A. D. MCCXXIV

UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II

Scuola di Dottorato in Ingegneria dell'Informazione
Dottorato di Ricerca in Ingegneria Informatica ed Automatica

Comunità Europea
Fondo Sociale Europeo

Indexing Techniques for Image and Video Databases: an approach based on Animate Vision Paradigm

Vincenzo Moscato

Tesi di Dottorato di Ricerca

Novembre 2005

via Claudio, 21- I-80125 Napoli - ☎ *[#39]* (0)81 768 3813 - *[#39]* (0)81 768 3816

A. D. MCCXXIV

UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II

Scuola di Dottorato in Ingegneria dell'Informazione
Dottorato di Ricerca in Ingegneria Informatica ed Automatica

Comunità Europea
Fondo Sociale Europeo

Indexing Techniques for Image and Video Databases: an approach based on Animate Vision Paradigm

Vincenzo Moscato

Tesi di Dottorato di Ricerca

(XVIII ciclo)

Novembre 2005

Il Tutore

Prof. Angelo Chianese

Il Coordinatore del Dottorato

Prof. Lugi Cordella

Dipartimento di Informatica e Sistemistica

Table of Contents

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Professor Angelo Chianese, and Professors Antonio Picariello and Lucio Sansone for their academic guidance, critiques of my ideas, emotional support, and encouragement. My research could not have been finished reasonably without insightful advice from them.

Moreover, I would like to sincerely thank my PhD course coordinator Professor Luigi Cordella for his comments and useful suggestions and all my friends and colleagues, in particular Antonio, Carmine, Flora, Luigi, and Vittorio, for their support and suggestions.

Finally, I am deeply grateful to my family, especially my parents, grandparents and brothers for their love, support, and patience during these years. This thesis would not have been possible without their support and confidence.

List of Tables

List of Figures

ABSTRACT

English Version

In this dissertation some novel indexing techniques for video and image database based on `Animate Vision` Paradigm are presented and discussed.

From one hand, it will be shown how, by embedding within image inspection algorithms active mechanisms of biological vision such as saccadic eye movements and fixations, a more effective query processing in image database can be achieved.

In particular, it will be discussed the way to generate two fixation sequences from a query image I_q and a test image I_t of the data set, respectively, and how to compare the two sequences in order to compute a possible similarity (consistency) measure between the two images.

Meanwhile, it will be shown how the approach can be used with classical clustering techniques to discover and represent the hidden semantic associations among images, in terms of categories, which, in turn, allow an automatic pre-classification (indexing), and can be used to drive and improve the query processing.

Eventually, preliminary results will be presented and the proposed approach compared with the most recent techniques for image retrieval described in the literature.

From the other one, it will be discussed how by taking advantage of such foveated representation of an image, it is possible to partitioning of a video into shots.

More precisely, the shot-change detection method will be based on the computation, at each time instant, of the consistency measure of the fixation sequences generated by an ideal observer looking at the video.

The proposed scheme aims at detecting both abrupt and gradual transitions between shots using a single technique, rather than a set of dedicated methods.

Results on videos of various content types are reported and validate the proposed approach.

Italian Version

In questo lavoro di tesi vengono presentate e discusse delle innovative tecniche di indicizzazione per database video e di immagini basate sul paradigma della `Animate Vision` (Visione Animata).

Da un lato, sara' mostrato come utilizzando, quali algoritmi di analisi di una data immagine, alcuni meccanismi di visione biologica, come i movimenti saccadici e le fissazioni dell'occhio umano, sia possibile ottenere un query processing in database di immagini piu' efficace ed efficiente.

In particolare, verranno discussi, la metodologia grazie alla quale risulta possibile generare due sequenze di fissazioni, a partire rispettivamente, da un'immagine di query I_q ed una di test I_t del data set, e, come confrontare tali sequenze al fine di determinare una possibile misura della similarita' (consistenza) tra le due immagini.

Contemporaneamente, verra' discusso come tale approccio unito a tecniche classiche di clustering possa essere usato per scoprire le associazioni semantiche nascoste tra immagini, in termini di categorie, che, di contro, permettono un'automatica pre-classificazione (indicizzazione) delle immagini e possono essere usate per guidare e miglioare il processo di query.

Saranno presentati, infine, dei risultati preliminari e l'approccio proposto sara' confrontato con le pi recenti tecniche per il recupero di immagini descritte in letteratura.

Dall'altro lato, sara' mostrato come utilizzando la precedente rappresentazione "foveata" di un'immagine, risulti possibile partizionare un video in shot.

Piu' precisamente, il metodo per il rilevamento dei cambiamenti di shot si basera' sulla computazione, in ogni istante di tempo, della misura di consistenza tra le sequenze di fissazioni generate da un osservatore ideale che guarda il video.

Lo schema proposto permette l'individuazione, attraverso l'utilizzo di un'unica tecnica anziche' di piu' metodi dedicati, sia delle transizioni brusche sia di quelle graduali.

Vengono infine mostrati i risultati ottenuti su varie tipologie di video e, come questi, validano l'approccio proposto.

Chapter 1

Introduction

1.1 Thesis Purposes

Managing in an efficient way multimedia information in database systems represent by now an open challenge of research activity on multimedia.

Today, the management of multimedia information such as images, graphics, video, audio, and text, is of great interest in a lot of application fields like: Information Retrieval, Office Automation, E-learning, Virtual Museums, Newspaper and Magazines production, Video and Cinema Editing, Medical and Bio-informatics Applications (e.g., Radiographic and DNA-Sequences Archives), Geographical Information Systems Management, Biometric, Security Applications (including Video Surveillance), Remote Sensing and Meteorology and so on.

The spatial, temporal, storage, retrieval, integration, and presentation features of multimedia data calls for new processing beyond the ability of traditional database architecture. For these reasons, in the past decade, the first `MultiMedia Database Management Systems (MMDBMS)` were carried out with the aim of managing in a more efficient way heterogeneous data like image, text, audio or video.

For the MMDBMS to serve its expected purposes, it must meet certain special requirements:

- Huge capacity storage management
- Information retrieval capabilities
- Media integration, composition, and presentation
- Multimedia query support
- Multimedia interface and interactivity
- High performances

In addressing these requirements, in this work some novel indexing techniques for video and image database based on Animate Vision Paradigm are discussed.

From one hand, it will be shown how, by embedding within image inspection algorithms active mechanisms of biological vision such as saccadic eye movements and fixations, a more effective query processing in image database can be achieved. In particular, it will be discussed the way to generate two fixation sequences from a query image I_q and a test image I_t of the data set, respectively, and how to compare the two sequences in order to compute a possible similarity (consistency) measure between the two images. Meanwhile, it will be shown how the approach can be used to discover and represent the hidden semantic associations among images, in terms of categories, which, in turn, allow an automatic pre-classification (indexing), and can be used to drive and improve the query processing. Eventually, preliminary results will be presented and the proposed approach compared with the most recent techniques for image retrieval described in the literature.

From the other one, it will be discussed how by taking advantage of such foveated representation of an image, it is possible to partitioning of a video into shots. More precisely, the shot-change detection method will be based on the computation, at each time instant, of the consistency measure of the fixation sequences generated by an ideal observer looking at the video. The proposed scheme aims at detecting both abrupt and gradual transitions between shots using a single technique, rather than a set of dedicated methods. Results on videos of various content types are reported and validate the proposed approach.

1.2 Outline of Thesis

The thesis is organized as following:

- in the second chapter an overview on requirements and issues in managing multimedia information inside database system is presented;
- in the third chapter the state of the art of open challenges in developing image and video database system is reported;
- in the fourth chapter the Animate Vision model for image analysis is illustrated;
- in the fifth chapter a Content Based Image Retrieval System for context-sensitive queries is discussed;
- in the sixth chapter a Video Segmentation system based on a foveated video analysis is described;
- in the seventh chapter conclusions and final remarks are discussed.

Chapter 2

Multimedia Databases Management Systems

2.1 Introduction

The spatial, temporal, storage, retrieval, integration, and presentation requirements of multimedia data differ significantly from those for traditional data. A MultiMedia Database Management System (MMDBMS) has to provide for the efficient storage and manipulation of multimedia data in all its varied forms.

In this chapter, the basic nature of multimedia data has looked into, highlighting the need for MMDBMSs, and discussing the requirements and issues necessary for developing such systems.

2.2 Towards the first Multimedia Database Management Systems

In the last years, the database research field has been quite active for discovering, from one hand, more efficient methods to manage traditional alphanumeric data and, from the other one, to deal with new types of data as images, audio and video.

When multimedia data were first brought into a database environment, they underwent a natural transformation in order to assume a representable shape for existing architectures. Thus, when images were first managed in a database, numerous techniques for representing them, first in a relational architecture, then in an object-oriented architecture have been proposed.

In the relational architecture, a multimedia object and its content are represented by means of sets of tuples over several relations. Researchers initially believed that such kind of representation was suitable for most of the classic relational techniques developed for indexing, query optimization, buffer management, concurrency control, security and recovery. It was only after some experiences working with these new types of data that this approach was shown to have an inherent weakness [46]: *a mismatch between the nature of data and the way both the user and system were forced to query and operate on it.*

Object SQL queries and operations were not very suitable for multimedia data, for which browsing is in important paradigm, and, standard indexing approaches do not work for content-based queries of multimedia data. Other modules of database systems likewise have to be changed in order to manage multimedia data efficiently [46].

It has been realized that an evolution of standard database modules has to be done in order to cope with multimedia data features. Commercial object-relational database systems are at the moment the state of art for implementing multimedia database systems, but even these systems leave much to be desired in such areas as management and intuitive querying environment. This presses to develop separate multimedia data management modules to be integrated in such architectures.

As discussed in [46], over the past 20 years, managing multimedia data in a database environment has evolved through the following sequences of conceptual and performance insights:

- *Multimedia data was first transformed into relations in ad-hoc ways. Only certain types of queries and operations were efficiently supported. Initially, a query, such as "Find all images contained the person shown dancing in this video", was extremely difficult, if not impossible, to respond efficiently.*

- *When the weakness of this approach become apparent, researcher asked what types of information should be extracted from images and videos and how this information should be represented to support content-based queries most efficiently. The result was a large body of work on multimedia data models.*

- *Since these data models specified the types of information that could be extracted from multimedia data, the nature of multimedia queries was also discussed. Earlier work on feature matching from the field of image interpretation was brought to bear, helping launch the field of multimedia indexing. Multimedia indexing, in turn, started the ball rolling towards multimedia query optimization techniques.*

- *A multimedia query was seen as quite different from a standard database query and closer to queries in information-retrieval setting. The implication of this important concept have still not played themselves out.*

Today the management of multimedia information such as images, graphics, video, audio, and text, is of great interest in a lot of application fields like: Information Retrieval, Office Automation, E-learning, Virtual Museums, Newspaper and Magazines production, Video and Cinema Editing, Medical and Bio-informatics Applications (e.g., Radiographic and DNA-Sequences Archives), Geographical Information Systems Management, Biometric, Security Applications (including Video Surveillance), Remote Sensing and Meteorology and so on.

By now, as underlined in a lot of important works [46, 2, 98, 59], the multimedia information features (e.g., large data size, structure, and time dependencies, etc...) calls for new processing beyond the ability of traditional database architecture. For these reasons, in the past decade, the first MultiMedia Database Management Systems were carried out with the aim of managing in a efficient way heterogeneous data like image, text, audio or video.

2.3 Basic Features of a Multimedia Database Management System

A multimedia database management system is the heart of each multimedia information system. It allows the integration of different multimedia data types from multiple sources.

Traditionally, a database consists of a controlled collection of data related to a given entity, while a database management system, or DBMS, is a collection of interrelated data with the set of programs used to define, create, store, access, manage, and query the database. Similarly, it is possible to view a multimedia database as a controlled collection of multimedia data items, such as text, images, graphic objects, sketches, video, and audio.

In a such context, as suggested in [2], *a MMDBMS provides support for multimedia data types, plus facilities for the creation, storage, access, query, and control of the multimedia database. More in details, a multimedia database management system provides a suitable environment for using and managing multimedia database information. Therefore, it must support the various multimedia data types, in addition to providing facilities for traditional DBMS functions like database definition and creation, data retrieval, data access and organization, data independence, privacy, integration, integrity control, version control, and concurrency support.*

The different data types involved in multimedia databases require special methods for optimal storage, access, indexing, and retrieval and a MMDBMS should accommodate these special requirements by providing high level abstractions to manage the different data types, along with a suitable interface for their presentation [2].

Before detailing the capabilities expected of a multimedia DBMS and the requirements such systems should meet, the characteristic nature of multimedia information are first considered. Then the requirements and issues facing MMDBMSs are discussed.

2.4 Nature of multimedia data

The composition and characteristics of multimedia data can be analyzed from several perspectives like: information overload, inadequacy of textual descriptions, multiplicity of data types, spatial and temporal characteristics, and huge volumes of data.

The data types found in a typical multimedia database include:

- text;
- images: color, black and white, photographs, maps, and paintings;
- graphic objects: ordinary drawings, sketches, and illustrations, or 3D objects;
- animation sequences: images or graphic objects, (usually) independently generated;
- video: also a sequence of images (called frames), but typically recording a real-life event and usually produced by a video recorder;
- audio: generated from an aural recording device;
- composite multimedia: formed from a combination of two or more of the above data types, such as an intermix of audio and video with a textual annotation.

Some multimedia data types such as video, audio, and animation sequences also have temporal requirements, which have inevitable implications on their storage, manipulation, and presentation. The problems become more acute when various data types from possibly disparate sources must be presented within or at a given time. Similarly, images, graphics, and video data have spatial constraints in terms of their content. Usually, individual objects in an image or a video frame have some spatial relationship between them. Such relationships usually produce some constraints when searching for objects in a database [2].

Huge volumes of data also characterize multimedia information. For instance, to store an uncompressed image or video, the requested storage capacity is of the Mbytes or Gbytes order respectively. The potential for huge volumes of data involved in multimedia information systems become apparent when very large video repositories are considered.

However, representing multimedia information as pictures or image sequences poses some problems also for information retrieval due to the limitations of textual descriptions of a multimedia experience and the massive information available from it. The potential information overload means that users may find it difficult to make precise requests during information retrieval. The limitations of textual descriptions also imply the need for content-based access to multimedia information. Users need multiple cues (such as shape, color, and texture) that are relevant to the multimedia content [2].

Another characteristic of multimedia information is that interaction with such information types usually involves long-duration operations (such as with video data), and sometimes, with more than a single user (as is typical in collaborative support environments). However, in collaborative environments, it is expected that most multimedia data are likely to be accessed in a readonly mode. This assumption can be used to facilitate the provision of concurrency control algorithms [2].

Moreover, multimedia data is quite different from standard alphanumeric data in terms of both presentation and semantics [46]. From a presentation point of view, multimedia data is huge and involves time-dependent characteristics that must be taken into account for coherent viewing.

From the other point of view, because of its complex structure, multimedia data requires complex processing to derive semantics from its content. Real word object shown in images, video, animations, or graphics and discussed in audio participate in meaningful events whose nature is often the subject of queries. Using state of the art techniques from the fields of image processing and speech recognition, systems can often made to recognize similar real-word object and events by extracting certain information from the corresponding multimedia objects, also called "features", which usually less complex and voluminous than the multimedia object themselves.

How the logical and physical representation of multimedia objects are defined and relate to each other, as well as what features are extracted from these objects and how extraction is accomplished, is in the domain of multimedia data modeling (see [104, 68] for more details).

2.5 Purposes of a Multimedia Database Management System

The functions of a MMDBMS basically resemble those of a traditional DMBS. However, the nature of multimedia information makes new demands. As proposed in [2], using the general functions provided by a traditional DBMS as a guide, it is possible to describe the purposes of a MMDBMS as follows:

- *"Integration": ensures that data items need not be duplicated during different program invocations requiring the data.*
- *"Data independence": separation of the database and the management functions from the application programs.*
- *"Persistence": the ability of data objects to persist (survive) through different transactions and program invocations.*
- *"Concurrency control: ensures multimedia database consistency through rules, which usually impose some form of execution order on concurrent transactions.*
- *"Privacy": restricts unauthorized access and modification of stored data.*

- *"Integrity control": ensures consistency of the database state from one transaction to another through constraints imposed on transactions.*
- *"Recovery": methods needed to ensure that results of transactions that fail do not affect the persistent data storage.*
- *"Query support": Ensures that the query mechanisms are suited for multimedia data.*
- *"Version control": organization and management of different versions of persistent objects, which might be required by applications.*

In concurrency control, a transaction is a sequence of instructions executed either completely or not at all. In the latter case, the database is restored to its previous state. Defining the appropriate granularity for concurrency is a problem in multimedia databases.

Traditional databases use a single record, a table or a part of it as the unit of concurrency; multimedia databases typically use a single object (or composite object) as the logical unit of access. Thus the single multimedia object could form the unit of concurrency [2]. In achieving persistence, a simple method is to store the multimedia files in some operating system files or as object database "blob". However, the huge data volumes make this approach costly to implement. Moreover, the system also needs to store the multimedia "metadata" and possibly composite multimedia objects. Most MMDBMSs classify the data as either persistent or transient and store only persistent data after transaction updates. Transient data are used only during program or transaction execution and are removed afterwards.

Traditionally, a query selects a subset of the data objects based on the user's description (usually apposite query language are used to help the user in the query expression) of what data to access. A query usually involves various attributes, possibly keyword-based or content-oriented, and is usually interactive. Thus, functions for relevance feedback and query formulation, similarity (rather than exact) matches, and mechanisms for displaying ranked results are important in a MMDBMS [2].

Version control becomes important when a persistent multimedia object is updated or modified, as some applications might need to access previous states of the object. A MMDBMS provides such access through versions of the persistent objects.

The special nature of multimedia data also makes it important to support new special functions. These include object composition and decomposition, management of huge volumes of multimedia data, effective storage management, and information retrieval and handling of spatial and temporal data objects [2].

2.6 Requirements and Issues of Multimedia Database Management System

For the MMDBMS to serve its expected purpose, it must meet certain special requirements and issues [2]. They are divided into the following broad categories:

- Multimedia data modeling
- Huge capacity storage management
- Information retrieval capabilities
- Media integration, composition, and presentation
- Multimedia query support
- Multimedia interface and interactivity
- Multimedia indexing
- High Performances
- Distributed multimedia database management

In addressing these requirements when building a multimedia database system, one must also address several other questions, as reported in [2], to achieve full functionality, including:

- *How to build a multimedia database system that encompasses several application domains (that is not restrictive in terms of its domain applicability)*?

- *What are the levels of granularity for information decomposition, storage, and management? And how the underlying techniques and structures can be mapped and used on the units of data*?

- *Knowing the data compositions of a multimedia database, how can one reliably and efficiently develop a query language that supports the myriad access methods associated with and necessary for the diverse object types? How will the query language support the multimedia datas different characteristics and morphologies*?

- *What kind of presentation infrastructure will the multimedia system have to accommodate the diverse presentation requirements and modes for the different multimedia data? How can one synchronize presentations to support the temporal and spatial requirements of the different multimedia data*?

- *Given that different media types have differing modification and update requirements, how will the system update different components of the multimedia session? What levels of granularity will those updates have*?

To respond to classical data management requirements, the architecture of a standard database system consists of well know modules for query processing, transaction management, buffer management, file management, recovery and security. These modules suffer of inevitable modifies in the case of a multimedia database [46].

For what concerns query processing, in multimedia database querying is quite different from querying in standard alphanumeric databases. Besides the fact that browsing takes on added importance in a multimedia environment, queries can be of different nature and contain multimedia objects input by user. The result of such queries are based on similarity matches, not exact matches. To this purpose, indexes of standard, usually hash-based or utilizing B-tree variants, are unsuitable for similarity matching.

In multimedia databases a generic indexing technique is to extract n numerical-valued features from a multimedia object and represent these n values by a n-dimensional point. A spatial index that supports nearest-neighbor searching is used for similarity matching and query optimization became the process of choosing the optimal access path to answer a query [46].

For the transaction management, as already discussed, conventional concurrency control algorithms can be used to satisfy the four ACID properties. However, the concurrency of the overall system would suffer, since in this environment,transaction tend to be long, compute-intensive, interactive and cooperative. For example, if a video is locked for an update transaction, then many thousands of images frames are also locked.

In order to increase system concurrency in such environment, new transaction models defined for object-oriented environments. In particular, the traditional ACID properties have been generalized: *Atomicity* is changed to *recovery*, which refers to placing the database in a correct state in the event of a database failure or transaction abortion. To this aim recovery models for long-running transaction have to be developed. *Consistency* need not depend on the traditional concept of serializability ; a non-serializable schedule can still leave the database in a consistent state. *Isolation* is changed to *visibility*: transaction are allowed to view the results of other transactions. And finally, *durability* is changed to *permanence* [46].

Continuous media presentation for many concurrent users require also sophisticated buffer management techniques to deliver information on demand, scheduling the buffering in order to maximize sharing and support interactivity without violating the synchronization requirements [46].

Eventually, for the storage management the challenge is to serve multiple requests for multiple media streams so as guarantee the process do not starve, while minimizing the buffer space needed and the time between an initial request for service and the data fruition. Techniques as striping/interliving, data compression, data contiguity, and storage hierarchies have been employed to reduced this bottleneck [46].

Figure 2.1 shows a sample high-level architecture for a MMDBMS that addresses some of the requirements that have been discussed [2].

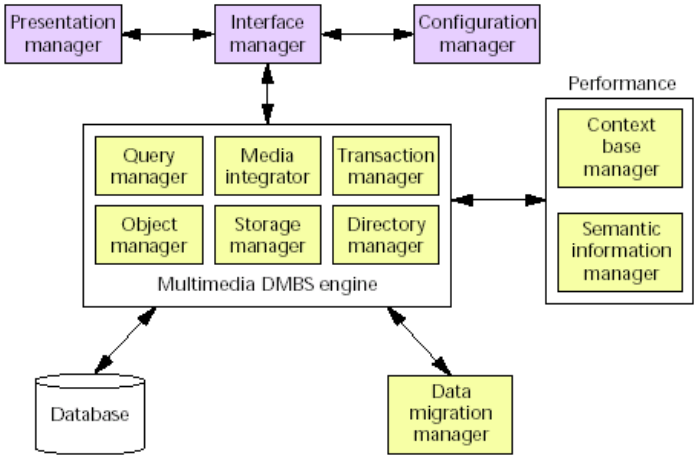


Figure 2.1: MMDBMS Architecture

In such architecture most of the management modules associated with a traditional DBMS are reported. In addition, it contains some of the modules that are required specifically for multimedia data management, such as the media integrator and object manager. However, most of the additions to the traditional DBMS are external to the core of the MMDBMS. These include the presentation, interface, and configuration managers. The configuration also includes a context-base and semantic information manager, which are part of the performance module. In the following the main requirements of a MMDBMS are described like discussed in [2].

2.6.1 Multimedia data modeling

In standard database systems, a data model is a collection of abstract concepts that can be used to represent real-word objects, their properties, their relationships to each other, and the operation defined over them. These abstract concepts are capable of being physically implemented in the given database system.

Through the mediation of this data model, queries and other operations over real-world objects are transformed into operations over abstract representations of these objects, which are, in turn, transformed into operations over the physical implementation of the abstract representations.

Data models are central to multimedia database systems. A data model must isolate and hide to users from the details of storage device management and storage structures. It requires the development of appropriate data models to organize the various data types typically found in a multimedia database system.

Multimedia data models capture the static and dynamic properties of the database contents, and thus provide a formal basis for developing the appropriate tools needed in using the multimedia data. The static properties could include the objects that make up the multimedia data, the relationships between the objects, the object attributes, and so on. Examples of the dynamic properties include interaction between objects, operations on objects, user interaction, and so forth.

However, the unique nature of multimedia data requires certain new considerations when choosing the data model. For instance, some multimedia data types (such as video) or group of types (example, video and images) might require special data models for improved modeling efficiency and flexibility. Moreover, the importance of interactivity in multimedia systems makes their support by the data model an important issue. Furthermore, it may be necessary to consider new integrity constraints in the context of multimedia databases. Various data models, such as network, relational, semantic, and object-oriented models already exist for traditional databases, and a few have been proposed for multimedia databases.

Two basic approaches have been used in modeling multimedia data. The first involves building a multimedia data model on top of an underlying traditional database data model (usually relational or object-oriented databases) by using appropriate interfaces for the multimedia data. The problem with this approach is that the underlying structures are not designed for multimedia data. Often, the significant differences between the requirements of the traditional and multimedia data make the interface a bottleneck in the overall system.

These problems led to the second method, which opts to develop true multimedia-specific data models from scratch, rather than on top of appropriate data models for individual multimedia data types (such as video, images, or visual data), uniform modeling of arbitrary data types, and supporting huge volumes of multimedia data, multimedia interactivity, and content-based information using these models. Some authors have gone so far as to claim that the data model for a MMDBMS can only be fully achieved by object-oriented technology.

2.6.2 Huge capacity storage management

The storage requirements in multimedia systems can be characterized by their huge capacities and the storage systems hierarchical (pyramidal) organization (see Figure 2.2) [2].

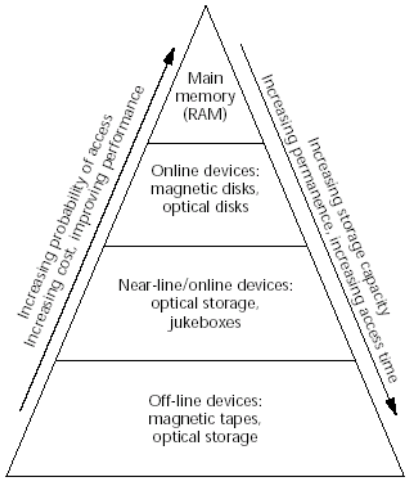


Figure 2.2: Hierarchically organized storage for multimedia database

Hierarchical storage places the multimedia data objects in a hierarchy of devices, either online, near-line, or off-line. In general, the highest level provides the highest performance, highest cost, smallest storage capacity, and least permanence. Note, however, that permanence improves at significant additional costwith the use of nonvolatile random access memory. Another unique use of this hierarchical storage organization is that the higher levels of the hierarchy can be used to store smaller abstractions (or representations) of the actual multimedia data, which can be used to facilitate faster browsing and previewing of the database content. Cost and performance (in terms of access time) decrease as we go down the hierarchy (pyramid), while storage capacity and permanence increase. Typically, in most multimedia storage systems the highest level of storage is (volatile) random access memory, followed by magnetic disk drives. These provide online services. Optical storage devices provide the next level of storage. Online in some cases, they are near-line (like jukeboxes) in most cases. The lowest level in the storage hierarchy represents off-line storage devices, including magnetic tapes, optical disks, and so forth. These may or may not be directly connected to the computer. They offer the highest storage capacity and permanence but provide the least performance in terms of access time. A MMDBMS must therefore manage and organize multimedia data stored at any level in the hierarchy. It must have mechanisms for automatically migrating multimedia data objects from one level of the storage hierarchy to another and for managing data compression and decompression.

2.6.3 Query support and information retrieval

Querying in multimedia databases can involve different multimedia data types, keywords, attributes, content, or even contextual information. Because of the different ways in which users think about multimedia data, multimedia query can simultaneously involve multiple cues, necessitating multiple or multidimensional indices.

Moreover, queries are usually imprecise. Because of this and the difficulty of ensuring exact matches between multimedia data items, retrieval usually involves comparing data items for similarity or partial (rather than exact) matching.

Thus, since queries might not yield exact matches, there is the need of facilities for ranking the retrieved results according to how closely they match the given query. Similarly, methods to prune results that do not seem to satisfy the query are required. Doing so reduces the potentially enormous computation needed for further matching.

With the ranking, the MMDBMS should also support browsing the various retrieved items. It is possible also to want to retrieve similar items based on one or more of the already retrieved items. A true MMDBMS also needs a facility to support incomplete information. More importantly, since the information extracted to index the multimedia data or from the user query might contain errors, query interpretation should provide for uncertainties in the information. This might require an iterative search mechanism and a relevance feedback mechanism along with techniques for query reformulation.

Among the issues involved in multimedia query support is the availability of a multimedia query language capable of supporting both the various media types encountered in a typical multimedia database and new requirements such as fuzzy query predicates. Such query models should also provide mechanisms for users to reformulate their queries, perhaps based on the already retrieved results.

Query by example is the primary method used to enter queries in multimedia databases, especially in those involving images. Here, the user makes a request using an existing example (for example, similar images). Thus, the interface used to enter the query into the system becomes an issue.

Since different multimedia data types may require different query interfaces, the problems to consider include how to integrate the various interfaces in an integrated multimedia database system. Other problems to be resolved include querying spatial data and content-based video query, which could involve temporal and spatial information.

2.6.4 Media integration, composition, and presentation

Given the multiplicity and heterogeneity of data types supported, the MMDBMS should also provide facilities for integrating data items (from possibly disparate media types) to form new composite multimedia types and for presenting such data at a given site within the required time.

Multimedia integration, composition, and presentation are exacerbated by the often continuous (temporal) nature of multimedia data especially video, animation, and audio. Moreover, certain applications, such as geographic information systems, may require a MMDBMS to address spatial information. All these factors put together make multimedia composition and presentation a complex process that the MMDBMS must support to meet the diverse user's needs.

Unlike traditional data, multimedia data have presentation constraints. These mainly result from the continuous nature of some multimedia data types, which requires presenting certain amounts of data within a given time for the presentation to seem natural to the user. When multimedia data are distributed and transported over networks, the problems of presentation become even more acute. Here, one can easily experience network problems, such as limited bandwidth and statistical network delays.

Continuous media by definition are time-dependent, so timing becomes an important factor in their delivery and presentation. Therefore, in MMDBMSs the response to a query is often judged by both the correctness and the quality of the retrieved results. From the users point of view, the QoS parameter specifies, qualitatively, the acceptable levels of performance for the various services provided by the multimedia system and may affect the results of the multimedia presentation. Thus, to support multimedia presentations where a user can specify various QoS levels for different services, the MMDBMS must support the specified QoS levels and a QoS management service. This typically involves providing an appropriate mapping from the users QoS to the systems QoS and vice versa.

When presenting different types of multimedia data such as video and audio together, problems of media integration and synchronization also become important.

The MMDBMS must provide a mechanism to ensure good synchronization of the presented data while still meeting other requirements such as the data availability rates and the QoS. In some situations, the MMDBMS may have to rely on an explicit synchronization manager to ensure synchronization within a given data type and between different data types.

2.6.5 Multimedia interface and interactivity

The diverse nature of multimedia data calls for different and apposite interfaces for interacting with the database. Each media data type has its own method for access and presentation and, for instance, video and audio data will need different user interfaces for presentation and query.

Moreover, for some multimedia applications, especially those involving continuous media, the user often expects the interactive facilities of a VCR or tape recorder (such as fast forward and reverse). When a multimedia system provides such services, it has implications for the database, especially retrieval of the needed multimedia objects, their integration, and their synchronization.

