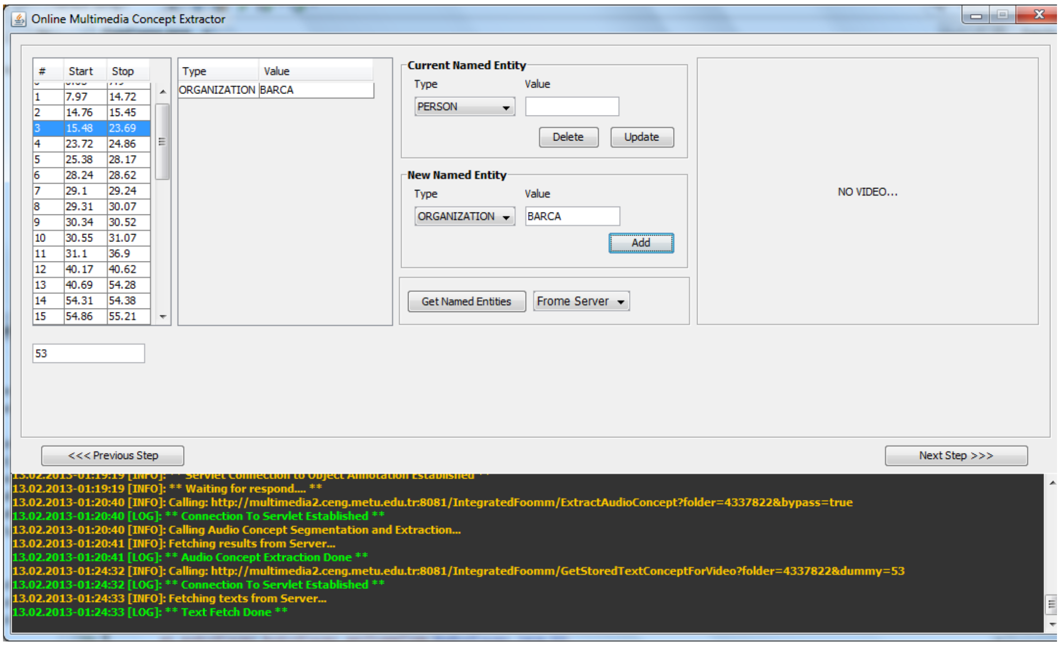
Figure 4-13 User interface for event extraction

4.5.5 Named Entity Recognition

Since most of news videos has subtitle, utilizing these subtitles provide us with valuable data and concepts. The extraction of subtitle is somehow a relatively out of bound of this thesis so, we just extract them manually and process them. The module for text processing gets some sentences and after processing them the result is a list of named entities inside text. We have 7 types of named entities in this architecture which are as follow:

- PERSON
- LOCATION
- ORGANIZATION
- TIME
- DATE
- MONEY
- PERCENT

After getting results from servlet, suitable GUIs are provided that make manipulation of named entities possible. User can play each shot and add new named entity type with any desired value. The extracted named entities can also be removed and edited (Figure 4-14).



#	Start	Stop
1	7.97	14.72
2	14.76	15.45
3	15.48	23.69
4	23.72	24.86
5	25.38	28.17
6	28.24	28.62
7	29.1	29.24
8	29.31	30.07
9	30.34	30.52
10	30.55	31.07
11	31.1	36.9
12	40.17	40.62
13	40.69	54.28
14	54.31	54.38
15	54.86	55.21

53

Type	Value
ORGANIZATION	BARCA

Current Named Entity
Type: PERSON
Value:

New Named Entity
Type: ORGANIZATION
Value: BARCA

Frome Server

```
13.02.2013-01:19:19 [INFO]: ** Servlet Connection to Object Annotation Established **
13.02.2013-01:19:19 [INFO]: ** Waiting for respond.... **
13.02.2013-01:20:40 [INFO]: Calling: http://multimedia2.ceng.metu.edu.tr:8081/IntegratedFoomm/ExtractAudioConcept?folder=4337822&bypass=true
13.02.2013-01:20:40 [LOG]: ** Connection To Servlet Established **
13.02.2013-01:20:40 [INFO]: Calling Audio Concept Segmentation and Extraction...
13.02.2013-01:20:41 [INFO]: Fetching results from Server...
13.02.2013-01:20:41 [LOG]: ** Audio Concept Extraction Done **
13.02.2013-01:24:32 [INFO]: Calling: http://multimedia2.ceng.metu.edu.tr:8081/IntegratedFoomm/GetStoredTextConceptForVideo?folder=4337822&dummy=53
13.02.2013-01:24:32 [LOG]: ** Connection To Servlet Established **
13.02.2013-01:24:33 [INFO]: Fetching texts from Server...
13.02.2013-01:24:33 [LOG]: ** Text Fetch Done **
```

Figure 4-14 User interface for named entity recognition

4.5.6 Data Fusion

At final step, the results of all previous steps are fused to create new scores for detected semantic concepts. All results are transferred to server and stored in object oriented database to be indexed and queried. Mentioned steps can be done for any desired number of videos. We perform these steps for some videos and after populating the database, we utilize them to evaluate our query performance which is discussed in upcoming chapter.

CHAPTER 5

RETRIEVAL MODEL AND SUPPORTED QUERIES

5.1 Retrieval Models

In this section a short overview of some well-known methods in Information Retrieval (IR) are described. We start by looking at two examples of widely used systems: “Exact Match” and “Partial Match”. Then the Boolean Model for exact match and Vector Space Model for partial match are discussed. At the end, supported query types are described.

5.1.1 Boolean Model

The interest for information retrieval has existed long before the Internet. The Boolean retrieval is the most simple and common of these retrieval methods and relies on the use of Boolean operators. The terms in a query are linked together with AND, OR and NOT. This method is often used in search engines on the Internet because it is fast and can therefore be used online. This method has also its problems. The user has to have some knowledge to the search topic for the search to be efficient, e.g., a wrong word in a query could rank a relevant document non-relevant. Furthermore, retrieved documents are all equally ranked and the number of retrieved documents can only be changed by reformulating the query. In general, exact match has the following advantages and disadvantages [80]:

- Advantages of exact match
    - Can be implemented very efficiently
    - Predictable and easy to explain
    - Structured queries for finding precise documents
    - Work well when we know exactly (or roughly) what the collection contains and what we’re looking for
- Disadvantages of exact match
    - Query formulation difficulty for most users
    - Indexing vocabulary same as query vocabulary
    - Acceptable precision generally means unacceptable recall
    - Hard to compare best- and exact-match (No ranking)

The Boolean retrieval has been extended and refined to solve these problems. Expanded term weighting operations make ranking of documents possible, where the terms in the document can be weighted according to their frequency in the document [81]. Boolean information retrieval has been combined with content-based search using concept lattices, where shared terms from previously attained documents are used to refine and expand the query in [82]. The Boolean operators have been replaced with fuzzy operators in [83]. Weighted query expansion using a thesaurus stated in [84]. A model based on fuzzy set theory allows the interpretation of a user query with a linguistic descriptor for each term also proposed in [85].

5.1.2 Vector Space Model

In information retrieval societies, best-match or ranking models are now more common. The Vector Space Model is an example of this category which is described generally in upcoming sections.

The vector space model procedure can be divided into three stages. The first stage is the document indexing where content holding terms are extracted from the document text. The second stage is the weighting of the indexed terms to enhance retrieval of document relevant to the user query. The last stage ranks the documents with respect to the query according to a similarity measure.

5.1.2.1 Document Indexing

It is obvious that many of the words in a document do not describe the content such as *the* and *is*. By using automatic document indexing these non-significant words (function words) are removed from the document vector, so the document will only be represented by content bearing words [81]. This indexing can be based on term frequency, where terms that have both high and low frequency within a document are considered to be function words [81][86]. In practice, term frequency has been difficult to implement in automatic indexing. Instead, the use of a stop list which holds common words to remove high frequency words (stop words) [81][86] is more common which makes the indexing method language dependent. In general, 40-50% of the total number of words in a document is removed with the help of a stop list [81].

Non-linguistic methods for indexing have also been implemented. Probabilistic indexing is based on the assumption that there is some statistical difference in the distribution of content bearing words, and function words [86]. Probabilistic indexing ranks the terms in the collection with respect to the term frequency in the entire collection. The function words are modeled by a "Poisson distribution" over all documents, but in this model content bearing terms cannot be modeled. The use of Poisson model also was expanded to Bernoulli model in [87]. Recently, an automatic indexing method which uses serial clustering of words in text was introduced [88]. The value of such clustering is an indicator if the word is content bearing. Since in our work we utilize concepts' names as key words, we do not need to remove the non-significant words. Each concept name in each modality is regarded as important word for document. In the scope of this thesis each shot equals to a document.

5.1.2.2 Term Weighting

Term weighting has been explained by controlling the exhaustivity and specificity of the search, where the exhaustivity is related to recall and specificity to precision [86]. The term weighting for the vector space model has entirely been based on single term statistics. There are three main term weighting's factors: term frequency factor, collection frequency factor and length normalization factor. These three factors are multiplied together to make the resulting term weight.

A common weighting method for terms within a document is to use the frequency of occurrence as stated in [89]. The term frequency is somewhat content descriptive for the documents and is generally used as the basis of a weighted document vector. Using binary document vector is also possible, but the results have not been as good compared to term frequency when using the vector space model [90].

There are various weighting schemes to discriminate one document from the other. In general, these factors are called collection frequency document. Most of them, e.g. the inverse document frequency, assume that the importance of a term is proportional with the number of document the term appears in [81]. Experimentally it has been shown that these document discrimination factors lead to a more effective retrieval, i.e., an improvement in precision and recall [90].

The third possible weighting factor is a document length normalization factor. Long documents have usually a much larger term set than short documents, which make long documents more likely to be retrieved than short documents [90].

Different weight schemes have been investigated and the best results, with regard to recall and precision, are obtained by using term frequency with inverse document frequency and length normalization [90],[87]. In this work, since the number of key words (Concepts' name) are limited and fixed for each document, we use score of each concept as its weight. These scores come from each modal's concept classifications.

5.1.2.3 Similarity Coefficients

The similarity in vector space models is decided by using associative coefficients based on the inner product of the query vector and document vector, where word overlap indicates similarity. The inner product is usually normalized. The most popular similarity measure is the cosine coefficient (1), which measures the angle between the document vector and the query vector. Other measures are e.g., Jaccard (2) and Dice coefficients (3) [91].

To sum up about partial match retrieval method following good points and weak points can be mentioned [80] :

- Advantages of partial match

        - Significantly more effective than exact match
        - Uncertainty is a better model than certainty
        - Easier to use (supports full text queries)
        - Similar efficiency (based on inverted file implementations)
    - Disadvantages of partial match
        - More difficult to convey an appropriate cognitive model (“control”)
        - Full text does not mean natural language understanding (no “magic”)
        - Efficiency is always less than exact match (cannot reject documents early)

$$similarity = \cos(\theta) = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (1)$$

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

$$DS = \frac{2C}{A+B} = \frac{2|A \cap B|}{|A| + |B|} \quad (3)$$

We represent each shot in our retrieval model as a 166 dimensional vector in the vector space model of all shots:

$$\boldsymbol{S}_{\mathrm{i}} = (\underbrace{\overset{\mathrm{V}}{v_0, v_1, \ldots, v_{48}}}_{\text{Visual concepts}}, \underbrace{\overset{\mathrm{A}}{a_0, a_1, \ldots, a_{16}}}_{\text{Audio Concepts}}, \underbrace{\overset{\mathrm{T}}{t_0, t_1, \ldots, t_{99}}}_{\text{Text Concepts}})$$

Each element in mentioned vector ($\boldsymbol{S}_{\mathrm{i}}$) is mapped to a concept in a specific modality. The value of element is equal to the weight of concept’s classification score in that modality. In textual modality, we represent each text concept (word) as 0 or 1 which is equivalent to absence or presence of that word. Each query is also represented by a vector in the same vector space. For those constraints included in query string we put 1 in related element and for others that are not presented as query constraint we put 0. In this work, we apply cosine similarity in order to retrieve and rank similar shots.

5.2 Supported Query Types

5.2.1 Query by Concept

In this type of query, the aim is to retrieve shots that include the concepts from single modal or multi modal by joining modalities, concepts with logical operators. Each modality can have more than one query terms and these terms joint by logical operators between them.

Any combination of all 4 modalities can take part in query composition process. The sequence diagram for this type of query is presented in Figure 5-1.

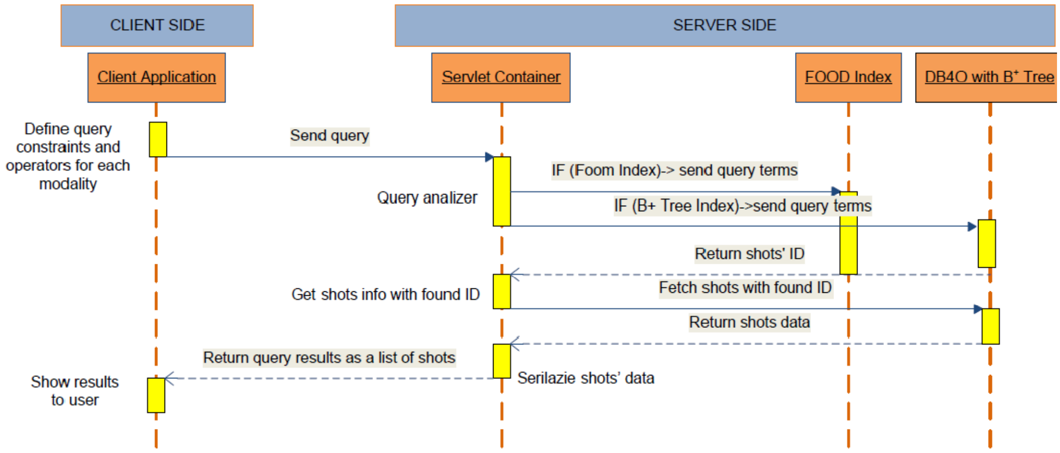

CLIENT SIDE
SERVER SIDE
Client Application
Servlet Container
FOOD Index
DB4O with B+ Tree
Define query constraints and operators for each modality
Send query
IF (Foom Index)-> send query terms
Query analizer
IF (B+ Tree Index)->send query terms
Return shots' ID
Get shots info with found ID
Fetch shots with found ID
Return shots data
Return query results as a list of shots
Serilazie shots' data
Show results to user