2.6.6 Multimedia indexing

As in traditional databases, multimedia information can be retrieved using identifiers, attributes, keywords, and their conjunctions using conditional statements. Keywords are by far the predominant method used to index multimedia data. A human typically selects keywords from a set of specialized vocabulary. While simple and intuitive, this method usually creates problems when applied to multimedia data: it is basically manual and time consuming, and the resulting indices are highly subjective and limited depending on the vocabulary. Another method, content-based access, refers either to the actual contents of the multimedia database or to derived contextual information. Intensive research has focused on content-based indexing in recent years, with the goal of indexing the multimedia data using certain features derived directly from the data.

Various features, such as color, shape, texture, spatial information, symbolic strings, and so on, have been used to index images. Deriving such features requires automatic analysis of the multimedia data. The primary methods used for image and video data are image processing, image understanding, and video sequence analysis. With video data, the video sequence is first separated into its constituent scenes, then representative abstractions (usually key frames) are selected to represent each scene. Further indexing on the video is based on the key frame, as in the case for images. For audio data, content-based indexing could involve analysis of the audio signal or automatic speech recognition followed by keyword-based indexing. On the other hand, indexing can be based on other information depending on the type of audio data. For example, some developers have used rhythm signature, chord, and melody for content-based indexing of music data.

Similarly, methods for content-based search and retrieval of audio data have been proposed based on the characteristics of audio data, as indicated by its perceptual and acoustic features. Using content-based indexing implies the consideration of certain issues. First, the same multimedia data could mean different things to different people. Second, users typically have diverse information needs. Thus, it is evident that a single feature may not be sufficient to completely index a given multimedia data type.

Therefore, it becomes difficult to identify the features that are most appropriate in any given environment. Another problem has to do with efficiency: making the indexing fast and storing the indices efficiently for easy access, since multimedia data typically come in huge volumes. Because of the diverse content inherent in multimedia data, indexing has not been completely automated. For example, while the computer can easily analyze a picture containing works of art, it is almost impossible for the computer to automatically determine the meaning of the art object. Only a human can provide such information.

2.6.7 Performance

High performance is an important requirement for a MMDBMS. It includes efficiency, reliability, real-time execution, guaranteed and synchronized delivery of multimedia presentations, and quality-of-service (QoS) acceptable to the users.

2.6.8 Distributed multimedia database management

Distributed MMDBMS loosely refers to a collection of various (possibly) independent multimedia database management systems, located in disparate locations, that can communicate and exchange multimedia data over a network. Multimedia systems are usually distributed in the sense that a single multimedia interaction often involves data obtained from distributed information repositories. This is typically the case in collaborative multimedia environments, where multiple users in possibly disparate physical locations manipulate and author the same multimedia document. Moreover, issues like storage problems and data generation may also force multimedia system designers to place multimedia data in different physical locations. To support the information required in such distributed and collaborative environments, a distributed MMDBMS must address the general problems in distributed databases, such as distributed and parallel query processing, distributed transaction management, data location transparency, data security, and so forth. In addition, network issues such as limited bandwidth and network delays become important considerations, since they could have adverse effects on the QoS supported.

Unlike in the traditional DBMS, data replication is often not encouraged in a distributed MMDBMS due to the huge data volumes. The client-server computing model, in which a server application services multiple client applications with the clients and server residing in possibly different machineshas proven suitable for multimedia systems in general and distributed multimedia DBMSs in particular.

2.7 Some motivating examples

In the following two examples in video and image data managing are reported in order to underline the requirements for MMDBMS environments and motivate such dissertation.

- **Video Database System**. Let us consider a large repository of soccer-match videos: in a such environment an interesting query should be "find all goal-actions scenes in a given video" (see figure 2.3).

  To respond in an efficient manner to the user query, the system has to present the following features:

  - A Data Model that supports an high-level video abstraction in terms of scenes and their contents is necessary.
  - The videos have to be indexed (in an automatic or semi-automatic manner) in according to the Data Model and by means of apposite textual description (metadata) inside the repository.
  - The Interface and Presentation Manager modules have to guarantee a suitable video fruition that satisfies the temporal constraints and supports some interactive facilities (as fast forward and reverse).
  - The Object, Storage, Directory Manager modules have to serve the requests of multimedia streams optimizing the storage access.
  - The Transaction Manager has to resolve a possible concurrency access to video resources.
  - The Query Manager has to offer to the user a friendly interface to build the query in according to a given query language.

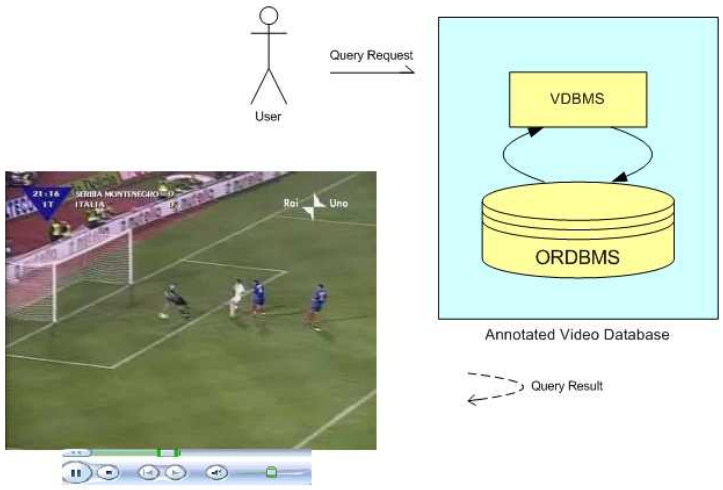


Figure 2.3: An example of query by content in a video database

- **Image Database System**. Let us consider a large repository of images: in a such environment an interesting query should be "find all airplanes images similar to a given target image and with a blue-sky color predominance" (see figure 2.4).

  To respond in an efficient manner to the user query, the system has to present the following features:

  – A Data Model that supports an high-level image abstraction in terms of image description and content is necessary.

  – The images have to be indexed (in an automatic or semi-automatic manner) in according to the Data Model and by means of apposite textual description (metadata) inside the repository.

  – The Interface and Presentation Manager modules have to guarantee a suitable image fruition without viewing latency and supports some interactive facilities (as zoom, rotation, etc...).

  – The Object, Storage, Directory Manager modules have to serve the requests of multimedia data optimizing the storage access.

– The Transaction Manager has to resolve a possible concurrency access to image resources.

– The Query Manager has to offer to the user a friendly interface to build the query in according to a given query language and a matching module for computing images similarity and selecting the most relevant results.

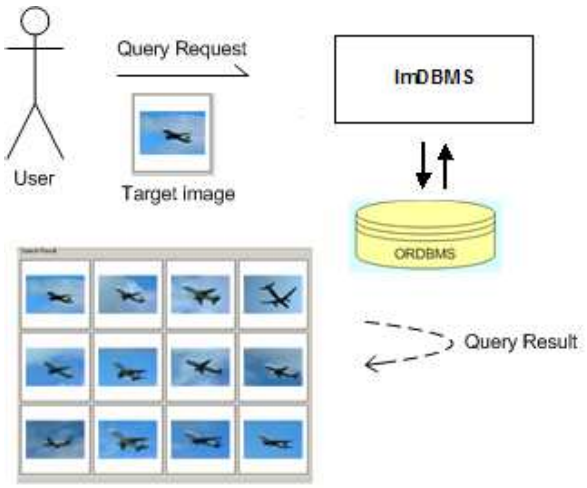


Figure 2.4: An example of query by content and example in an image database

As can be observed in the two examples, a traditional DBMS is in general not able to manage multimedia information and support possible user queries. The need of a novel architectures and techniques to respond to such requirements motivates this work.

Chapter 3

Video and Image Database Systems

3.1 Introduction

Among multimedia information, video and image are the most common ones and their efficient managing in a MMDBMS environment represents an open challenge of research activity on multimedia.

Fast access to multimedia information requires the ability to search and organize the information. While, the technology to search text has been available for some time - and in the form of web search engines is familiar to many people - the technology to search images and videos, is much more challenging due to different nature of multimedia information.

In such area the main objective of the researchers is to index in an automatic way video and image data on the base of their content in order to facilitate and make more effective and efficient the query processing.

In the following, supported by the related state-of-the-art, we describe the major challenges in developing reliable image and video database systems.

3.2 Image Databases

3.2.1 Introduction

With the steady growth of computer power, rapidly declining cost of storage, and ever-increasing access to the Internet, digital acquisition of information has become increasingly popular in recent years. For this reason, during the last years, fast retrieval of relevant and accurate images respect to the user query in very large repositories or huge digital libraries has been one of the most important research issue.

Libraries have traditionally used manual image annotation for indexing and then later retrieving their image collections. However, by now, manual image annotation is an expensive and labor intensive procedure and hence there has been great interest in coming up with automatic ways to retrieve and index images based on content.

To this purpose, several researchers have investigated techniques to retrieve images based on their content, but many of these approaches require the user to query based on image concepts like color or texture, which most people are not familiar with.

In general, people would like to pose semantic queries using textual descriptions and find images relevant to those semantic queries. In particular, the automatic derivation of semantically-meaningful information from the content of an image has the focus of interest for the most recent research on image databases.

The images semantics, i.e., the meanings of an image, has several levels. From the lowest to the highest, these levels can be roughly categorized as:

1. semantic types (e.g., landscape photograph, clip art);
2. object composition (e.g., a bike and a car parked on a beach, a sunset scene);
3. abstract semantics (e.g., people fighting, happy person, objectionable photograph);
4. detailed semantics (e.g., a detailed description of a given picture)

As underlined in [28], in the state-of-the-art approaches, the retrieval task may be efficient for some queries in which the semantic content of the query can be easily translated into visual features. For example, finding images of fires is simple because fires are characterized by specific colors (yellow and red). However, it is not efficient in other application fields in which the semantic content of the query is not easily translated into visual features. For example, finding images of birds during migrations is not easy because the system has to understand the query semantic. In the query, the basic visual features may be useful (a bird is characterized by a texture and a color), but they are not sufficient. What is missing is the generalization capability. Birds during migrations belong to the same repository of birds, so they share common associations among basic features (e.g., textures and colors) that the user cannot specify explicitly.

There is the need of an approach that discovers hidden associations (these associations discriminate image repositories) among features during image indexing and allow a user-friendly query formulation.

3.2.2 Content based Image Retrieval (CBIR)

The goal of an image retrieval system is to find images from an image database while processing a query provided by a user. In the last decade, most of researches are focused on `Content Based Image Retrieval (CBIR)`. The CBIR is characterized by the ability of a system in retrieving relevant information on the base of image visual content and semantics expressed by means of simple search-attributes or keywords [112].

The relevance of retrieved information can be judged in different ways by different users, although the query was formulated in the same manner. In other terms, the relevance-concept is dynamical, subjective and user and context dependent.

A CBIR system has to be capable of becoming adaptable respect to user query formulation and interesting information content. Actually, there are two different methodologies to perform a search in image database on the base of image information content:

- `textual-based` methodologies, usually using descriptive metadata (title, description, format, resolution, etc...);
- `feature-based` methodologies, usually based on image processing algorithms able to extract from image sets of content features (color, shape, texture, objects, image samples, etc...).

In the first case the retrieval is more simple but presents some limitations due to the manual annotation process: for example, two user can describe the same image using different descriptions. In the opposite, the second methodology is more complex and requires, form on hand, the choice of suitable perceptive features to represent an image and, form the other one, the presence of image processing algorithms to extract the perceptive content in an automatic and robust manner.

For these reasons, the actual trend, as already described, is to combine the two approaches and allow to pose queries using textual descriptions or image samples, where the semantically-meaningful information has to be automatically obtained from the content of an image.

3.2.2.1 Image representation and feature selection problems

An important research issue in the field of multimedia data analysis is that of choosing the right representations for the data (images, sounds, video, etc). Whether it is for searching, indexing, comparison, etc., it is clear that the way the multimedia data are represented can significantly influence the performance for the various data analysis tasks.

Approaches for content-based image querying typically extract a single signature from each image based on color, texture, or shape features. The images returned as the query result are then the ones whose signatures are closest to the signature of the query image.

Traditionally, the problem of query by content or, alternately, that of retrieving images that match a given query image from a large database of images has been solved by computing a `feature signature` for each image, mapping all signatures to d-dimensional points in some metric space (usually reducing dimensionality in the process), and building an index on all signatures for fast retrieval. An appropriate distance function (e.g., Euclidean distance) is then defined for each pair of signatures and, given a query, the index is used to efficiently locate signatures close to the query point. The set of images corresponding to the signatures are then returned to the user and constitute the result of the query.

Typical methods for computing image signatures include color histograms [6], which can be used to characterize the color composition of an image, regardless of its scale or orientation. Color histograms, however, do not contain any shape, location, or texture information. As a result, two images with similar color composition may in fact contain very different shapes and, thus, be completely unrelated semantically. The most diffuse opposite approach is to use the wavelets coefficients for an image as its signature, since wavelets capture shape, texture, and location information in a single unified framework [67]. In Fig. 3.1 an example of color histograms and Wavelet transform of a given image is reported.

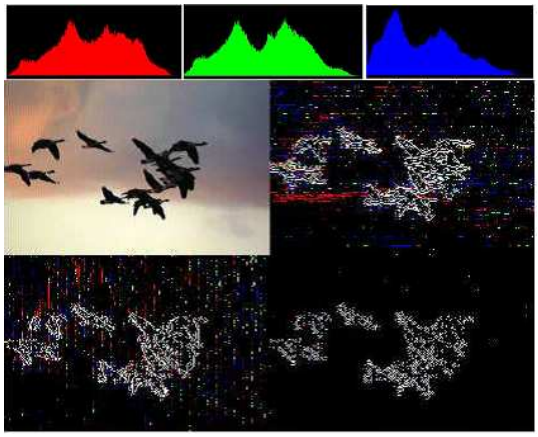


Figure 3.1: Color histograms and Wavelet Transform

Features can be extracted from the whole image or from particular regions of image itself, in the second case, a crucial part of any region-based image query system is the region extraction procedure (`image segmentation`). A number of strategies for decomposing an image into its individual objects have been proposed in the literature, however, extracting regions from an image is a difficult problem to solve [17]. Approaches that involve manual object extraction can be extremely tedious and time-consuming and are therefore impractical for large image collections. Consequently, most image segmentation techniques rely on being able to identify region boundaries, sharp edges between objects, and on a number of other factors, such as color, shape, connectivity, etc. However, besides being computationally expensive, the schemes are frequently inaccurate in identifying objects and the used methods are generally not robust with respect to object granularity.

The reason for this is that the definition of an object is largely subjective as a result, a single method cannot successfully identify the correct objects for all applications and may decompose what the user perceives as a single object into several smaller objects. A number of image segmentation techniques therefore utilize domain-specific constraints and are thus application-specific. Similarly, there is a need for region extraction specifically for the purpose of sub-image indexing and retrieval.

3.2.2.2 Possible query formulation in image database

The formulation of a query in a modern CBIR system can be executed using three different approaches or by a their combination.

- **Query By Example** (QBE): in a such approach a user specifies a `target image` and the system respond retrieving from the database the most similar images in according to a similarity criterion.
- **Query By Features** (QBF): in a such approach a user specifies the wanted image features (e.g., images with a red color predominance, images with a circular object, etc..) by means of an apposite graphical interface.
- **Query By Attributes** (QBA): in this approach the classical textual annotations are used as keys for searching in the database.

3.2.3 Similarity query and access methods for very large database

The emerging technologies based on repositories of heterogeneous information (such as images, video, audio, time series and DNA sequences) require general search models and algorithms in order to deal with such complex and large-scale multidimensional data sets. In this context a "key-problem" is the development of fast and efficient access techniques, especially for what concerns very large image repositories.

A viable solution to perform queries on multimedia and complex data is the introduction, in the objects domain, of a distance function δ, pointing out the dissimilarity between two objects belonging to the class of objects $\mathcal{O}$. Formally:

$$\delta : \mathcal{O} \quad \times \quad \mathcal{O} \rightarrow \mathcal{R}^{+} \tag{3.1}$$

In the multimedia realm, to make possible a comparison between any two objects, a "feature-based" solution, as already described, is usually proposed. The basic idea is to extract important features from the multimedia objects, represent the above features by high-dimensional vectors and search the database objects having the most similar features. Thus, we assume that objects (for example images or video-frame) are mapped into points of a "multi dimensional features vector space" with a fixed and finite dimension d.

The introduced distance function must satisfy some particular properties (non-negativity, symmetry and triangular inequality) that characterize a metric. Formally a metric space is a pair $\mathcal{MS} = (\mathcal{S}, \delta)$ where $\mathcal{S}$ is a domain of feature values and δ is a distance function having the following properties:

1. $\delta(\mathcal{O}^x, \mathcal{O}^y) = \delta(\mathcal{O}^y, \mathcal{O}^x)$(symmetry);

2. $\delta(\mathcal{O}^x, \mathcal{O}^y) > 0$ with $(\mathcal{O}^x \neq \mathcal{O}^y)$(non negativity); and $\delta(\mathcal{O}^x, \mathcal{O}^x) = 0$

3. $\delta(\mathcal{O}^x, \mathcal{O}^y) < \delta(\mathcal{O}^x, \mathcal{O}^z) + \delta(\mathcal{O}^z, \mathcal{O}^y)$ (triangle inequity).

When a distance function with the above features is defined, it is said to be a metric function, the domain is said to be metric space, and metric access methods can be used to indexing and retrieving data by means of the "similarity query" search paradigm. Its essence is to find in a given collection of objects those which better fit (i.e., which are more similar to) a given query specification.

More in details, the types of similarity queries that can be usefully used to search objects in a generic metric space are defined as follows.

DEFINITION 1 (Range Query).

$$RangeQuery(DB, \mathcal{O}^q, \varphi, \mathcal{M}) = \{\mathcal{O} \in DB | \delta_{\mathcal{MS}}(\mathcal{O}, \mathcal{O}^q) \leq \varphi\}$$

where DB is a set of n points in a d-dimensional data space, $\mathcal{O}^q$ is the query object, φ is a distance value and $\mathcal{MS}$ is a generic metric space.

The result of this function is the set of all object-points having a distance smaller than or equal to φ from $\mathcal{O}^q$, according to the metric δ

DEFINITION 2 (Nearest-Neighbor Query).

$$NNQuery(DB, \mathcal{O}^q, \mathcal{M}) \subseteq \{\mathcal{O} \in DB | \forall \mathcal{O}^{'} \in DB : \delta_{\mathcal{MS}}(\mathcal{O}, \mathcal{O}^q) \leq \delta_{\mathcal{MS}}(\mathcal{O}^{'}, \mathcal{O}^q)\}$$

where DB is a set of n points in a d-dimensional data space, $\mathcal{O}^q$ is the query object and $\mathcal{MS}$ is a generic metric space.

The result is an object-point chosen among those points having minimal distance from the query object $\mathcal{O}^q$.

In particular if a user wants to get the first k closest points to the query object, the notion of Nearest-Neighbor query could be expanded with the definition of k-NN Query.

DEFINITION 3 (K-Nearest-Neighbor Query).

$$kNNQuery(DB, \mathcal{O}^q, k, \mathcal{M}) = \{\{\mathcal{O}^1..\mathcal{O}^k\} \in DB| \neg\exists i,\ 0 \leq i < k : \delta_{\mathcal{MS}}(\mathcal{O}^i, \mathcal{O}^q) > \delta_{\mathcal{MS}}(\mathcal{O}^{'}, \mathcal{O}^q)\}$$

where k indicates the number of closer points to the query point Q.

Note that the queries introduced above can be performed using either a "sequential" scan of the objects present in the database or a "smart scan" that permits to locate and analyze only the relevant objects. A drawback of sequential scanning is time complexity, which is directly related to the size and number of stored objects.

To improve retrieval efficiency, features should be organized into indexing data structures that support efficiently the query process. Generally speaking, an index consists of a collection of entities, one for each object, containing the key for that object, and a reference pointer which allows immediate access to that object [19].

Different indexing mechanisms have been proposed in the literature to facilitate fast feature-based retrieval of multimedia objects in very large database. A formal definition is:

DEFINITION 4 (Index mechanism for feature based retrieval). *Let Σ be a set of multimedia objects (O_i) and $\Omega = \omega_1, \omega_2, ..., \omega_m$ a set of m classes to which Σ is to be classified where ω satisfies the following:*

1. $\omega_i \neq \Sigma \forall\, i = 1, 2, 3...m$;
2. $\cup_{1 \leq i \leq m} \omega_i = \Sigma$;
3. $\omega_i \neq \omega_j \, for \, i \neq j$;

the indexing process consists of the application of a mapping $\Sigma \rightarrow \Omega$ denoted by $T = \eta(R, \Omega)$, where R is a set of parameters to define the mapping, and classes in Ω represent the categories of multimedia object set Σ.

If the clusters are organized according to the classical tree structure, we have the following indexing association $\{\omega_1, \omega_2, ..., \omega_m\} \rightarrow \{N_1, N_2, ..., N_m\}$, N_i being a generic tree-node.

In particular, an efficient index for a large data set, where data are described in high dimensional feature space, should allow to prune comparisons between data during the similarity search process by taking advantage from distance properties. To this end, the overall data distributions can be considered for "grouping" similar objects in the same "similarity-class", and each class represented by a particular object called "centroid", "pivot" or "routing object". Then, during retrieval, it is possible to reduce necessary comparisons by calculating the similarity between the query objects and each class through their representatives. In this way, complex and multimedia data retrieval can be accelerated and improved using both classes-based index structures and the similarity query concept.

Probably, the first general solution to search in metric spaces was presented by Burkhard and Keller [15]. They propose a tree (thereafter called the "Burkhard Keller Tree", or BKT), which is suitable for discrete-valued distance functions. From the opposite the "vantage-point trees" or VPT is proposed in [111] as a tree data structure designed for continuous distance functions. The "bisector trees" (BST) are proposed in [54] as binary trees built recursively segmenting the data space. In [102], is also proposed the "generalized-hyperplane tree" (GHT), identical in construction to a BST. The GHT is extended in [14] to an m-ary tree, called GNAT ("geometric near-neighbor access tree"), keeping the same essential idea, but also using a Voronoi-like partition (an example of GNAT is reported in Figure 3.2). Eventually, the "M-tree" (MT) data structure is presented in Ciaccia et al., [19] aiming at providing dynamic capabilities and good I/O performance in addition to few distance computations (an example of MTREE generated in a $[0, 1]^2$ domain by L_1 metric is reported is reported in Figure 3.3).

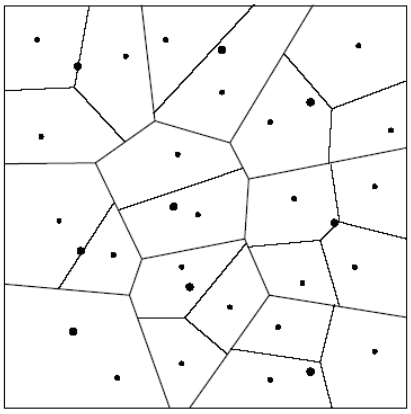


Figure 3.2: A GNAT example using Voronoi-like partition

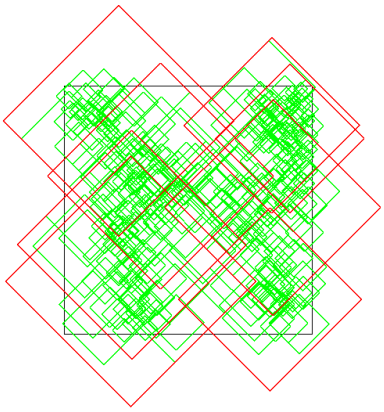


Figure 3.3: An example of generated MTREE on a $[0,1]^2$ domain by L_1 metric

From the opposite, clustering [52] represents the most diffused analysis technique for discovering interesting data patterns in the underlying data set: given a set of n data points in a d-dimensional metric space, a clustering approach assigns the data points to l ($l << n$) classes or groups, maximizing object similarity within the same class. The clustered data structure can be efficiently used (e.g., by means of a recursive application on the data space) to build indexes (e.g., search-trees) for high dimensional data sets, which support efficient queries. Interestingly enough, many statistical clustering techniques (e.g., k-means, fuzzy k-means) can be considered as special cases of the Expectation-Maximization (EM) algorithm [11].

Different approaches were proposed in literature for representing clustered data by indexing structures. In [43], the clusters are organized in a tree structure CF^*tree and a representative called "clusteroid" is chosen from each cluster. While searching, the query object is compared against the clusteroid and the associated cluster is eliminated from consideration in the case in which a similarity criterion does not hold. The problems connected to the inserting of new objects are solved by introducing a "inter-cluster" distance. An advantage of the clustering approach respect to the other ones is the possibility of generating classes of objects that are independent: such feature can be used to simplify the pruning conditions in the query process.

Eventually, in [113] a new indexing approach to representing clusters generated by any existing clustering algorithm in a tree structure called "ClusterTree" is presented.

3.2.4 A short overview of the most diffused CBIR systems

In the past decade, systems for retrieval by visual content (CBIR) have been presented in the literature proposing visual features that, together with similarity measures, could provide an effective support of image retrieval ([26], and for a detailed survey, [89]).

Three commercial CBIR systems are now available, IBM's Query by Image Content (QBIC) [38], Virage's VIR Image Engine [47], and EXACALIBUR's Image RetrievalWare [32]. They mostly rely on low-level image features such as color, texture, shape features [38], [47], [32], where shapes can be described using simple cues like shape area, circularity, eccentricity, major axis orientation, algebraic moment invariants [38], relative orientation, curvature and contrast of lines [32]. Matching is performed through weighted Euclidean distance [38], or user supplied similarity functions [47], [32].

Also, a number of experimental/research systems have been proposed, beyond early systems like Chabot [78], including MIT's Photobook [81], Columbia University's VisualSeek [96, 97], Pichunter [22], PICASSO [24, 25], Blobworld [17], SIMPLIcity [105], El Nino [90], [92].

Features exploited relate to color, [22], [81], [25],[92], spatial properties [96, 97], [24], faces, 2-D shapes and textures [81],[92], sketches [24]. Segmentation is specifically accounted for by PICASSO [25], SIMPLIcity [105] and Blobworld [17]. PICASSO exploits multiresolution color segmentation [25]. In SIMPLIcity, the k-means algorithm is used to cluster regions, while in Blobworld regions (blobs) are segmented via the EM algorithm and features used for querying are color, texture, location, and shape of regions (blobs) and of the background. In these systems, matching is accomplished through a variety of ways, using distance between eigenimage representations [81], color set similarity, region absolute location, spatial extent [96, 97] L^1 distance [22], quadratic or Euclidean distance [17], specific color distances [25], feature contrast metrics [92], integrated region matching through wavelet coefficients [105].

3.3 Video Databases

3.3.1 Introduction

The rapid advances in video technology, the widespread diffusion of video products - such as digital cameras, camcorders and storage devices - and the explosive growth of the internet have quickly made of digital video an essential component of today multimedia applications, including video-on-demand, video conferences, multimedia authoring systems and so on.

Then, thanks to the development of multimedia compression techniques, we have observed an exponential increase in the amount of available digital video data. While the amount of video data is rapidly increasing, multimedia applications are still very limited in content management capabilities. There is a growing demand for new techniques that can enable efficient processing, modeling and management of video contents.

Searching information based on content is difficult for audio-visual content, as no generally recognized or standardized description of this material exist. To this end, in the last years, MPEG (the Moving Pictures Experts Group) has been setting a standard called "Multimedia Content Description Interface (MPEG7)", that will extend the limited search capabilities of today to allow efficient retrieval of multimedia information.

Today, the major bottleneck limiting a wider use of digital video is the ability of quickly finding desired information from a huge database using content. A reliable way to resolve this problem and enable fast access to video clips is to properly index video sequences using suitable descriptors.

Traditional video database use keywords as index to quickly access to a great deal of data. However, this kind of data representation requires a burdening manual processing. It is widely agreed that video analysis refers to the computerized understanding of the semantic meanings of a video sequence. Tools that enable such automated analysis are becoming indispensable to be able to efficiently access and retrieve video information.

A typical indexing and retrieval scenario of video content is shown in Figure 3.4. First, input videos and images are segmented into temporal consistent units. Visual and audio features are then extracted from these segments to build indices and summaries. And finally videos or images are browsed and retrieved based on these features and structures.

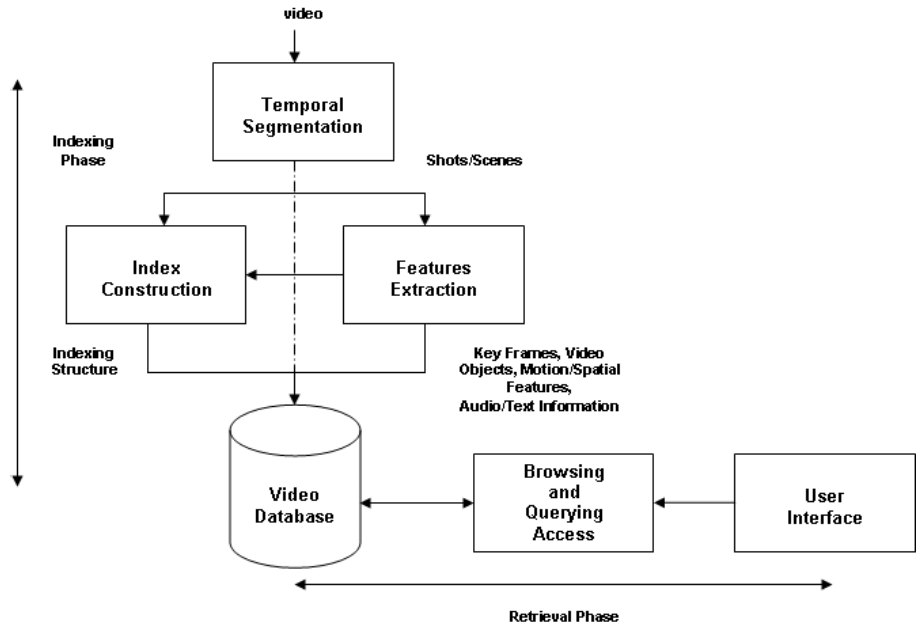


Figure 3.4: Block diagram of a video database management sytsem for content-based video indexing and retrieval

The main feature of a video management system should be the presence of an efficient indexing system to enable fast access to the stored data. This could be achieved by a set of semantic indices for meaningfully describing video scenes and a query capability for flexible specification and efficient retrieval.

Within this scenario, the content-based technique is one key component that provides feature based similarity search of pictorial data.

In order to achieve such purposes a common approach and an essential step in video indexing and content-based retrieval is to segment a video sequence in atomic units called `shots`. In this work our efforts will be devoted to develop new techniques for the video shots segmentation process.

3.3.2 The video segmentation problem

The first step in an automatic video indexing process is the so called `video shot segmentation`. The objective of video shot segmentation is to partition a video into basic, meaningful parts called shots. Each video shot represents a meaningful event, or a continuous sequence of actions, and it corresponds, in according to the most diffused definition in the literature, to a sequence of frames captured from a unique and continuous record by a camera. Further scene analysis and interpretation could then be performed on such units. Semantic annotations, determined either by an human or by means of some automatic or semiautomatic content analysis techniques, can be associated to each shot to complete the indexing process. The segmented video sequences could also be used for browsing, in which only one or a few representative frames (`key-frames`) of each shot are displayed.

In a given video, different kinds of transitions may occur. The basic distinction is between `abrupt` and `gradual` ones. An abrupt transition occurs in a single couple of frames, when stopping and restarting the video camera. A gradual transition is obtained using some spatial, chromatic or spatio-chromatic effects, such as `fade in(out)` - i.e. a gradual increase (decrease) in brightness resulting in a solid color frame - or `dissolve` - i.e. a gradual super-imposition of two consecutive shots. Gradual transitions include also other video editing special effects as wiping, tinning and so on.

Generally, abrupt transitions are very easy to detect because the two successive frames involved in the transition are totally uncorrelated. On the contrary gradual transitions are harder to detect from a data-analysis point of view because the difference between consecutive frames is substantially reduced. A number of considerable works has been reported on the detection of abrupt transitions. The majority of the proposed techniques evaluate some similarity measure between successive frames and assume that a cut occurs when the value returned by the measure is lower than a fixed threshold. The comparison of successive frames is not useful to detect gradual transitions because the difference between them is very small.

One of the main issue in segmenting a video sequence into shots is the ability to distinguish between scene breaks and normal changes into the scene. Moreover camera movements, such as panning, tilting and zooming, present similar features to transition effects such as dissolves.

A reliable video segmentation algorithm must be able to recognize dissolve effects without misinterpreting camera movements as gradual transitions. Although many progresses have been made in scene cut detection, existing systems still lack the following capabilities: (i) detect gradual transitions reliably; (ii) achieve real-time (or faster than real-time) processing performance; and (iii) to handle special situations such as flashes or sudden lightening variances.

After shot detection preliminary stage, an efficient process for content-based video retrieval requires an effective scene segmentation technique to divide a video into meaningful high-level aggregates of shots called `scenes`. Each scene has an autonomous semantic content and can be used as starting point in the video classification and annotation work. Such task involves the segmentation of the video into semantically meaningful units, classifying each unit into a predefined scene type, and indexing and summarizing the video for efficient retrieval and browsing.

3.3.3 An overview of issues and existing techniques for video shot segmentation

3.3.3.1 Main objectives in a video shot segmentation process

The development of shot-boundary detection algorithms has the longest and richest history in the area of content-based video analysis and retrieval longest, because this area was actually initiated some decade ago by the attempts to detect hard cuts in a video, and richest, because a vast majority of all works published in this area so far address in one way or another the problem of shot-boundary detection. This is not surprising, since detection of shot boundaries provides a base for nearly all video abstraction and high-level video segmentation approaches. Therefore, solving the problem of shot-boundary detection is one of the major prerequisites for revealing higher level video content structure.

As underlined in [49], despite countless proposed approaches and techniques so far, robust algorithms for detecting various types of shot boundaries have not been found yet. We relate here the attribute "robust" to the following major criteria:

1. excellent detection performance for all types of shot boundaries (hard cuts and gradual transitions);

2. constant quality of the detection performance for any arbitrary sequence, with minimized need for manual fine tuning of detection parameters in different sequences.

3.3.3.2 Modeling the video shot segmentation process

Assume as input to a segmentation system a video sequence, that is a finite sequence of time parameterized images, $(f(t_0), f(t_1), \ldots f(t_N))$, where each image $f(t_n)$ is called a frame. Each frame is a color image, namely a mapping from the discrete image support $\Omega \subseteq Z^2$ to an m-dimensional range, $f : \Omega \rightarrow Q \subseteq Z^m$; in other terms, it is a set of single-valued images, or channels, sharing the same domain, i.e., $f(x, y) = (f_i(x, y))^T$, where the index $i = 1, .., m$, defines the i-th color channel and (x, y) denotes a point in the Ω lattice. $Q = \{q_1, ..., q_N\}$ is the set of colors used in the image. Each frame displays a view, a snapshot, of a certain visual configuration representing an original world scene.

A time segmentation of a video f defined on the time interval $[t_0, t_N]$ is a partition of the video sequence into N_b subsequences or blocks. One such partition can be obtained in two steps. First, a mapping $\mathcal{T} : Z^m \rightarrow F$ of the frame $f(t_n) \in Z^m$ to a representation $\mathcal{T}(f(t_n)) \in F$, F being a suitable feature space, is performed. Then, given two consecutive frames $f(t_n)$ and $f(t_{n+l})$, where $l \geq 1$ is the skip or inter-frame distance, a discriminant function $\mathcal{D} : F \times F \rightarrow R^+$ is defined to quantify the visual content variation between $\mathcal{T}(f(t_n))$ and $\mathcal{T}(f(t_{n+l}))$, such that a boundary occurs at frame $f(t_n)$ if $\mathcal{D}(\mathcal{T}(f(t_n)), \mathcal{T}(f(t_{n+l}))) > T$, where T is a suitable threshold.

Thus, in principle, to solve the shot detection problem three steps must be undertaken: choose an appropriate mapping $\mathcal{T}$; define a robust discriminant function $\mathcal{D}$; devise a (universal) threshold T.

As regards the first two points, different techniques have been used: pixel based methods, such as the mean absolute value of intensity between frames [56],[80], or block matching [93] [49], histograms difference [110],[41], [42], [73], [116] motion difference [33, 115, 83] and perceived motion energy [66], differential geometry [64].

For what concerns the third point, heuristically chosen thresholds have been proposed [73],[80]. However a fixed thresholding approach is not feasible especially when considering gradual transitions. In particular, dissolve effects are reputed the most common ones, but also the most difficult to detect [35], [100]. A dissolve can be obtained as a combination of fade-out and fade-in, superimposed on the same film strip; fade-out occurs when the visual information gradually disappears, leaving a solid color frame, while fade-in takes place when the visual information gradually appears from a solid color frame.

Dissolve detection is still a challenging problem. Few techniques have been published [61]. Variable thresholding has been proposed in [110] and [116], the latter relying on gaussian distribution of discontinuity values. For instance in [116] the twin-comparison approach using a pair of thresholds, for detecting hard cuts and gradual transitions, respectively, has been introduced. More significant improvements have been achieved by recasting the detection problem in a statistical framework. A novel and robust approach has been presented by Lienhart [62], which relies on multi-resolution analysis of time series of dissolve probabilities at various time scales; experiments achieved a detection rate of 75% and a false alarm rate of 16% on a standard test video set. Further, it has been recently argued that a statistical framework incorporating prior knowledge in model based [48], statistical approaches leads to higher accuracy for choosing shot boundaries [64], [49].

In the following we describe, more in details, some of the most diffused techniques in the literature for video shot detection.

3.3.3.3 Analysis of methods for detecting abrupt transitions

As already described, considerable work has been reported on detecting abrupt transitions both in uncompressed and compressed video domain.

The first methods for shot detection [56, 80] were methods based on a pixelwise difference. in particular, the distance, $\mathcal{D}(\mathcal{T}(f(t_n)), \mathcal{T}(f(t_{n+l})))$, used for establishing the similarity between two successive frames, is reported in equations 3.2 and 3.2 respectively.

$$\mathcal{D}(\mathcal{T}(f(t_n)), \mathcal{T}(f(t_{n+l}))) = \frac{\sum_{x=1}^{X} \sum_{y=1}^{Y} |P_n(x,y) - P_{n+1}(x,y)|}{XY} \tag{3.2}$$

for gray-levels images,

$$\mathcal{D}(\mathcal{T}(f(t_n)), \mathcal{T}(f(t_{n+l}))) = \frac{\sum_{x=1}^{X} \sum_{y=1}^{Y} \sum_{c} |P_n(x,y,c) - P_{n+1}(x,y,c)|}{XY} \tag{3.3}$$

for color images.

Where $f(t_n)$ e $f(t_{n+l})$ are the two successive frames of XY size, $P_n(x,y)$ is the intensity value of (x,y) pixel, c is the index for color components (for example in the RGB space, $c \in \{R,G,B\}$), and $P_n(x,y,c)$ is the color component of (x,y) pixel.

All methods based on image pixel difference difference are sensible to camera or frame objects motion that can produce false alarms. For this reason a more motion-tolerant (*block-based comparison*) was introduced [49, 93].

In such approach a given frame is subdivided into b blocks that are then compared with the related blocks in the successive frames in according to equation 3.4:

$$\mathcal{D}(\mathcal{T}(f(t_n)), \mathcal{T}(f(t_{n+l}))) = \sum_{k=1}^{b} c_k \cdot DP(n, n+1) \tag{3.4}$$

where c_k is a matching coefficient for block k and $DP(n, n+1, k)$ is the local distance between the $k-th$ blocks in the two successive frames.

An example of block-based comparison is the *likelihood ratio* proposed in [55] and reported in equation 3.5:

$$\lambda = \frac{[\frac{\sigma_n+\sigma_{n+1}}{2} + (\frac{\mu_n-\mu_{n+1}}{2})^2]^2}{\sigma_i \cdot \sigma_{n+1}} \tag{3.5}$$

where μ_n e μ_{n+1} and σ_n e σ_{n+1} are respectively the mean and variance values for homologous blocks in the $n-th$ and $n+1-th$ frames.

All block-based methods are more robust but requires more time for video analysis.

The *histogram comparison* based methods try to resolve robustness and performance problems using color histograms of an image [42, 116, 73, 41, 110]. A color histogram is a $m-$ dimensional vectors $\mathcal{T}(f(t_n)) = H_n(j)$, $j = 1, .., m$, where m is the number of color levels and $H_i(j)$ is the number of pixel belonging to the $n-th$ frame and having a color level equal to j.

The most simple histogram-based metric uses the gray-levels histograms as shown in equation 3.6:

$$\mathcal{D}(\mathcal{T}(f(t_n)), \mathcal{T}(f(t_{n+l}))) = \sum_{j=1}^{n} |H_n(j) - H_{n+1}(j)| \tag{3.6}$$

Gargi e al. [42] evaluate the performances of histogram-based methods using six color spaces: *RGB*, *HSV*, *YIQ*, *L*a*b**, *L*v*u** and Munsell. From this analysis the authors have underlines as the best performances in terms both of precision and computational cost are a characteristic of *YIQ*, *L*a*b** and *HSV* spaces.

Authors agree that pixel-based methods are highly sensitive to motion of objects, so they generate an high rate of false detection. On the contrary histogram-based methods provide a better trade-off between accuracy and speed, and the performances of such methods are good for the case of abrupt scene changes. However, color histograms provide information about the color composition of images, but not about the spatial distribution of color, so different images could have similar histograms.

For these reasons hybrid approaches [73] based on *local histogram comparison* have been proposed in according to 3.7 and 3.7:

$$\mathcal{D}(\mathcal{T}(f(t_n)), \mathcal{T}(f(t_{n+l}))) = \sum_{k=1}^{b} DP(n, n+1, k) \tag{3.7}$$

with:

$$\mathcal{D}(\mathcal{T}(f(t_n)), \mathcal{T}(f(t_{n+l}))) = \sum_{j=1}^{n} |H_n(j,k) - H_{n+1}(j,k)| \tag{3.8}$$

Moreover, in order to overcome the limitation due to the use of single comparison technique, in [83] Philips and Wolf propose a multi-attribute algorithm for detecting cuts in video programs: the algorithm uses a motion metric to identify a set of cuts, then uses luminance histograms to eliminate false cuts. Motion vector analysis is also used in [34], while an edge tracking is used in [115] to detect both gradual and abrupt transitions.

The introduction of the MPEG standard has redirected the research efforts for the video segmentation in the compressed-domain. The most common approaches in this domain are based on DC-components [70, 117], DC-sequences [110] and number of interpolated blocks [33].

In such methods the DCT coefficients of the different $8x8$ are merged frame blocks are grouped into a particular vector $\mathcal{T}(f(t_n)) = V_n = dct_1, dct_2, ..$ and to evaluate the frames difference the equation 3.9 is used :

$$\mathcal{D}(\mathcal{T}(f(t_n)), \mathcal{T}(f(t_{n+l}))) = \frac{V_n \cdot V_{n+1}}{|V_n| \cdot |V_{n+1}|} \tag{3.9}$$

Eventually, in [36] a unified approach to scene change detection in both uncompressed and compressed video is proposed.

3.3.3.4 Analysis of methods for detecting gradual transitions

Also the detection of gradual transitions has been widely investigated. At the best of our knowledge, a lot of techniques and algorithms have been proposed for fading (fade-in, fade-out) regions detection [35], while there are few works about the detection of other special effects such as dissolve and wipe.

Xiong et al. [107] use a classical *block-based comparison* based on a *Step-variable* algorithm for detecting gradual transitions. In this approach the comparison is not performed on consecutive frames, but on frames at a variable distance. Similarly, Yeo e Liu [110] use a *plateau detection* method comparing frames at a fixed distance.

Zhang e al.[116] propose a particular *histogram based comparison* called *twin-comparison*[9]. In such methods the histograms of frames belonging to a hypothetic dissolve region are investigated using two separate thresholds for abrupt and gradual transition. Li e al. [57] propose a *Model-based Video Segmentation* algorithm based on the analysis of frame-blocks gray-levels. By means of the use of two two separate thresholds, in a first step abrupt transitions are detected, then in the second step dissolves are isolated.

Other important approaches [3, 35] works on frame statistical features as dissolve variance respect to frame intensity and its derivative function. In a dissolve region the variance will have a parabolic behavior.

Lienhart [62] casts the problem of automatic dissolve detection as a pattern recognition and learning problem. Nam and Tewfik [74] use a technique based on B-spline polynomial curve fitting. In [58] Li and Wei propose a dissolve detection method based on the analysis of Joint probability Images, while in [60] a motion-tolerant algorithm for dissolve detection is presented.

3.3.3.5 A short outline on video scenes detection techniques

In the literature, several automatic techniques for video scene detection have been proposed.

The majority of such methods uses audio and visual information jointly for accomplishing the above task. In [104], the main audio and visual features that can effectively characterize scene content and some algorithms for video segmentation and classification are reported. In [40] some visual useful metrics for scene change detection based on scene lighting and intensity distribution are presented; in opposite [65] focuses the attention on the associated audio information for video scene analysis. In [106] a framework to group shots based on the analysis of video content (in terms of visual, position, camera, motion and audio features) continuity is performed. In [99] video scenes are detected on the base of cromacity, lighting conditions and ambient sound video properties. In [4] a Markov model approach for scene detection based on audio and visual video analysis is proposed. Eventually, in [82] a scheme for identifying scenes on the base of video genre has been developed.

Chapter 4

A Model for a Foveated Image and Video Analysis

4.1 Introduction: the Animate Vision approach

Content Based Image Retrieval (CBIR) systems and some video segmentation approaches rely upon the effectiveness of image (frame) similarity models, consequently, the effectiveness of querying and shot detection processes largely depends on the strategy adopted to analyze and represent the image content.

Assessing the similarity between two images can be reformulated as a task of visual search: *given a target image I_q and a test image I_t, is there an instance of the target in the test image*?

To this end, it is of relevance that in most biological vision systems, only a small fraction of the information registered at any given time reaches levels of processing that directly influence behavior and, indeed, attention seems to play a major role in this process. Notwithstanding, classical approaches in computer vision and in CBIR, consider the input image as a static entity, to be processed in a passive way.

To correct such trend, the well-know `Animate Vision paradigm` has been proposed [7] and used in a lot of applications.

The term `attention` captures the cognitive functions that are responsible for filtering out unwanted information and bringing to consciousness what is relevant for the observer [51].

When analyzing an image, human subjects mainly concentrate inspection on a subset of points (regions). A basic example is the generation of saccades, i.e. ocular movements that allow to acquire high resolution images (foveation) of the most relevant part of the scene. More precisely an average of three eye fixations per second generally occurs during active looking; these eye fixations are intercalated by rapid eye jumps, the saccades, during which vision is suppressed.

Noton and Stark [77] claimed that when a particular visual pattern is viewed, a particular sequence of eye movements is executed and this sequence is important in accessing the visual memory for the pattern. An example is provided in Figs. 4.1, 4.2.

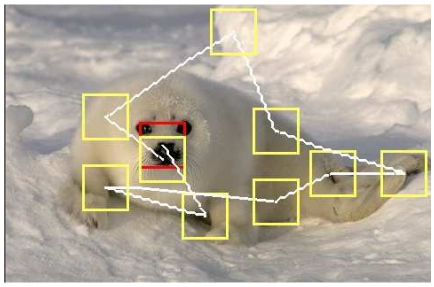


Figure 4.1: Eye movements sequence over a sample image

Figure 4.2: Fovea view of the sample image

In other words the points of an image have not the same importance: human eye attention is captured only from certain points, called `saliency points` and *Animate Vision is the visual biological system capacity of quickly detecting interesting regions of visual stimulus, a computational counterpart of using gaze shifts to enable a deictic perceptual-motor strategy on an image* [7].

In this note, we argue that animate vision mechanisms should be taken into account in CBIR, since they allow to concentrate the visual process on a circumscribed region of the visual field, the "focus of attention" (`FOA`), which is sequentially shifted across the scene either in a bottom-up, saliency driven fashion, and/or in a top-down, model driven way.

To the best of our knowledge, the animate perspective has never been considered for the CBIR problem which still adheres to a passive approach. However, it may play a twofold role for the purposes of this work:

1. An attentional scheme has as its main goal the selection of certain aspects of the input stimulus while causing the effects of other aspects of the stimulus to be minimized;

2. Attention introduces a third dimension, beyond features and relationships, namely the dimension of times: features and relationships are not established as static structures, but are incrementally set-up along the visual inspection task. In other terms, attention "linearizes" the 2D structure, naturally reducing visual matching complexity.

In particular, in the following, we discuss the way to generate two fixation sequences from a query image I_q and a test image I_t, respectively, and how to compare the two sequences in order to compute a similarity measure of the two images.

4.2 Outline of the model for attentive/foveated image analysis: the mapping in the WW space

Indeed, at the heart of our ability to detect changes and similarities between images is an intriguing and remarkable information selection/reduction process.

As already described, through an attentive visual inspection, we view scenes in the real world by moving our eyes three to four times each second, and integrating information across subsequent fixations (`foveation points`). Each fixation defines a focus of attention (FOA) on a certain region of the scene, and the FOA sequence is denoted `scanpath` [77]. According to scanpath theory, patterns that are visually similar, give rise to similar scanpaths when inspected by the same observer under the same viewing conditions (current task or context). Such animate visual behavior [6] can be computationally modeled as described in Fig. 4.3.

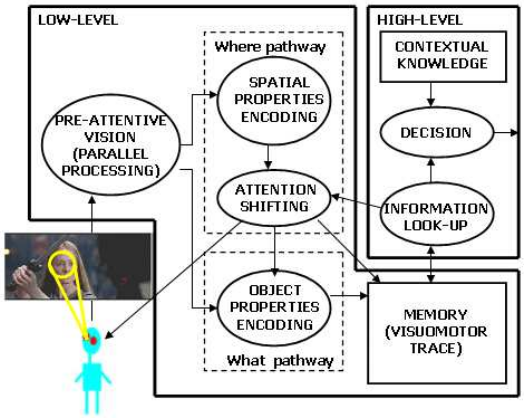


Figure 4.3: A general model of attentive/foveated image analysis.

At a lower level, the observer generates visuomotor patterns, related to the images content. At a higher level, the observer evaluate images similarity by judging his own visuomotor behavior in the context of prior knowledge available (for CBIR it could be the database image categories, in the opposite, in a video segmentation task, it could be the kinds of possible transitions).

In the `pre-attentive stage`, salient features are extracted by specific detectors operating in parallel for all points of the image, at different spatial scales, and organized in the form of contrast maps. In order to obtain such a representation different methods can be adopted, e.g., [86], [50], [37].

It is worth remarking, that the model presented in this work is unaffected by the method chosen to implement such pre-attentive stage. We experimented with schemes proposed in [37], [50], and opted for the latter due to simplicity and limited computational complexity.

Precisely, low-level vision features are derived from the original color image decomposed at several spatial scales using Gaussian and oriented pyramids (via convolution with Gabor filters). Note that pyramid computation is an $O(|\Omega|)$ method, where $|\Omega|$ represents the number of samples in the image support Ω.

The features considered are:

- brightness *(I)*;
- color channels tuned to red (R), green (G), blue (B) and yellow (Y) hues;
- orientation *(O)*.

From color pyramids, red/green (RG) and blue/yellow (BY) pyramids are derived by subtraction. Then, from each pyramid a contrast pyramid is computed encoding differences between a fine and a coarse scale for a given feature. As a result, one contrast pyramid encodes for image intensity contrast, four encode for local orientation contrast, and two encode for red/green *(RG)* and blue/yellow *(BY)* contrast (see [50], for details).

From a computational point of view, the generation of a scanpath under free viewing conditions, can be accomplished in three steps: (i) Selection of interesting regions; (ii) Features extraction from the detected regions; (iii) Search of the next interesting region.

To this aim, the pre-attentive representation undergoes specialized processing through a ‘‘`Where"` system devoted to localizing regions (objects) of interest, and the ‘‘`What"` system tailored for analyzing them (`WW Space`). Clearly, tight integration of these two information pathways is essential, and indeed attentive mechanisms play a fundamental role. A plausible assumption is that, in the "What" pathway, early layers provide feature extraction modules, whose activity is subjected to temporal modulation by the "Where" pathway and the related attention shifting mechanism, so that unmodulated responses are suppressed.

4.2.1 The Where system: from pre-attentive features to attention shifting

In the "Where" pathway, the pre-attentive contrast maps are combined into a `master or saliency map` [50], [71], [1], which is used to direct attention to the spatial location with the highest saliency through a `winner take-all (WTA)` network (`attention shifting` stage). The region surrounding such location represents the current focus of attention (FOA), F_s . By traversing spatial locations of decreasing saliency, a scanpath, $(F_s)_{s=1,2,..}$ is obtained by connecting a sequence of FOAs, and stored. An example of a scanpath is reported in Fig.4.4.

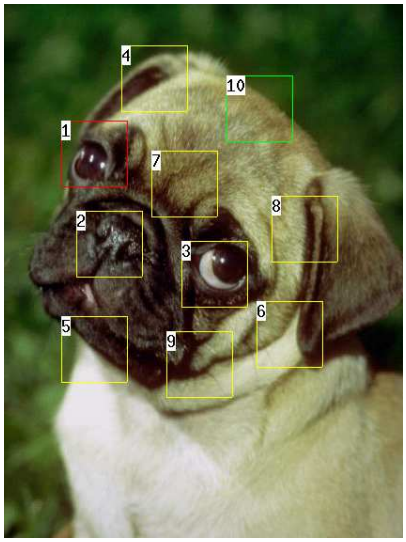


Figure 4.4: Example of a scanpath

It is important to note that, in general and specifically in this work, a "working memory" retains either a representation of a set of visual features (measured at FOAs) and a motor map of how such features have been explored; indeed, the memory of an attentive system is a `visuomotor trace` of a world view [79],[45], rather than a classical feature representation of the original image, and any subsequent information-lookup task entails a prediction/confirmation upon such visuomotor scheme.

More in details, the goal of the "Where" system is to build a saliency map of the image and define over this map the motor trace, that is the sequence of FOAs $(F_s)_{s=1,2,...,N_f}$. To this end, the contrast features for intensity, color and orientation, obtained from the preattentive stage, are summed across scales (pyramid levels) into three separate contrast maps, one for intensity, one for color and one for orientation. Eventually, the three maps, normalized between 0 and 100, are linearly summed into a unique master map (for simplicity, we compute the latter as the average of the three maps), or saliency map (SM). By using the SM map, the attention shifting mechanism could be implemented through a variety of ways (e.g., [44], [86], [50], [101]). One intuitive method for traversing spatial locations of decreasing saliency, is to use a winner-take-all (WTA) strategy [50], [101], in which the most salient location "wins" and determines the setting of the FOA; the winner is subsequently inhibited in order to allow competition among less salient locations, for predicting the next FOA.

In Fig. 4.5 an overview of the results coming from application of the Animate Vision process for generating the scanpath of a given image is shown.

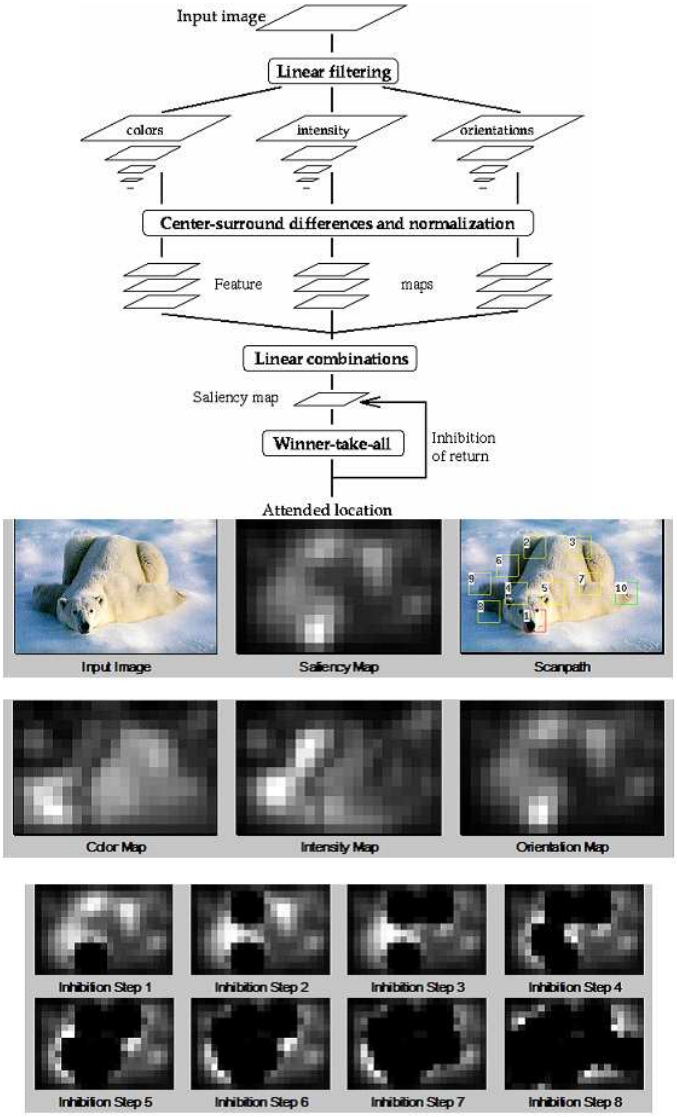


Figure 4.5: From features maps to scanpath

A simple and efficient way of implementing such strategy is through a WTA neural network, e.g. an array of integrate-and-fire neurons with global inhibition [50]. It is worth noting that a WTA algorithm, due to fast convergence properties, has $O(n)$ time complexity, n being the number of processing elements (neurons) of the network. In our scheme the number of neurons is constrained by the number of samples in the saliency map (each point of the map, represents the input to one neuron). Since the map resides at an intermediate scale between the highest and the lowest resolution scales, namely at scale 4, a reduction factor 1:16 is achieved with respect to the original image, thus the time complexity of the WTA stage is given by $|\Omega|/16$ time units.

This solution has the advantage of providing information on the fixation time spent on the FOA (the firing time of WTA neurons) and our model, differently from others proposed in the literature, explicitly exploits such information.

After the "Where" processing, an image I_i is represented by a spatio-temporal, or motor trace, representing the stream of foveation points $(F_s^i(p_s;\tau_s))_{s=1,2,\ldots,N_f}$, where $p_s = (x_s, y_s)$ is the center of FOA s, and the delay parameter τ_s is the observation time spent on the FOA before a saccade shifts to F_{s+1} [88], provided by the WTA net.

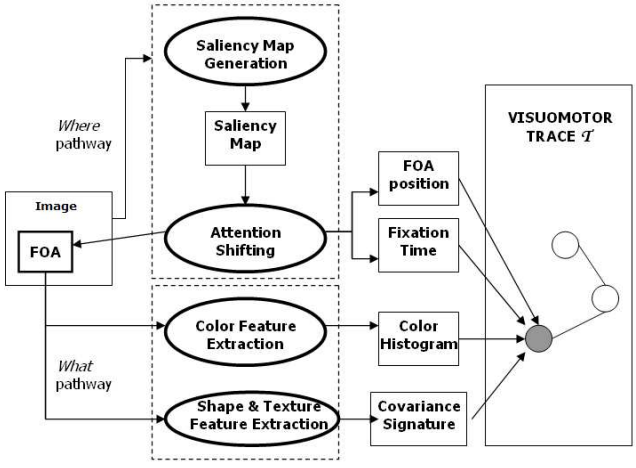


Figure 4.6: Generation of the motor trace of a given image

As outlined in Fig. 4.6, the generation of spatio-temporal information is basically an information reduction step in which we assume that the "Where" system "projects" towards the "What" system and signals the FOA to be analyzed. The scheme shows the selection of a FOA by the "Where" pathway, and the extraction of FOA information by the "What" pathway. For visualization purposes, the trace is represented as a graph-like structure: each node corresponds to a single FOA, and the arc joining two FOAs denotes a saccade.

4.2.2 The What pathway: properties encoding

In the "What" pathway, features are extracted from each highlighted FOA, relative to color, shape and texture. A FOA is represented in the intensity and color opponent pyramids, at the highest resolution scale. Note that in biological vision, the spatial support of the FOA is usually assumed as circular; here, for computational purposes, each FOA is defined on a square support $D_{p_s} \subseteq \Omega$, centered on p_s, of dimension $|D_{p_s}| = \frac{1}{36}|\Omega|$ (we drop the τ parameter for sake of simplicity).

In our case, for each FOA F_s^i, the "What" pathway extracts two specific features, the color histogram $h_b(F_s^i)$ and the edge covariance signature $\Xi_{F_s^i}$ as described in the following.

- `Color features`. A color image, is a mapping from the discrete image support $\Omega \subseteq Z^2$ to an m-dimensional range, $I : \Omega \to Q \subseteq Z^m$; in other terms, it is a set of single-valued images, or channels, sharing the same domain, i.e., $I(x, y) = (I_i(x, y))^T$, where the index $i = 1, .., m$, defines the i-th color channel and (x, y) denotes a point in the Ω lattice. $Q = \{q_1, ..., q_N\}$ is the set of colors used in the image. Thus, given a set of representative colors $Q = \{q_1, ..., q_B\}$, a color histogram $h(F(p)) = \{h_b\}$ of the FOA $F(p)$ is defined on bins b ranging in $[1, B]$, such that h_b given for any pixel in D_p, is the probability that the color of the pixel is $q_b \in Q$. Here, $B = 16 \times 16 \times 16$ is used. For a three channel frame, the FOA histogram calculation time is $|D_p| \times 3$.