Figure 5-1 Sequence diagram for concept query

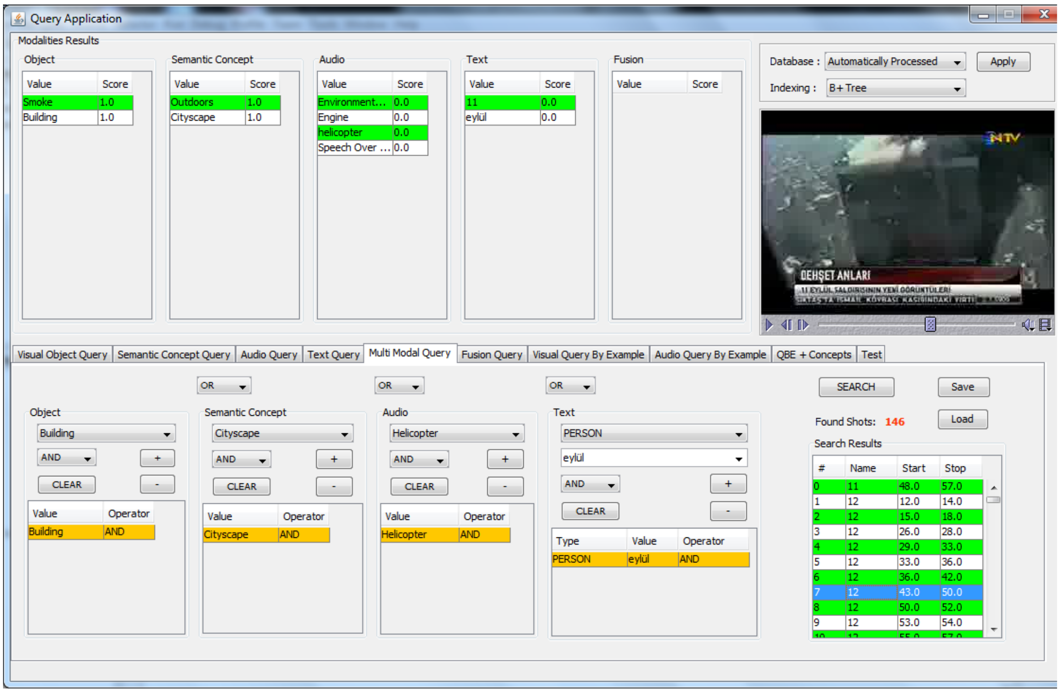
The user starts by selecting query constraints with desired operators to join them then, adds them to query list via GUIs facilities. The constraints can be removed or cleared from the list to start new query. These steps repeated for any desired modalities along with some logical operators between them. After selecting all query terms, they are passed to server as soon as user presses “SEARCH” button in prepaid form. All queries sent to “Query Analyzer” servlet to be processed for next step. In this step, each term for each modality along with their joint operators are extracted and formatted to be sent to data fetcher servlets. The data fetcher utilizes two indexing options which is defined by user during query composition step. First one is embedded B$^+$ Tree indexing which is activated and created inside DB4O beforehand and updated on any data manipulation and the second one is using customized high dimensional indexing system. Depending on the indexing system selection query terms are sent to the storage system and the returned data are potential shots’ IDs. Then these IDs are used to fetch extra information and metadata about specific shot such as all the concepts lists for modalities and iFrames lists and so on. These results then return back to serializer servlet and this servlet reformat and serialize the list of the shots and send them to client as a result for query. On client side, the received object is deserialized to original list of shots and presented to the user as a result. Finally, user can select a desired shot, view its frames and concepts as well as play that shot. Snapshots for these steps are presented in APPENDIX A (Figure 11-1, Figure 11-2, Figure 11-3, Figure 11-4). In the multimodal mode, different combination of the modalities can be used to compose a query.

Any number of constraints can be selected from “Visual Object”, “Semantic Concept”, “Audio” and “Text” modalities to fulfill all kind of attributes that a potential shot may have.

These steps are managed via five tabs in user interfaces. In four of them the query can be composed on single modalities and one tab is for multimodality. These tabs are:

- Visual Object Query
- Semantic Concept Query
- Audio Query
- Text Query
- Multi Modal Query

Another form of this query is on fused data. Fused data consists of a final scored result by using results from all modalities. In this type, only the general constraints are used to compose query, no matter these constraints belong to which category or modality. The query composition starts with selecting multipurpose key words and constraints as well as joining them by any desired logical operator. Then this query is submitted to query analyzer servlet and this servlet in turn decides about indexing type determined by the user. Consequently, potential shots are fetched from fused results by means of the defined index (Figure 5-2).



Object

Value	Score
Smoke	1.0
Building	1.0

Semantic Concept

Value	Score
Outdoors	1.0
Cityscape	1.0

Audio

Value	Score
Environment...	0.0
Engine	0.0
helicopter	0.0
Speech Over ...	0.0

Text

Value	Score
11	0.0
eylül	0.0

Fusion

Value	Score

Database : Automatically Processed

Indexing : B+ Tree

OR OR OR

Object: Building, AND

Value	Operator
Building	AND

Semantic Concept: Cityscape, AND

Value	Operator
Cityscape	AND

Audio: Helicopter, AND

Value	Operator
Helicopter	AND

Text: PERSON, eylül, AND

Type	Value	Operator
PERSON	eylül	AND

Found Shots: 146

Search Results

#	Name	Start	Stop
0	11	48.0	57.0
1	12	12.0	14.0
2	12	15.0	18.0
3	12	26.0	28.0
4	12	29.0	33.0
5	12	33.0	36.0
6	12	36.0	42.0
7	12	43.0	50.0
8	12	50.0	52.0
9	12	53.0	54.0
10	12	55.0	57.0

Figure 5-2 User interface for multimodal query

5.2.2 Query by Content

5.2.2.1 Audio Query by Content

In this type of query an example audio file is selected from data base (Figure 5-3). This example then is sent to query analyzer servlet. The servlet processes the query and decides about the mechanism of retrieval. The audio low-level features that match to selected audio features are fetched from storage. These low-level features are sent to high-dimensional indexing system as example content to be searched. Along with the low-level features, a number (*N*) also send to system which determines the nearest neighborhood parameter. The indexing system, by means of its internal mechanism, finds the closest audio segments which are similar to given audio example. Top *N* numbers of result that have the most similarity to the content of given example audio (in terms of low-level features) are returned as a most similar list. The shot's IDs of these similar audio segments are returned as a result. Finally, the shots are fetched from database according to their IDs and after formatting and serialization, are send to client application as a java object.

At client side, this object deserialized and formatted then presented to user via rich user interfaces. The result shots can be played instantly and related concepts are presented as soon as a shot is selected (Figure 5-4).

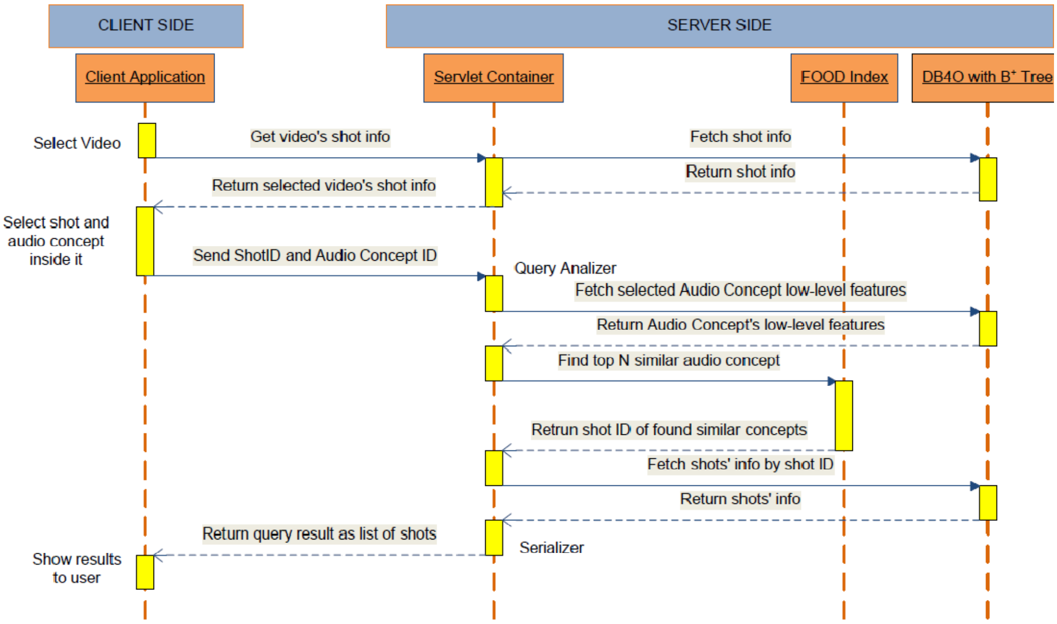
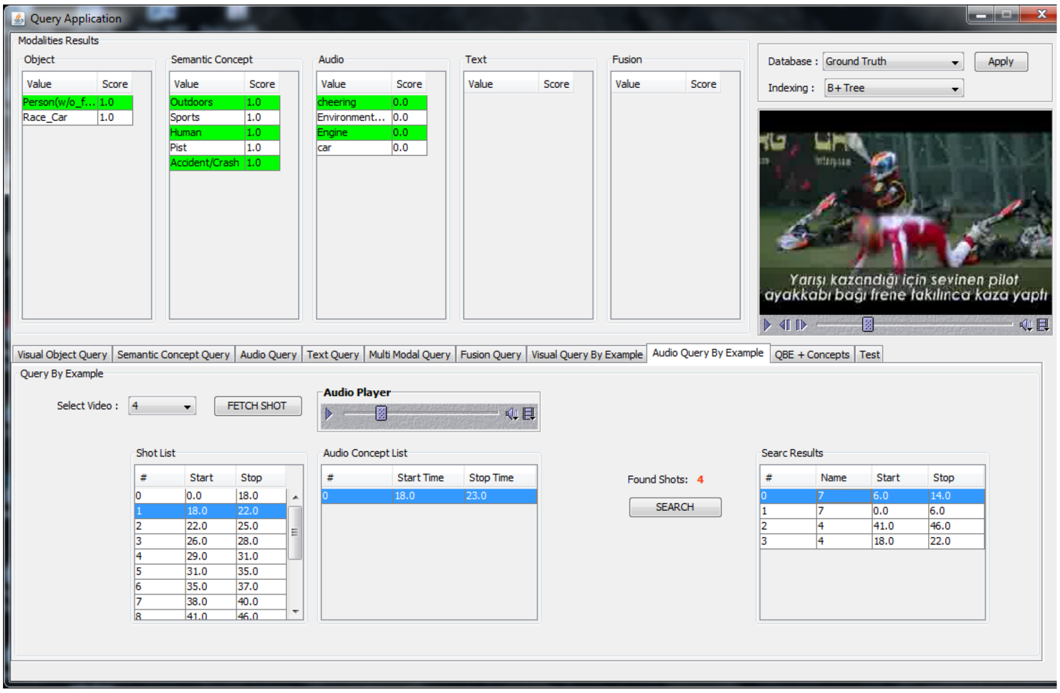

CLIENT SIDE
SERVER SIDE
Client Application
Servlet Container
FOOD Index
DB4O with B+ Tree
Select Video
Get video's shot info
Fetch shot info
Return shot info
Return selected video's shot info
Select shot and audio concept inside it
Send ShotID and Audio Concept ID
Query Analizer
Fetch selected Audio Concept low-level features
Return Audio Concept's low-level features
Find top N similar audio concept
Retrun shot ID of found similar concepts
Fetch shots' info by shot ID
Return shots' info
Return query result as list of shots
Serializer
Show results to user

Figure 5-3 Sequence diagram for audio query by content

The internal process in high-dimensional indexing starts with converting low-level features to a single vector. Then, it searches for the closest vectors for this example in database by using distance function. This process ends with returning top *N* shots with the nearest distance as a result. The distance function applied in proposes indexing system is *Euclidean Distance* that can be replaced by any other distance function.



Database : Ground Truth

Indexing : B+Tree

Object

Value	Score
Person(w/o_f...	1.0
Race_Car	1.0

Semantic Concept

Value	Score
Outdoors	1.0
Sports	1.0
Human	1.0
Pist	1.0
Accident/Crash	1.0

Audio

Value	Score
cheering	0.0
Environment...	0.0
Engine	0.0
car	0.0

Text

Value	Score

Fusion

Value	Score

Select Video : 4

Shot List

#	Start	Stop
0	0.0	18.0
1	18.0	22.0
2	22.0	25.0
3	26.0	28.0
4	29.0	31.0
5	31.0	35.0
6	35.0	37.0
7	38.0	40.0
8	41.0	46.0

Audio Concept List

#	Start Time	Stop Time
0	18.0	23.0

Found Shots: 4

Searc Results

#	Name	Start	Stop
0	7	6.0	14.0
1	7	0.0	6.0
2	4	41.0	46.0
3	4	18.0	22.0

Figure 5-4 User interface for audio QBE

5.2.2.2 Visual Query by Content

Handling this type of query is partly equivalent to the method mentioned in *audio query by content* from mechanism perspective. In this type, a region is selected from an iFrame which belongs to a determined shot. Then this region along with its iFrame number, shot number and video name are send to query analyzer servlet (Figure 5-5).

The servlet processes the request and fetches the visual low-level features that belong to the selected region. These features are passed to the index structure then this structure converts them to a single vector of features. This vector of features compared to all available low-level features via a distance function, the top *N* shots' IDs that have the nearest distance to this visual object returned as potential result. Then these IDs are used to fetch relative shots from database. After reformatting and serializing, the final result of query is sent to client side. Client application receives the result and after deserializing and reformatting, represents the result as a list of potential shots to user via GUI. Here, the user can play them or view their modalities, concepts by clicking on each shot (Figure 5-6). As in the previous method, the distance measure used here is *Euclidean Distance* that can be substitute by any other measure as well.

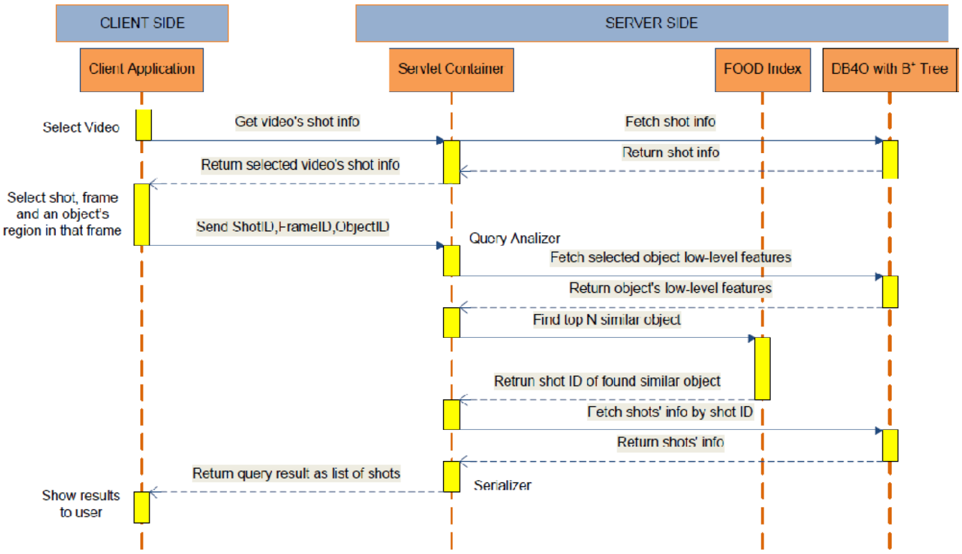
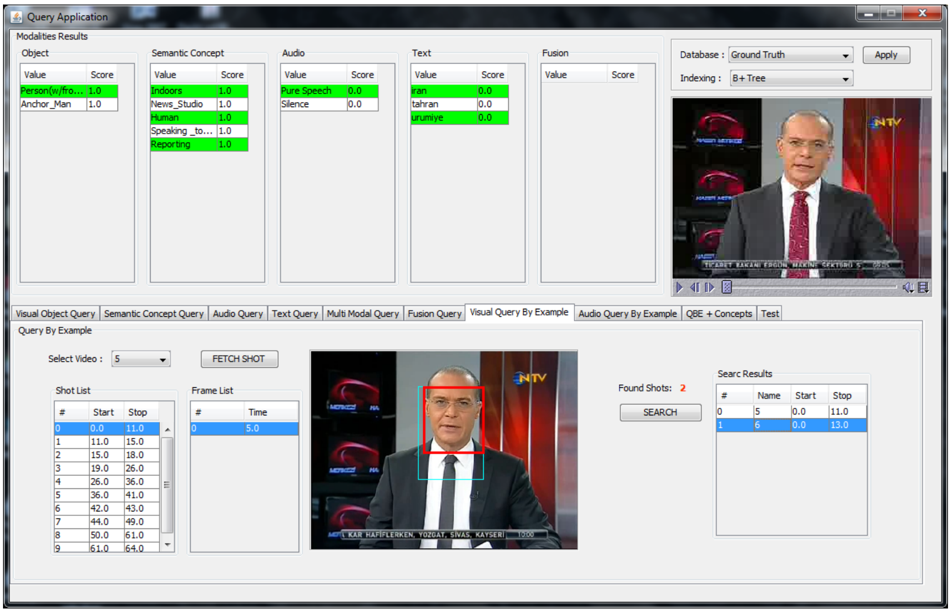