- `Shape and texture features` A wavelet transform (WT) of the FOA has been adopted [67]. Denote the wavelet coefficients as $w_l^k(x,y)$, where $(x,y) \in D_p$, l indicates the decomposition level and k indexes the sub-bands. In our case, due to the limited dimension of the FOA, only a first level decomposition ($l = 1$) is considered, and in the sequel, for notational simplicity the index l is dropped. Decomposition gives rise to 4 subregions of dimension $|D_p|/4$. Then, only detail components of the WT are taken into account, in order to characterize shape (edges) and texture. Namely, for $k = 1, 2, 3$, the detail sub-bands contain horizontal, vertical and diagonal directional information, respectively, and are represented by coefficient planes $\left[\{w^k(x,y)\}\right]_{k=1,2,3}$. The wavelet covariance signature is computed as the feature vector of coefficient covariances $\Xi_{F_s^m} = \{\xi_{X,Y}\}$, where:

$$\xi_{X,Y} = \sum_{x,y}\{\frac{1}{|D_p|/4}\sum_{k=1}^{3} X_k(x,y)Y_k(x,y)\}. \tag{4.1}$$

  The pair (X_k, Y_k) is in the set of coefficient plane pairs $\{(w_i^k, w_j^k)\}$, i and j being used to index the three channels, and (x,y) span over the sub-band lattice of dimension $|D_p|/4$ [67]. Clearly, $|\Xi| = 18$.

Eventually, the saccadic movements together with their resultant fixations, and feature analysis of foveated regions, allow the formation of the trace $\mathcal{T}(I_i)$ in the WW space, briefly $\mathcal{T}(i)$, of the view observed in the image I_i:

$$\mathcal{T}(i) = (\mathcal{T}_s^i)_{s=1,\ldots,N_f} \tag{4.2}$$

where $\mathcal{T}_s^i = (F_s^i, h_b(F_s^i), \Xi_{F_s^i})$. Note that the process described above obtains a visuomotor trace as generated under free-viewing conditions (i.e., in the absence of an observation task), which is the most general scanpath that can be recorded. Clearly, according to different viewing conditions an image may be represented by different maps in such space; such "biased" maps can be conceived as weighted $\mathcal{T}$s, or sub-paths embedded in the context-free one, as Yarbus' seminal experiments have shown [109].

In Fig. 4.7 an example of the features extracted in the "What" pathway is reported.

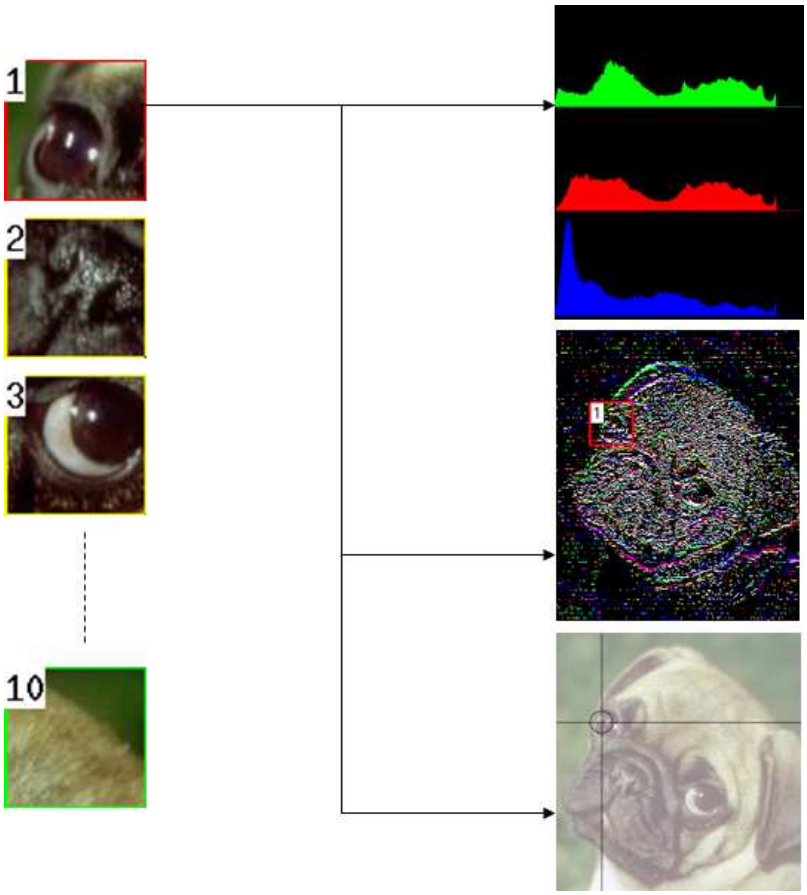


Figure 4.7: Features from "What" pathway

4.3 Evaluating images similarity by attention consistency

For defining a similarity function $\mathcal{M}$ between two images, we rely upon a given assumption: *the visuomotor generation process performed on a pair of similar images under the same viewing conditions will generate similar traces*, a property that we denote `Attention Consistency`.

In Fig. 4.8 two similar images with respective visuomotor traces are shown.

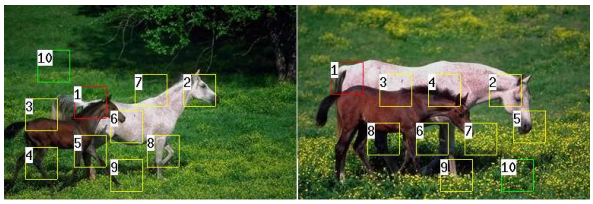


Figure 4.8: Similar images with visuomotor traces

Hence, the image-matching problem can be reduced to a visuomotor traces matching; in fact, experiments performed by Walker and Smith [103], provide evidence that when observers are asked to make a direct comparison between two simultaneously presented pictures, a repeated scanning, in the shape of a FOA by FOA comparison, occurs [103].

Thus, in our model, two images are similar if `homologous` FOAs have similar color, texture and shape features, are in the same spatial regions of the image, and are detected with similar times. The procedure, is a sort of inexact matching, which we denote `animate matching` and is summarized in Fig. 4.9.

Given a fixation point $F_r^t(p_r;\tau_r)$ in the test image I_t, the procedure selects the homologous point $F_s^q(p_s;\tau_s)$ in the query image I_q among those belonging to a local temporal window, that is $\tau_s \in [s-H, s+H]$. The choice is performed by computing for the pair F_r^t and F_s^q:

$$\mathcal{M}^{r,s} = \alpha_a \mathcal{M}^{r,s}_{spatial} + \beta_a \mathcal{M}^{r,s}_{temporal} + \gamma_a \mathcal{M}^{r,s}_{visual}, \tag{4.3}$$

where $\alpha_a, \beta_a, \gamma_a \in [0,1]$, and by choosing the FOA s as $s = \arg\max\{\mathcal{M}^{r,s}\}$.

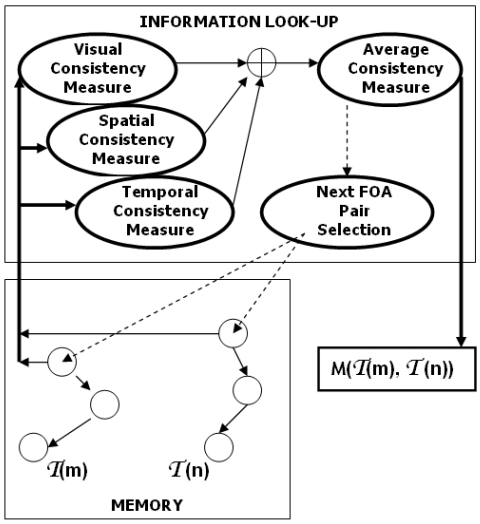


Figure 4.9: Animate matching between two images represented as visuomotor traces $\mathcal{T}(m), \mathcal{T}(n)$ in the WW space

In other terms, the choice of the new scanpath is top-down driven, so as to maximize the similarity of the query image respect to the test image; the analyzing scanpath results to be a sub-path of the original free-viewed one. This "best fit" strategy has been chosen in order to reduce the sensitivity of the algorithm both to the starting FOA point and to the fact that, in similar images, some FOAs could be missing due to lighting changes and noise. Such "best fit" is retained and eventually used to compute the consistency $\mathcal{M}(\mathcal{T}^t, \mathcal{T}^q)$ as the average consistency of the first $N_f^{'}$ consistencies:

$$\mathcal{M} = \frac{1}{N'_f} \sum_{f=1}^{N'_f} \mathcal{M}_f^{r,s}, \tag{4.4}$$

where $N'_f <= N_f$. Right-hand terms of Eq. 4.3, namely $\mathcal{M}_{spatial}^{r,s}$, $\mathcal{M}_{temporal}^{r,s}$, $\mathcal{M}_{visual}^{r,s}$, account for local measurements of spatial temporal and visual consistency, respectively.

These are calculated as follows.

- `Local spatial consistency`. $\mathcal{M}^{r,s}_{spatial}$ is gauged through the ℓ^1 distance between homologous FOA centers, $d(p_r, p_s) = |x_r - x_s| + |y_r - y_s|$. The distance is "penalized" if, for the two images, the displacement between the current FOA and the next one is not in the same direction:

$$\hat{d}(p_r, p_s) = d(p_r, p_s) \cdot e^{-\Delta(p_r, p_s)}, \tag{4.5}$$

Δ being the difference of direction between two FOAs, $\Delta = \zeta \cdot \mathrm{sgn}[(x_r - x_{r-1}) \cdot (x_s - x_{s-1})] \cdot \mathrm{sgn}[(y_r - y_{r-1}) \cdot (y_s - y_{s-1})]$, where ζ is a penalization constant. Thus, after $\hat{d}(p_r, p_s)$ normalization:

$$\mathcal{M}^{r,s}_{spatial} = 1 - \hat{d}(p_r, p_s). \tag{4.6}$$

- `Local temporal consistency`. $\mathcal{M}^{r,s}_{temporal}$ takes into account the difference of time that the observer gazes at two different fixation points. To this end the ℓ^1 distance is introduced, $d(\tau_r, \tau_s) = |\tau_r - \tau_s|$. The distance is normalized with respect to the maximum fixation time of the scanpath. Then temporal consistency is calculated as

$$\mathcal{M}^{r,s}_{temporal} = 1 - d(\tau_r, \tau_s). \tag{4.7}$$

- `Local visual consistency`. $\mathcal{M}^{r,s}_{visual}$ is defined using either color and texture/shape properties. Evaluation of consistency in terms of color is performed by exploiting well known histogram intersection, which again is an ℓ^1 distance on the color space [6]: given the two color histograms $h(F^m_r)$ and $h(F^n_s)$, defined on the same number of bins $b = [1, \ldots, B]$, $d^{r,s}_{col} = \sum_{b=1}^{B}(\min(h_b(F^m_r), h_b(F^n_s)))/\sum_{b=1}^{B} h_b(F^m_r)$, where $\sum_{b=1}^{B} h_b(F^m_r)$ is a normalization factor. Then,

$$\mathcal{M}^{r,s}_{col} = 1 - d^{r,s}_{col}. \tag{4.8}$$

By using the ℓ^1 distance $\frac{1}{R}\sum_{i=1}^{|\Xi|}\frac{|\Xi_{F_r^m}[i]-\Xi_{F_s^n}[i]|}{\min(|\Xi_{F_r^m}[i]|,|\Xi_{F_s^n}[i]|)}$, shape and texture consistency is measured as:

$$\mathcal{M}_{tex}^{r,s} = 1 - \frac{1}{R}\sum_{i=1}^{|\Xi|}\frac{|\Xi_{F_r^m}[i] - \Xi_{F_s^n}[i]|}{\min(|\Xi_{F_r^m}[i]|, |\Xi_{F_s^n}[i]|)}, \tag{4.9}$$

where R is a normalization factor to bound the sum in $[0, 1]$, and $|\Xi|$ the number of features in the feature vector Ξ computed through Eq. 4.1. Eventually, FOA's visual content consistency is given from the weighted mean of terms calculated via Eqs. 4.8 and 4.9:

$$\mathcal{M}_{visual}^{r,s} = \mu_1\mathcal{M}_{col}^{r,s} + \mu_2\mathcal{M}_{tex}^{r,s}. \tag{4.10}$$

The computation cost of Eq. 4.3 is approximately linear in the number of histogram bins B, since $|\Xi| = 18$, and Eqs. 4.6 and 4.7, are performed in constant time units. Thus, the matching algorithm requires $(2H + 1)BN_f^{'}$ operations, which means that, once H and B have been fixed as in our case, the AC algorithm is linear in the number of FOAs $N_f^{'}$. The algorithm 1 reported in the following summarizes the animate matching procedure

Algorithm 1 Attention Consistency (AC) Algorithm

Given $\mu_1, \mu_2, \alpha_a, \beta_a, \gamma_a, \mathcal{T}^q$ and $\mathcal{T}^t$
$\mathcal{M} \leftarrow 0$
for $i = 1, ..., N_f^{'}$ **do**
  $j =$ `selectFOA`$(F_i^q, \mathcal{T}^t, H)$ using the best fit strategy
  Compute `local spatial consistency` $\mathcal{M}_{spatial}^{i,j}$ between F_i^q and F_j^t
  Compute `local temporal consistency` $\mathcal{M}_{temporal}^{i,j}$ between F_i^q and F_j^t
  Compute `local visual consistency` $\mathcal{M}_{visual}^{i,j}$ between F_i^q and F_j^t
  $\mathcal{M}^{i,j} \leftarrow \alpha_a\mathcal{M}_{spatial}^{i,j} + \beta_a\mathcal{M}_{temporal}^{i,j} + \gamma_a\mathcal{M}_{visual}^{i,j}$
  $\mathcal{M} \leftarrow \mathcal{M} + \mathcal{M}^{i,j}$
$\mathcal{M} \leftarrow \frac{\mathcal{M}}{N_f^{'}}$
Return $\mathcal{M}$

Chapter 5

Context-sensitive Queries for Image Retrieval

5.1 Introduction

In spite of recent important research efforts, image indexing and retrieval of images in large databases is still a challenging task. Clearly, the greatest difficulty is to find features that effectively represent image content, while adopting image data structures that organize efficiently the feature space.

In the framework of Content Based Image Retrieval (CBIR), Query By Example (QBE) is considered a suitable approach because the user handles an intuitive query representation: the form of the query, namely an image, is that of the data to be evaluated. However, a hallmark all too easily overlooked is that when the user is performing a query, he is likely to have some semantic specification in mind, e.g. "I want to see a portrait", and the portrait example provided to the query engine is chosen to best represent the semantics. However, traditional image databases are not able to express either such semantics or similarity rules consistent with semantics; this problem is known as "semantic gap" [28],[112], [20].

In this chapter, it will be shown how, by embedding within image inspection algorithms active mechanisms of biological vision such as saccadic eye movements and fixations, a

more effective query processing in image database can be achieved. In particular, it will be discussed how the model for foveated image analysis can be used to discover and represent the hidden semantic associations among images, in terms of categories, which in turn drive the query process. Also, such associations allow an automatic pre-classification, which makes query processing more efficient and effective. Preliminary results will be presented and the proposed approach compared with recent techniques described in the literature.

5.2 The Context-sensitive approach

As pointed out by Santini *et al.* [92], in traditional databases information is encoded in symbols that have a syntax and a semantics relying upon a distinction between structurally atomic and structurally composed representation.

Similarly, current image databases mainly work within the framework of a syntactical description of the image (a scene composed of objects, that are composed of parts, etc.). Consequently, the only meaning that can be attached to an image is its similarity with the query image, namely the meaning of the image is determined by the interaction between the user and the database.

The main issue here is that perception indeed is a relation between the perceiver and its environment, which is determined and mediated by the goals it serves (i.e., context): we perceive the world as made of objects because the goal of our perception is to help us act upon the world [31]. Thus, considering for instance Leonardo's Mona Lisa (Fig. 5.1): should it be classified as a portrait or a landscape?

Clearly, the answer depends on the context at hand. In this perspective, it is useful to distinguish between the "What" and "Where" aspects of the sensory input and to let the latter serve as a scaffolding holding the would-be objects in place [31]. Such distinction offers a solution to the basic problem of scene representation - what is where - by using the visual space as its own representation and avoids the problematic early commitment to a rigid designation of an object and to its crisp segmentation from the background (on demand problem, binding problem) [31]. Consider again Fig. 5.1 and let Leonardo's Mona Lisa represent one target image I_t. An ideal unconstrained observer would scan along free viewing the picture by noting regions of interest of either the landscape and the portrait, mainly relying on physical relevance (color contrast,etc). However this is unlikely in real observations, since the context (goals) heavily influence the observation itself.

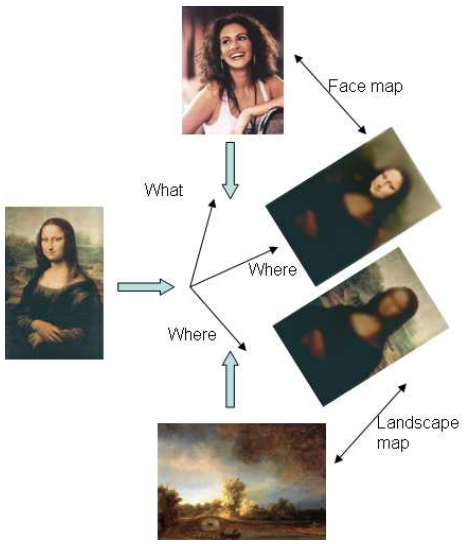


Figure 5.1: The "What-Where" similarity space: the "Where" dimension (corresponding to the image location) and the two "What" dimensions (similarity to a face image and to a landscape image) are shown. Switching to one "What" dimension or to the other one, depends on the context/goal provided, represented in the image by a face example and a landscape example

For example, in a face detection context, the goal is accomplished when "those" eye features are encountered "here" above "these" mouse features. On the other hand, when a landscape context is taken into account, the tree features "there" near river features "aside" may better characterize the Mona Lisa image. Clearly, in the absence of this active binding, the Mona Lisa picture can either be considered a portrait or a landscape; *per se*, it has no meaning at all.

This dynamic binding of context-sensitive components is an example of a deictic strategy [85]. Linguists classify words like "this" or "that" as deictic because they constrain the listener's attention to a specific target from a set of candidate targets. Such a strategy is known to be used in vision to circumscribe the possible interpretations of perceptual feedback to the current context.

Visual fixations, following eye movements is one example of deictic strategy [85]. The act of fixating on an object centers the target in the retinotopic array, potentially simplifying cognitive process to deal with the "object I'm looking at" rather than with the attributes that distinguish that particular objects from all others.

The computational counterpart of using gaze shifts to enable a deictic perceptual-motor strategy is named, as already seen, `Animate Vision`.

In according to the model presented in the previous chapter, we propose a representation scheme in which the "What" entities are coded by their similarities to an ensemble of reference features, and, at the same time the "Where" aspects of the scene structure are represented by their spatial distribution with respect to the image support domain.

This is obtained by generating a perceptual-motor trace of the observed image, which we have denoted `visuomotor trace` $\mathcal{T}$. Thus, the similarity of a query image I_q to a test image I_t of the data set can be assessed within the "What+Where" (WW) space, or equivalently by comparing their $\mathcal{T}$s (`Animate Matching`).

In this sense we agree with [92] that the meaning that can be attached to an image is its similarity with the query image. In fact, by providing a query image, we can "shape" the WW space by "pinning features to a corkboard", which, in some way, corresponds to shape the geometric structure of the feature space. In computer vision terms, we are exploiting "top-down" information to perform the matching.

Clearly, the approach outlined above assumes the availability of a context, and of a representation of such context in order to drive the perceptual actions in the WW space. There is a wealth of research in neurophysiology [72] and in psychology [39] showing that humans interact with the world with the aid of categories. When faced with an object or person, an individual activates a category that according to some metric best matches the given object, and in turn the availability of a category grants the individual the ability to recall patterns of behavior (stereotypes as built on past interactions with objects in a given category. In these terms, an object is not simply a physical object but a view of an interaction.

Thus, differently from [92], we allow for the possibility of providing the database with a preliminary context in the form of a tunable WW space, where an image is not definitely classified but, according to a finite number of different viewing strategies, is represented in terms of the likelihood to belong to a finite number of pre-specified categories.

In the proposed system, AICQ, we functionally distinguish these basic components:

1. A component which performs a "free-viewing" analysis of the images, corresponding to "bottom-up" analysis mainly relying on physical features (color, texture, shape) and derives their $\mathcal{T}$s as described in chapter 4.

2. A WW space in which different WW maps may be organized according to some selected categories; any image is to be considered the support domain upon which different maps ($\mathcal{T}$s) can be generated, according to viewing purposes.

3. A query module (high level component) which acts upon the WW space by considering "top-down" information, namely, context represented through categories, and exploits animate matching to refine the search.

A functional outline of AICQ is depicted in Fig. 5.2.

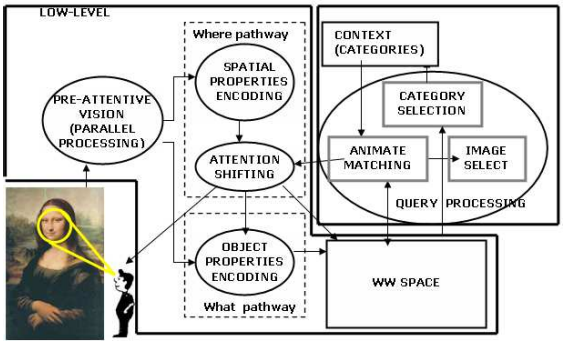


Figure 5.2: A functional view of the system at a glance

5.3 Endowing the WW space with context: category representation

An observer will exhibit a consistent attentive behavior while viewing a group of similar images. This is consistent with the fact that we can categorize objects in categories, where each category represents a stereotyped view of the interaction with a class of objects [39]. Thus, in our case an image category, say $\mathcal{C}_n$, can be seen as a group of images from which, under the same viewing conditions, similar $\mathcal{T}$s are generated.

In the following we first discuss in general terms how categories can be structured in a set of clusters from a probabilistic standpoint, by using the Expectation-Maximization (EM) algorithm. Then in order to achieve a balanced solution of the clustering procedure, which is desirable for a large database, a balanced variant of the EM algorithm will be introduced. The balanced clustering obtained in this way is appealing, since allowing category representation in terms of a balanced tree.

To this end, the AICQ system requires an initial step in which an initial image set and the associated category classification have been pre-selected, through a supervised process [30].

5.3.1 Probabilistic learning of category clusters

We use a probabilistic framework in order to allow the association of each image to different categories, according to the concept of WW space, thus avoiding a hard assignment to a single $\mathcal{C}_n$.

What we need is a procedure capable to estimate the likelihood $P(\mathcal{T}^i|\mathcal{C}_n)$ that a visuomotor trace (VMT) $\mathcal{T}^i$ is generated by observing a test image, which belongs to one of the categories $\mathcal{C}_n, n = 1, \cdots, N$.

However, the browsing of all database categories and the selection of the "best match", the most evident solution, is computationally prohibitive for a very large database, the number of comparisons being equal to the total number of images in the data set. A more efficient solution is to subdivide/cluster the images belonging to a given category $\mathcal{C}_n$ into subgroups called `category clusters`, $\mathcal{C}_n^l$.

Note that a VMT can be thought of as a feature vector so that the goal of clustering [52] is to assign a label l to the different VMTs (images), where $l \in [1, \ldots, L_n]$ identifies a particular category cluster $\mathcal{C}_n^l$, which can be selected with a certain probability $P(l)$.

Denote $\mathcal{T} = \{\mathcal{T}^1, \cdots, \mathcal{T}^N\}$ the VMT data set generated by sampling independently from the following generative model, namely a mixture model:

$$p(\mathcal{T}|\Theta) = \sum_{l=1}^{L} \alpha_l p_l(\mathcal{T}|\theta_l), \tag{5.1}$$

where $\Theta = \{\alpha_1, \cdots, \alpha_L, \theta_1, \cdots, \theta_L\}$, α_l being the mixing proportions subject to constraints $\alpha_l \geq 0$, $\sum_{l=1}^{L} \alpha_l = 1$ and the distribution $p_l(\mathcal{T}|\theta_l)$ is a single multivariate gaussian distribution with parameters (mean and covariance) $\theta_l = \{\mathbf{m}_l, \mathbf{\Sigma}_l\}$, more precisely

$$p_l(\mathcal{T}^i|\mathbf{m}_l, \mathbf{\Sigma}_l) = \frac{exp(-\frac{1}{2}(\mathcal{T}^i - \mathbf{m}_l)^T \mathbf{\Sigma}_l^{-1}(\mathcal{T}^i - \mathbf{m}_l))}{(2\pi)^{(D/2)}|\mathbf{\Sigma}_l|^{1/2}}. \tag{5.2}$$

here D denotes the dimension of the feature space.

Let $\mathcal{Z} = \{z_1, \cdots, z_N\}$ be the corresponding set of hidden random variables such that $z_i = l$ when $\mathcal{T}^i$ has been generated following $p_l(\mathcal{T}|\theta_l)$. Then, the `complete` log-likelihood of the observed data is given by:

$$\log \mathcal{L} = \log p(\mathcal{T}, \mathcal{Z}|\Theta) = \sum_{i=1}^{N} \log(\alpha_{z_i} p_{z_i}(\mathcal{T}^i|\theta_{z_i})), \tag{5.3}$$

from which maximum likelihood parameter estimates can be obtained. Since, in terms of the mixture model we are dealing with an incomplete data problem (i.e., we must simultaneously determine the labelling $p(\mathcal{Z}|\mathcal{T}, \Theta)$ given distribution parameters Θ and viceversa), a suitable choice for the maximization of the likelihood is the EM algorithm [27].

The EM algorithm starts with some initial guess at the maximum likelihood parameters θ_l^0, and then proceeds to iteratively generate successive estimates, $\theta_l^1, \theta_l^2, \cdots$ by repeatedly applying the following two steps for $t = 1, 2, \cdots$ [75]: **E-step**: compute a distribution $\widetilde{p}^t$ over the range of $\mathcal{Z}$ such that $\widetilde{p}^t = p(\mathcal{T}|\mathcal{Z}, \Theta^{t-1})$; **M-step**: set Θ^t to maximize the expectation of the complete log–likelihood $E_{\widetilde{p}^t}[\log p(\mathcal{T}, \mathcal{Z}|\Theta)]$

Note that the expectation of $\log \mathcal{L}$ over the given distribution p can be obtained from Eq. 5.3, through simple manipulations as [11]:

$$E_p[\log p(\mathcal{T}, \mathcal{Z}|\Theta)] = \sum_{i=1}^{N}\sum_{l=1}^{L} h_{il} \log(\alpha_l) + \sum_{i=1}^{N}\sum_{l=1}^{L} h_{il} \log(p_l(\mathcal{T}^i|\theta_l)) \tag{5.4}$$

where $h_{il} = p(l|\mathcal{T}^i, \Theta)$ denotes the posterior distribution of the hidden variables given the set of parameters Θ and the observed $\mathcal{T}^i$.

Under this setting, the standard EM algorithm for Gaussian mixtures with parameters $\theta_l = \{\mathbf{m}_l, \mathbf{\Sigma}_l\}$ can be computed in close form. The E-step computes the distribution of hidden variables as:

$$h_{il}^t = \frac{\alpha_l^t p(\mathcal{T}^i|l, \mathbf{m}_l^t, \mathbf{\Sigma}_l^t)}{\sum_l \alpha_l^t p(\mathcal{T}^i|l, \mathbf{m}_l^t, \mathbf{\Sigma}_l^t)}, \tag{5.5}$$

while the M-step, given the distribution of the hidden variables, obtains parameters θ_l that maximize the expectation of $\log \mathcal{L}$ as:

$$\alpha_l^{t+1} = \frac{1}{N}\sum_i h_{il}^t, \mathbf{m}_l^{t+1} = \frac{\sum_i h_{il}^t \mathcal{T}^i}{\sum_i h_{il}^t}, \mathbf{\Sigma}_l^{t+1} = \frac{\sum_i h_{il}^t [\mathcal{T}^i - \mathbf{m}_l^{t+1}][\mathcal{T}^i - \mathbf{m}_l^{t+1}]^T}{\sum_i h_{il}^t}. \quad (5.6)$$

Both steps are iterated until convergence criteria are met. It can be shown that the `incomplete` data log-likelihood $\log p(\mathcal{T}|\Theta)$ is non-decreasing at each iteration of the update [27]; thus, a suitable convergence criterion is $\Delta_{log} = |\log \mathcal{L}^{(t+1)} - \log \mathcal{L}^{(t)}| < \epsilon$, where ϵ is a threshold experimentally determined.

Once the learning step is completed and the parameters Θ of the Gaussian mixture model have been obtained, the images $I_i$of a given category $\mathcal{C}_n$ can be partitioned in clusters $\mathcal{C}_n = \{\mathcal{C}_n^1, \mathcal{C}_n^2, \ldots, \mathcal{C}_n^{L^n}\}$, where each image I_i, represented through $\mathcal{T}^i$, is assigned to the cluster $\mathcal{C}_n^l$ with probability $p(l|\mathcal{T}^i, \Theta)$

5.3.2 Balanced EM learning

The cluster representation previously discussed, beyond its generality, has some drawbacks when exploited for a very large database. On the one hand the labeling of the image bears a computational cost which is linear in time with the number of clusters L^n in the category. On the other hand, for retrieval purposes, such solution is not efficient with respect to indexing issues, since the clusters obtained are in general unbalanced. To overcome such drawbacks, we introduce a variant of the EM algorithm which provides a balanced clustering of the observed data, so that clusters can be organized in a suitable data structure, namely a balanced tree.

The idea is to constrain, along the E-step, the distribution of the hidden variables so as to provide a balanced partition of the data, and then perform a regular M-step. This is equivalent to provide a mapping $p(\mathcal{Z}|\mathcal{T},\Theta) \rightarrow q(\mathcal{Z}|\mathcal{T},\Theta)$ so that $\log p(\mathcal{T}|\Theta)$ is non-decreasing at each iteration of the update. To make this clear define, following Neal and Hinton [75], the free energy:

$$F(\widetilde{p},\Theta) = E_{\widetilde{p}}\left[\log p(\mathcal{T},\mathcal{Z}|\Theta)\right] + H(\widetilde{p}), \tag{5.7}$$

where $H(p) = E_p\left[\log p(\mathcal{Z}|\mathcal{T},\Theta)\right] = \sum_{i=1}^{N}\sum_{l=1}^{L} h_{il}\log(h_{il})$ is the entropy of the hidden variables. It has been shown [75] that this function is maximized by the E and M steps.

In particular when the distribution of the hidden variable is computed according to the standard E-step as in Eq. 5.5, then $\widetilde{p} = p$ gives the optimal value of the function, which is exactly the `incomplete` data log-likelihood $F(p,\Theta) = \log p(\mathcal{T}|\Theta)$. For any other distribution $\widetilde{p} = q \neq p$ over the hidden variables, $F(q,\Theta) \leq F(p,\Theta) = \log p(\mathcal{T}|\Theta)$

Basically, what we need is to design a distribution q so that information paths $\mathcal{T}^i$ are assigned to clusters where each hidden variable has a distribution with probability 1 for one of the mixture component and zero for all the others [8]. Denote $\mathcal{Q}$ this class of distributions.