CLIENT SIDE
SERVER SIDE
Client Application
Servlet Container
FOOD Index
DB4O with B⁺ Tree
Select Video
Get video's shot info
Fetch shot info
Return shot info
Return selected video's shot info
Select shot, frame and an object's region in that frame
Send ShotID,FrameID,ObjectID
Query Analizer
Fetch selected object low-level features
Return object's low-level features
Find top N similar object
Retrun shot ID of found similar object
Fetch shots' info by shot ID
Return shots' info
Return query result as list of shots
Serializer
Show results to user

Figure 5-5 Sequence diagram for visual query by content



Modalities Results

Object

Value	Score
Person(w/fro...	1.0
Anchor_Man	1.0

Semantic Concept

Value	Score
Indoors	1.0
News_Studio	1.0
Human	1.0
Speaking _to...	1.0
Reporting	1.0

Audio

Value	Score
Pure Speech	0.0
Silence	0.0

Text

Value	Score
iran	0.0
tahran	0.0
urumiye	0.0

Fusion

Value	Score

Database : Ground Truth

Indexing : B+ Tree

Query By Example

Select Video : 5

Shot List

#	Start	Stop
0	0.0	11.0
1	11.0	15.0
2	15.0	18.0
3	19.0	26.0
4	26.0	36.0
5	36.0	41.0
6	42.0	43.0
7	44.0	49.0
8	50.0	61.0
9	61.0	64.0

Frame List

#	Time
0	5.0

Found Shots: 2

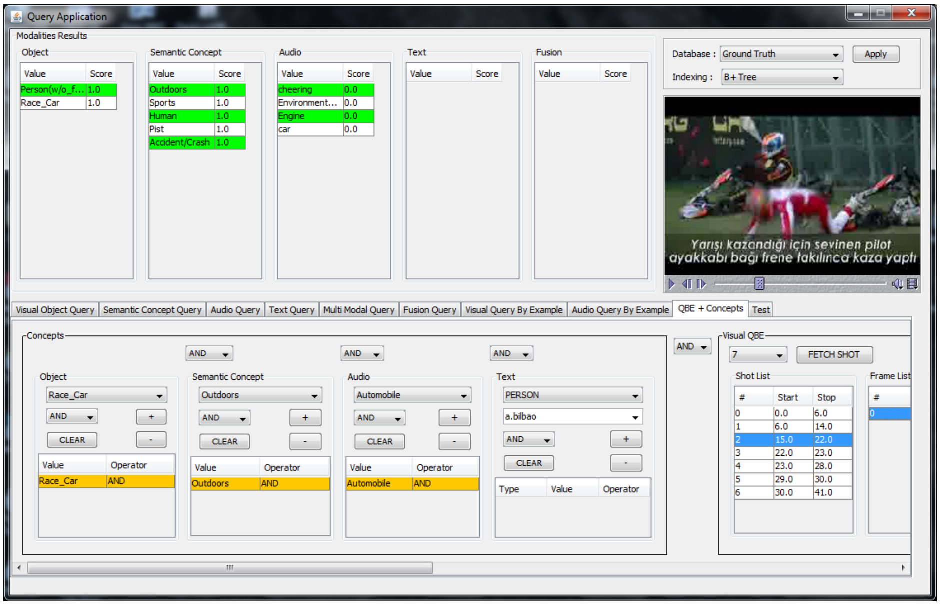
Searc Results

#	Name	Start	Stop
0	5	0.0	11.0
1	6	0.0	13.0

Figure 5-6 User interface for visual QBE

5.2.3 Query by Concept and Content

This type of query is a combination of *Query by Concept* and *Query by Example* which are joined together by means of some logical operators (Figure 5-8). Here we use concept query and content query for audio and visual. In concept query section, the user starts with selecting query constraints with desired operators and adds them to the query list via GUIs. The constraints can be removed form list or totally cleared to start new query. These steps are repeated for any desired modalities along with some logical operators to join modals. In audio query by content section, an audio concept selected from database and a desired part of audio selected as an example. In visual query by example, a video is selected from available videos in the database and sent to server. The server in turn responds to the client with information about selected video which includes shot lists, iFrames lists, objects' region that annotated in each frame and their temporal information. At this point, a region is selected from an iFrame which belongs to a determined shot (Figure 5-7).



Object

Value	Score
Person(w/o_f...	1.0
Race_Car	1.0

Semantic Concept

Value	Score
Outdoors	1.0
Sports	1.0
Human	1.0
Pist	1.0
Accident/Crash	1.0

Audio

Value	Score
cheering	0.0
Environment...	0.0
Engine	0.0
car	0.0

Text

Value	Score

Fusion

Value	Score

Database : Ground Truth

Indexing : B+ Tree

Concepts

AND AND AND

Object: Race_Car, AND

Value	Operator
Race_Car	AND

Semantic Concept: Outdoors, AND

Value	Operator
Outdoors	AND

Audio: Automobile, AND

Value	Operator
Automobile	AND

Text: PERSON, a.bilbao, AND

Type	Value	Operator

AND

Visual QBE: 7

Shot List

#	Start	Stop
0	0.0	6.0
1	6.0	14.0
2	15.0	22.0
3	22.0	23.0
4	23.0	28.0
5	29.0	30.0
6	30.0	41.0

Frame List

#
0

Figure 5-7 User interface for query by concept and content

Now all constraints for contents and concepts are ready. After determining the joint operators, the query structure is sent to the query analyzer servlet. This query structure normally includes a list of visual objects' names which associated together by means of some logical operations. Furthermore, selected audio concepts along with its video name, shot number and audio concept number as well as the selected object's region and its video name, shot number and frame number are included in query structure. Query analyzer gets this structure and processes it then, decides on mechanism of retrieving result. The visual low-level features for selected object region and audio low-level feature along with concept list send to high-dimensional index and the returning result are shot IDs that fulfill the query constraints. Consequently, the shot information fetched from data base according to returning shot IDs and after reformatting, are serialized and sent to client. The client application gets this object and deserialized it and present to user via graphical user interfaces.

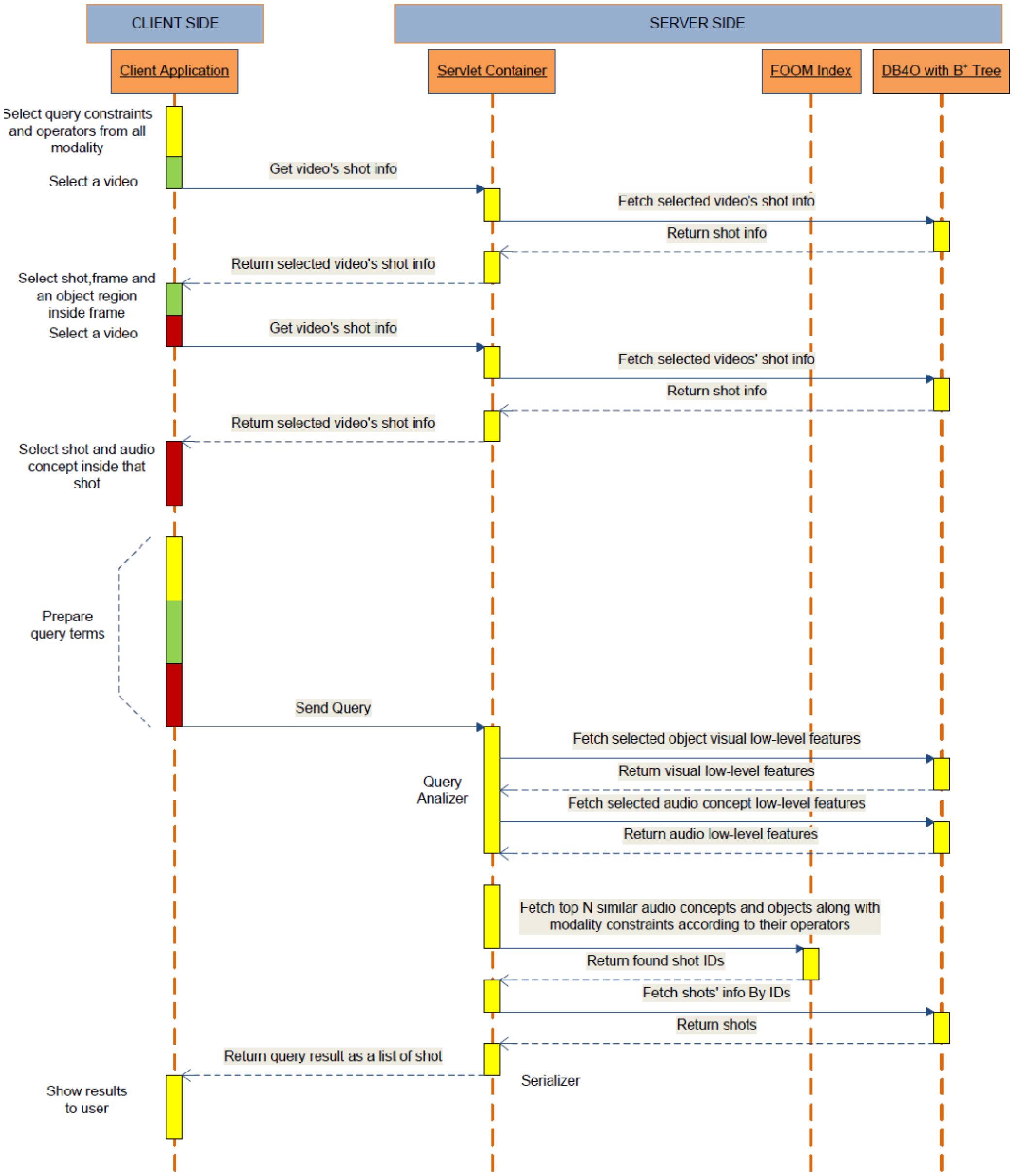
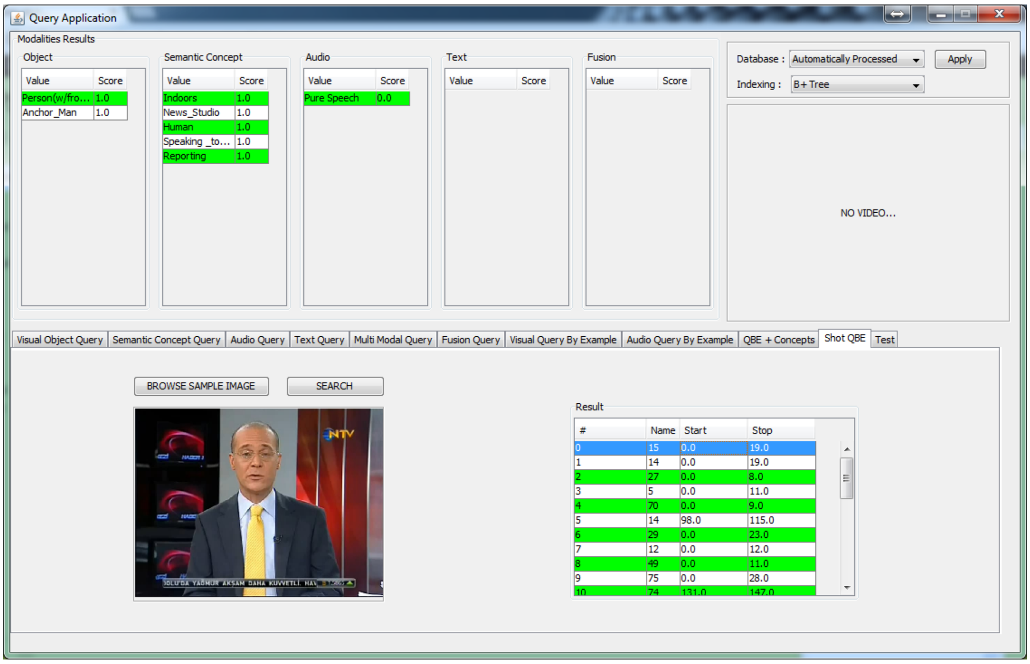

CLIENT SIDE
SERVER SIDE
Client Application
Servlet Container
FOOM Index
DB4O with B+ Tree
Select query constraints and operators from all modality
Select a video
Get video's shot info
Fetch selected video's shot info
Return shot info
Return selected video's shot info
Select shot,frame and an object region inside frame
Select a video
Get video's shot info
Fetch selected videos' shot info
Return shot info
Return selected video's shot info
Select shot and audio concept inside that shot
Prepare query terms
Send Query
Fetch selected object visual low-level features
Return visual low-level features
Query Analizer
Fetch selected audio concept low-level features
Return audio low-level features
Fetch top N similar audio concepts and objects along with modality constraints according to their operators
Return found shot IDs
Fetch shots' info By IDs
Return shots
Return query result as a list of shot
Serializer
Show results to user

Figure 5-8 Sequence diagram for query by concept and content

5.2.4 Query by Example

In previously mentioned query by content, we focus on sample objects or regions in the shots. We look for shots that contain similar objects or regions to the example. Here, our goal is retrieving shots that contain combination of regions and object provided by sample image. This image can be a particular frame of intended video or sub region of such a frame that contains multiple objects.

In client side, user selects a sample image then this image is sent to the server. At server this image's low-level features are extracted and compared with previously extracted low-level features of all frames in database. By using Euclidean distance, top 20 shots that have the least distance from sample provided by user, returned to the client as a result (Figure 5-9). An indexing mechanism can be utilized to increase the retrieval speed.



Database : Automatically Processed

Indexing : B+ Tree

Object

Value	Score
Person(w/fro...	1.0
Anchor_Man	1.0

Semantic Concept

Value	Score
Indoors	1.0
News_Studio	1.0
Human	1.0
Speaking _to...	1.0
Reporting	1.0

Audio

Value	Score
Pure Speech	0.0

Text

Value	Score

Fusion

Value	Score

Result

#	Name	Start	Stop
0	15	0.0	19.0
1	14	0.0	19.0
2	27	0.0	8.0
3	5	0.0	11.0
4	70	0.0	9.0
5	14	98.0	115.0
6	29	0.0	23.0
7	12	0.0	12.0
8	49	0.0	11.0
9	75	0.0	28.0
10	74	131.0	147.0

Figure 5-9 Frame-based QBE GUI

CHAPTER 6

QUERY LEVEL FUSION

In this chapter, we describe proposed query level fusion. According to Pearson Correlation Coefficient matrix for inter-modal and intra-modal as depicted in Figure 6-1 and APPENDIX B, we come up with the idea that by using these correlations we can increase the retrieval efficiency. In order to obtain better retrieval performance we exploit the correlations among modalities as well as inside single modalitiy to enrich our query. To analyze correlation among modals we use linear and non-linear models which respectively are CCA and KCCA. Our method is described in details as follows.