Remark that for $q \in \mathcal{Q}$, a partition of $\mathcal{T}^1,\cdots,\mathcal{T}^N$ is defined where for each $\mathcal{T}^i$, there exists $l(1 \leq l \leq L)$ such that $q(l|\mathcal{T}^i,\Theta) = 1$, thus $q(l|\mathcal{T}^i,\Theta)\log q(l|\mathcal{T}^i,\Theta) = 0$ for all $1 \leq l \leq L$ and $1 \leq i \leq N$ (since $0\log 0 = 0$, [23]). Hence $H(q) = 0$ and we have:

$$F(q,\Theta) = E_q\left[\log p(\mathcal{T},\mathcal{Z}|\Theta)\right] \leq F(p,\Theta) = \log p(\mathcal{T}|\Theta) \tag{5.8}$$

which shows that the expectation over q lower bounds the likelihood of the data. Further, it has been shown [8] that for some choices of q (e.g., $q = 1$, if $l = \arg\max_{l'} p(l|\mathcal{T}^i,\Theta)$ and $q = 0$ otherwise) is a tight lower bound, $E_p\left[\log p(\mathcal{T},\mathcal{Z}|\Theta)\right] \leq E_q\left[\log p(\mathcal{T},\mathcal{Z}|\Theta)\right]$.

Thus, due to Eq. 5.8, we can set up an E-step in which the free energy is maximized by maximizing $E_{\widetilde{p}}[\log p(\mathcal{T}, \mathcal{Z}|\Theta)]$ given Θ where, taking into account Eq. 5.4, and discarding the penalty term $\sum_{i=1}^{N}\sum_{l=1}^{L} h_{il}\log(\alpha_l)$,

$$E_{\widetilde{p}}[\log p(\mathcal{T}, \mathcal{Z}|\Theta)] = \sum_{i=1}^{N}\sum_{l=1}^{L} h_{il}\log(p_l(\mathcal{T}^i|\theta_l)) \tag{5.9}$$

Following [118], balanced partitioning can be achieved by solving the optimization problem:

$$\max E_{\widetilde{p}} = \max_{h} \sum_{i=1}^{N}\sum_{l=1}^{L} h_{il}\log(p_l(\mathcal{T}^i|\theta_l)), \tag{5.10}$$

subject to $\sum_{l=1}^{L} h_{il} = 1, \forall i$, $\sum_{i=1}^{N} h_{il} = \frac{N}{L}, \forall l$, and $h_{il} \in \{0,1\}, \forall i, l$.

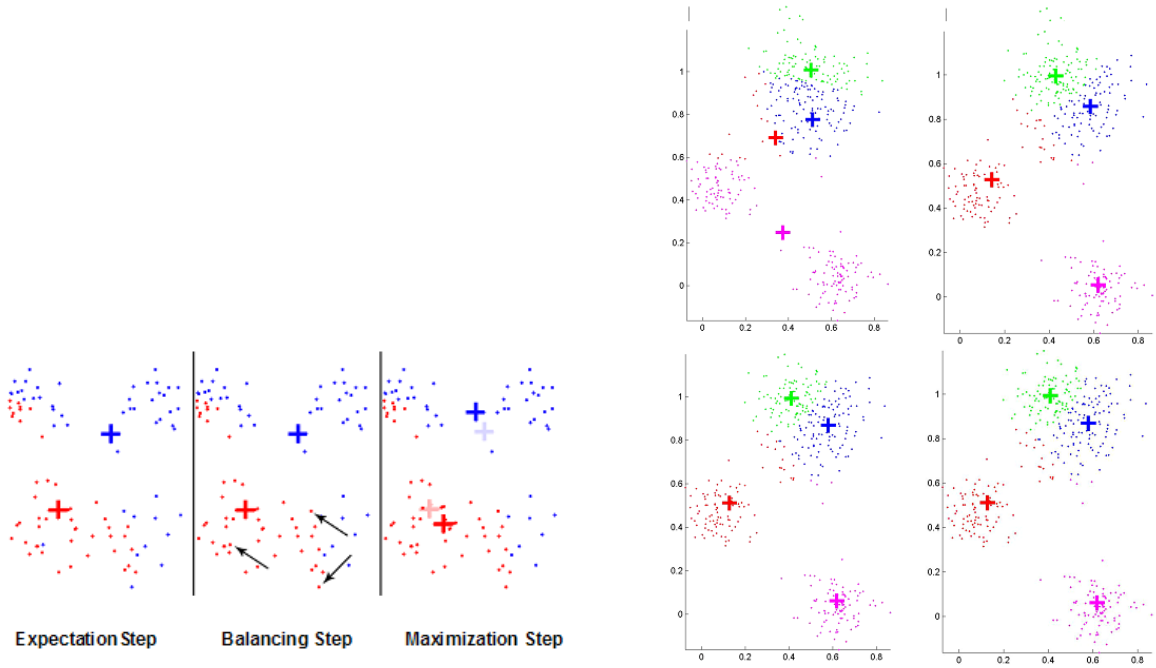
Algorithm 2 Balanced EM

Initialize all $\alpha_l, \theta_l, l = 1, \cdots, L$
$t \leftarrow 1$
repeat
  {E-step}
  for $(i = 1, ..., N)$ **do**
    for $(l = 1, ..., L)$ **do**
      $h_{il}^t \leftarrow \frac{\alpha_l^t p(\mathcal{T}^i|l, \mathbf{m}_l^t, \mathbf{\Sigma}_l^t)}{\sum_l \alpha_l^t p(\mathcal{T}^i|l, \mathbf{m}_l^t, \mathbf{\Sigma}_l^t)}$
      $q_{il}^t \leftarrow 1$ if h_{il}^t is in the N/L highest values for class l, $q_{il}^t \leftarrow 0$ otherwise
  {M-step}
  for $(l = 1, ..., L)$ **do**
    $\alpha_l^{t+1} \leftarrow \frac{1}{N}\sum_i q_{il}^t$
    $\mathbf{m}_l^{t+1} \leftarrow \frac{\sum_i q_{il}^t \mathcal{T}^i}{\sum_i q_{il}^t}$
    $\mathbf{\Sigma}_l^{t+1} \leftarrow \frac{\sum_i q_{il}^t [\mathcal{T}^i - \mathbf{m}_l^{t+1}][\mathcal{T}^i - \mathbf{m}_l^{t+1}]^T}{\sum_i q_{il}^t}$
  Compute $\log \mathcal{L}^{(t+1)}$
  $t \leftarrow t + 1$
until $|\log \mathcal{L}^{(t+1)} - \log \mathcal{L}^{(t)}| < \epsilon$

Since the solution of such optimization problem is an integer programming problem, which is NP-hard in general, in [118] a greedy heuristics has been suggested which gives a locally optimal solution to such problem. The procedure assign N/L data samples to one of the L clusters at each iteration, by selecting the first N/L samples with higher h_{il} probability with respect to the cluster. For instance, for $L = 2$, this gives a $\{N/2, N/2\}$ bipartition that maximizes $E_{\widetilde{p}}$. Eventually, the given partition provides the distribution $q \in \mathcal{Q}$. Denote $q_{il} = q(l|\mathcal{T}^i, \theta_l^t)$.

The balanced EM algorithm (BEM) is detailed in Algorithm 2. Note that the algorithm introduces a sort of classification within the E-step in the same vein of the CEM algorithm[18].

In Fig. 5.3 an application of expectation, balancing and maximization steps and an example of BEM algorithm evolution are shown.

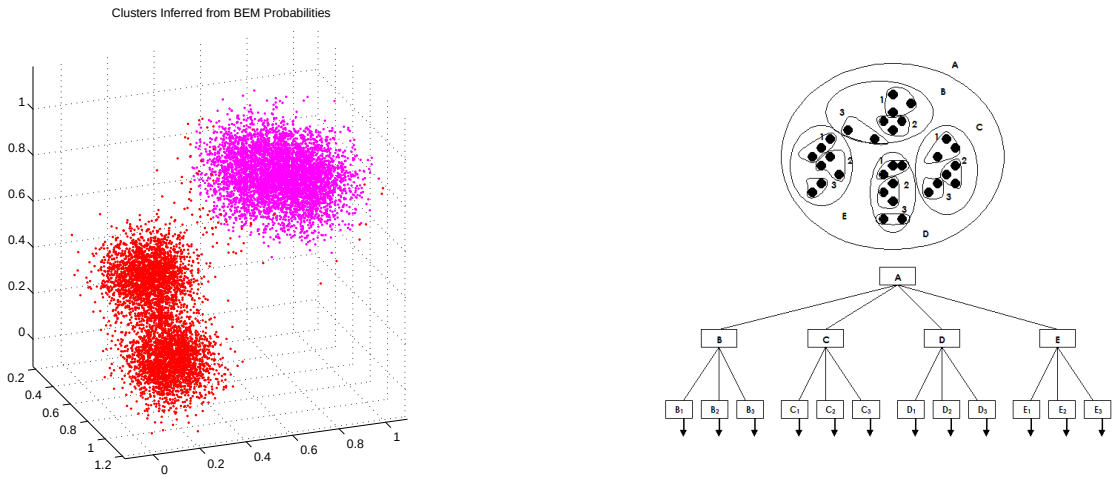


a. BEM Steps b. BEM evolution

Figure 5.3: BEM behavior

5.3.3 Balanced Cluster Tree representation

At this point, each category can be represented in terms of clusters by mapping the cluster space onto the tree-structure shown in Fig. 5.4, which we denote Balanced Cluster Tree (BCT).



a. Balanced Clusters obtained by BEM in a 3-dimensional space b. A 2-D example of BCT

Figure 5.4: Generating BCT

Given a category $\mathcal{C}_n$ a BCT of depth Υ is obtained by recursively applying the balanced EM algorithm at each level $\upsilon = 0, \cdots, \Upsilon - 1$ of the tree. Each tree node of level υ is associated with one of the discovered clusters at the $(\upsilon + 1)$-th iteration of the BEM algorithm. New discovered clusters are recursively partitioned until each category cluster contains a number of VMTs lower than a fixed threshold c, representing the desired filling-coefficient (capacity) of tree leaves. This induces a coarse-to-fine representation, namely $\mathcal{C}_n(\upsilon) = \{\mathcal{C}_n^1(\upsilon), \mathcal{C}_n^2(\upsilon), \ldots, \mathcal{C}_n^{L^n}(\upsilon)\}_{\upsilon=0,\cdots,\Upsilon-1}$. The category sub-tree level can be calculated as $lev_n = log_{L^n}(\frac{N_n}{c})$, N_n being the number of category indexing objects, and L^n the number of clusters generated at n-th BEM iteration.

In particular, as shown in Fig. 5.4, the root node is associated with the whole category $\mathcal{C}_n$, and the tree maintains a certain number of entry points for each node dependent on the number L^n of wanted clusters for each tree-level; we represent the non-leaves node $\{\mathcal{C}_n^1(\upsilon), \mathcal{C}_n^2(\upsilon), \ldots, \mathcal{C}_n^{L^n}(\upsilon)\}_{\upsilon=0,\cdots,\Upsilon-1}$, at level υ by using the parameters $\mathbf{m}_n^l(\upsilon)$, and, the cluster radius $|\mathbf{\Sigma}_n^l(\upsilon)|$, whereas leaves contain the image pointers. In Fig. 5.5 an UML representative diagram of BCT nodes is reported.

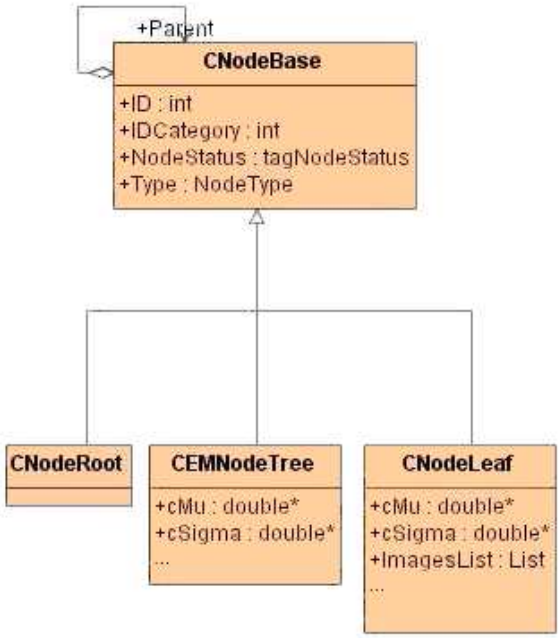


Figure 5.5: BCT Nodes: a representative diagram

Formally, we can define the tree-nodes ("pivots","routing nodes") and the leaves of our structure nodes as $\rho = \langle \mathbf{m}, |\mathbf{\Sigma}|, Ptr \rangle$ and $\iota = \langle \Gamma \rangle$, respectively, where $(\mathbf{m}, |\mathbf{\Sigma}|)$ are the features representative of the current routing node, Ptr is the pointer to the parent tree-node and Γ is the set of pointer to the objects on the secondary storage system. In this manner, the procedure to build our tree can be outlined by algorithm 3.

Algorithm 3 Building BCT

Given the current level υ and the pointer Ptr to the parent node
$\Upsilon = \left[log_{|\mathcal{C}(\upsilon-1)|}(\frac{N_n}{c})\right]$
if $\upsilon \leq \Upsilon - 1$ **then**
  for $(i = 1, ..., |\mathcal{C}_n(\upsilon)|$ **do**
    $\mathbf{m}_n^i(\upsilon), |\mathbf{\Sigma}_n^i(\upsilon)| \leftarrow BEM_{Algorithm}$
    $\rho_n^i(\upsilon) \leftarrow \{\mathbf{m}_n^i(\upsilon), |\mathbf{\Sigma}_n^i(\upsilon)|, Ptr\}$
    Building Cluster Tree $(\upsilon + 1, Ptr(\rho_n^i(\upsilon)))$
else
  $\iota_n^i(\upsilon) \leftarrow \Gamma$

At this point to perform the category assignment process, we can obtain the probability, at level υ, that a test image I_t belongs to a category $\mathcal{C}_n$ as $P(\mathcal{C}_n(\upsilon)|\mathcal{T}^t) \simeq P(\mathcal{T}^t|\mathcal{C}_n(\upsilon))P(\mathcal{C}_n(\upsilon))$, which, due to independency of clusters guaranteed by the EM algorithm, can be reformulated as:

$$P(\mathcal{C}_n(\upsilon)|\mathcal{T}^t) \simeq P(\mathcal{C}_n(\upsilon)) \prod_{l \in L_p} p(\mathcal{T}^t|\mathcal{C}_n^l(\upsilon)). \tag{5.11}$$

The category discovery process can be carried out by comparing the image map VMT with the category clusters in the WW space at a coarse scale ($\upsilon = 1$) and by choosing the best categories on the base of belonging probabilities of the image to the database categories obtained by Eq. 5.11. Each image I_t, is associated to probabilities of being within given categories as $\langle I_t = P(\mathcal{C}_1|\mathcal{T}^t), \cdots, P(\mathcal{C}_N|\mathcal{T}^t)\rangle$. On the other hand, given the category $\mathcal{C}_n$ to which the image belongs, the search of the images can be performed by exploiting the BCT structure.

Eventually, to evaluate the clustering goodness, in Fig. 5.6 we propose a comparison between a balanced and unbalanced tree-solution obtained by BEM and EM application, respectively.

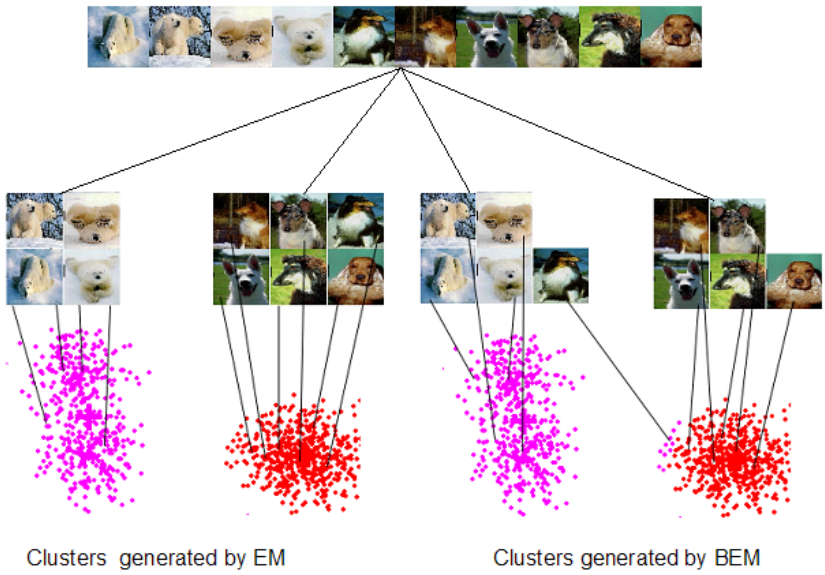


Figure 5.6: Goodness of clustering with BEM: a comparison with EM

5.4 The Animate query process

Given a query image I_q and the dimension of the desired results set, the T_k most similar images are retrieved in the following steps:

1. Map the image in the WW space by computing the image path under free viewing conditions, $I_q \mapsto \mathcal{T}^q$.

2. Discover the best $K < N$ categories that may describe the image by using Eq. 5.11, but substituting I_q for I_t.

3. For each category $\mathcal{C}_n$ among the best K discovered, by traversing the BCT associated to $\mathcal{C}_n$, retrieve the N_I target images I_t within the category at minimum distance from the query image.

4. Refine results by choosing the T_K images most similar to the query image by performing a sequential scanning of the previous set of KN_I images and evaluating the similarity $\mathcal{M}(\mathcal{T}^t, \mathcal{T}^q)$ between their VMTs.

Fig. 5.7 summarizes the animate query process.

Thus, in order to perform step 3 we need to efficiently browse the BCT while step 4, requires the specification of the similarity function $\mathcal{M} \in R^+$ used to refine the results of query process. The first issue is addressed in the following, while the second has already discussed in chapter 4.

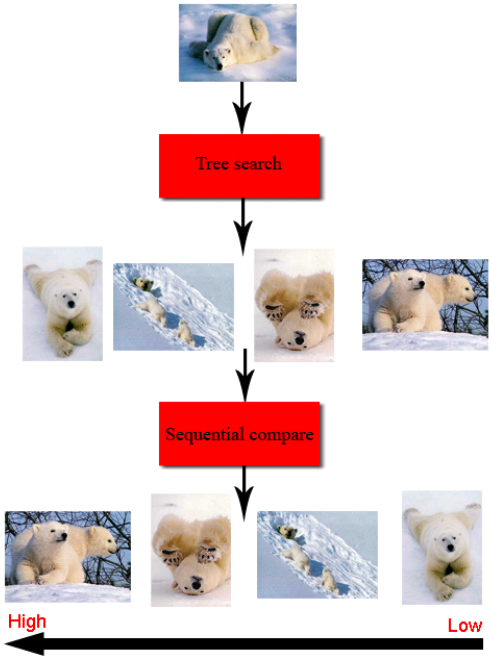


Figure 5.7: Animate Query Process

5.4.1 Category browsing using the BCT

When a query image I_q is proposed, the BCT representing category $\mathcal{C}_n$ can be traversed for retrieving the N_I target images I_t, by evaluating the similarity between $\mathcal{T}^q$ and clusters $\mathcal{C}_n^l(\upsilon)$ at the different levels υ of the tree. Recall that each cluster $\mathcal{C}_n^l(\upsilon)$ is represented through its mean and covariance, respectively $\mathbf{m}_n^l(\upsilon)$, $\mathbf{\Sigma}_n^l(\upsilon)$. To this end, it is possible to define the distance $d(\mathcal{T}^q, \mathcal{C}_n^l(\upsilon))$ as the distance between $\mathcal{T}^q$ and the cluster center $\mathbf{m}_n^l(\upsilon)$ weighted by covariance $\mathbf{\Sigma}_n^l(\upsilon)$ [95]:

$$d(\mathcal{T}^q, \mathcal{C}_n^l(\upsilon)) = e^{-(\mathcal{T}^q - \mathbf{m}_n^l(\upsilon))^T \mathbf{\Sigma}_n^l(\upsilon)^{-1} (\mathcal{T}^q - \mathbf{m}_n^l(\upsilon))}. \tag{5.12}$$

It is easy to verify that such distance indeed is real-valued, finite and nonnegative and satisfies symmetry and triangle inequality properties, so that d is a metric on the information path space $\mathcal{T}$ and the pair $(\mathcal{T}, d)$ is a metric space. In other terms the BCT is a metric balanced tree and, as such, is suitable to support operations of classic multidimensional access methods [19].

Recall that a viable search technique is the range query [19], which returns the objects of our distribution that have a distance lower than a fixed range query radius $r(\mathcal{I}P^q)$ with respect to the query object $\mathcal{T}^q$. In such approach the tree-search is based on a simple concept: the node related to the region having as center $\mathbf{m}_n^l(\upsilon)$ is visited only if $d(\mathbf{m}_n^l(\upsilon), \mathcal{T}^q) \leq r(\mathcal{T}^q) + r(\mathbf{m}_n^l(\upsilon))$, where $r(\mathbf{m}_n^l(\upsilon))$ is the radius of the analyzed region.

The range query algorithm starts form the root node and recursively traverses all paths which cannot be excluded from leading to objects because satisfying the above inequality. The $r(\mathcal{T}^q)$ value is usually evaluated in an experimental way [19].In our case, due to cluster independency guaranteed by the EM procedure, the classical problem of range query strategy [19] is avoided. In Fig. 5.8 an example of a range query is shown.

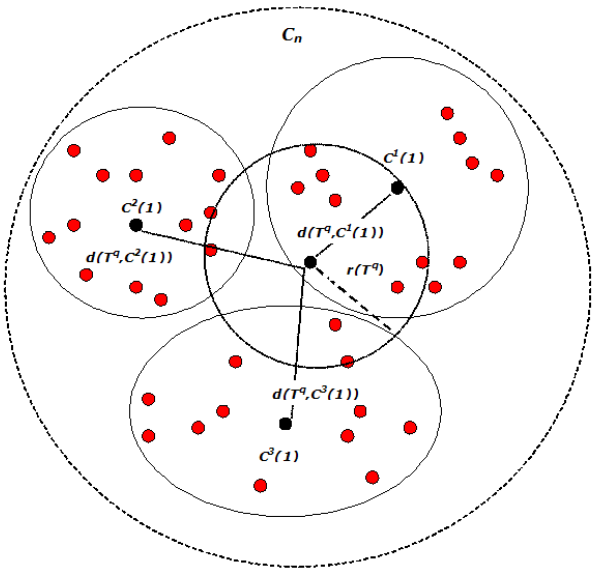


Figure 5.8: Range Query inside a given category C_n

For a given tree level $\upsilon >= 1$, clearly, it is not convenient to have a fixed value of $r(\mathcal{T}^q)$, which rather should depend on the distribution of cluster centers surrounding the query object, at a certain level of the BCT (cfr. Fig. 5.8). Thus, for each level, we consider the maximum and the minimum distances between the query object and each cluster center, $d^q_{min}(\upsilon)$ and $d^q_{max}(\upsilon)$, respectively , where distance is computed via Eq. 5.12.

Denote for simplicity, $\mathbf{m}^l = \mathbf{m}^l_n(\upsilon)$ the center of the l-th cluster of category n, $l = 1, \cdots, L^n$, surrounding the query point, and d^l the distance between the latter and cluster l. By increasing the radius through discrete steps, $j = 1, 2, \cdots$, within the interval $[d^q_{min}(\upsilon), d^q_{max}(\upsilon)]$ and counting the number of clusters occurring within the area spanned by the radius, $a_j = \{\#\mathbf{m}^l | d^l \leq r_j\}$, a step-wise function $s = \{\overline{a}_1, \overline{a}_2, \ldots, \overline{a}_k\}$ is obtained, where normalization $\overline{a}_j = \frac{a_j}{\max_j a_j}$ constrains s to take values within the interval $[0, 1]$. Each s value is thus related to the number of BCT nodes we want to explore for a given query object. In other terms, given a query object $\mathcal{T}^q$, by choosing a value s_q, which specifies the span of the search, we can automatically decide, at each level of the BCT, the range query radius at that level by using the inverse mapping $s \mapsto r$; for instance, by setting $s_q = 1$ exploration is performed on all cluster nodes available at that level. We have experimentally verified that such mapping is well approximated by a sigmoid function, namely $\frac{1}{1+\exp(-\rho \cdot (s_q - .5))}$, where $\rho = 0.2$ provides the best fit.

A possible procedure to exploit range query is reported by algorithm 4.

Algorithm 4 Range Query on Cluster $\mathcal{C}(\upsilon)$

```
Given s_q, T^q and υ
Compute d^q_max(υ) and d^q_min(υ)
φ = 1/(1+exp(−ρ·(s_q−.5)))
for (i = 1, ..., |C_n(υ)| do
  if υ = ϒ − 1 then
    Save Object Pointers Γ
    break
  else if d(T^q, C^l_n(υ)) < φ then
    Range Query(C^l_n(υ + 1))
```

Eventually, it is worth remarking that, for what concerns the tree updating procedures, a naive strategy would simply re-apply the classification step of BEM algorithm. However, a more elegant and efficient solution is to exploit the category detection step to assign the new item to category $\mathcal{C}_n$ and then exploit an on-line, incremental version of the BEM algorithm to update the related tree; the incremental procedure updates the sufficient statistics of the expected log-likelihood only as a function of the new data item inserted in the database, which can be done in constant time [75], [108].

Summing up, in the approach proposed here, a user specifies a query image I_q, from which the image path under free viewing conditions, $I_q \mapsto \mathcal{T}^q$ (query object) is extracted, the related $r(\mathcal{T}^q)$ and the dimension T_k of the results set (step 1). By means of the range query technique, the BCT is traversed to discover all related categories and the images similar to the query image are retrieved (step 2 and 3). Eventually (step 4), a sequential animate matching is performed on the results set to select the T_k most similar images.

5.5 Experimental results

5.5.1 Methodological foreword

Retrieval effectiveness is usually measured in the literature through recall and precision measures [28]. For a given number of retrieved images (the result set rs), the recall $R = |rl \cap rs|/|rl|$ assesses the ratio between the number of relevant images within rs and the total number of relevant images rl in the collection, while the precision $P = |rl \cap rs|/|rs|$ provides the ratio between the number of relevant images retrieved and the number of retrieved images.

Unfortunately, on the one hand, from a bare practical standpoint, when dealing with large databases it is difficult to estimate even approximately [105] the recall, and, in particular, the number of relevant results that have to be retrieved. On the other hand and most important, the concept of "relevant result" is often ill-defined or, at least problematic (see [21] and [91] for an in-depth discussion).

More generally, it is not easy to evaluate a system that takes into account properties like perceptual behaviors and categorization, since this necessarily involves comparison with human performance. This entails in our case the evaluation of the matching relying upon attention consistency and categorization capabilities along the query step.

To this end, for what concerns the animate matching step, we consider the following issues: 1) Robustness of the matching method, with respect to physical variations of the image; 2) Consistency of image similarity proposed by the matching with respect to human judgement of similarity. The first issue does not involve comparison with human performance, while the second experiment entails that the human subject is used as a measuring instrument [91].

Categorization effectiveness has been tested in terms of: 1) Performance with respect to recall and precision figures of merit; 2) Performance with respect to human categorization. The former experiment was performed, similarly to [105], on a subset of the COREL database, where precision and recall can be directly evaluated. The latter relies upon a weighted precision metric accounting for human performance. Eventually, query performance in terms of retrieval efficiency is discussed.

5.5.2 Experimental setting

Our data set consists of about 50000 images collected from the Internet, experimental databases and several commercial archives. In particular a subset of 1000 pictures has been obtained from the COREL archive and used only for the evaluation of categorization performance in terms of precision. Images are coded in the JPEG format at different resolution and size, and stored, together with the related VMTs, into a commercial object relational DBMS.

The images have been grouped into 300 categories. In order to associate the set of images to each proposed category, twenty `naive` observers were asked to perform the task on the data set, and eventually the classification has been accomplished by grouping into a category those images that the majority of observers judged to belong to such category.

The VMT as provided `tout court` by the "What" and "Where" streams gives rise to a high dimensional feature space spanning a 2-D subspace representing the set of FOA spatial coordinates, a 768-D (256 for component) space which represents the set of FOA HSV color histograms, a 1-D subspace which represents the set of FOA WTA fire-times and a 18-D subspace which represents the set of FOA covariance signatures of the wavelet transform.

To exploit the BEM algorithm, each image is represented more efficiently by performing the following reduction: the color histogram is obtained on the HSV components quantized by using $16, 8, 8$ levels for H S and V components, respectively; the covariance signatures of wavelet transform are represented through using 18 components.

Eventually the clustering space becomes a $53N_f$-D space, $N_f = 20$ being the number of FOAs in free viewing conditions.

The different BCTs related to each category have been joined by means of a root node that represents the whole space of images; thus, each node of the first tree level contains the images related to a given database category.

For what concerns the BCT building step, at each level $\upsilon > 1$ of the tree (we assume the root node related to level 0), a number $L = 3$ was used in the recursive application of BEM algorithm due to efficiency and effectiveness aims in the retrieval task. Moreover, for each category sub-tree the total number of level lev was chosen considering a leaf filling coefficient $c = 15$. Note that we assume L fixed, in that we are not concerned here with the problem of model selection, in which case L may be selected by Bayesian information criterion (BIC, [69]).

Note that at BCT level $\upsilon = 1$, a characterization (in terms of mean and covariance) of each category is not available, so for determining the distances between query object and clusters in the range query process, mean and covariance of the whole category IP distribution are considered.

For what concerns the BEM algorithm, non uniform initial estimates were chosen for $\alpha_k^{(0)}, \mu_l^{(0)}, \mathbf{\Sigma}_l^{(0)}$ parameters; $\{\mathbf{m}_l^{(0)}\}$ were set in the range from minimal to maximal values of $\mathcal{T}^i$ in a constant increment; $\{\mathbf{\Sigma}_l^{(0)}\}$ were set in the range from 1 to $\max\{\mathcal{T}^i\}$ in a constant increment; $\{\alpha_l^{(0)}\}$ were set from $\max\{\mathcal{T}^i\}$ to 1 in a constant decrement and then normalized, $\sum_l \alpha_l^{(0)} = 1$.

We found that convergence rate is similar for both methods, convergence being achieved after $t = 300$ iterations (with $\epsilon = 0.1$). Fig. 5.9 shows how the `incomplete` data log-likelihood $\log p(\mathcal{T}|\Theta)$ as obtained by the BEM algorithm is non-decreasing at each iteration of the update, and that convergence is faster than with classic EM.

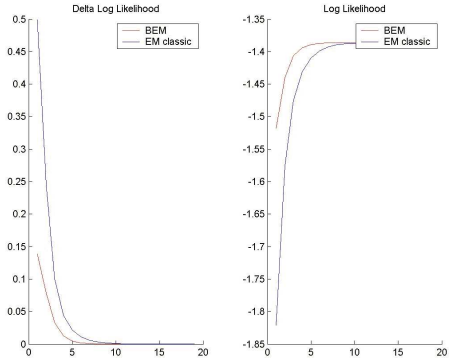


Figure 5.9: Behavior of Δ_{log} (left) and of $\log p(\mathcal{T}|\Theta)$ vs. number of iterations of the BEM algorithm compared with standard EM

As regards the animate matching step (see chapter 4), the value of $N'_f = 10$ was chosen either because, in this way, each FOA is only visited once, and for the importance of earliest FOAs. The local temporal window used in the image matching algorithm was set to the fixed size 4, as an experimental trade-off between retrieval accuracy and computational cost. For what concerns the setting of equation parameters, considering again Eq. 4.3, we simply use $\alpha_a = \beta_a = \gamma_a = 1/3$, granting equal informational value to the three kinds of consistencies; similarly, we set $\mu_1 = \mu_2 = 1/2$ in Eq. 4.10.

5.5.3 Matching robustness

In this experiment we evaluate the robustness of the matching algorithm used in the query refining stage, with respect to image alterations: brightness and contrast variation, noise corruption, rotation and translation operations. An example is provided in Fig. 5.10

.

Figure 5.10: An example of Information Path changing due to image alterations: (1,1) Original Image; (1,2) Brighten 10%; (1,3) Darken 10%; (2,1) More Contrast 10%; (2,2) Less Contrast 10%; (2,3) Noise Adding 5%; (3,1) Horizontal Shifting 15%; (3,2) Rotate 90; (3,3) Flip 180

In Fig. 5.11, the average attention consistency distance between 50 target images, randomly chosen from different categories and the same images after alterations, is plotted for increasing brightness and contrast variations and noise (uniform noise was considered).

Corresponding curves, for rotations and translations are not reported, since we have experimentally observed that, on the one hand, for rotation operations, the average attention consistency does not suffer of significant alterations (lower than 0.3) for variations inside the intervals: $[-15^o, 15^o]$,$[90^o-15^o, 90^o+15^o]$, $[180^o-15^o, 180^o+15^o]$ and $[270^o-15^o, 270^o+15^o]$;

on the other hand, the consistency measure is definitely robust to horizontal translations.

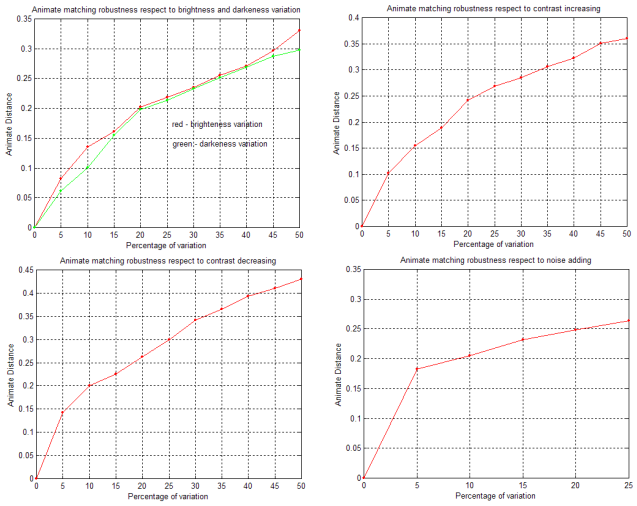


Figure 5.11: Robustness of the animate matching algorithm with respect to image alterations

Provided that the matching procedure results to be quite robust to imaging conditions, what is important to notice here is that, clearly, like it happens for human observers, some transformations should be invariant only to some extent. For instance consider the simple image of an horse, depicted at the center of an image (e.g., Fig. 5.10). If the object (horse) translates, the semantic content of the image should be comparable, and actually the Information path provides a similar "shape". A different effect should play scale variations: if the same horse is reduced to a small patch at the bottom right of the picture, is the image still an horse image? It is likely that if a large region of grass is represented, it would be better classified as a landscape. This is easy to verify, by performing eye-tracking experiments with human observers. Thus scale invariance in many case could be a wrong issue to address. The same holds for occlusions: assume that an elephant is half-occluding the horse. In this case the IP and related matching will dramatically change (as well as for human observers) providing a different classification.

5.5.4 Matching effectiveness

This set of experiments aims at comparing the ranking provided by our system using the proposed similarity measure (attention consistency $\mathcal{M}$) with the ranking provided by a human observer. To such end we have slightly modified a test proposed by Santini [91]. in order to obtain a quantitative measure of the difference between the two performed rankings ("treatments", [91]) in terms of hypothesis verification. Consider a weighted displacement measure defined as follows [91]. Let q be a query on a database of N images that produces n results. There is one ordering (usually given by one or more human subjects) which is considered as the ground truth, represented as $L_t = \{I_1, \ldots, I_n\}$. Every image in the ordering has also associated a measure of relevance $0 \leq S(I, q) \leq 1$ such that (for the ground truth), $S(I_i, q) \geq S(I_{i+1}, q)$, $\forall i$. This is compared with an (experimental) ordering $L_d = \{I_{\pi_1}, \ldots, I_{\pi_1}\}$, where $\{\pi_1, \ldots, \pi_n\}$ is a permutation of $1, \ldots, n$. The displacement of I_i is defined as $d_q(I_i) = |i - \pi_i|$. The relative weighted displacement of L_d is defined as:

$$W_q = \frac{\sum_i S(I_i, 1) d_q(I_i)}{\Omega} \tag{5.13}$$

where $\Omega = \lfloor \frac{n^2}{2} \rfloor$ is a normalization factor. Relevance S is obtained from the subjects asking them to divide the results in three groups: `very similar` ($S(I_i, q) = 1$), `quite similar` ($S(I_i, q) = 0.5$) and `dissimilar` ($S(I_i, q) = 0.05$).

In our experiments, on the basis of the ground truth provided by human subjects, treatments provided either by humans or by our system are compared. The goal is to determine whether the observed differences can indeed be ascribed to the different treatments or are caused by random variations. In terms of hypothesis verification, if μ_i is the average score obtained with the i^{th} treatment, a test is performed in order to accept or reject the null hypothesis H_0 that all the averages μ_i are the same (i.e., the differences are due only to random variations); clearly the alternate hypothesis H_1 is that the means are not equal, that is the experiment actually revealed a difference among treatments.

The acceptance of H_0 hypothesis can be checked with the F ratio. Assume that there are m treatments and n measurements (experiments) for each treatment. Let w_{ij} be the result of the j^{th} experiment performed with the i^{th} treatment in place. Define : $\mu_i = \frac{1}{n}\sum_{j=1}^{n} w_{ij}$ the average for treatment i, $\mu = \frac{1}{m}\sum_{i=1}^{m}\mu_i = \frac{1}{nm}\sum_{i=1}^{m}\sum_{j=1}^{n} w_{ij}$ the total average, $\sigma_A^2 = \frac{n}{m-1}\sum_{i=1} m(\mu_i - \mu)^2$ the between treatments variance, $\sigma_W^2 = \frac{1}{m(n-1)}\sum_{i=1} m\sum_{j=1} n(w_{ij} - \mu_i)^2$ the within treatments variance. Then, the F ratio is:

$$F = \frac{\sigma_A^2}{\sigma_W^2}. \tag{5.14}$$

A high value of F means that the between treatments variance is preponderant with respect to the within treatment variance, that is, that the differences in the averages are likely to be due to the treatments. In our case we have used 8 subjects selected among undergraduate student. Six students randomly chosen among the 8 were employed to determine the ground truth ranking and the other two served to provide the treatments to be compared with that of our system. Four query images have been used, and for each of them a query was performed in order to provide a result set of 12 images, for a total of 48 images. Each result set was then randomly ordered and the two students were asked to rank images in the result set with respect to their similarity to the query image. Each subject was also asked to divide the ranked images in three groups: the first group consisted of images judged `very similar` to the query, the second group consisted of images judged `quite similar` to the query, and the third of `dissimilar` to the query. The mean and variance of the weighted displacement of the two subjects and of our system with respect to the ground truth are reported in Table 5.1.

	Human 1	Human 2	IP Matching
μ_i	0.0209	0.0203	0.0190
σ_i^2	$7.7771e^{-4}$	$8.1628e^{-4}$	$8.5806e^{-4}$

Table 5.1: Average (μ_i) and variance (σ_i^2) of the weighted displacement for the three treatments (two human subjects and system

Then, the F ratio for each pair of distances,in order to establish which differences were significant, was computed according to Eq. 5.14.

As can be noted from Table 5.2 the F ratio is always less then 1 and since the critical value F_0, regardless of the confidence degree (the probability of rejecting the right hypotesis), is greater then 1, the null hypothesis can be statistically accepted. It is worth noting that the two rankings provided by the observers are consistent with one another and the attention consistency ranking is consistent with both.

F	Human 1	Human 2	IP Matching
IP Matching	0.3021	0.7192	0
Human 2	0.0875	0	
Human 1	0		

Table 5.2: The F ratio measured for pairs of distances (human vs. human and human vs. system)

5.5.5 Query performance via recall and precision

In this experiment we evaluate recall and precision parameters, following the systematic evaluation of image categorization performance provided by Wang et al. [105]. A subset composed of 10 images categories, each containing 100 pictures has been chosen from the COREL database and described in Table 5.3. In particular such testing database has been downloaded from http://www-db.stanford.edu/IMAGE/ web site (the images are stored in JPEG format with size 384 x 256 or 256 x 384). The 10 categories reflect different semantic topics.

Table 5.3: The COREL subdatabase used for query evaluation

ID	Category Name	Number of Images
1	Africa people and villages	100
2	Beach	100
3	Building	100
4	Buses	100
5	Dinosaurs	100
6	Elephants	100
7	Flowers	100
8	Horses	100
9	Mountains and glaciers	100
10	Food	100

Within such data set a retrieved image can be considered a match respect to the query image if and only if it is in the same category as the query. In this way it easy to estimate precision parameter within the first 100 retrieved images for each query, and, moreover in these conditions recall is identical to precision. In particular, for recall and precision evaluation every image in the sub-database was tested as query image and the retrieval results obtained. In Fig. 5.12, the achieved performance is reported for each category in terms of precision and weighted precision ($\bar{p} = \frac{1}{100}\sum_{k=1}^{100}\frac{n_k}{k}$, where $k = 1...100$ and n_k is the number of matches in the first k retrieved images).

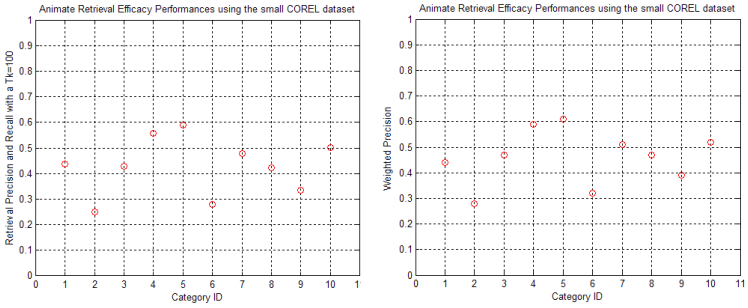
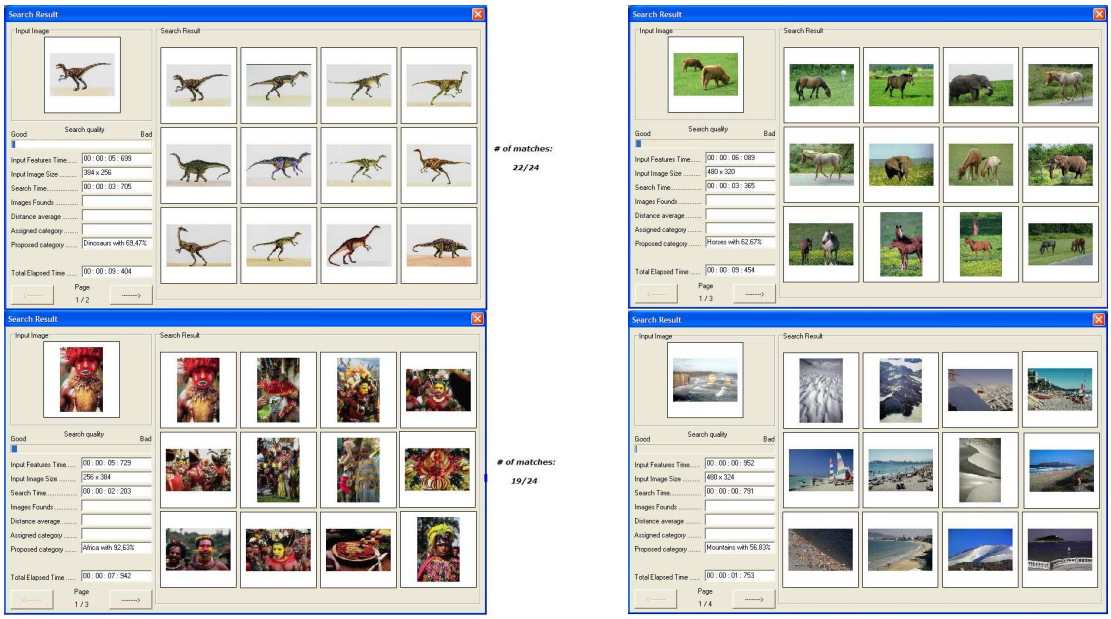


Figure 5.12: Precision of retrieval on the COREL subdatabase

For performing the previous experiment, a number of clusters equal to 3 for each tree level, a max tree level equal to 6, a leaf fan-out equal to 15 and a range query strategy using $s_q = 0.5$ have been set in the BEM tree building and traversing steps. The average precision and weighted precision obtained are respectively 0.438 and 0.460; the obtained results can also be compared with those obtained by SIMPLICITY and Color Histogram methods discussed in [105].

Fig. 5.13 a) shows the top 12 results related to 2 inside query cases with the number images belonging to the same query category among the first 24 proposed ones and, and Fig. 5.13 b), the top 12 results related to 2 outside query cases using a Top-K of size 100.

For the inside query, the category belonging score computed from maximum probability $P(\mathcal{C}_n|\mathcal{T}_t)$ resulted to be 69.47% corresponding to $\mathcal{C}_n$="Dinosaurs" for the top image and 92.63% corresponding to $\mathcal{C}_n$="Africa" for the bottom image. For queries performed with outside images the maximum category belonging score resulted to be 62.67% corresponding to $\mathcal{C}_n$="Horses" followed by 61.45% score corresponding to $\mathcal{C}_n$="Elephants" for the top image, and 56.83% corresponding to $\mathcal{C}_n$="Mountains" followed by a 56.33% score corresponding to $\mathcal{C}_n$="Beaches" for the bottom image. In the latter case, note that the top query presents image with cows and the system retrieves images from the data set by chosing "Horses" and "Elephants" categories which are most likely to represent, with respect to other categories, the semantics of the query.



a. Query Results for inside images. b. Query Results for outside images

Figure 5.13: Query results on the COREL subdatabase using either query images present within the data set (a) or outside the data set (b)

5.5.6 Query performance with respect to human categorization

The goal is the evaluation of the retrieval precision of the system, with respect to the possible categories that the user has in mind when a query is performed. This measure is evaluated with respect to the whole database (50000 images), and the following protocol has been adopted.

Given a test set of 4 outside images $I_q, q = 1...4$ (see Fig. 5.14), randomly selected out of 50 images, ten observers $u_j, j = 1...10$ (different from those that performed category identification), were asked to perform the task of choosing for each query image I_q, the three most representative categories, say $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ among those describing the database. To this end, images in all categories have been presented in a hierarchial way (e.g., animals: horses, cows, etc..), to speed-up the selection process.

Meanwhile, each user was asked to rank the three categories in terms of a representativeness score, within the interval $[0, 100]$, namely $R_1^{(u_j,q)}(\mathcal{C}_1|I_q), R_2^{(u_j,q)}(\mathcal{C}_2|I_q), R_3^{(u_j,q)}(\mathcal{C}_3|I_q)$; the three scores were constrained to sum to 100 (e.g., a user identifies categories $1, 2, 3$ for image 2 with scores $60, 30, 10$)

Figure 5.14: Query examples

For each image, the three most relevant categories have been chosen,according to a majority vote, by considering those that received the highest number of "hits" Nh_c, $c = 1, 2, 3$, from the observers, and each category was assigned the average score $R_c^q(\mathcal{C}_c|I_q) = \frac{1}{Nh_c}\sum_{j=1}^{Nh_c} R_c^{(u_j,q)}(\mathcal{C}_c|I_q)$. Results are reported in Table 5.4.

Table 5.4: Representativeness score $R_c^q(\mathcal{C}_c|I_q)$ for each query image of Fig.5.14

Image	**User Scores**
1	Sunset (40%), Beaches (35%), Coasts (25%)
2	Horses (45%), People (40%), Landscapes (15%)
3	Cows (0.60%), Landscapes (0.25%), Mountains (0.15%)
4	Buildings (55%), Mountains (30%), Landscapes (15%)

The scores $R_c^q(\mathcal{C}_c|I_q)$ are then normalized within the range $[0, 1]$ to allow comparison with category belonging probabilities computed by the system, and the perceptually weighted precision has been calculated:

$$P_w^q = \frac{1}{T_K}\sum_{k=1}^{T_K}\frac{wn_k^q}{k}, \tag{5.15}$$

where wn_k^q represents, for the query q, the weighted average match of the k retrieved image with respect to user score $R_c^q(\mathcal{C}_c|I_q)$ and belonging probability $P_c^k(\mathcal{C}_c|I_k)$ provided by the system:

$$wn_k^q = 1 - \frac{\sum_{c=1}^{3} w_c|R_c^q(\mathcal{C}_c|I^q) - P_c^k(\mathcal{C}_c|I_k)|}{\sum_{c=1}^{3} w_c} \tag{5.16}$$

Note that a perfect match is obtained only for $wn_k^q = 1$, that is for $|R_c^q(\mathcal{C}_c|I^q) - P_c^k(\mathcal{C}_c|I_k)| = 0, \forall c$. Relevance distance weights w_c have been chosen as the decreasing values $\{1, 0.5, 0.25\}$.

In this way the perceptually weighted precision on the whole data set of 50000, considering the first 100 retrieved images, as in the previous experiment, resulted to be 0.788.

Also, a query was performed for each image I_q, by considering a variable T_K of images. Fig. 5.15, for each query case, values P_w^q plotted at T_k variation. As shown in the figure, the three category belonging scores returned by system decrease to the T_K size variation, but it is possible to notice that the related proportions between system scores and user probabilities are preserved.

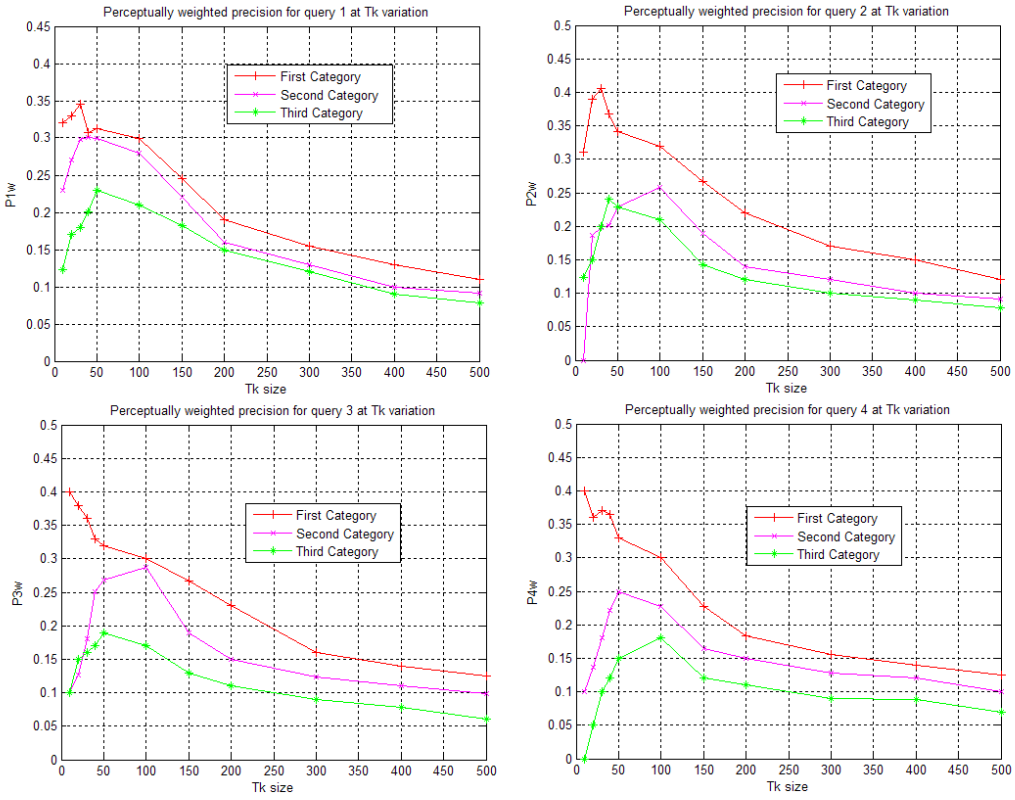


Figure 5.15: Perceptually weighted precision P_w^q plotted as a function of T_K, for queries $q = 1, 2, 3, 4$.

5.5.7 Retrieval efficiency

The retrieval efficiency can be evaluated in terms of time elapsed between query formulation and presentation of results. For AICQ the total search time t_Q is obtained from the tree search (traversing) time t_{tree} and the query refining time t_{qref} as

$$t_Q = t_{tree} + t_{qref}. \tag{5.17}$$

Due to the indexing structure adopted, the parameters that affect the total search time are the range query radius, obtained via the s_q value, the number of clusters L, which is fixed for each level of the BCT, the tree capacity c and the number of images within the i-th category N_i. Thus, by fixing L, c, N_i, the times t_{tree} and t_{qref} are expected to increase for increasing s_q within the interval $[0, 1]$. The upper bounds on such quantities can be estimated as follows.

The tree search time accounts for the CPU time t_{CPU} to compute the range query distances while traversing the tree, and the I/O time t_{IO} needed to retrieve from the disk the image VMTs (the storage on disk of each VMT requires 32Kb) and to transfer them to central memory:

$$t_{tree} = t_{CPU} + t_{IO}. \tag{5.18}$$

By allocating the images of a leaf node in contiguous disk sectors (by exploiting the appropriate operating system primitives) it is possible to reduce the number of disk accesses, so that $t_{CPU} >> t_{IO}$, and $t_{tree} \approx t_{CPU}$ holds. In the worst case, $s_q = 1$,

$$t_{tree} \approx \sum_{i=1}^{N_c} \cdot \sum_{k=0}^{[log_L(\frac{N_i}{c})]} t_d \cdot L^k, \tag{5.19}$$

$$t_{qref} = t_{sim} \cdot \sum_{i=1}^{N_c} [\frac{N_i}{N_{leaves}}] \cdot N_{leaves} \tag{5.20}$$

N_c being the number of database categories. Here t_d is the time for computing a single distance via Eq. 5.12, N_{leaves} the number of tree leaves. Eq. 5.20 takes into account the fact that our tree is balanced and each leaf contains approximately the same number of images, in general $[\frac{N_i}{N_{leaves}}] <= c$.

Eqs. 5.19 and 5.20 provide upper bounds in the sense that the number of evaluated distances, in the tree traversing step, is greater than the average case since, to simplify, we are not considering that in practice at each tree-level many pruned nodes occur. In fact, by setting $s_q = 1$, all nodes of the tree are explored: thus, the number of evaluated distances is equal to the total number of such nodes and the number of retrieved leaves that satisfy the range query is equal to the total number of tree leaves; on the contrary, by choosing $s_q < 1$, at each tree-level there are many pruned nodes and the number of retrieved leaves is lower than N_{leaves}.

The actual variations of times t_{tree} and t_{qref} for an increasing range query radius are plotted in Fig. 5.16 (here, $c = 15, L = 3$).

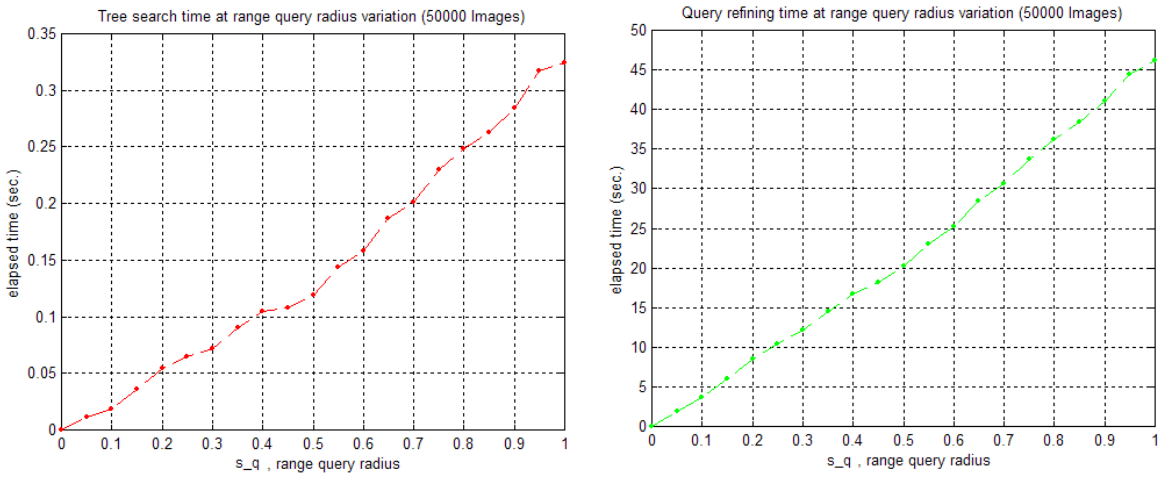


Figure 5.16: Tree search and query refining time at s_q variation

The experimental curves have been obtained by using a PENTIUM IV 3GHz Server (1 GB RAM), under the Windows 2003 Server operating system. To compute the VMT features (about 0.6 sec. for each image) and create the full BCT index (about 1 min. for each category) on the entire database (50000 images subdivided in about 300 categories) our system requires about 14 hours. Moreover for such hardware configuration the time required for computing t_d is about $0.3e-4$ secs. (about 25000 CPU floating operations are necessary), and the time required for computing t_{sim} is about $1e-3$ secs. Such results refer to the case in which the query image is present in the database; on the contrary, one extra second of CPU time is approximately spent to extract from the query image features related to the VMT.

In order to have an idea of BCT performances respect to other access methods, in figure 5.17 we report the index construction time and index size at d (space-dimension) variation.

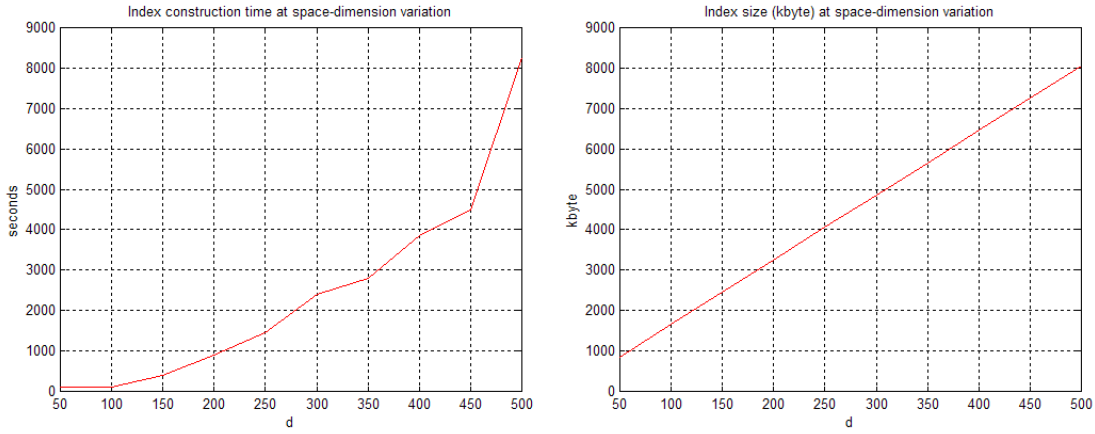


Figure 5.17: Index Construction Time and Index Size at d variation

We compared BEM access method with M-Tree one in terms of `Query Processing Time`, using our data set (50000 images) and metric. To this end, we have observed a performance improvement of 0.05 sec. respect to M-tree and of about 2 sec. respect to sequential scan approach.

By considering Eqs. 5.19 and 5.20, it is possible to estimate the scalability of our system and the total search times for a very large database. Assuming a database of 1000000 images subdivided in 2000 categories (500 images for each category), and choosing $L = 3, c = 25$, we have a tree search time of about 3 sec. and a query refining time of about 1000 sec., in other terms, in the worst case, our system would spend about 15 minutes to execute a user query.

Chapter 6

Foveated Shot Detection for Video Segmentation

6.1 Introduction

Detection of shot boundaries provides a base for nearly all video abstraction and high level video segmentation methods [73], [57]. In this work, we propose a novel approach to partitioning of a video into shots based on a foveated representation of the video.

A shot is usually conceived in the literature as a series of interrelated consecutive frames taken contiguously by a single camera and representing a continuous action in time and space. In other terms, a shot is a subsequence generated by the camera from the time it "starts" recording images, to the time it "stops" recording [42]. However, shot segmentation is ill-defined. On the one hand, a video is generated by composing several shots by a process called `editing`, and due to edit activity different kinds of transitions from one shot to another, either `abrupt` or `gradual`, may take place. An abrupt transition, or hard cut, occurs between two consecutive frames and is the most common type. An example is provided in Fig. 6.1.

Gradual transitions such as fades, wipes and dissolves (see Fig. 6.2 below) are spread over several frames and are obtained using some spatial, chromatic or spatio-chromatic effect; these are harder to detect from a purely data analysis point of view because the difference between consecutive frames is smaller.

Figure 6.1: An example of hard cut effect. An abrupt transition occurs between the second and the third frame

Figure 6.2: An example of dissolve effect

It has been observed [5] from a study of video production techniques, that the production process originates several constraints, which can be useful for video edit classification in the framework of a model based approach to segmentation. But the use of such constraints implies high costs in designing shot models due to the high number of degrees of freedom available in shot production (for review and discussion, see [57], [42]).

On the other hand, for the purposes of video retrieval, one would like to mark the case of any large visual change, whether camera stops or not (e.g., a large object entering the scene). Thus, from a general standpoint, shot detection should rely on the recognition of any significant discontinuity in the visual content flow of the video sequence [42]. Meanwhile, the detection process should be unaffected by less significant changes within the same shot, like object/camera motion and lighting changes, which may contribute to missed or false detections. In such a complex scenario, despite the number of proposals in the literature, robust algorithms for detecting different types of boundaries have not been found, where robustness is related to both detection performance and stability with minimum parameter tuning [49].

6.2 The attentive model for video segmentation

At the heart of our ability to detect changes from one view of a scene to the next is the mechanisms of visual attention.

Film makers have long had the intuition that changes to the visual details across cuts are not detected by audiences, particularly when editing allows for smooth transitions [94]. In the movie *Ace Ventura: When Nature Calls* the pieces on a chess board disappear completely from one shot to the next. In *Goodfellas* a child is playing with blocks that appear and disappear across shots. In fact, almost every movie, and almost every cut, has some continuity mistake, yet, most of the time people are blind to these changes. It has been noted that change blindness is evident when mistakes occur far from the viewer's focus of attention [94].