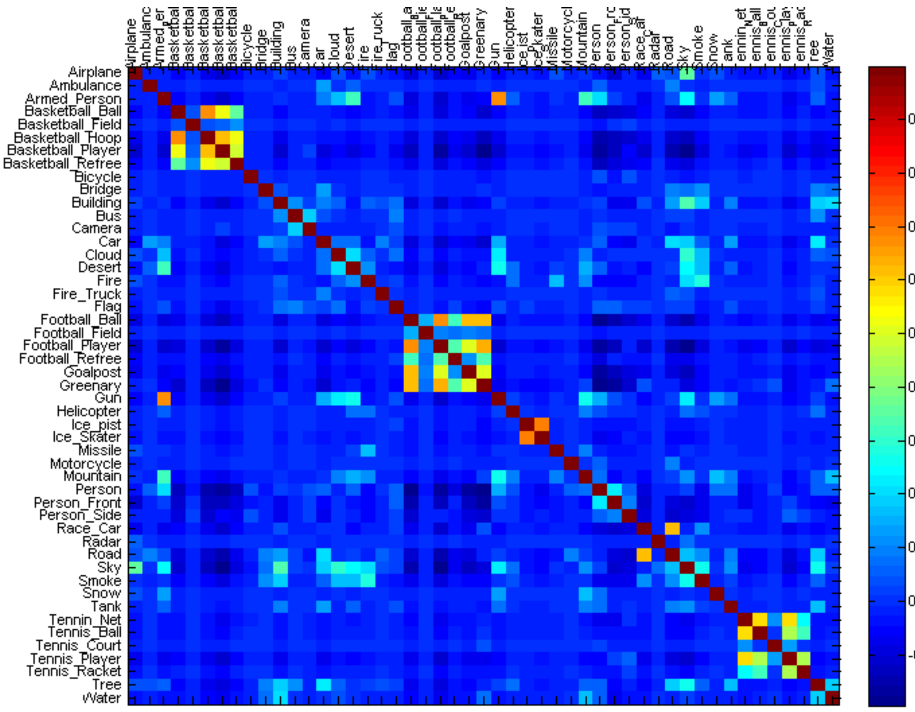


Figure 6-1 Correlation coefficient matrix for visual modal

6.1 Inter-Modal

For each modal, we utilize correlation coefficient among its terms to update weight of specific terms in query. Since values in our dataset are intervals of real numbers, we use Pearson correlation coefficient to analyze correlation among terms (Table 6-1).

Table 6-1 Data Related Correlation Coefficients

Data Types	**Nominal**	**Ordinal**	**Interval**
Nominal	Phi	Rank Biserial	Point Biserial
Ordinal	Rank Biserial	Spearman Rank Corr.	
Interval	Point Biserial		Pearson's *r*

Correlation coefficients that their magnitudes are between 0.9 and 1.0 indicate variables which can be considered very highly correlated**.** Those coefficients whose magnitudes are between 0.7 and 0.9 indicate variables which can be considered highly correlated. We can readily see that $0.9 < |r| < 1.0$ corresponds with $0.81 < r^2 < 1.00$ and $0.7 < |r| < 0.9$ corresponds with $0.49 < r^2 < 0.81$. Since we aim to exploit both negative and positive correlations, we use r^2 values throughout this work. We present each shot by vector $\boldsymbol{S}_i$:

$$\boldsymbol{S}_{\mathrm{i}} = (\underbrace{\overset{\text{V}}{v_0, v_1, \ldots, v_{48}}}_{\text{Visual concepts}}, \underbrace{\overset{\text{A}}{a_0, a_1, \ldots, a_{16}}}_{\text{Audio Concepts}}, \underbrace{\overset{\text{T}}{t_0, t_1, \ldots, t_{99}}}_{\text{Text Concepts}})$$

We consider q as our query which consists of three parts; visual, audio and text which are represented subsequently by q_{v}, q_{a} and q_{t}. We introduce V_i^q as a set of visual concepts which are in q_{v} so that: $\mathrm{V}_i^q = \{\mathrm{v}_1^q, \mathrm{v}_2^q, \mathrm{v}_3^q, \ldots, \mathrm{v}_i^q\}$ (Same definition is valid for audio, A_i^q and text, T_i^q). We also compose a function $f_V(\mathrm{v_i})$ which obtain correlation coefficients of $\mathrm{v_i}$ with all $\mathrm{v_j} \in \mathrm{V}$ so that $\mathrm{i} \neq \mathrm{j}$. To put it formally: $f_V = \{corrcoef(\mathrm{v_i}, \mathrm{v_j}) \mid \mathrm{v_j} \in \mathrm{V}$ *and* $\mathrm{i} \neq \mathrm{j}\}$ where $corrcoef(\mathrm{x,y})$ returns Pearson Correlation Coefficient for x and y and $r^2 > 0.49$. We use the following algorithm (1) to construct new vector for visual part of initial query's visual part (q_{v}).

$$\begin{array}{l} \textit{for all } v_i \in \mathrm{V}^q \\ \qquad \mathrm{q}_i^{tmp} = f_V(\mathrm{v_i}), \\ \mathrm{q}^{total} = \left(\sum_{j=1}^{i} \mathrm{q}_j^{tmp}\right) / \, i, \\ \mathrm{q}_v^{final} = q^{total} + q, \\ \textit{Normalize } \mathrm{q}_v^{final}, \end{array} \tag{1}$$

In order to propagate correlation, we apply this algorithm to all single modals separately. As a result, we have a new query vector q^{final} which is enriched by means of correlation among terms inside each individual modal.

After performing inter-modal correlation propagation, we train two models, CCA and KCCA. We exploit these models to grab the correlation between modalities. We then map our vectors to different dimension that best represents this correlation in reduced dimension. CCA captures linear correlation and KCCA non-linear and in the following segments these methods are elaborated.

6.2 Intra-Model

In this part, we start with describing some concepts and then explain our proposed method in details based on these concepts.

6.2.1 Canonical Correlation Analysis Detail

Canonical correlation analysis is the most generalized member of the family of multivariate statistical techniques. It is directly related to several dependence methods. Similar to regression, canonical correlation's goal is to quantify the strength of the relationship, in this case between the two sets of variables (independent and dependent). It corresponds to factor analysis in the creation of composites of variables. It also resembles discriminant analysis in its ability to determine independent dimensions (similar to discriminant functions) for each variable set and in this situation with the objective of producing the maximum correlation between the dimensions [92].

Thus, canonical correlation identifies the optimum structure or dimensionality of each variable set that maximizes the relationship between independent and dependent variable sets. Canonical correlation analysis deals with the association between composites of sets of multiple dependent and independent variables. In doing so, it develops a number of independent *canonical functions* that maximize the correlation between the linear composites, also known as *canonical variates*, which are sets of dependent and independent variables. Each canonical function is actually based on the correlation between two canonical variates, one variate for the dependent variables and one for the independent variables.

Another unique feature of canonical correlation is that the variates are calculated to maximize their correlation. Moreover, canonical correlation does not stop with the calculation of a single relationship between the sets of variables. Instead, a number of canonical functions (pairs of canonical variates) may be calculated [92].

During recent years there have been advances in data learning using kernel methods. Kernel representation offers an alternative learning to non-linear functions by projecting the data into a high dimensional feature space to increase the computational power of the linear learning machines.

In our example, for finding correlation between modals even if strong linear relationships between variables are used, these relationships might not be visible as correlation. CCA may not extract useful relations of the data because of its linearity. Kernel CCA (KCCA) offers an alternative solution by first projecting the data into a higher dimensional feature space before performing CCA in the new feature space, in order to expose us some latent correlations. Of course the issue of choosing the optimal parameters or the kernel function in a way that improves performance is still open. We describe the method of selecting parameters in our work in the upcoming section [75].

Applying CCA on two sets $A_{\mathrm{n,a}}$ and $B_{\mathrm{n,b}}$ in which n is the number of elements (samples) and a, b are the dimensions of samples, produce two matrices $\breve{A}_{n,c}$ and $\breve{B}_{n,c}$ in which c=Min(a,b). The result matrices are our original data that are represented in different dimension. In our work, $A_{\mathrm{n,a}}$ and $B_{\mathrm{n,b}}$ are equal to our Visual and Audio modals, n is the number of shots and a,b represent the number of concepts in Visual and Audio modals.

For all shots, we initially apply CCA for audio and visual modal then using these new values we apply CCA on text modal. According to our test on all combination, using visual and audio combination at first step and then their combination with text give the best results.

After applying all these steps on each shot the 166 dimensional vector which represents a shot is transferred to a 68 dimensional space. Here, by solving Eigen problem for the data in 68 dimensional space and selecting maximum Eigen Values (both negatives and positives); we reduce our final vector to 61 dimensional. After normalizing these values, each shot is represented by a 61 dimensional vector in the new vector space. The next step is transferring the query vector to the similar dimension in order to calculate Cosine similarity. The q^{final} which is returned from Inter-Modal correlation analysis phase is transferred to new vector space by means of learned *canonical functions.* At the next step, Cosine similarity is calculated among query vector and shot vectors and the most similar shots are ranked as a result. The performance measures are discussed at the end of current chapter.

The second model for Intra-Model is using KCCA. In this model we use Gaussian as our kernel function. As in the general CCA, we first transfer audio and visual modals to higher dimension then use them to apply KCCA on text modal. Since transferring data to higher dimension would push us to higher computation time, we used maximum Eigen values and their Eigen vectors to reduce dimension as much as the redundant dimensions can be wiped out. To find most important Eigen vectors, we select the top most λ^2. For instance, the λ^2 diagram for visual/audio [Figure 6-2] shows us that the Eigen vectors for $\lambda < 2697$ can be wiped out. Therefore we can reduce our 3142 dimensional data to 445 dimensional.

6.2.2 Parameters Selection

There are three parameters involved in the entire procedure of KCCA. One is the choice of kernel, the second is the kernel window width and the last is the regularization parameter.

Throughout this thesis we use *Gaussian* as our choice of kernel and the window width is set to $\sqrt{10S}$ where S is the one-dimensional samples' variance. Such a choice is based on empirical experience which results in good normality checks on kernel data [93]. Since our data is multi-dimensional we transfer data to one dimensional space using PCA by means of greatest λ^2 and their related Eigen vectors. We obtain these values for kernel window width parameter; Visual=1.8795, Audio=2.0333 and Text=1.6105. The window width $\sqrt{10S}$ is a universal rule of thumb suggestion. Though it might not be optimal it gives robust and satisfactory results. The regularisation parameter is considered 1E-5 (0.000001) as recommended in [75].

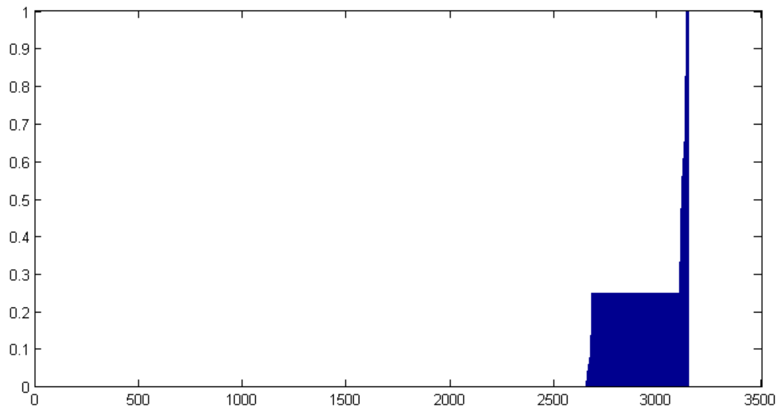


Figure 6-2 λ^2 diagram for visual/audio

6.2.3 Incomplete Cholesky Decomposition

In this work we use complete decomposition to calculate the kernel matrix. Since complete decomposition of a kernel matrix is an expensive task and causes the curse of dimensionality it should be avoided with very large data [75]. Incomplete Cholesky Decomposition as described in [94] is one solution to deal with this problem. In this work we also use this technique by providing an extra parameter to KCCA and then we evaluate the retrieval performance.

6.3 Calculation steps for a sample query

In this section we describe the steps along with intermediate values for performing query level fusion on a given sample query.

We suppose that a query is submitted by a user that asks for "all shots that contain Explosion, Libya". Query fusion analyzer detects that "Explosion" belongs to visual modal as well as "Libya" belongs to text modal and no terms are provided for audio modal. In query vector we put "1" for provided terms' index. Then we start applying inter-modal as well as intra-modal correlation analysis on query vector. Our initial vector is as follows:

V_{1*49}	A_{1*17}	T_{1*100}
(0,0,…,0,0,1,0,0…,0,0)	(0,0,…,0,0,0,0,0…,0,0)	(0,0,…,0,0,0,1,0…,0,0)

In inter-modal for visual according to the Pearson correlation coefficient following concepts are correlated with the initial query concepts: Fire (0.88), Fire-Truck (0.61), Helicopter (0.72), and Missile (0.76). These concepts are added to the visual modal terms' vector and their weights are updated according to their indices:

V_{1*49}	A_{1*17}	T_{1*100}
(0,0.88,0.61,…,0.72,0,1,0,0…,0.76,0)	(0,0,…,0,0,0,0,0…,0,0)	(0,0,…,0,0,0,1,0…,0,0)

For text modal via analyzing Pearson correlation we realize that the following concepts have strong correlations with initial terms which are provided by user: Savaş (0.71), Birleşmiş milletler (0.63), KADDAFİ (0.91), raket (0.64) and Uçuş (0.61). These concepts are added to text modal terms' vector and their weights are updated according to their indices:

V_{1*49}	A_{1*17}	T_{1*100}
(0,0.88,0.61,…,0.72,0,1,0,0…,0.76,0)	(0,0,…,0,0,0,0,0…,0,0)	(0,0.71,…,0.61,0.63,0,1,0.91,…,0.64,0)

Now we start to apply intra-modal step. In this step we utilize CCA to calculate the weight vectors in order to correlate terms between two separate modalities. Since the internal mechanism of CCA fulfills the calculation of the correlations among the terms of two different modalities by transferring the query vector to a different dimensional vector in another space, presenting the weight updates and intermediate values in the another space may be confusing. So we present the intra-modal steps in the initial dimension as a proof of concept.

The query terms that are added to visual part so far are correlated with some terms in audio modal. They can be listed as: Bomb (0.71), Gun (0.78), Helicopter (0.63) and violence (0.77). These terms are added to the audio modality and their weights are updated according to their indices in query vector's audio modal:

V_{1*49}	A_{1*17}	T_{1*100}
(0,0.88,0.61,…,0.72,0,1,0,0…,0.76,0)	(0,0.71,…,0,0.78,0,0.63,0…,0.77,0)	(0,0.71,…,0.61,0.63,0,1,0.91,…,0.64,0)

By propagating correlations inside each modality as well as among modalities we calculate our 166 dimensional query vector. Each shot in database also is presented with a 166 dimensional vector that the value of each index is a score for that concept that is calculated in the data level fusion. Sample vector for a shot is:

V_{1*49}	A_{1*17}	T_{1*100}
(0.6,0.56,0.61,…,0.4,0.1,0.7,…,0.46,0.89)	(0.1,0.7,…,0.23,0.12,0.24,…,0.34,0.23)	(0.3,0.71,…,0.61,0.23,0,1,0.9,…,0.64,0.7)

By using cosine similarity we find the most similar shots to our extended query vector.

6.4 Results and Evaluation

6.4.1 Indexing

First results are related to time evaluation for B^+ Indexing. Total numbers of 12437 objects are used in this evaluation. We start by 2416 objects and increased them in seven steps to cover all objects. It is visible that in large number of objects the time measurement tends to be reduced as shown in Figure 6-3 and Figure 6-4.