As already seen in chapter 4, the term attention captures the cognitive functions that are responsible for filtering out unwanted information and bringing to consciousness what is relevant for the observer [51]. Visual attention, in turn, is related to how we view scenes in the real world: moving our eyes (saccade) three to four times each second, and integrating information across subsequent fixations [109]. Saccades represent overt shifts of spatial attention that can be performed either voluntarily (top-down), or induced automatically (bottom-up) by salient targets suddenly appearing in the visual periphery and allow an observer to bring targets of interest onto the fovea, the retinal region of highest spatial resolution. Eye movements, though being characterized by some degree of randomness are likely to occur in a specific path (the scanpath, [77]) so as to focus areas that are deemed important. The scanpath can be conceived as a visuomotor pattern resulting from the perceptual coupling of observer and observed scene.

An example generated on the third frame of Fig. 6.1 is illustrated in Fig. 6.3. The scanpath has been graphically overlapped on the original image: circles represent fixations, and lines trace displacements (saccades) between fixations.

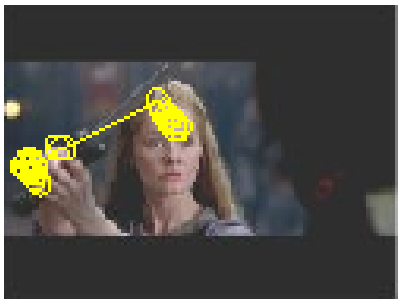


Figure 6.3: Scanpath eye-tracked from a human observer while viewing the third frame presented in Fig. 6.1.

In the course of a scan we have a rich visual experience from which we abstract the meaning or gist of a scene. During next scan, if the gist is the same our perceptual system assumes the details are the same. Clearly, this "sketch" representation not only serves the information reduction purpose of filtering unwanted information, but also, by integrating the gist from one view to the next, to achieve the impression of a stable world. However, the lack of a detailed representation of the outside world from one view to the next can rise failures of change detection [94].

The background question which motivates this work is whether these mechanisms that are useful to prevent audiences noticing the transitions, can conversely be exploited to detect such transitions, and thus help for video segmentation. Intuitively, one could argue that if the playback speed is reduced (or, equivalently, the saccade rate increased) change blindness effects would be reduced too. This corresponds to introducing an ideal observer or agent, capable of tuning his saccadic rate. In some sense, this is akin to Gargi's experimental study of human ground-truthing, where most consistent results in marking shot changes were obtained when subjects performed such task at half speed after viewing the sequence once at full speed [42].

The rationale behind our approach is that perceptual capacity of an observer can be defined at two levels [79]. At the first level there is the ability of the agent to explore the scene in ways mediated by knowledge of patterns of visuomotor behavior, that is the ability to exploit the interdependence between incoming sensory information and motor behavior (eye movements). At the second, higher level there is the accessing by the observer of information related to the nature of observer's own exploration.

For example, while viewing a video sequence, it is reasonable that in the presence of similar visual configurations, and in the absence of an habituation mechanism, an observer should consistently deploy attention to visually similar regions of interest and by following a similar motor pattern; clearly, when the gist of the world observed undergoes a significant change, the visuomotor pattern cannot be exploited further, since inconsistent, and a new scanpath will be generated (see Fig. 6.4). Such an intuitive assumption can be theoretically motivated on the basis that after an abrupt transition the video signal is governed by a new statistical process [63].

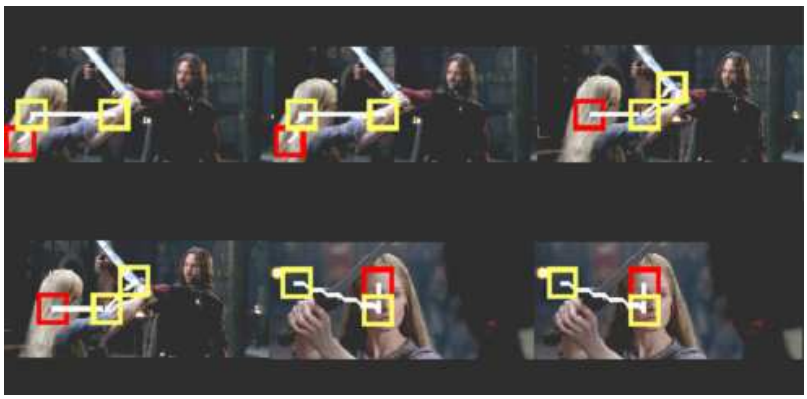


Figure 6.4: Traces generated on six frames embedding an hard cut. The first four FOAs are shown for each frame. The red rectangle represents the first FOA of the trace. The trace sequence abruptly changes between frame 3 and 4

Indeed, it has been shown [12] that gaze-shift is strongly constrained by structure and dynamics of the underlying random field modeling the image. Quantitatively, if a measure $\mathcal{M}$ of `attention consistency` is defined, $\mathcal{M}$ should decrease down to a minimum value. For instance, this is what is likely to occur when a view abruptly changes.

On the other hand, a view change may occur across long delay intervals, as in gradual transitions. In this case, $\mathcal{M}$ should account for a behavior similar to that experienced in change blindness experiments, where subjects fail to detect a slow, global spatio-chromatic editing of a sequence presenting the same image [79], but suddenly succeed when the frame rate of presentation is increased, due to the reduction of the time lag between the first and the last frames of the transition. In this case the $\mathcal{M}$ function should vary smoothly across the interval, while decreasing rapidly if measured on the first and the last frames of the same interval. It is worth remarking that shots involved in a dissolve transition may have similar color distribution, which a color histogram would hardly detect [63], while differing in structural information that can be detected by appropriate algorithms (e.g., edge based).

As in the case of hard cuts, the sequence of attention shifts can be suitably exploited, since its dynamics [12] is strongly intermingled with the complexity of the statistical process modelling the signal (e.g., two-source model for a dissolve [63]).

As regards the second level, namely the evaluation of information about the nature of visual exploration itself, it can be stated as an inference drawn by the observer from its own sensorimotor behavior under prior knowledge available. On such assumption, the problem of detecting a shot change given the change of the observer's behavior $\mathcal{M}$, naturally leads to a Bayesian formulation, and can be conceived as a signal detection problem where the probability that a shot boundary B occurs, given a behavior $\mathcal{M}$, is compared against the probability that a shot boundary is not present.

The introduction of this approach has several advantages, both theoretical and practical. First it allows to find a uniform method for treating both abrupt and gradual transitions. As discussed above, this result stems from relations occurring between the dynamics of gaze-shifts and statistical processes modelling the observed image [12]; also, the method is well grounded in visual perception theories [77], [79].

As such, it is suitable to overcome usual shortcomings of other simpler techniques proposed so far (e.g, histogram manipulations). In this sense, higher robustness can be achieved, as regards performance and stability in detecting important visual changes while discarding negligible ones. Then, once the distinctive scanpath has been extracted from a frame, subsequent analysis needs only to process a sparse representation of the frame. Eventually, attentive analysis can, in perspective, provide a sound and unitary framework at higher levels of video content analysis. For instance, key frame selection/generation could be conceived in terms of average scanpath of shot frames; multimodal processing for deriving semantic properties of a scene, can be stated in terms of attentive audio/visual integration. Summarizing, the discussed video model , in according to the suggestions reported in chapter 4, has two levels of analysis:

- At a lower level, the observer generates the visuomotor patterns, related to the content of the video sequence (we denote $\mathcal{T}(f(t))$ the visuomotor trace (simply, the trace) of frame $f(t)$).

- At a higher level, the observer detects scene changes by judging his own visuomotor behavior in the context of prior knowledge available. More in details, the observer evaluates the information regarding the nature of visual exploration itself and infers the presence of a shot boundary from its own sensorimotor behavior under prior knowledge available on the kinds of transitions he is dealing with. To this end, given two frames $f(t)$ and $f(t+l)$ (for notational simplicity, $t = t_n$), the animate matching procedure (see chapter 4) is used to compute the function $\mathcal{M}(t)$) which gauges the consistency between the two traces $\mathcal{T}(f(t))$ and $\mathcal{T}(f(t+l))$. The behavior of the $\mathcal{M}$ function, is then used by a detection module, based on Bayesian decision theory, which, under prior contextual knowledge available, infers from $\mathcal{M}$ the presence of a scene transition, either abrupt or gradual.

While the low level of video analysis has been largely exploited in chapter 4, some aspects of high level need a separate discussion for the video segmentation problem.

6.3 Using attention consistency and prior knowledge for detecting shot transitions

The observer's behavior can be formalized as the attention consistency gauged over subsequences of the video sequence f. To this end, let us generalize the local attention consistency measure $\mathcal{M}$ to a parametrized family $\mathcal{M} : F \times F \times N^+ \rightarrow R^+$, which accounts for the attentive behavior over the full sequence f , namely $(\mathcal{M}(\mathcal{T}(i), \mathcal{T}(i+l)))_{i=0,l,\ldots,N/l}$, being l the considered frame distance.

In such framework, the problem of inferring a shot change given the change of observation behavior $\mathcal{M}(t)$ can be conceived as a signal detection problem where the probability that a shot boundary B occurs, given a behavior $\mathcal{M}(t)$, $P(B|\mathcal{M}(t))$, is compared against the probability that a shot boundary is not present, $P(\bar{B}|\mathcal{M}(t))$.

More precisely, the observer's judgement of his own behavior can be shaped in a Bayesian approach where detection becomes the inference between two hypotheses:

- $\mathcal{H}_0$: no shot boundary occurs between the two frames under analysis ($\bar{B}$)
- $\mathcal{H}_1$: a shot boundary occurs between the two frames (B)

In this setting the optimal decision is provided by a test where $\mathcal{H}_1$ is chosen if $p(\mathcal{M}(t)|B)P(B) > p(\mathcal{M}(t)|\bar{B})P(\bar{B})$ and $\mathcal{H}_0$ is chosen, otherwise. Namely a cut occurs if:

$$\mathcal{L}(t) > \frac{P(\bar{B})}{P(B)} = \frac{1 - P(B)}{P(B)} \tag{6.1}$$

where $\mathcal{L}(t) = \frac{p(\mathcal{M}(t)|B)}{p(\mathcal{M}(t)|\bar{B})}$ represents a likelihood ratio.

In general, the prior shot probability $P(B)$ models shot boundaries as arrivals over discrete, nonoverlapping temporal intervals, and a Poisson process seems an appropriate prior [64], [49], which is based on the number of frames elapsed since the last shot boundary.

Hanjalic has suggested [49] that the prior $P(B)$ should be more conveniently corrected by a factor depending upon the structural context of the specific shot boundary, gauged through a suitable function.

It is possible to generalize this suggestion resorting to contextual Bayesian analysis [53] in which an occurrence of the property B is detected by taking into account the behavior $\mathcal{M}(t)$ given a context E, that is a set of events $\{e_1, e_2, ..., e_n\}$ characterizing B. Namely, $\mathcal{H}_1$ is chosen if $p(\mathcal{M}(t)|B, E)P(S|E) > p(\mathcal{M}(t)|\bar{B}, E)P(\bar{B}|E)$.

Thus, a cut is detected according to the likelihood ratio:

$$\mathcal{L}(t) > \frac{1 - P(B|E)}{P(B|E)}, \tag{6.2}$$

where now the r.h.s. of Eq. defines the adaptive threshold:

$$T(t) = \frac{1 - P(E|B)P(B)}{P(E|B)P(B)}. \tag{6.3}$$

The prior probability $P(B)$ models the Poisson process of boundary arrival according to the cumulative probability $P(B) = \frac{1}{2} \cdot \sum_{w=0}^{\lambda(t)} \frac{\mu^w}{w!} \exp(-\mu)$ [49], where $\lambda(t)$ is the shot-length at the current frame and w is a frame-counter that is reset in correspondence of a detected shot boundary.

As regards $P(E|B)$, under weak coupling assumption [114] of structural events $e_1, e_2, ..., e_n$, we can set $P(e_1, e_2, ..., e_n|B) = \prod_i P(e_i|B)$. The events that constitute the structural context can be described as follows.

Consider the behavior of function $\mathcal{M}$ for both abrupt and gradual transitions. An example is depicted in Fig. 6.5 related to a video sequence characterized by the presence of two hard cuts embedding a dissolve.

The first event we deal with is a `shape` event: when the gist of the world observed abruptly changes (hard cut), $\mathcal{M}$ decreases down to a minimum value.

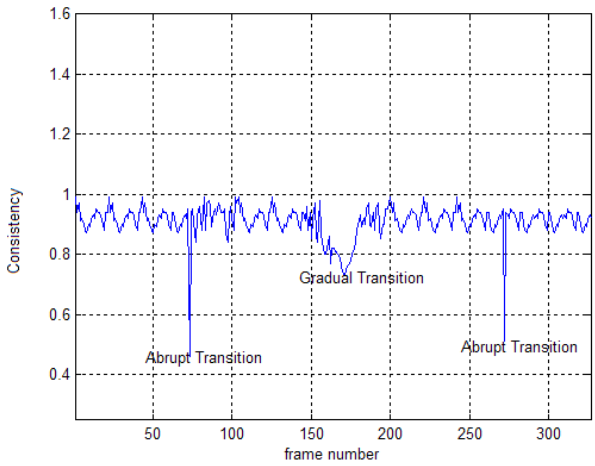


Figure 6.5: Plot of $\mathcal{M}(t)$ function for a sequence characterized by one a dissolve region embedded between two abrupt transitions

Thus, as regards hard cuts, to calculate the probability $P(E|B)$, we use a sliding window of dimension $W = 10$, centered on the frame $f(t)$, thus including all frames in the temporal interval $[t - \frac{W}{2}, t + \frac{W}{2}]$, chosen with the interframe distance $l = 5$. For each frame, we consider the probability that the difference between the first minimum of $\mathcal{M}$, $\mathcal{M}_{min1}$, and the second minimum $\mathcal{M}_{min2}$ detected within the temporal window, be significant:

$$P(E|B) = P(shape|B_{cut}) = \frac{1}{1 + \exp(\beta' \delta)} \tag{6.4}$$

where δ represents the normalized difference $(\mathcal{M}_{min1} - \mathcal{M}_{min2})/\mathcal{M}_{min1}$.

The Fig. 6.6 shows the advantages in using an adaptive threshold.

On the contrary, during a dissolve, the difference between consecutive frames is reduced, and a frame is likely to be similar to the next one. Thus, the consistency function will vary smoothly across the transition interval. Indeed, the behavior of $\mathcal{M}$ along a dissolve region is of parabolic type, and can be more precisely appreciated in Fig. 6.7, where $\mathcal{M}(t)$ decreases very slowly till a local minimum point (fade-out effect), then slowly increases (fade-in effect) in according to the model illustrated in section 3.3.5.

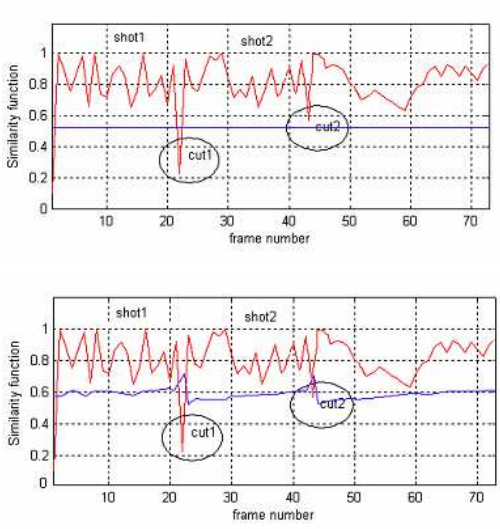


Figure 6.6: Abrupt transition detection: using a static threshold (top) a cut is not detected, in the opposite, using an adaptive threshold the previous missed shot boundary is detected

A second event, which we denote $d\mathcal{M}$, stems from the fact that the first derivative function of $\mathcal{M}$ is approximately constant and about zero in those frames characterized by dissolve effects (see Fig. 6.8).

Clearly, previous events are not sufficient to completely characterize the context of a dissolve region: in fact $\mathcal{M}$ could exhibit a similar trend, e.g. in shots featuring a slow zoom. Thus, the inconsistency between the edge frames, that is the first and last frames of an hypothetical dissolve region, must be taken into account. We denote this event a `change` event.

Summing up, in the case of dissolves we can assume:

$$P(E|B) = P(shape|B_{dis})P(d\mathcal{M}|B_{dis})P(change|B_{dis}) \tag{6.5}$$

To calculate the probability $P(E|B)$, we use a sliding window of dimension $W = 20$, centered on the frame $f(t)$, which includes all frames in the temporal interval $[t-\frac{W}{2}, t+\frac{W}{2}]$, chosen with the interframe distance $l = 5$. The first term on the r.h.s. of Eq. 6.5 is defined as:

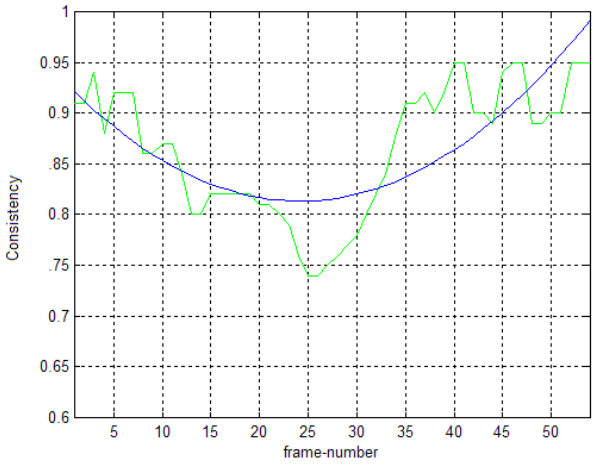


Figure 6.7: Attention consistency $\mathcal{M}$ in a dissolve region and its parabolic fitting

$$P(shape|B_{dis}) = \frac{1}{1+\exp(\beta'(d_{minP}))}, \tag{6.6}$$

where d_{minP} represents the distance between the absolute minimum of $\mathcal{M}$ within the temporal window and the minimum of the parabolic fitting performed on $\mathcal{M}$ values occurring in the same window.

The second term $P(d\mathcal{M}|B_{dis})$ accounting for the probability that derivative $\frac{d\mathcal{M}}{dt}$ be close to zero, is modelled as

$$P(d\mathcal{M}|B_{dis}) = \exp(-k|\frac{d\mathcal{M}}{dt} - \mu|), \tag{6.7}$$

where μ is the mean value of $\frac{d\mathcal{M}}{dt}$ within the time window. To compute derivatives, the $\mathcal{M}$ curve is preprocessed via median filtering [84] in order to avoid noise boost-up.

The third term $P(change|B_{dis})$, representing the probability that the first and the last frame of the dissolve be different, is given by

$$P(change|B_{dis}) = 1 - \frac{1}{1+\exp(-\beta(\mathcal{M}(\mathcal{T}(f_{start}),\mathcal{T}(f_{end})) - \delta'))}, \tag{6.8}$$

where f_{start} and f_{end} are the first and last frame of the sliding window, $f_{start} = f_{t-\frac{W}{2}}$ and $f_{end} = f_{t+\frac{W}{2}}$ respectively.

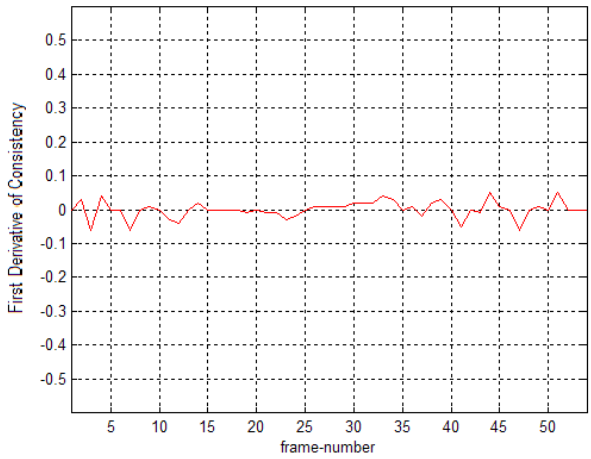


Figure 6.8: First derivative of $\mathcal{M}(t)$ in the same region shown in Fig. 6.7

The variation δ' is defined as:

$$\delta' = \mathcal{M}_{min} + (\mathcal{M}_{max} - \mathcal{M}_{min})/2 \tag{6.9}$$

where $\mathcal{M}_{min}$ and $\mathcal{M}_{max}$ represent the absolute minimum and maximum values of the $\mathcal{M}$ function within the window, respectively.

The likelihood in Eq. 6.2 is estimated, on training sequences, by computing the histograms of the $\mathcal{M}(t)$ values within a shot and at its boundaries, respectively; then, ideal distributions are derived in non parametric form through Parzen windows [29] using kernels $\xi(1-\mathcal{M})\exp(-(1-\mathcal{M}))$ (boundaries) and $\frac{1}{\sigma\sqrt{2\pi}}\exp(-((1-\mathcal{M})-\mu)^2/2\sigma^2)$ (within shot), where $\xi = 2.5$, $\mu = 1.1$, $\sigma = 0.4$, are the estimated parameters. In fig. 6.9 the normalized distributions of the $\mathcal{M}(t)$ values within a shot and at its boundaries and the related distributions derived by Parzen are reported for the case of abrupt transitions.

Eventually, the decision module can be outlined as in Fig. 6.10.

The input is represented by the $\mathcal{M}(t)$ sequence computed by applying the AC algorithm on the video sequence, together with contextual knowledge. Boundary detection is accomplished according to a two-step procedure, which we denote Inconsistency Detection (ID).

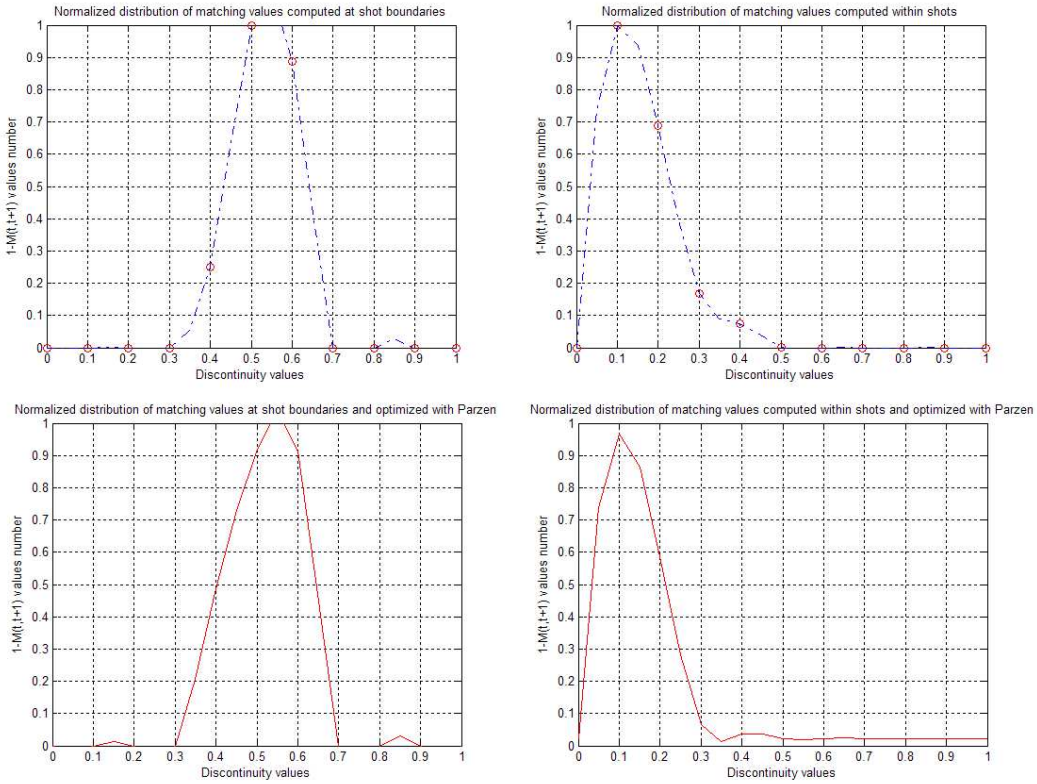


Figure 6.9: Normalized distributions of the $\mathcal{M}(t)$ values within a shot and

In a first step abrupt transitions are detected by means of Eqs. 6.2, 6.3, 6.4. At the end of this phase we obtain the positions of hard cuts, which partition the original video in a sequence of blocks representing candidate shots.

In a second step, the frames interested in dissolve effects are detected. For each block, dissolve regions are individuated by means of Eqs. 6.2, 6.3, 6.6, 6.7, 6.8, computed through a sliding window centered on each frame of the block, chosen according to an interframe distance $l = 5$. Eventually, the output of the system is represented by the list of shot boundary positions, defining the shot segmentation of the original video.

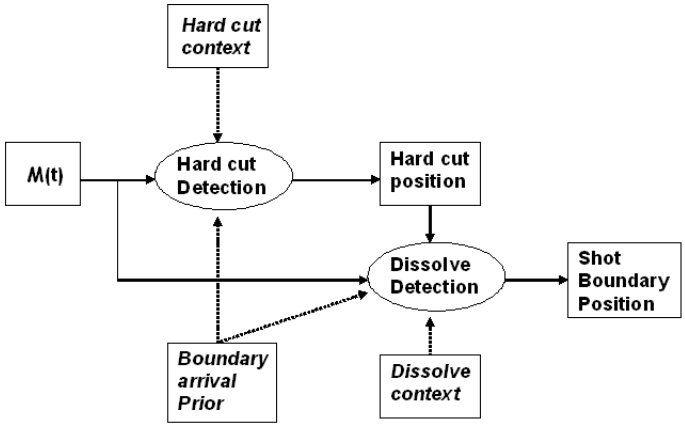


Figure 6.10: The decision module for inferring boundary presence from $\mathcal{M}(t)$ behavior and prior/contextual knowledge

The first step of the ID algorithm has complexity $O(N/l)$, N being the number of frames of the video sequence. The second step is $O(WN_bL_b/l)$, where W, N_b, L_b are the dimension of the sliding window, the number of blocks partitioned along the first step, and the maximum block length, respectively.

The dimensions of the sliding windows have been chosen by means of an analysis of *ROC* curves obtained for the training set in order to maximize true detections with respect to false alarms (in fig. 6.11 the ROC curve for dimensioning W in the case of dissolves is reported).

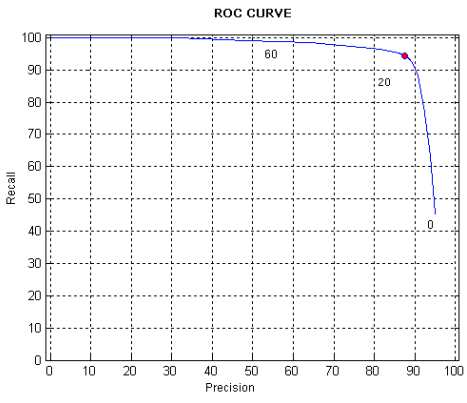


Figure 6.11: ROC curve for dimensioning W in the case of dissolves

Eventually, for what concerns the attention consistency algorithm, a value of $H = 2$ has been chosen for the best fit window provides suitable results, while the value of $N_f^{'} = 10$ was chosen either because, in this way, each FOA is only visited once, and for the bottom-up importance of earliest FOAs. For what concerns the setting of equation parameters, considering again Eq. 4.3, we simply use $\alpha_a = \beta_a = \gamma_a = 1/3$, granting equal informational value to the three kinds of consistencies; similarly, we set $\mu_1 = \mu_2 = 1/2$ in Eq. 4.10.

The algorithm 5 and 6, reported in the following, summarizes the ID procedures for detecting abrupt and gradual transitions.

Algorithm 5 Abrupt Transitions Detection

```
Given a video v, l and W
c[] ← 0
k ← 0
Detect all cuts-position c[]
for i = W/2, i = i + l, i < length(v) − W/2 do
    for j = i − W/2 ... i + W/2 − 1 do
        Compute M(f_j, f_{j+1})
    Compute L(i)
    if (L(i) > T_cut(i)) then
        c[k] ← i
        k++
Build intermediate video-blocks b[]
h ← 1
f_s ← f_1
f_e ← f_{length(v)}
for z = 1...length(c) do
    b_h = [f_s, ..., f_{c(z)−1}]
    f_s ← f_{c(z)}
    h++
b_h = [f_s, ..., f_e]
Return all intermediate video-blocks b[]
```

Algorithm 6 Gradual Transitions Detection

```
Given a set of video blocks b[], l and W
d[] ← 0
Detect all dissolve positions inside a block h, d_h[]
for h = 1...length(b[]) do
  k ← 0
  if length(b_h[]) > W then
    for i = W/2, i = i + l, i < length(b_h[]) − W/2 do
      for j = i − W/2 ... i + W/2 − 1 do
        Compute M(f_j, f_{j+1})
      Compute L(i)
      if (L(i) > T_diss(i)) then
        d_h[k] ← i
        k++
Build final video-blocks s[]
y ← 1
for h = 1...length(b[]) do
  if b_h[] is affected by a dissolve then
    f_s ← b_h(1)
    f_e ← b_h(length(b_h))
    for z = 1...length(d_h) do
      s_y = [f_s, ..., f_{d_h(z)−1}]
      f_s ← f_{d_h(z)}
      y++
    s_y = [f_s, ..., f_e]
    y++
  else
    s_y = b_h
    y++
Return all final video-blocks s[]
```

6.4 Experiments and results

To evaluate the performance of the proposed shot detection algorithm, a database of video sequences has been obtained from documentaries and news belonging to TREC01 video repository and from famous movies. The database represents a total of 1694 cuts and 440 dissolves in approximately 166 min. of video. The selected sequences are complex with extensive graphical effects. Videos were captured at a rate of 30 frames/sec, 640×480 pixel resolution, and stored in AVI format. These video sequences are also characterized for presenting significant dissolve effects. For each sequence a ground-truth was obtained by three experienced humans using visual inspection [42].

To obtain an estimate of parameters for detection (we have set $\beta'^{=0.075,\mu=50}$ for abrupt transitions and $\beta'^{=0.075,\beta=20,k=4.5,\mu=19}$ for gradual transition), the training set, shown in Table 6.1, has been used.

Table 6.1: Description of the video training set

Sequence	Dur.(sec.)	Transitions (Abrupt-Gradual)
The School of Athens (Docum.)	60	0-9
BOR03 (Doc. TREC01)	330	34-21
ANNI006 (Doc. TREC01)	366	41-28
The Time Machine (Movie)	118	15-6
The Life is Beautiful (Movie)	600	100-30
Moulin Rouge (Movie)	700	200-10
Total	36 min	390-104

Experiments for performance evaluation were carried out on a test set including a total of 1304 cuts and 336 dissolves in 130 min. of video, which is summarized in Table 6.2.

The comparison between the proposed algorithm's output and the ground truth relies on the well know *recall* and *precision* figures of merit [42]:

$$recall = detects/(detects + MD) \tag{6.10}$$

$$precision = detects/(detects + FA) \tag{6.11}$$

Table 6.2: Description of the video sequences in the test set

Sequence	Dur.(sec.)	Transitions (Abrupt-Gradual)
ANNI005 (Doc.TREC01)	245	38-8
BOR02 (Doc.TREC01)	328	20-9
BOR07 (Doc.TREC01)	420	45-22
BOR08 (Doc.TREC01)	350	42-18
NAD31 (Doc.TREC01)	516	51-19
NAD33 (Doc.TREC01)	310	41-8
NAD53 (Doc.TREC01)	692	62-36
NAD55 (Doc.TREC01)	485	49-24
NAD57 (Doc.TREC01)	420	43-23
SENSES111 (Doc.TREC01)	388	31-18
Desert Storm (News)	30	4-4
Mandela (News)	22	0-3
Dinosaurs (Movie)	600	205-50
Harry Potter (Movie)	661	176-21
Matrix (Movie)	617	162-0
The Fifth Element (Movie)	600	191-0
The Lord of the Rings II (Movie)	627	63-33
The Patriot (Movie)	500	81-47
Total	130 min	1304-336

where *detects* denotes the correctly detected boundaries, while MD and FA denote missed detections and false alarms, respectively.