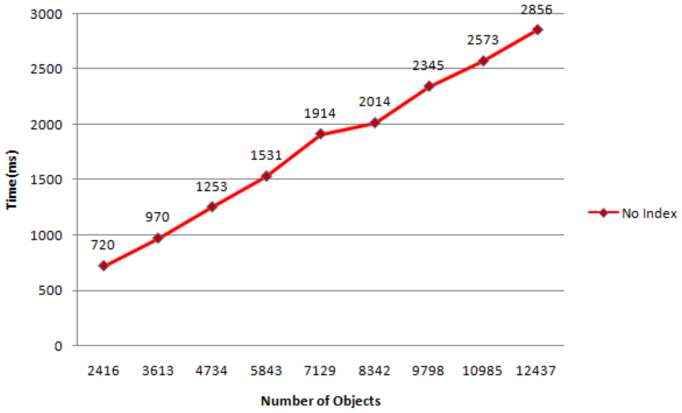


Figure 6-3 Retrieval performance without index

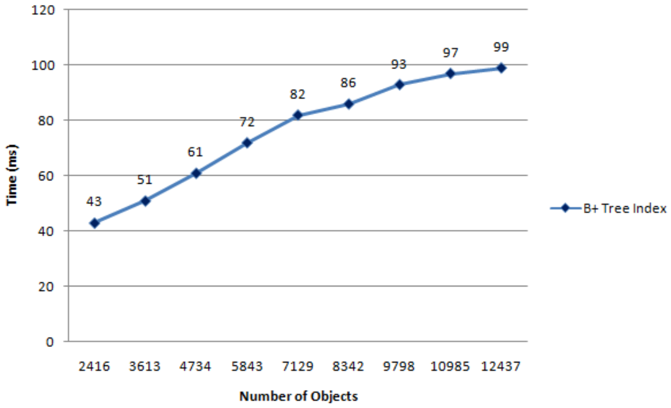


Figure 6-4 Retrieval performance with B^+ indexing

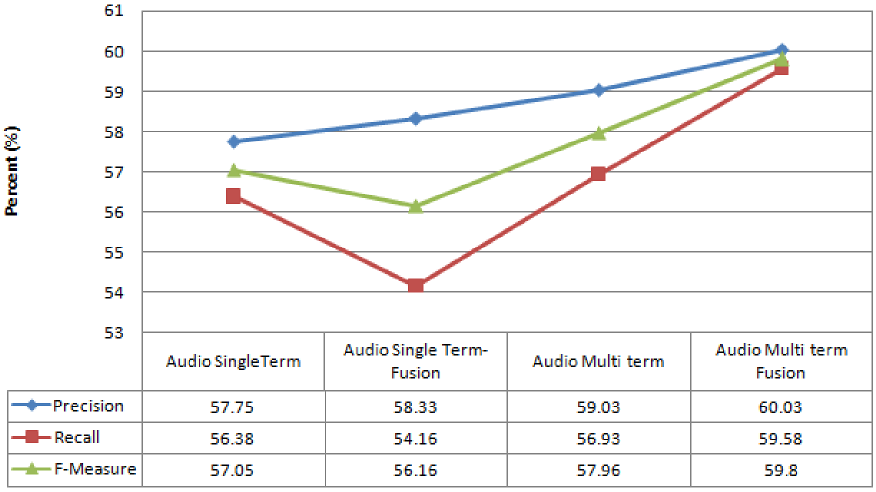
6.4.2 Query by Concept

Our results for query on single modal and multi modal in Boolean model are as following: In audio query by concept (Figure 6-5) we use Boolean model to retrieve queries. Single term uses just one constraint for audio concept and multi term uses “AND” operator between some constraints. The cases that we use fused data as our source are marked with “Fusion”.

As we can conclude from results multi term audio query from fused data gives us the best performance. The reason is that since we use automatic concept extraction and there are some noisy concepts, adding more constraints reduces the number of retrieved noisy shots. Furthermore, Using fused data reflects the intrinsic concepts of shots more correctly.

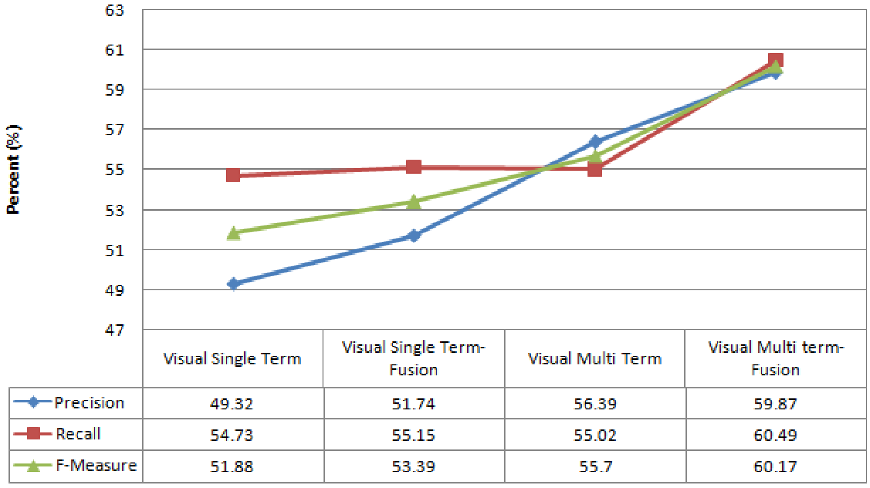
In visual query by concept same justification can be mentioned to define the best performance for combination of multi term and fused data as depicted in Figure 6-6.

Generally in information retrieval domain text retrieval has high performance. In our study this trend can be observed by looking at results in Figure 6-7. Since we use named entity in textual modal and key words in fusion, comparison- as mentioned in audio and visual- is not much reasonable.



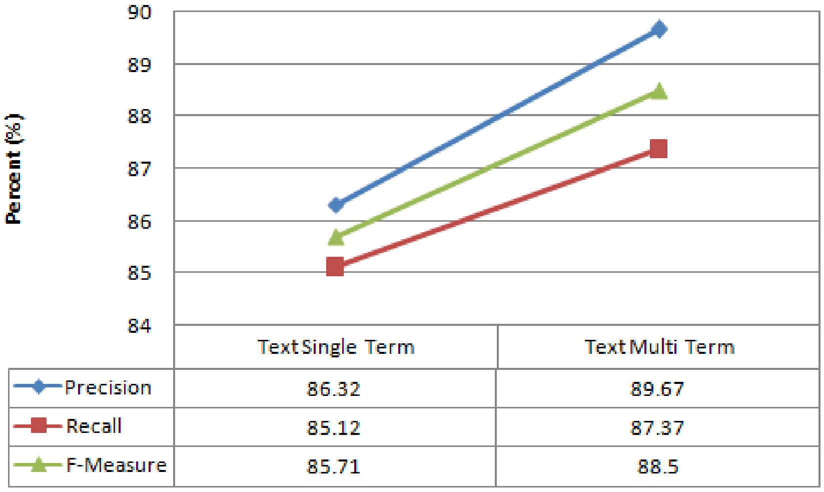
	Audio SingleTerm	Audio Single Term-Fusion	Audio Multi term	Audio Multi term Fusion
Precision	57.75	58.33	59.03	60.03
Recall	56.38	54.16	56.93	59.58
F-Measure	57.05	56.16	57.96	59.8

Figure 6-5 Audio query by concept, Boolean retrieval model



	Visual Single Term	Visual Single Term-Fusion	Visual Multi Term	Visual Multi term-Fusion
Precision	49.32	51.74	56.39	59.87
Recall	54.73	55.15	55.02	60.49
F-Measure	51.88	53.39	55.7	60.17

Figure 6-6 Visual query by concept, Boolean retrieval model



	Text Single Term	Text Multi Term
Precision	86.32	89.67
Recall	85.12	87.37
F-Measure	85.71	88.5

Figure 6-7 Text query, Boolean retrieval model

The overall results for multimodal retrieval in vector space model are summarized in the table shown in (

Figure **6-8**). In this table firs test which named as “Multi-Modal” is a case that all shots and queries are represented as the 166 dimensional vectors in VSM. “QFusion” is query level fusion which is proposed in this work. “LFusion” is data fusion (late fusion) which was proposed in [10].

Here, we examine each fusion separately and then in combination with other fusion method. We observe that combining query fusion and data fusion provides noticeable change in retrieval performance. Furthermore, we apply kernel version of our proposed query level fusion and we observe further increases in retrieval performance.

To our point of view, using data fusion and kernel version of query fusion simultaneously results in the best performance among those models that are used in this thesis. Since full kernel computation in KCCA is time consuming process and this should be considered in query retrieval time, we used Incomplete Cholesky Decomposition (ICD) for kernel matrix. This method is the base for final implementation in our application. In all of these tests we use top 30 shots result as our evaluation set. This value is highly dependent on data set and we decide to set this to 30 by examining retrieval performance on top-25, top-30 and top-35. Due to nature and size of our dataset top 30 give us the best retrieval performance.

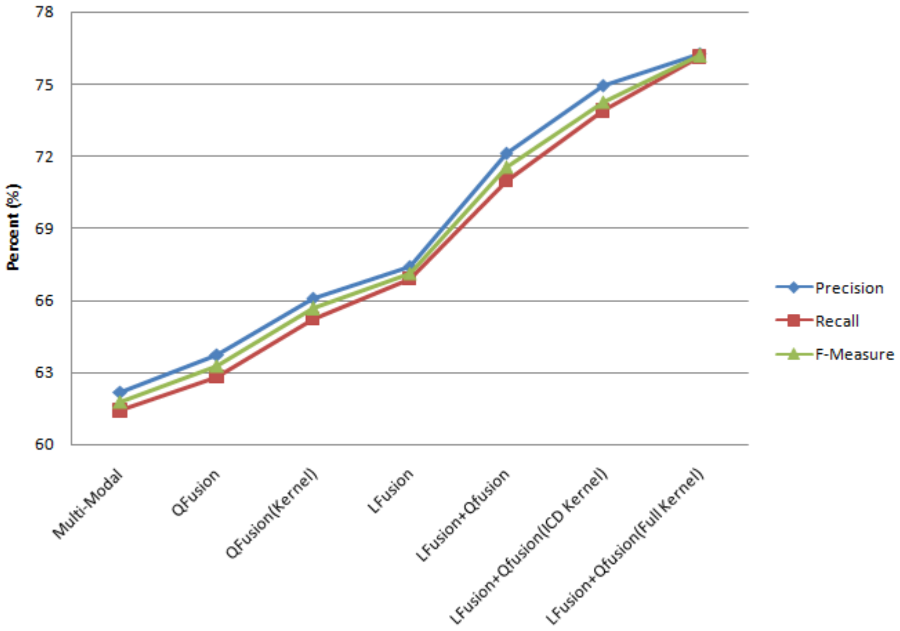
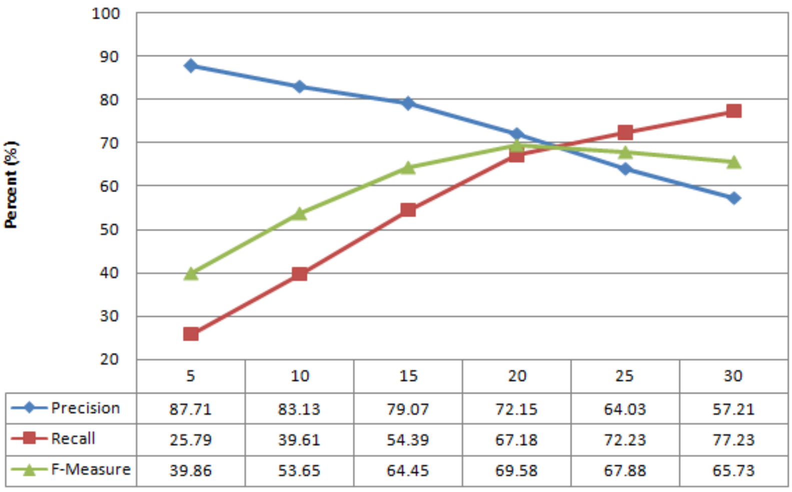


Figure 6-8 Multimodal retrieval in Vector Space Model, top 30

6.4.3 Query by Example

Our final result is for shot QBE. In this test we use our QBE retrieval on different top-*K* results and observe that top-20 is optimal value for our dataset in terms of performance [Figure 6-9]. It should be mention that the values for precision and recall are average of 5 query results. The average time for answering shot QBE in our test is 2.87 seconds.



	5	10	15	20	25	30
Precision	87.71	83.13	79.07	72.15	64.03	57.21
Recall	25.79	39.61	54.39	67.18	72.23	77.23
F-Measure	39.86	53.65	64.45	69.58	67.88	65.73

Figure 6-9 Shot QBE performance, top N

CHAPTER7

CONCLUSION AND FUTURE WORK

Due to recent improvements in multimedia technology and increasing popularity of digital video files, multimedia database systems have gained more importance lately. A multimodal multimedia database system which supplying automatic concept extraction from audio, video and text as well as providing services to store, index and retrieve them seems to be important with this increasing trend in the amount of multimedia data, especially digital videos.

In this thesis, we propose and integrate a multimedia system architecture that provides users with the capabilities to automatic semantic concept extracting, storing, indexing and retrieving of semantic concepts of digital videos. Proposed system allows users to perform various query types such as content and concept based queries. Since a typical video file consists of various data types, handling all modalities simultaneously is a significant necessity to have acceptable retrieval performance. Therefore, the proposed architecture also supports multimodal multimedia data which includes visual, audio and text modals.

We also develop a full software system which is an integration of different modules developed by the members of the multimedia database group. This system is equipped with some graphical user interfaces that provide users with capabilities to perform supported operations in the proposed architecture.

By using query level fusion and utilizing non-linear correlations we also improve query retrieval performance in proposed system. We come up with the hypothesis that by exploiting correlations inside modals and among them we can improve the query retrieval performance. Eventually, we obtain the results that confirm our hypothesis. Furthermore, we observe that the combination of data-level and query-level fusion has the best query retrieval performance among the models that we investigate in this work.

Despite the fact that we try to address some requirements and challenges regarding multimodal multimedia database as well as its architecture integration, the research in this area still can be improved with several approaches. We identify some of them that can extend the ability and performance efficiency of the proposed system:

- Since automatic concept extraction may result in some noisy data about shots' semantic contents, retrieving them via simple query matching would be inefficient. For solving that issue a mechanism to refine the results in advance is a useful idea that can be applied to this system. Although we utilize vector space retrieval model to compensate some of these problems, still some pre-preparation process can be applied to refine noisy data.

- For calculating correlation between modalities a multi-set CCA can be applied to observe if this has any effect on retrieval performance. Since there is not a usable and stable implementation for this method and due to complexity of implementing by ourselves, we are not able to exploit this method.
- Applying different methods of 'user feedback' to increase precision and recall in video retrieval is also recommended as a future work to improve the query retrieval performance.

REFERENCES

[1] M.S.d. Hacid, C. Decleir and J. Kouloumdjian, "A Database Approach for Modeling and Querying Video Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, pp. 729-750, 2000.

[2] T. Yilmaz, Y. Yildirim and A. Yazici, "A Genetic Algorithms Based Classifier for Object Classification in Images," in *ISCIS*, London, 2011, pp. 519-525.

[3] U. Demir, M. Koyuncu, A. Yazici, T. Yilmaz and M. Sert, "Flexible Content Extraction and Querying for Videos," in *FQAS 2011*, Ghent, Belgium, 2011, pp. 460-471.

[4] W. Jonker and M. Petkovic, "An Overview of Data Models and Query Languages for Content-based Video Retrieval," in *Advances in Infrastructure for E-Business, Science, and Education on the Internet*, l'Aquila, Italy, 2000.

[5] R. Datta, D. Joshi, J. Li and J. Wang, "Image retrieval: Ideas, influences, and trends of the new age," in *ACM Computing Surveys (CSUR),* 2008, pp. 1-60.

[6] G. Csurka, C. Dance, L. Fan, J. Willamowski and C. Bray, "Visual categorization with bags of keypoints," *Workshop on Statistical Learning in Computer Vision, ECCV*, vol. 1, p. 22, 2004.