In other terms, at fixed parameters, *recall* measures the ratio between right detected shot changes and total shot changes in a video, while *precision* measures the ratio between right detected shot changes and the total shot changes detected by algorithm.

Results obtained are provided in Tables 6.3 and 6.4 and summarized in Table 6.5.

Table 6.3: Abrupt transition performance of the foveated detection method

Video	Cuts	Detections	MD	FA
ANNI005	38	40	0	2
BOR02	20	19	1	0
BOR07	45	50	1	6
BOR08	42	41	2	3
NAD31	51	52	0	1
NAD33	41	41	0	0
NAD53	62	65	1	4
NAD55	49	49	0	0
NAD57	43	42	2	1
SENSES111	31	31	2	2
Desert Storm	4	4	0	0
Dinosaurs	205	210	3	8
Harry Potter	176	176	0	0
Matrix	162	170	2	10
The Fifth Element	191	189	3	1
The Lord of the Rings II	63	67	0	4
The Patriot	81	83	1	3
Total	1304	1322	18	45

The proposed method achieves a 97% recall rate with a 95% precision rate on abrupt transitions, and a 92% recall rate with a 89% precision rate on gradual transitions (Table 6.5). In order to provide an idea about the quality of this results, we refer to the discussion published by Hanjalic [49]. In particular, on dissolve detection, it is worth comparing with Lienahrt [62] and works therein reported, [63], [115].

Table 6.4: Gradual transition performance of the foveated detection method

Video	Dissolves	Detections	MD	FA
ANNI005	8	8	2	2
BOR02	9	10	3	4
BOR07	22	24	3	5
BOR08	18	16	2	0
NAD31	19	22	1	4
NAD33	8	8	0	0
NAD53	36	38	0	2
NAD55	24	27	2	5
NAD57	23	21	3	1
SENSES111	18	21	1	4
Desert Storm	4	4	0	0
Mandela	3	3	0	0
Dinosaurs	50	52	5	7
Harry Potter	21	22	1	2
The Lord of the Rings II	33	35	5	7
The Patriot	47	49	2	4
Total	336	360	30	43

Table 6.5: Performance of the method

Type of Transition	Average Recall	Average Precision	$F1$
Abrupt	0.97	0.95	0.93
Gradual	0.92	0.89	0.90

Also, Table 6.5 provides results in terms of the $F1$ metric, $F1 = 2 \times precision \times recall/(precision + recall)$, which is commonly used to combine *precision* and *recall* scores [87], $F1$ being high only when both scores are high. Summing up, the method proposed here achieves an overall average $F1$ performance of 0.91 when considering both kinds of transitions. This result can indicatively be compared to the performance of a recently proposed method [87] that uses global and block wise histogram differences, camera motion likelihood, followed by k-nearest neighbor classification. Such method achieves an $F1$ performance of 0.94 and 0.69, for hard cuts and gradual transitions, respectively, resulting in an average performance of 0.82; interestingly enough, this result is higher than average scores (0.82 and 0.79) obtained by the two best performing systems at 2001 TREC evaluation [87].

It is worth noting that, in our case, the overall score of 0.91 also accounts for results obtained by processing movies included in our test set, which eventually resulted to be the most critical. For completeness sake, by taking into account only TREC01 video sequences, the overall performance of our method is 0.925.

As regards the efficiency of the method, recall that to obtain the visuomotor trace of the frame, main effort is spent on pyramid and WTA computation, which can be estimated as an $O(|\Omega|)$ step, where $|\Omega|$ represents the number of samples in the image support Ω, while FOA analysis involves lower time complexity, since each of the N_f FOAs is defined on a limited support with respect to the original image ($1/36|\Omega|$) and only 10 FOAs are taken into account to form a trace. The AC algorithm is $O(N_f)$, that is linear in the number of FOAs. The first step of ID algorithm has complexity $O(N/l)$, N and l being the number of frames of the video sequence and the interframe distance, respectively. The second step is $O(WN_bL_b/l)$, where W, N_b, L_b are the dimension of the sliding window, the number of blocks partitioned along the first step, and the maximum block length, respectively.

From this analysis, by considering operations performed on a single frame, we can expect that most of the time will be spent in the low-level perception stage, while the AC and ID algorithms will have higher efficiency, the former only performing on a sparse representation of the frame ($N_f = 10$) and the latter working on $\mathcal{M}(t)$ values of the sliding window of dimension W. This is experimentally confirmed from the results obtained and reported in Table 6.6.

Table 6.6: Average frame processing time for each step

Steps	Low-level	AC algorithm	ID algorithm
Elapsed time (msec)	26	6.2	2.8

The system achieves a processing speed per frame of about 35 ms on the Pentium IV 2.4 GHz PC (1 GB RAM). It is worth noting that the current prototype has been implemented using the Java programming language, running in Windows XP operating system, without any specific optimization.

Clearly, for time critical applications, the bottleneck of the proposed method, that is the computing of visuomotor traces, could be easily reduced by resorting to existing hardware implementation of pyramidal representations ([16]) and more efficient realizations of the WTA scheme (e.g., in [9] a network is presented, which has $O(\lg n)$ time complexity).

Chapter 7

Final Remarks and Conclusions

7.1 Image retrieval task

A novel approach to QBE has been presented. We have shown how, by embedding within image inspection algorithms active mechanisms of biological vision such as saccadic eye movements and fixations, a more effective processing can be achieved. Meanwhile, the same mechanisms can be exploited to discover and represent hidden semantic associations among images, in terms of categories, which in turn drives the query process along an animate image matching. Also, such associations allow an automatic pre-classification, which makes query processing more efficient and effective in terms of both time (the total time for presenting the output is about 4 sec.) and precision results.

Note that the proposed representation allows the image database to be endowed with semantics at a twofold level, namely, both at the set-up stage (learning) and at the query stage. In fact, as regards the query module it can in principle work on the given WW space learned along the training stage or by further biasing the WW by exploiting user interaction in the same vein of [92].

A feasible way could be that of using an interactive interface where the actions of the user (pointing, grouping, etc.) provide a feedback that can be exploited to tune on the fly parameters of the system, e.g. the category prior probability $P(C_n)$ or, at a lower level, the mixing coefficients in Eq. 4.3 to grant more information to color as opposed to texture, for instance. Current research is devoted to such improvements as well as to extend our experiments to very large image databases.

7.2 Video segmentation task

We have defined a novel approach to partitioning of a video into shots based on a foveated representation of the video. To the best of our knowledge, foveation mechanisms have never been taken into account for video segmentation, while there are some recent applications to video compression (refer to [13]). The motivation for the introduction of this approach stems from the fact that success or failure in the perception of changes to the visual details of a scene across cuts are related to the attentive performance of the observer [94]. By exploiting the mechanism of attention shifting through saccades and foveations, the proposed shot-change detection method computes, at each time instant, a consistency measure $\mathcal{M}(t)$ of the foveation sequences generated by an ideal observer looking at the video. The problem of detecting a shot change given the change of consistency $\mathcal{M}(t)$ has been conceived as a Bayesian inference of the observer from his own visual behavior.

The main results achieved can be summarized as follows. The proposed scheme allows the detection of both cuts and dissolves between shots using a single technique, rather than a set of dedicated methods. Also, it is well grounded in visual perception theories and allows to overcome usual shortcomings of many other techniques proposed so far. In particular, features extracted are strictly related to the visual content of the frame; this, for instance is not true for simpler methods, such as histogram based methods, where, in general, totally different frames may have similar histograms (e.g., a frame generated by randomly flipping the pixels of another frame has the same histogram of the original one). Further, the FOA representation is robust with respect to smooth view changes: for instance, an object translating with respect a a background, gives rise to a sequence of similar visuomotor traces. Meanwhile, a large object entering the scene would be recognized as a significant discontinuity in the visual content flow of the video sequence; in this sense, the approach accounts for the more general definition of shot as a sequence of frames that was, or appears to be, continuously captured from the same camera [42].

Once the distinctive scanpath has been extracted from a frame, subsequent feature analysis need only to process a sparse representation of the frame; note that for each frame, we consider 10 FOAs, each FOA being defined on a square support region whose dimension is 1/36 of the original image; further reduction is achieved at the detection stage, where only the $\mathcal{M}$ function is processed (cfr. Table 6.6). Last, the perceptual capacity of an observer to account for his own visual behavior, naturally leads, in this framework, to a Bayesian decision formulation for solving the detection problem, in a vein similar to [64], [49]. In particular, by resorting to recently proposed contextual Bayesian analysis [53], we have generalized some suggestions introduced in [49] for exploiting structural information related to different types of transitions.

It is worth remarking that, with respect to the specific problem of gradual transitions, the present work focuses on dissolve detection. However, the detection scheme can be easily extended to other kinds of transitions; for instance, preliminary experiments performed on wipes (not reported here, because out of the scope of this paper) show a behavior of the $\mathcal{M}$ function characterized by a damped oscillatory pattern. Also, beyond the context of video segmentation, the proposed technique introduces some novelties *per se* with respect to the "Where" and "What" integration problem, the explicit use of the fixation time in building a visuomotor trace, and as regards the way to exploit the extracted information for comparing different views (information look-up problem).

Results on a test set representing a total of 1304 cuts and 336 dissolves in 130 min. of video, including videos of different kinds are reported and validate the proposed approach. The performance of the currently implemented system is characterized by a 97% recall rate with a 95% precision rate on abrupt transitions, and a 92% recall rate with a 89% precision rate on gradual transitions. Meanwhile it exhibits a constant quality of detection for arbitrary complex movie sequences with no need for tuning parameters. Interestingly enough, the system has been trained on a small data set with respect to the test set used.

However, the introduction of an attention based approach not only is motivated by performance in shot-detection, but in perspective it could constitute an alternative to traditional approaches, and overcome their limitations for high-level video segmentation. Consider, for instance, the issue of scene change detection by jointly exploiting video and audio information. Audio and pictorial information play different roles and, to some extent, complementary. When trying to detect a scene decomposition of the video sequence, the analysis of visual data may provides candidate cuts, which are successively validated through fusion with information extracted from audio data. How to perform such fusion, in a principled way, is unclear. However, behavioral studies and cognitive neuroscience have remarked the fundamental role of attention in integrating multimodal information [10]; and the approach proposed here could serve as a sound basis for such integration. In this way, the low level and high level video analysis could share the processing steps, making the entire content analysis process more effective and efficient.

Bibliography

[1] B. Adams, C. Breazeal, R.A. Brooks, and B. Scassellati, "Humanoid Robots: A New Kind of Tool", *IEEE Int. Systems*, 2000, pp. 25–31.

[2] D. A. Adjeroh, and K. C. Nwosu, "Multimedia Database Management - Requirements and Issues",*IEEE Transaction Multimedia*, pp. 24-33, July-September 1997.

[3] A.M. Alattar, "Detecting and Compressing Dissolve Regions in Video Sequences witha DVI Multimedia Image Compression Algotithm", *IEEE International Symposium on Circuits and Systems (ISCAS)*, Vol.1, pp.13-16, May 1993.

[4] A. A. Altan, A. Akansu, and W. Wolf, "Multi-Modal dialog Scene Detection using Hidden Markov Models for Content-Based Multimedia Indexing", *Multimedia Tools and Application Journal*, Kluwer Pub., 2001.

[5] S. Antani, R. Kasturi, and R. Jain, "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video", *Pattern Recognition*, 2002, vol. 35, pp. 945–965.

[6] M.J. Swain, and D.H. Ballard, "Color indexing", *Int. Journal of Computer Vision*, vol. 7, n. 1, 1991, pp. 11–32.

[7] D. Ballard, "Animate Vision", *Artificial Intelligence*, vol. 48, pp. 57-86, 1991.

[8] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, "Clustering on hyperspheres using Expectation Maximization", *Technical Report TR-03-07*, Department of Computer Sciences, University of Texas, February 2003.

[9] Barnden and J.A. Srinivas, "Temporal winner-take-all networks: a time-based mechanism for fast selection in neural networks", *IEEE Transactions on Neural Networks*,vol. 4, 1993, pp. 844–853.

[10] A. Berthoz, "Le sens du mouvement", Ed. Odile Jacob, 1997.

[11] J. A. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models", *Technical Report*, U.C. Berkeley, April 1998

[12] G. Boccignone and M. Ferraro, "Gaze shift as a constrained random walk", Physica A, vol. 331, (2004), pp. 207–218.

[13] S. Lee, M. S. Pattichis, and A. C. Bovik, "Foveated Video Compression with Optimal Rate Control", *IEEE Trans. on Image Processing*, vol. 10, no. 7, 2001, pp. 977–992.

[14] S. Brin, "Near neighbor search in large metric spaces", *In Proc. of VLDB95*, 574-584, Switzerland, 1995.

[15] W. A. Burkhard, and R. M. Keller, "Some approaches to best-match file searching", *Comm. of the ACM*, 16(4):230236, 1973.

[16] P.J. Burt, "A Pyramid-Based Front-End Processor for Dynamic Vision Applications", *Proc. of the IEEE*, vol. 90 (7), 2002, pp. 1188–1200.

[17] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: image segmentation using expectation-maximization and its application to image querying", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1026-1038, 2002.

[18] G. Celeux, and G. Govaert, "A classification EM algorithm for clustering and two stochastic versions", *Computational Statistics and Data Analysis*, vol. 14, pp. 315-332, 1992.

[19] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient Access Method for Similarity Search in Metric Spaces", *In Proc. of 23rd International Conference on VLDB*, 426-435, 1997.

[20] C. Colombo, A. Del Bimbo, and P. Pala, "Semantics in Visual Information Retrieval", *IEEE MultiMedia*, vol. 6, no. 3, pp. 38-53, 1999

[21] J.M. Corridoni, A. Del Bimbo, and P. Pala, "Image retrieval by color semantics", *Multimedia Systems*, vol. 7, no. 3, 175-183, 1999.

[22] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos, "The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments", *IEEE Trans. Image Processing*, vol. 9, no. 3, pp. 20-37, 2000.

[23] T.M. Cover, and J.A. Thomas, "Elements of Information Theory", Wiley-Interscience, 1991.

[24] A. Del Bimbo and P. Pala, "Visual image retrieval by elastic matching of user sketches", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 121-132, 1997.

[25] A. Del Bimbo, M. Mugnaini, P. Pala, and F. Turco, "Visual querying by color perceptive regions", *Pattern Recognition*, vol. 31, no. 9, pp. 1241-1253, 1998.

[26] C. Colombo, and A. Del Bimbo, "Visible image retrieval", In L. Bergman and V. Castelli, eds., *Image Databases, Search and Retrieval of Digital Imagery*, Chapter 2, pp. 11-33, Wiley 2002.

[27] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum likelihood from incomplete data", *Journal of the Royal Statistical Society*, vol. 39, pp. 1-38, 1977.

[28] C. Djeraba, "Association and Content-Based Retrieval", *IEEE Trans. on Knowledge and Data Engineering*, vol. 15, no. 1, pp. 118-135, 2003.

[29] R.O. Duda, P.E. Hart, and D.G. Stork, "Pattern Classification", Wiley and Sons, N.Y, 2001.

[30] Pinar Duygulu, Kobus Barnard, Nando de Freitas, and David Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary", *Seventh European Conference on Computer Vision*, 97:112, 2002.

[31] S. Edelman, "Constraining the neural representation of the Visual World", *Trends in Cognitive Science*, vol. 6, no. 3, pp. 125-131, 2002.

[32] J. Feder, "Towards image content-based retrieval for the World-Wide Web", *Advanced Imaging*, vol. 11, no. 1, pp. 26-29, 1996.

[33] W.A.C. Fernando, C.N. Canagarajah, D.R. Bull, "Sudden Scene Change Detection in MPEG-2 Video Sequences", *in Proc. IEEE International Workshop on Multimedia Signal Processing*, Copenhagen, Denmark, Sep. 1999, pp. 259-264.

[34] W.A.C. Fernando, C.N. Canagarajah, D.R. Bull, "Video Segmentation and Classification for Content Based Storage and Retrieval Using Motion Vectors", *in Proc. SPIE Conference on Storage and Retrieval for Image and Video Databases VII*, San Jose, CA, USA, Jan. 1999, pp. 687-698.

[35] W.A.C. Fernando, C.N. Canagarjah, and D.R. Bull, "Fade and Dissolve Detection in Uncompressed and Compressed Video Sequences", *IEEE International Conference on Image Processing*, vol.3, 1999, pp. 299–303.

[36] W.A.C. Fernando, C.N. Canagarajah and D.R. Bull, "A Unified Approach to Scene Change Detection in uncompressed and compressed video", *IEEE Trans. on Consumer Electronics*, vol. 46, n. 3, Aug. 2000.

[37] M. Ferraro, G. Boccignone and T. Caelli, "Entropy-based representation of image information", *Patt. Recognit. Lett.*, vol. 23, 2002, pp. 1391–1398.

[38] M. Flickner et al., "Query by image and video content: the QBIC system", *IEEE Computer* vol. 28, no. 9, pp. 23-32, 1995.

[39] R.G. Fryer, and M.O. Jackson, "Categorical Cognition: A Psychological Model of Categories and Identification in Decision Making", *NBER Working Paper No. W9579*, March 2003.

[40] R. M. Ford, C. Robson, Daniel Temple, and M. Gerlach, "Metrics for Scene Change Detection in digital Video Sequences", IEEE Int. Conf. on Multimedia Computing and Systems, 1997.

[41] B. Furht, S. W. Smoliar, and H. Zhang, "Video and Image Processing in Multimedia Systems", Norwell, MA Kluwer, 1995.

[42] U.Gargi, R. Kasturi and S.H. Strayer, "Performance characterization of video-shot change detection methods," *IEEE Trans. on Circ. Sys. for Video Tech.* vol.10, no. 1, 2000, pp. 1–13.

[43] V. Ganti, R. Ramakrishnan, J. Gehrke, A. Powell, and J. French, "Clustering large datasets in arbitrary metric spaces", *In the Proceedings of International Conference on Data Engineering*, 1999.

[44] G.J. Giefing, H.Janssen, and H. Mallot, "Saccadic Object Recognition with an Active Vision System", *Proc. 10th Eur. Conf. Art. Intell.*, 1992, pp. 803–805.

[45] M. A. Goodale and G. K.Humphrey, "The objects of action and perception", *Cognition*, vol. 67, 1998, pp. 181–207.

[46] W. I. Grosky, "Managing Multimedia Information in Database Systems", *Comm. of the ACM*, vol. 40, no. 12, December 1997

[47] A. Gupta, et al, "The Virage image search engine: an open framework for image management", *Storage and Retrieval for Image and Video Databases IV*, Proc SPIE 2670, pp 76-87, 1996.

[48] A. Hampapur, R. Jain, and T.Weymouth, "Digital video segmentation, in *Proc. ACM Multimedia94*, 1994, pp. 357-364.

[49] A. Hanjalic, "Shot-Boundary Detection: Unraveled and Resolved?", *IEEE Trans. Circuits Syst. on Video Technol.*, vol. 12, 2002, pp. 90–105.

[50] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, 1998, pp. 1254–1259.

[51] L. Itti and C. Koch, "Computational modelling of visual attention", *Nature Reviews*, vol. 2, 2001, pp. 1–11.

[52] Anil K. Jain, and Richard C. Dubes, "Algorithms for Clustering Data", Prentice Hall, Englewood Cliffs, New Jersey, 1998.

[53] W. Richards, A. Jepson, and J. Feldman, "Priors, preferences and categorical percepts", in *Perception as Bayesian Inference*, D.C. Knill, W. Richards, eds., Cambridge University Press, MA, 1996, pp. 93–122.

[54] I. Kalantari, and G. McDonald, "A data structure and an algorithm for the nearest point problem", *IEEE Transactions on Software Engineering*, 9(5), 1983.

[55] R. Kasturi, R. Jain, "Dynamic vision", *Computer Vision: Principles*, IEEE Computer Society Press, Washington DC, 1991, pp. 469-480.

[56] T. Kikukawa and S. Kawafuchi, "Development of an automatic summary editing system for the audio visual resources", *Trans. Inst. Electron., Inform., Commun. Eng.*, vol. J75-A, no. 2, 1992, pp. 204-212.

[57] D. Li and H. Lu, "Model based video segmentation", *IEEE Trans. on Circuits and Systems for Video Technology*, no. 5, 1995, pp. 533–544.

[58] Z.N. Li and J. Wei, "Spatio-temporal Joint probability Images for Video Segmentation", *Proc. IEEE Int. Conf. on Image Processing*, Vancouver, BC, Canada, Sep.2000, pp.295-298.

[59] W. Li, S. Candan, K. Hirata, and Y. Hara, *Supporting efficient multimedia database exploration*, The VLDB Journal, pp. 312-326, 2001.

[60] H.Y.M. Liao, L.H. Chen, C.W. Su and H.R. Tyan, "A Motion-tolerant Dissolve Detection Algorithm", *Proc. IEEE Int. Conf. on Multimedia and Expo*, Lausanne, Switzerland, Aug. 2002, pp. 225-228.

[61] R. Lienhart, "Comparison of Automatic Shot Boundary Detection Algorithms", *Proc. of SPIE 3656-29 Image and Video Processing*, vol. VII, 1999, pp. 1–12.

[62] R. Lienhart, "Reliable Dissolve Detection", *Proc. SPIE 4315*, 2001, pp. 219–230.

[63] R. Lienhart, "Reliable transition detection in videos: a survey and practicioner's guide", *International Journal of Image and Graphics*, Vol. 1, No. 3, 2001, pp. 469–486.

[64] A. Lippman and N. Vasconcelos, "Statistical models of video structure for content analysis and characterization", *IEEE Trans. Image Processing*, vol. 9, Jan 2000, pp. 3-19.

[65] Z. Liu, Y. Wang, and T. Chen, "Audio Feature Extraction and Analysis for Scene Segmentation and Classification", *Journal of VLSI Signal Processing*, Kluwer Pub., 1998.

[66] T. M. Liu, H. J. Zhang, and F. H. Qi, "A novel Video Key Frame Extraction Algorithm", *IEEE International Symposium on Circuits and System*, vol. 4, 2002, pp. 149–152.

[67] S. Mallat, "A wavelet tour of signal processing", Academic Press, NY, 1998.

[68] S. Marcus, and V.S. Subrahmanian, "Foundations of Multimedia Database Systems", *Journal of the ACM*, vol. 43, n. 3, pp. 474-523, May 1996.

[69] D.J.C. MacKay, "Information Theory, Inference, and Learning Algorithms", *Cambridge University Press*, UK, 2003.

[70] J. Meng, Y. Juan and S.F. Chang, "Scene Change Detection in an MPEG Compressed Video Sequence", SPIE, Alg. and Tech., vol. 2419, Feb. 1995.

[71] R. Milanese, S. Gil, and T. Pun, "Attentive mechanisms for dynamic and static scene analysis", *Opt. Eng.*, vol. 34, 1995, pp. 2428-2434.

[72] E.K. Miller, D.J. Freedman, and J.D. Wallis, "The prefrontal cortex: categories, concepts and cognition", *Phil. Trans. R. Soc. Lond* B, 357, 1123-1136.

[73] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-videosearch for object appearances", *Visual Database Systems II*, 1995, pp.113–127.

[74] J. Nam and A.H. Tewfik, "Dissolove Transition Detection Using B-Splines Interpolation", *EEE Int. Conf. on Multimedia and Expo*, July 2000.

[75] R.M. Neal and G.E. Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants", *Learning in Graphical Models*, M.J. Jordan ed., 355-368, MIT Press, 1998.

[76] E. Niebur and C. Koch, "Computational architectures for attention", The MIT Press, Cambridge, MA, 1998.

[77] D. Noton, and L.Stark, "Scanpaths in the saccdice eye movements during pattern perception", *Visual Research*, vol. 11, pp. 929-942, 1990.

[78] V. E. Ogle, and M. Stonebraker, "Chabot: Retrieval from a relational database of images", *IEEE Computer*, vol. 28, no. 9, pp. 40-48, 1995.

[79] K. O'Regan, "Solving the 'real' mysteries of visual perception: The world as an outside memory", *Can. J. of Psychology, vol. 46*, no. 3, 1992, pp. 461–488.

[80] K. Otsuji, Y. Tonomura, and Y. Ohba, "Video browsing using brightness data", in *Proc. SPIE-IST VCIP91*, vol. 1606, 1991, pp. 980-989.

[81] A. Pentland, et al., "Photobook: tools for content-based manipulation of image databases", *International Journal of Computer Vision* vol. 18, no. 3, pp. 233-254, 1996.

[82] S. Pfeiffer, R. Lienhart, and W. Effelsberg, "Scene Determination based on Video and Audio Features", *Multimedia Tools and Application Journal*, Kluwer Pub.

[83] M. Philips and W. Wolf, "A multi-attribute shot segmentation algorithm for video programs", *Telecomm. Sys.*, n. 9, pp. 393402, 1998.

[84] I. Pitas and A. N. Venetsanopoulos, "Non-linear Filters", Kluwer, 1989.

[85] P.K. Pook, and D. H. Ballard, "Deictic human/robot interaction", *Robotics and Autonomous Systems*, vol. 18, pp. 259-269, 1996.

[86] C. M. Privitera and L. W. Stark, "Algorithms for Defining Visual Regions-of-Interest: Comparison with Eye Fixations", *IEEE Trans. on Pattern Analysis ans Machine Intelligence*, vol. 22, no. 9, 2000, pp. 970–982.

[87] Y. Qi, A. Hauptmann, and T. Liu, "Supervised Classification for Video Shot Segmentation", *IEEE Conference on Multimedia & Expo (ICME03)*, 2003.

[88] R. P. N. Rao, and D. H. Ballard, "Dynamic model of visual recognition predicts neural response properties in the visual cortex", *Neur. Comp.*, vol. 9, pp. 721-763, 1997.

[89] C. Remco and M.T. Veltkamp, "Content-Based Image Retrieval Systems: A Survey", http://www.aa-lab.cs.uu.nl/cbirsurvey/cbir-survey/cbir-survey.html

[90] S.Santini, and R. Jain, "Integrated Browsing and Querying for Image Databases", *IEEE MultiMedia*, vol. 7, no. 3, pp. 26-39, 2000.

[91] S.Santini, "Evaluation Vademecum for Visual Information Systems", *Proc. of SPIE*, vol. 3972, San Jose, USA, 2000.

[92] S.Santini, A. Gupta and R. Jain, "Emergent Semantics through Interactions in Image Databases", *IEEE Trans. on Knowledge and Data Engineering*, vol. 13, pp. 337-351, 2001.

[93] B. Shahraray, "Scene change detection and content-based sampling of video sequences", in *Proc. IST-SPIE*, vol. 2419, Feb. 1995, pp. 2-13.

[94] D.J. Simons and D.T. Levin, "Change blindess", *Trends in Cog. Sc.*, no. 7, 1997, pp. 261–267.

[95] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early Years", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1349-1379, 2000.

[96] J. R. Smith, "Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis", *PhD thesis, Graduate School of Arts and Sciences, Columbia University*, February 1997.

[97] J. R. Smith, and S. F. Chang, "Querying by color regions using the VisualSEEk content-based visual query system", *Intelligent Multimedia Information Retrieval*, (Maybury, M T, ed), AAAI Press, Menlo Park, CA, 23-41, 1997.

[98] V. S. Subrahmanian, *Principles of Multimedia Database Systems*, Morgan Kaufmann 1998.

[99] H. Sundaram, and S. Chang, "Determing Computable scenes in Films and their Structures using Audio-Visual Memory", ACM Multimedia, 2000.

[100] B.T. Truong, C. Dorai, and S. Venkatesk, "New Enhancements to Cut, Fade, and Dissolve detection Processes in Video Segmentation", *ACM Multimedia*, 2000, pp. 219–227.

[101] J. K. Tsotsos, et al., "Modeling visual-attention via selective tuning", *Art. Intell.*, vol.78, 1995, pp. 507–545.

[102] J. Uhlmann, "Satisying general proximity/similarity queries with metric trees", *Information Processing Letters*, 40, 175-179, 1991.

[103] G.J. Walker-Smith, A.G. Gale, and J.M. Findlay, "Eye movement strategies involved in face perception", *Perception*, vol. 6, pp. 313-326, 1997.

[104] Y. Wang, Z. Liu, and J. Huang, "Multimedia Content Analysis", IEEE Signal Processing Magazine, pp. 12-36, November 2002.

[105] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-Sensitive Integrated Matching for Pictures LIbraries", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1-16, Sept. 2001.

[106] J. Wang, and T. Chua, "A Framework for Video Scene Boundary Detection", *ACM Multimedia*, 2002.

[107] , W. Xiong and J.C.-M. Lee, "Efficient scene change detection and camera motion annotation for video classification", *Computer Vision Image Understanding*, 71(2), 1998, 166-181.

[108] K. Yamanishi, J-I. Takeuchi,G. Williams, and P.Melne, "On-Line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms", *Data Mining and Knowledge Discovery*, vol. 8, pp. 275-300, 2004.

[109] A.L. Yarbus, "Eye movements and vision", Plenum Press, New York, NY, 1967.

[110] B.-L. Yeo and B. Liu, "Rapid scene analysis on compressed video", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, Dec. 1995, pp. 533-544.

[111] Peter N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces", *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, p.311-321, Jan. 25-27, 1993.

[112] A. Yoshitaka, T. Ichikawa, "A survey on content based retrieval for Multimedia Database", *IEEE Trans. on Knowledge and Data Engineering*, 11:81-93, 1999.

[113] D. Yu, and A. Zhang, "ClusterTree: Integration of Cluster Representation and Nearest-Neighbor Search for Large Data Sets with High Dimensions", *IEEE Trans. on KDE*, 15(5), 1316-1330, 2003.

[114] A.L. Yuille and H.H. Bulthoff, "Bayesian decision theory and psychophysics", in *Perception as Bayesian Inference*, D.C. Knill, W. Richards, eds., Cambridge University Press, MA, 1996, pp. 123–162.

[115] R. Zabih, J. Miller, and K. Mai, "A Feature-Based Algorithm, for Detecting and Classifying Scene Breaks", *Proc. ACM Int. Conf. Multimedia*, 1995, pp. 189–200.

[116] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video", *Multimedia Syst.*, vol. 1, 1993, pp. 10-28.

[117] H.J. Zhang, L.Y. Ghong and S.W. Smoliar, "Video Parsing Using Compressed Data", in Proc. IS&T/SPIE Conf. on Image and Video Processing II, 1994.

[118] S. Zhong, and J. Ghosh, "A Unified Framework for Model-based Clustering", *Journal of Machine Learning Research*, vol. 4, pp. 1001-1037, 2003.