[7] K. Barnard, P. Duygulu, N.D. Freitas, D. Forsyth, D.M.Blei and M.I. Jordan, "Matching Words and Pictures," *Machine Learning Research*, vol. 3, pp. 1107–1135, 2003.

[8] D. Blei, A. Ng and M. Jordan, "Latent Dirichlet Allocation," *JMLR*, vol. 3, pp. 993–1022, 2003.

[9] S. Deerwester, S. Dumais, G. Furnas, T. Landauer and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for*, vol. 41, no. 6, pp. 391-404, 1990.

[10] E. Gulen, T. Yilmaz and A. Yazici, "Multimodal Information Fusion for Semantic Video Analysis," *International Journal of Multimedia Data Engineering and Management*, vol. 3, no. 4, pp. 52-74, 2012.

[11] M.A. Bhatti and U. Rashid, "Exploration and Management of Web Based Multimedia Information Resources," in *International Conference on Systems, Computing Sciences and Software Engineering*, Bridgeport, USA, 2007.

[12] Ch. Dorai, A.Mauthe, F. Nack, L. Rutledge, Th. Sikora and H. Zettl, "Media Semantics: Who Needs It and Why?," in *Multimedia'02, ACM*, Juan-les-Pins, France, 2002.

[13] L. Dunckley, *Multimedia Databases – An Object-Relational Approach*.: ISBN # 0 201 78899 3, 2003.

[14] T. Rolleke, T. Tsikrika and G. Kazai, "A general matrix framework for modeling information retrieval," in *Proceedings of the ACM SIGIR MF/IR*, 2003.

[15] H. Eidenberger, C. Breiteneder and M. Hitz, "A Framework for Visual Information Retrieval," *Visual Languages & Computing*, vol. 14, no. 5, pp. 443-469, 2003.

[16] F. Moelaert, EL. Hadidy, H.J.G. de Poot and D.D. Velthaus, "Multimedia Information Retrieval Framework From theory to practice (ADMIRE)," in *IFIP TC2/WG2.6 Eighth Working Conference on Database Semantics- Semantic Issues in Multimedia Systems*, 1999, pp. 271-290.

[17] J.A. Sánchez and J.A. Arias, ""Content-Based Search and Annotations in Multimedia Digital Libraries," in *ENC'03, Fourth Mexican International Conference on Computer Science*, 2003, pp. 109-117.

[18] A. Kerne, E. Koh, B. Dworaczyk, J.M. Mistrot, H. Choi, S.M. Smith, R. Graeber and D. Caruso, "CombinFormation: A Mixed-Initiative System for Representing Collections as Compositions of Image and Text," in *JCDL'05, ACM/IEEECS Joint Conference on Digital Libraries*, 2006, pp. 11-20.

[19] I.H. Witten and D. Bainbridge, "Building Digital Library Collections with Greenstone," in *JCDL'05, 5th ACM/IEEECS Joint Conference on Digital Libraries*, 2005, pp. 425-425.

[20] M.L. Wilson, "Advanced Search Interfaces considering Information Retrieval and Human Computer Interaction," in *Agents and Multimedia*, Southampton, 2007.

[21] M.C. Schraefel, M. Wilson, A. Russel and D.A. Smith, "MSPACE: Improving Information Access to Multimedia Domains with Multimodal Exploratory Search," *Communication of The ACM*, vol. 49, no. 4, pp. 47-49, 2006.

[22] W. Dunn, "EVIADA: Ethnomusicological Video for Instruction and Analysis Digital Archive," in *JCDL'05, 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, 2005, p. 407.

[23] U. Rashid, I.A. Niaz and M.A. Bhatti, "Unified Multimodal Search Framework for Multimedia Information Retrieval," in *4th International Conference on Systems, Computing Sciences and Software Engineering, Springer*, Bridgeport, USA, 2007.

[24] R. Yan, "Probabilistic Models for Combining Diverse Knowledge Sources in Multimedia Retrieval," in *Ph.D. Thesis*. USA: School of Computer Science, Carnegie Mellon University, 2006.

[25] R. Manmatha, "Multimedia indexing and retrieval," in *Workshop on Challenges in Information Retrieval and Language Modeling*, 2002.

[26] T.Westerveld, T. Ianeva, L. Boldareva, A. P. de Vries and D. Hiemstra, "Combining infomation sources for video retrieval," in *NIST TRECVID-2003*, 2003.

[27] A. Amir, W. Hsu, G. Iyengar, C.-Y.Lin, M. Naphade, A. Natsev, C. Neti, H. J. Nock, J. R. Smith, B. L. Tseng, Y. Wu and D. Zhang, "IBM research TRECVID- video retrieval system," in *NIST TRECVID*, 2003.

[28] G. Gaughan, A. F. Smeaton, C. Gurrin, H. Lee and K. Mc-Donald, "Design, implementation and testing of an interactive video retrieval system," in *11th ACM MM Workshop on MIR*, 2003.

[29] M. Rautiainen and et al., "TRECVID 2004 experiments at mediateam oulu," in *Proc. of TRECVID*, 2004.

[30] Ch. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic and W. Equitz, "Efficient and effective querying by image content," *Journal of Intelligent Information Systems*, vol. 3, no. 3/4, pp. 231–262, 1994.

[31] R. Yan, A. Hauptmann and R. Jin, "Multimedia search with pseudo-relevance feedback," in *Proc. of Intl Conf on Image and Video Retrieval*, 2003, pp. 238-247.

[32] R. Yan, J. Yang and A. G. Hauptmann, "Learning query-class dependent weights in automatic video retrieval," in *Proceedings of the 12th annual ACM international conference on Multimedia*, 2004, pp. 548-555.

[33] T. S. Chua, S. Y. Neo, K. Li, G. H. Wang, R. Shi, M. Zhao, H. Xu abd S. Gao and T.

L. Nwe, "Trecvid 2004 search and feature extraction task by nus pris," in *NIST TRECVID*, 2004.

[34] P. Over and A.F. Smeaton, "TRECVID: Benchmarking the effectiveness of information retrieval tasks on digital video," in *Proc. of the Intl. Conf. on Image and Video Retrieval*, 2003.

[35] B. Huurnink, "AutoSeek: Towards a fully automated video search system," in *M.Sc. Thesis.*: University of Amsterdam, 2005.

[36] J. Yuan, L. Xiao, D. Wang, D. Ding, Y. Zuo, Z. Tong, X. Liu, S. Xu, W. Zheng, X. Li, Z. Si, J. Li, F. Lin and B. Zhang, "Tsinghua university at TRECVID 2005," in *NIST TRECVID 2005*, 2005.

[37] L. Kennedy, P. Natsev and S.F. Chang, "Automatic discovery of query class dependent models for multimodal search," in *ACM Multimedia*, Singapore, 2005.

[38] R. Yan and A. G. Hauptmann, "Probabilistic latent query analysis for combining multiple retrieval sources," in *Proceedings of the 29th international ACM SIGIR conference*, Seattle, WA, 2006.

[39] J. Yu, Y. Cong, Z. Qin and T. Wan, "Cross-Modal Topic Correlations for Multimedia Retrieval," in *ICPR*, Tsukuba, Japan, 2012.

[40] Y. Song, L. Philippe Morency and R. Davis, "Multimodal Human Behavior Analysis: Learning Correlation and Interaction Across Modalities," in *ICMI*, Santa Monica, California, USA, 2012, pp. 22-26.

[41] W. Jiang and A. C. Loui, "Video concept detection by audio-visual grouplets," *Multimedia Information Retrieval*, vol. 1, pp. 223–238, 2012.

[42] W. Lin, T. Lu and F. Su, "A Novel Multi-modal Integration and Propagation Model for Cross-Media Information Retrieval," in *LNCS 7131, Springer*, 2002, pp. 740–749.

[43] J.D. Zeng, H.J. Zheng, C. Lu, T. Li and Ma. Wang, "ReCoM: Reinforcement Clustering of Multi-Type Interrelated Data Objects," in *ACM SIGIR*, Canada, 2003, pp. 274–281.

[44] X. Wang, X.J. Ma, W.Y. Xue and G.R. Li, "Multi-Model Similarity Propagation and its Application for Web Image Retrieval," in *ACM Multimedia*, 2004, pp. 944-951.

[45] H. Zhang, Y.T. Zhuang and F.H. Wu, "Cross-Modal Correlation Learning for Clustering on Image-Audio Dataset," in *ACM Multimedia*, 2007, pp. 273-276.

[46] Y. Yang, Y.T. Zuang, F. Wu and Y.H. Pan, "Harmonizing Hierarchical Manifolds for Multimedia Document Semantics Understanding and Cross-media Retrieval," in *IEEE Transactions*, 2008, pp. 437-446.

[47] D.M. Blei and M.I. Jordan, "Modeling Annotated Data," in *ACM SIGIR*, Toronto, Canada, 2003, pp. 127-134.

[48] M. Borga, "Learning Multidimensional Signal Processing," in *Ph.D. Thesis.*: Linkping Studies in Science and Technology, 1998.

[49] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, pp. 312-377, 1936.

[50] P. L. Lai and C. Fyfe, "Kernel and nonlinear canonical correlation analysis," *International Journal of Neural Systems*, 2001.

[51] Sh. Akaho, "A kernel method for canonical correlation analysis," in *International Meeting of Psychometric Society*, Osaka, 2001.

[52] A. Vinokourov, J.S.Taylor and N. Cristianini, "Inferring a semantic representation of text via cross-language correlation analysis," in *Advances of Neural Information Processing Systems*, 2002.

[53] J. M. Martinez. Overview of the MPEG-7 Standard, ISO/IEC-JTC1/SC29/WG11. http://mpeg.chiariglione.org/technologies/mpeg-7/mp07-rsw/index.htm, last visited on May 2013.

[54] Ç. Okuyucu, M. Sert and A. Yazıcı, "Environmental Sound Classification Using Spectral," in *IEEE*, Turkey, 2013.

[55] H.G. Kim, N. Moreau and T. Sikora, *MPEG-7 audio and beyond.*: Wiley Online Library, 2005.

[56] Object Database. http://en.wikipedia.org/wiki/Object_database, last visited on May 2013.

[57] DB4O. http://www.db4o.com/about/productinformation/db4o, last visited on May 2013.

[58] Naphade and et al., "A Large Scale Concept Ontology for Multimedia Understanding," Mitre Corporation, ppt presentation.

[59] M. R. Naphade and et al., "Large-Scale Concept Ontology for Multimedia," *IEEE MultiMedia*, vol. 13, no. 3, pp. 86-91, July-September 2006.

[60] wikipedia. ORACLE. http://en.wikipedia.org/wiki/Java_Media_Framework, last visited on May 2013.

[61] Java Web Start (JavaWS). http://en.wikipedia.org/wiki/Java_Web_Start, last visited on May 2013.

[62] ORACLE. Java Web Start Witepaper. http://java.sun.com/javase/technologies/desktop/webstart/, last visited on May 2013.

[63] ORACLE. JNLP. http://www.oracle.com/technetwork/java/javase/index-142562.html, last visited on May 2013.

[64] ORACLE. Java Network Launch Protocol. http://docs.oracle.com/javase/tutorial/deployment/deploymentInDepth/jnlp.html, last visited on May 2013.

[65] JNA wiki. http://en.wikipedia.org/wiki/Java_Native_Access, last visited on May 2013.

[66] JNI ORACLE. http://docs.oracle.com/javase/6/docs/technotes/guides/jni/, last visited on May 2013.

[67] jna java. http://jna.java.net/, last visited on May 2013.

[68] Servlet. ORACLE. http://www.oracle.com/technetwork/java/index-jsp-135475.html, last visited on May 2013.

[69] WiKi Servlet. http://en.wikipedia.org/wiki/Java_Servlet, last visited on May 2013.

[70] Tomcat. WiKi. http://en.wikipedia.org/wiki/Apache_Tomcat, last visited on May 2013.

[71] Tomcat. APACHE. http://tomcat.apache.org/, last visited on May 2013.

[72] Serialization. WiKi. http://en.wikipedia.org/wiki/Serialization, last visited on May

2013.

[73] Serialization.ORACLE.http://docs.oracle.com/javase/7/docs/platform/serialization/spec/serialTOC.html, last visited on May 2013.

[74] Y. Deng and B.S.Manjunath. UC Santa Barbara. http://vision.ece.ucsb.edu/segmentation/jseg/, last visited on May 2013.

[75] D. R. Hardoon, S. Szedmak and J.S. Taylor, "Canonical correlation analysis; An overview with application to learning methods," Royal Holloway, University of London, Technical Report CSD-TR-03-02, 2003.

[76] (2012, February) News channel of Turkey. http://www.ntvmsnbc.com/, last visited on May 2013.

[77] M.Aydınlılar and A. Yazici, "Semi-Automatic Semantic Video Annotation Tool," in *ISCIS 2012*, Paris, Turkey, 2012, pp. 303-310.

[78] Y. Yıldırım and A. Yazici, "Ontology-Supported Video Modeling and Retrieval," in *Adaptive Multimedia Retrieval* , Geneva, Switzerland, 2006, pp. 28-41.

[79] D. Küçük and A. Yazici, "Exploiting information extraction techniques for automatic semantic video indexing with an application to Turkish news videos," *Knowledge-Based Systems*, vol. 25, no. 6, pp. 844-857 , August 2011.

[80] J. Allan. Information Retrieval. http://ciir.cs.umass.edu/~allan/, last visited on May 2013.

[81] G. Salton, *Introduction to Modern Information Retrieval*.: McGraw-Hill, 1983.

[82] C. Carpineto and G. Romano, "Effective reformulation of Boolean queries with concept lattices.," *Datalogiske Skrifter, Univ. Roskilde*, vol. 78, pp. 83-95, 1998.

[83] K. Won Yong, K. Myoung Ho, L. Yoon Joon, and L. Joon Ho, "On the evaluation of boolean operators in the extended boolean retrieval framework," in *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Infofmation Retrieval*, 1993, pp. 291-297.

[84] O.W. Won, M.C. Kim and K.S. Choi, "Query expansion using domain-adapted, weighted thesaurus in an extended Boolean model," in *CIKM 94, Proceedings of the Third International Conference on Information and Knowledge Management*, 1994,

pp. 140-146.

[85] G. Bordogna, P. Carrara and G. Pasi, "Extending Boolean information retrieval: A fuzzy model based on linguistic variables," in *IEEE International Conference on Fuzzy Systems*, 1992, pp. 769-776.

[86] C. J. Van Rijsbergen, *Information retrieval*.: Butterworths, 1979.

[87] S. Chakrabarti, B. Dom, R. Agrawal and P. Raghavan, "Scalable feature selection, classification and signature generation for organizing large text databases into hierarchical topic taxonomies," *VLDB Journal*, vol. 7, no. 3, pp. 163-178, 1998.

[88] A. Bookstein, S. T. Klein and T. Raita, in *Detecting content bearing words by serial clustering. SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, 1995, pp. 319-327.

[89] H. P. Luhn, "The Automatic Creation of Literature Abstracts," *IBM Journal of Research and Development*, vol. 2, no. 2, pp. 159-165 and 317, 1958.

[90] G. Salton and Ch. Buckley, "Term Weighting Approaches in Automatic Text Retrieval," *Information Processing and Management*, vol. 32, no. 4, pp. 431-443, 1996.

[91] G. Salton, *Automatic Text Processing*.: Addison-Wesley Publishing Company, 1988.

[92] F. Hair Joseph and et al., *Multivariate Data Analysis*, 5th, Ed.: Prentice Hall, 1998.

[93] S.Y. Huang and C.R. Hwang, "Kernel Fisher discriminant analysis in Gaussian reproducing kernel Hilbert space," Institute of Statistical Science, Academia Sinica, Taiwan, Technical report 2005.

[94] F. Bach and M. Jordan, "Kernel independent component analysis," *Journal of Machine Leaning Research*, vol. 3, pp. 1-48, 2002.

[95] JNI WiKi. http://en.wikipedia.org/wiki/Java_Native_Interface, last visited on May 2013.

[96] P. Duygulu, D. Forsyth, N. De Freitas, D. Blei, and M. Jordan and K. Barnard, "Matching words and pictures," *JMLR*, vol. 3, pp. 1107–1135, 2003.

[97] W. Jiang, "Advanced Techniques for Semantic Concept Detection in General

Videos," in *Ph.D. Thesis*.: COLUMBIA UNIVERSITY, 2010.

[98] T.S. Chua, S.Y. Neo, H.K. Goh, M. Zhao, Y. Xiao and G. Wang, "Trecvid 2005 by nus pris," in *NIST TRECVID-2005*, 2005.

[99] R. Nevatia and P. Natarajan, "EDF: A framework for Semantic Annotation of Video," in *Tenth IEEE International Conference on Computer Vision Workshops (ICCVW'05)*, 2005, p. 1876.

[100] D. L. Lee, H. Chuang and K. Seamons, "Document ranking and the vector-space model," *IEEE Software*, vol. 14, no. 2, pp. 67-75, 1997.

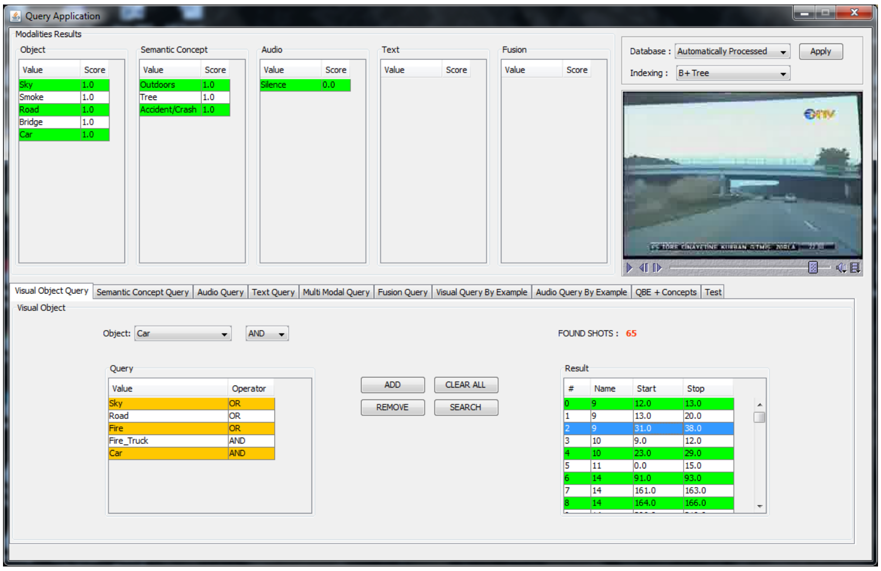
[101] Y. Wang, Z. Liu and J.C. Huang, "Multimedia content analysis-using both audio and visual clues," *Signal Processing Magazine, IEEE*, vol. 17, no. 6, pp. 12-36, 2000.

[102] B. Furht and O. Marques, "MUSE: A Content-Based Image Search and Retrieval System Using Relevance Feedback," *Multimedia Tools Appl.*, vol. 17, pp. 21-50, 2002.

[103] A. Ekin, A. M. Tekalp and R. Mehrotra, "Integrated Semantic-Syntactic Video Modeling for Search and Browsing," *IEEE Transactions on Multimedia*, vol. 6, p. 839.

APPENDICES

A: Application User Interfaces



Database : Automatically Processed

Indexing : B+ Tree

Object

Value	Score
Sky	1.0
Smoke	1.0
Road	1.0
Bridge	1.0
Car	1.0

Semantic Concept

Value	Score
Outdoors	1.0
Tree	1.0
Accident/Crash	1.0

Audio

Value	Score
Silence	0.0

Text

Value	Score

Fusion

Value	Score

Object: Car AND

FOUND SHOTS : 65

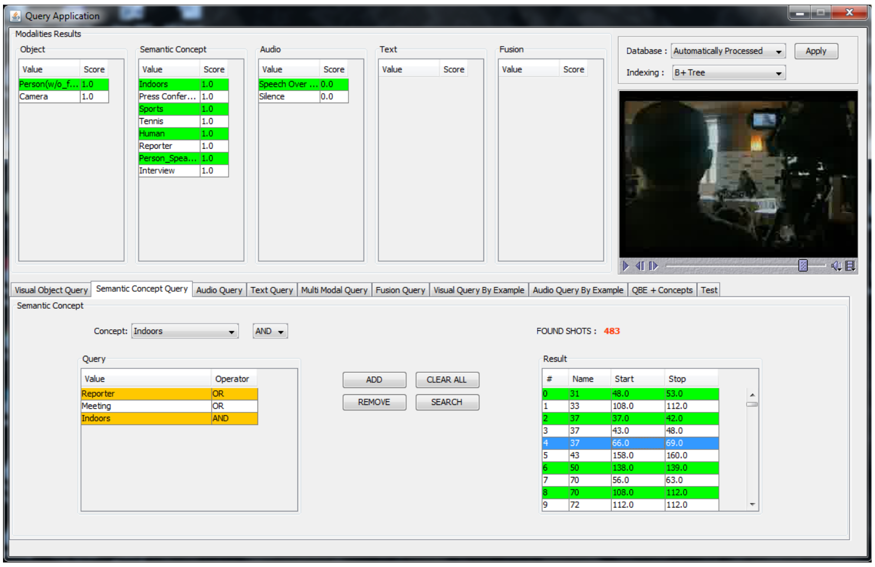
Query

Value	Operator
Sky	OR
Road	OR
Fire	OR
Fire_Truck	AND
Car	AND

Result

#	Name	Start	Stop
0	9	12.0	13.0
1	9	13.0	20.0
2	9	31.0	38.0
3	10	9.0	12.0
4	10	23.0	29.0
5	11	0.0	15.0
6	14	91.0	93.0
7	14	161.0	163.0
8	14	164.0	166.0

Figure 9-1 Object query by concept



Database : Automatically Processed

Indexing : B+ Tree

Object

Value	Score
Person(w/o_f...	1.0
Camera	1.0

Semantic Concept

Value	Score
Indoors	1.0
Press Confer...	1.0
Sports	1.0
Tennis	1.0
Human	1.0
Reporter	1.0
Person_Spea...	1.0
Interview	1.0

Audio

Value	Score
Speech Over ...	0.0
Silence	0.0

Text

Value	Score

Fusion

Value	Score

Concept: Indoors AND

FOUND SHOTS : 483

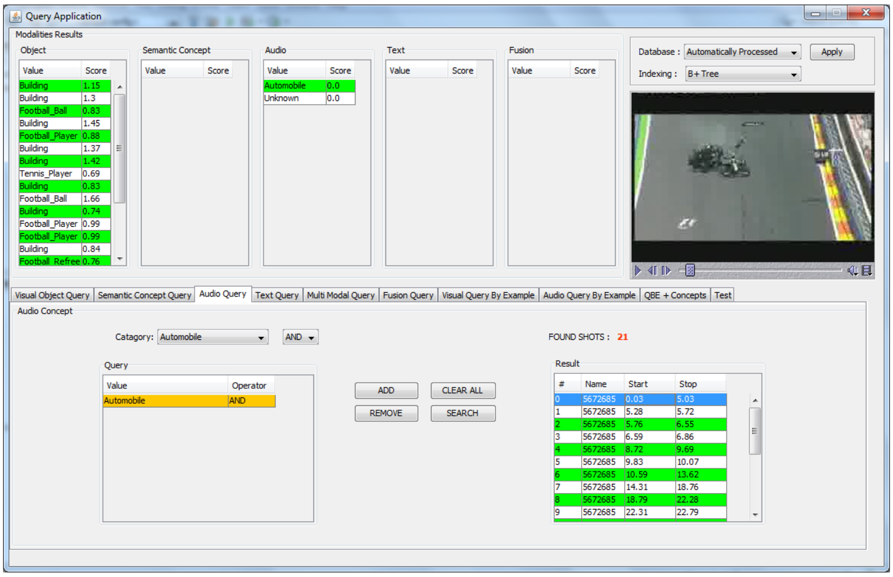
Query

Value	Operator
Reporter	OR
Meeting	OR
Indoors	AND

Result

#	Name	Start	Stop
0	31	48.0	53.0
1	33	108.0	112.0
2	37	37.0	42.0
3	37	43.0	48.0
4	37	66.0	69.0
5	43	158.0	160.0
6	50	138.0	139.0
7	70	56.0	63.0
8	70	108.0	112.0
9	72	112.0	112.0

Figure 9-2 Semantic Query by concept



Object

Value	Score
Building	1.15
Building	1.3
Football_Ball	0.83
Building	1.45
Football_Player	0.88
Building	1.37
Building	1.42
Tennis_Player	0.69
Building	0.83
Football_Ball	1.66
Building	0.74
Football_Player	0.99
Football_Player	0.99
Building	0.84
Football_Refree	0.76

Semantic Concept

Value	Score

Audio

Value	Score
Automobile	0.0
Unknown	0.0

Text

Value	Score

Fusion

Value	Score

Database : Automatically Processed

Indexing : B+ Tree

Audio Concept

Catagory: Automobile AND

FOUND SHOTS : 21

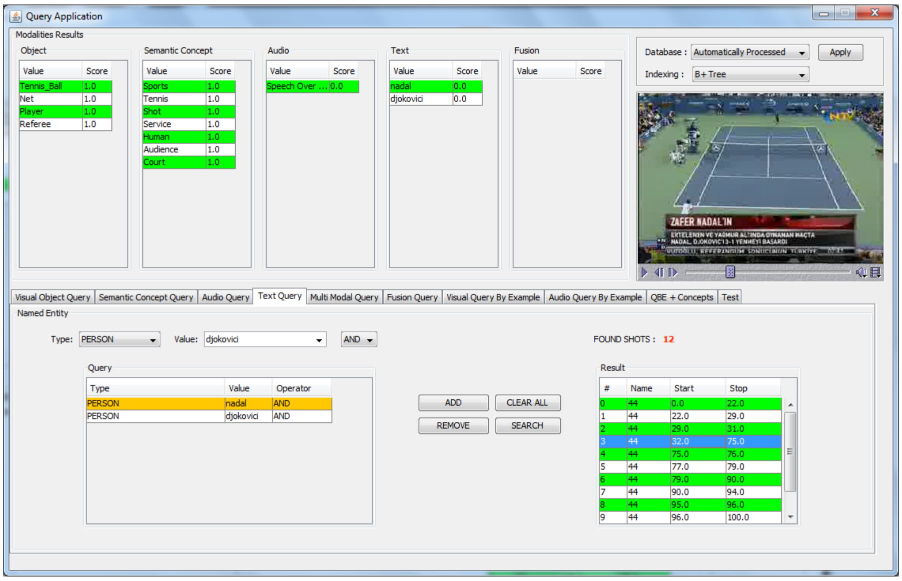
Query

Value	Operator
Automobile	AND

Result

#	Name	Start	Stop
0	5672685	0.03	5.03
1	5672685	5.28	5.72
2	5672685	5.76	6.55
3	5672685	6.59	6.86
4	5672685	8.72	9.69
5	5672685	9.83	10.07
6	5672685	10.59	13.62
7	5672685	14.31	18.76
8	5672685	18.79	22.28
9	5672685	22.31	22.79

Figure 9-3 Audio query by concept



Object

Value	Score
Tennis_Ball	1.0
Net	1.0
Player	1.0
Referee	1.0

Semantic Concept

Value	Score
Sports	1.0
Tennis	1.0
Shot	1.0
Service	1.0
Human	1.0
Audience	1.0
Court	1.0

Audio

Value	Score
Speech Over ...	0.0

Text

Value	Score
nadal	0.0
djokovici	0.0

Fusion

Value	Score

Database : Automatically Processed

Indexing : B+ Tree

Named Entity

Type: PERSON Value: djokovici AND

FOUND SHOTS : 12

Query

Type	Value	Operator
PERSON	nadal	AND
PERSON	djokovici	AND

Result

#	Name	Start	Stop
0	44	0.0	22.0
1	44	22.0	29.0
2	44	29.0	31.0
3	44	32.0	75.0
4	44	75.0	76.0
5	44	77.0	79.0
6	44	79.0	90.0
7	44	90.0	94.0
8	44	95.0	96.0
9	44	96.0	100.0

Figure 9-4 Text query by concept

B: Correlation Matrices

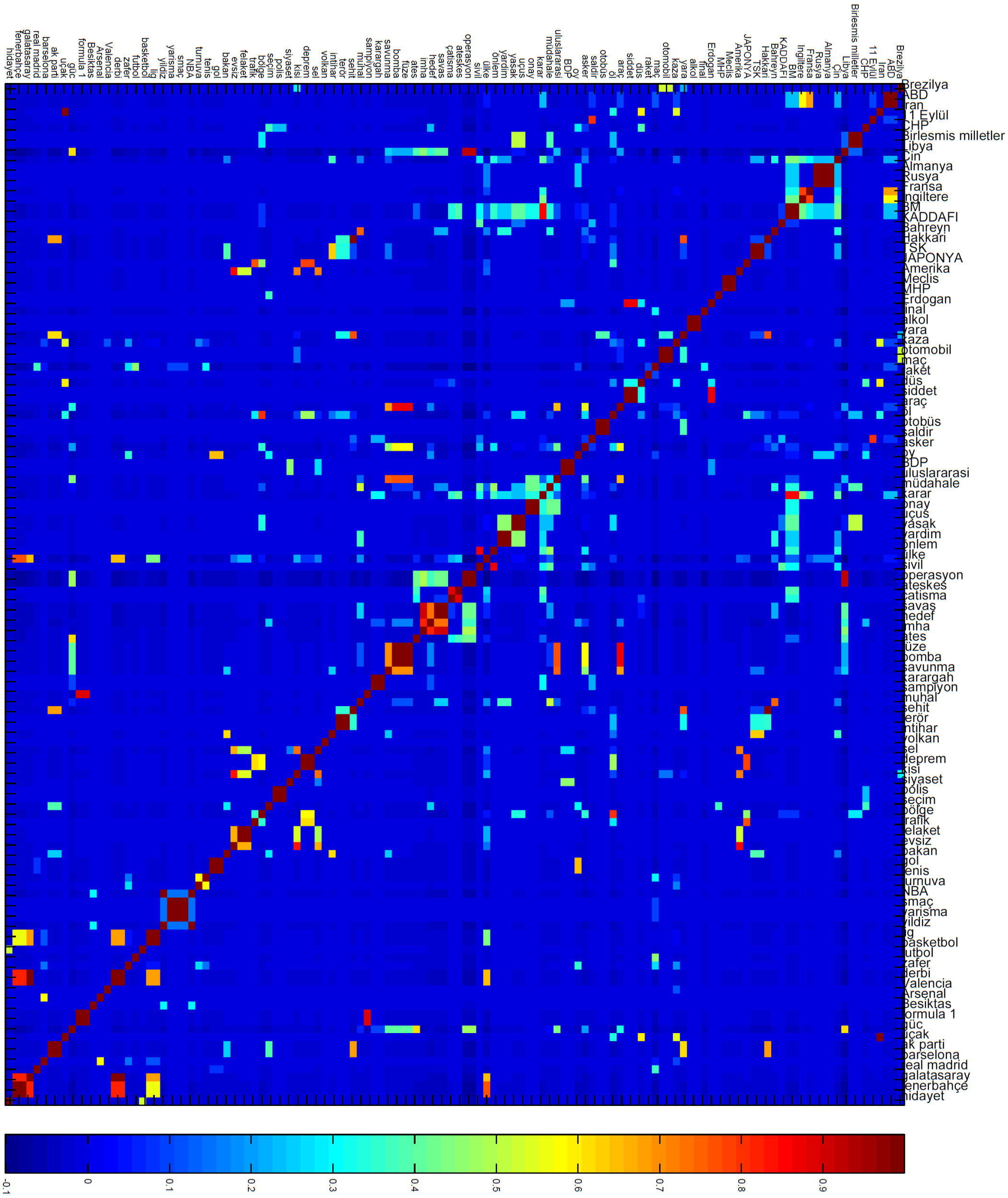


Figure 10-1 Text correlation matrix

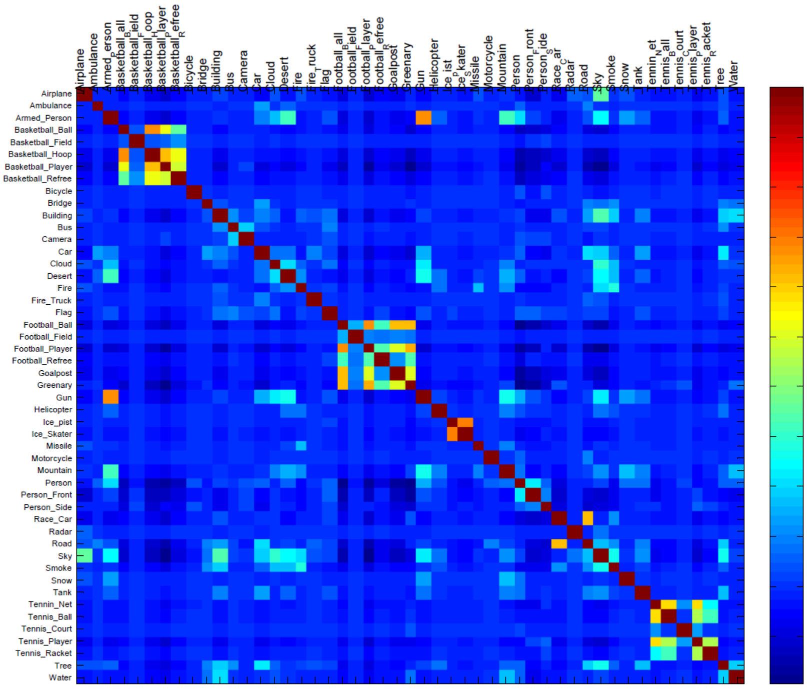


Figure 10-2 Visual correlation matrix

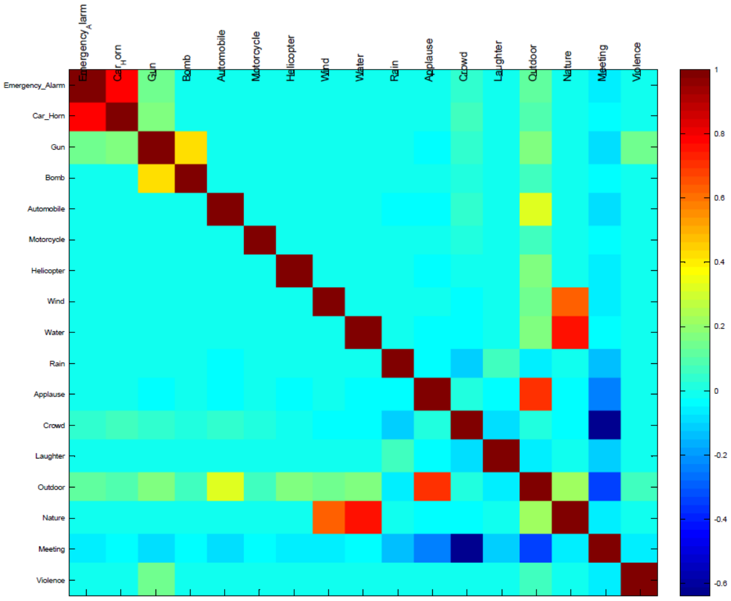


Figure 10-3 Audio correlation matrix

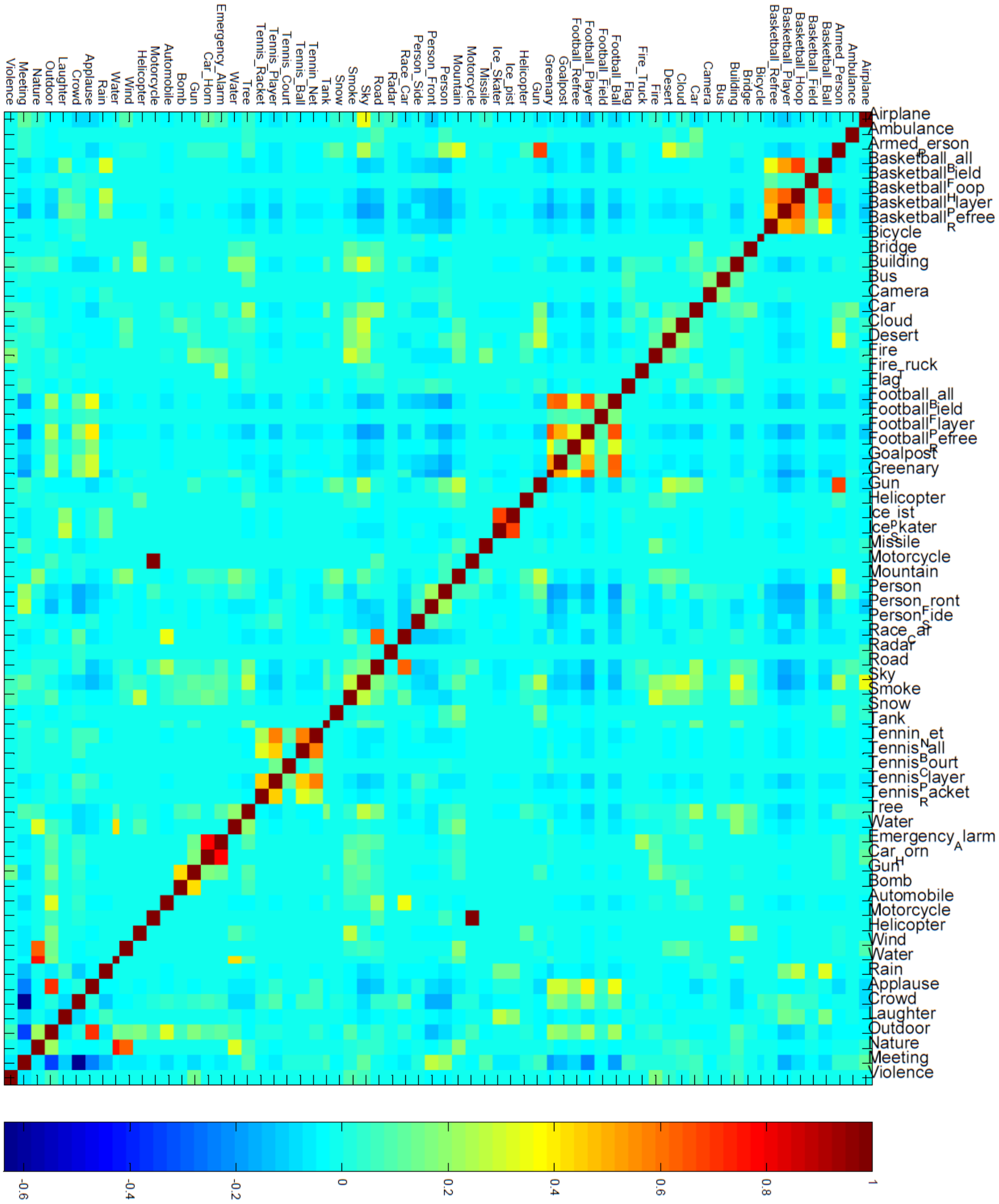


Figure 10-4 Visual-Audio correlation matrix

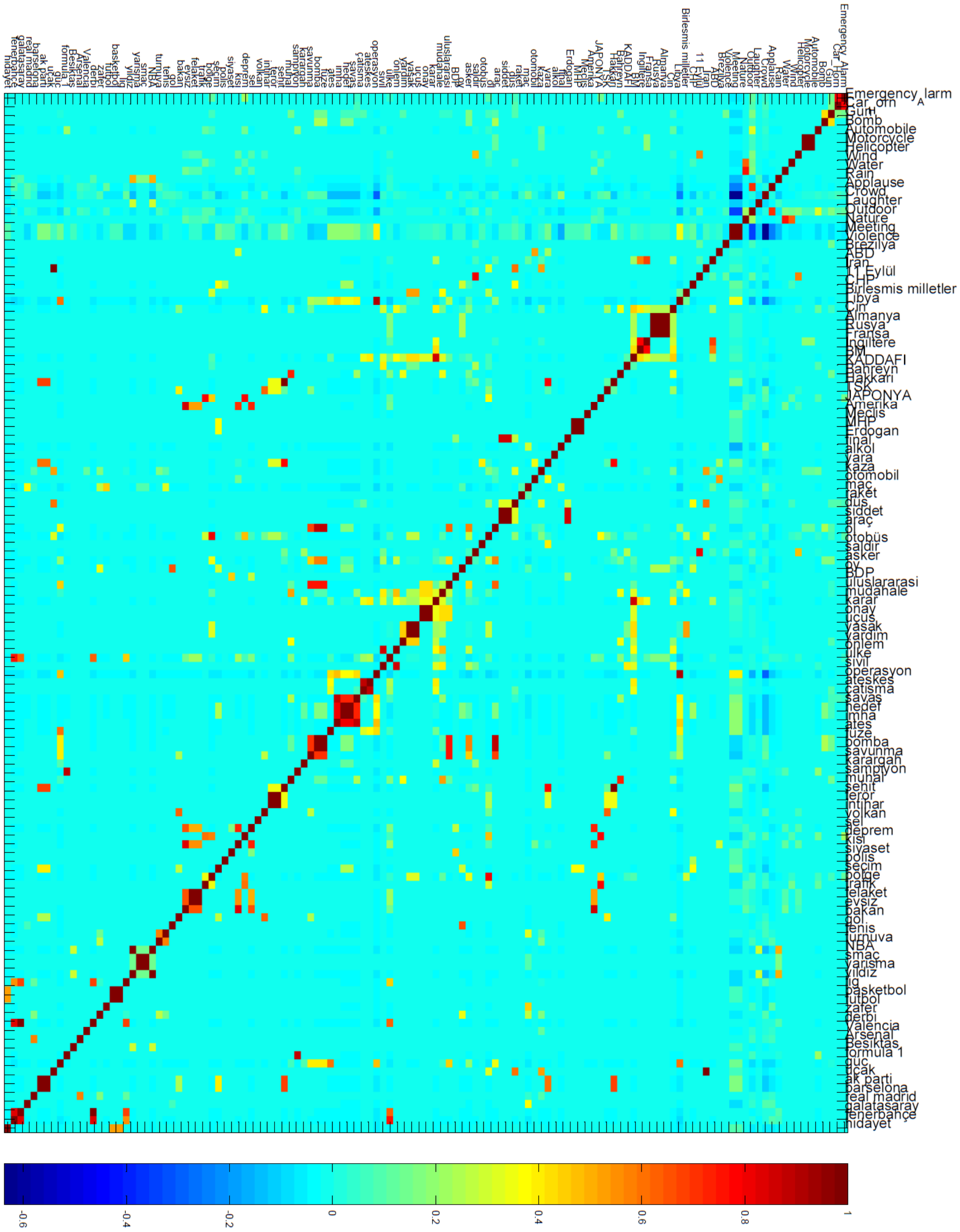


Figure 10-5 Audio-Text correlation matrix

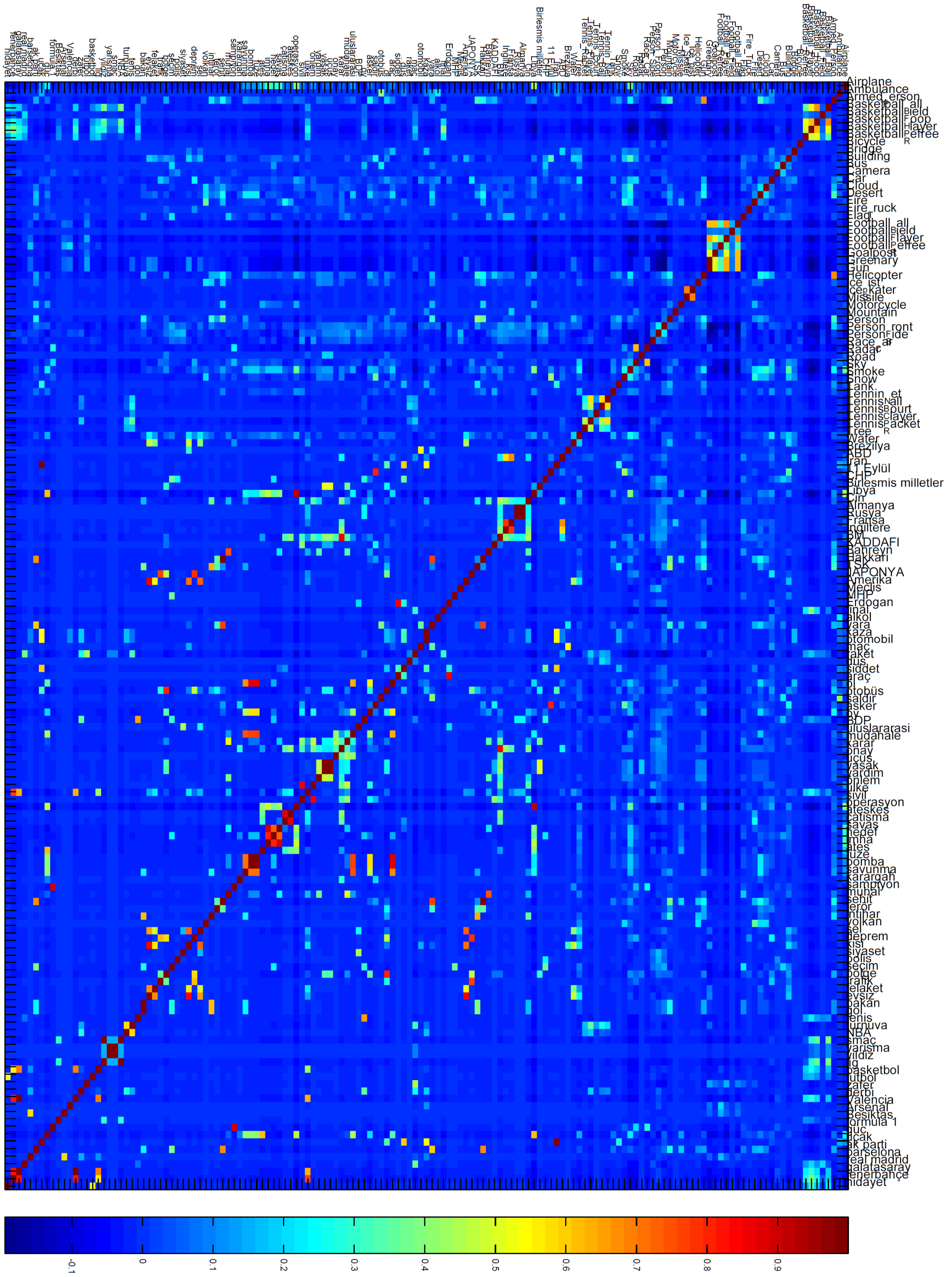


Figure 10-6 Visual-Text correlation matrix