EURECOM
Department of MultiMedia Communications
2229, route des Crêtes
B.P. 193
06904 Sophia-Antipolis
FRANCE

Research Report RR-10-231

Representation and Analysis of Video Content for Automatic Object Extraction

23, June, 2008

Eric Galmar

Tel : (+33) 4 93 00 81 00
Fax : (+33) 4 93 00 82 00

Supervised by Dr Benoit Huet. (benoit.huet@eurecom.fr)

EURECOM's research is partially supported by its industrial members: BMW Group Research & Technology - BMWGroup Company, Bouygues Télécom, Cisco Systems, France Télécom, Hitachi Europe, SFR, Sharp, STMicroelectronics, Swisscom, Thales.

Abstract

The recent explosion of multimedia applications has called for an increasing demand of advanced search and indexing of multimedia information. Among them, digital video content is certainly one of the most complex to analyze and represent. From this point of view, video objects are considered as essential elements for handling video contents, as they provide accurate and flexible representation for numerous applications such as semantic content analysis or video coding.

Automated object extraction from videos is a difficult task that has been widely addressed in the past years in the context of MPEG-4 video coding. Methods developed so far mostly rely on motion estimation to define the object model and adapt this models frame to frame. However there is an agreement that robustness of motion and the accuracy of the support are dependent to each other.

In this thesis, we first introduce a framework for video object modeling based on a spatiotemporal representation with graphs. The model describes both the internal structure of object regions and their spatiotemporal relationships inside the shot. This approach is fully supported by the MPEG-7 multimedia standard, where the information is structured hierarchically in scenes, shots, objects and regions.

As the next step, we propose a 2D+T scheme for the extraction of spatiotemporal volumes. The method we developed uses local and global properties of the volumes to propagate them coherently in space and time. At this point we investigate grouping of spatiotemporal regions into complex objects using motion models. To address the difficulty of building motion models, we propose a method to propagate and match moving objects to areas where motion information is less relevant.

In a third step, we investigate the benefit of semantic knowledge for spatiotemporal segmentation and labeling of video shots. For this purpose we extend a knowledge-based system providing fuzzy semantic labeling of image regions to video shots. The shot is split into smaller block units and, for each block, volumes are sampled temporally into frame regions that receive semantic labels. The semantic labels are then propagated within volumes and a consistent labeling of the shot is finally obtained by joint propagation and re-estimation of the semantic labels between the temporal segments.

Finally, we explore the capabilities of the representation for indexing and retrieval tasks. We first consider the context of a region-based indexing framework called the Vector Space Model. We present a study of the model properties and show that the spatiotemporal representation gives more robustness to the visual signatures compared to the traditional

keyframe representation. This dissertation concludes by proposing a strategy to compare efficiently object graphs. To this aim we introduce a similarity measure between graphs that we further use to search for a given object.

Résumé

L'explosion récente des applications multimédia génère une demande croissante d'indexation et de recherche de l'information multimédia. Parmi les différents types de contenu, l'information présente dans les vidéos est certainement l'une des plus complexes à analyser et à représenter. De ce point de vue, les objets vidéo apparaissent comme essentiels à une bonne gestion des contenus vidéo, car ils constituent une représentation souple et précise pour de nombreuses applications comme l'analyse du contenu sémantique ou le codage vidéo.

L'extraction automatique d'objets dans les vidéos est une tâche difficile sur laquelle on s'est penché ces dernières années dans le cadre du codage MPEG-4. Les méthodes développées à ce jour repose sur l'estimation de mouvement afin de définir un modèle et de l'adapter ensuite image par image. Cependant on s'accorde à dire que la robustesse du mouvement et la précision du support sont interdépendantes.

Nous présentons dans cette thèse un cadre pour la modélisation des objets vidéo et basée sur une représentation spatiotemporelle par graphes. Le modèle décrit à la fois la structure interne des régions de l'objet et leurs relations spatiotemporelle à l'intérieur du plan vidéo. Cette approche peut être entièrement intégrée dans le standard de description multimédia MPEG-7, où l'information est structurée hiérarchiquement en scènes, plans, objets et régions.

Dans un deuxième temps, nous proposons un schéma de type 2D+T pour l'extraction des volumes spatiotemporels. La méthode développée utilise les propriétés locales et globales des volumes pour les propager de manière cohérente dans l'espace et le temps. Nous examinons alors le groupement de volumes spatiotemporels en objets plus complexes à partir de modèles de mouvement. La construction de modèles de mouvement pouvant s'avérer délicate dans certaines zones, nous proposons une méthode pour propager et mettre en correspondance les objets en mouvement dans les zones ou l'information de mouvement n'est pas suffisamment significative.

Dans un troisième temps, nous étudions l'apport d'une connaissance sémantique supplémentaire pour la segmentation spatiotemporelle et l'annotation des plans vidéo. A cette fin nous étendons un système à base de connaissances permettant l'étiquetage de régions dans une image à la totalité du plan vidéo. Le plan est découpé en un ensemble de blocs d'images et les volumes à l'intérieur de chaque bloc sont échantillonnés temporellement en régions spatiales, lesquelles sont alors interprétées par le système. L'information sémantique est propagée à travers les volumes et l'étiquetage final est obtenu par propagation et ré-estimation

des étiquettes sémantiques entre les segments temporels.

Finalement nous examinons les aptitudes de la représentation proposée vis à vis des tâches d'indexation et de recherche. Nous considérons d'abord un cadre d'indexation région basé sur le modèle vectoriel. Nous étudions les propriétés de ce modèle et montrons que la segmentation spatiotemporelle permet d'améliorer la robustesse des signatures visuelles par rapport à une représentation traditionnelle par image clé. Ce mémoire s'achève en proposant une stratégie pour comparer efficacement des graphes objet. Dans ce but nous introduisons une mesure de similarité que nous utilisons pour la recherche d'objets.

Contents

Abstract i

Résumé iii

Contents v

List of figures ix

List of Tables xiii

List of abbreviations xv

Introduction 1

1 Modeling video content 5

1.1 Hierarchical modeling in the context of MPEG-7 5

1.1.1 The MPEG-7 framework 6

1.1.2 Description of video shots 14

1.1.3 Example of representation with the MPEG-7 24

1.2 Spatiotemporal representation of video objects 29

1.2.1 Visual processing 29

1.2.2 Basic graph terminology 31

1.2.3 Spatiotemporal volumes 34

1.2.4 Adjacency graphs 36

1.2.5 Hierarchical representation 36

1.2.6 Merging and grouping 38

1.3 Applications of the proposed representation 39

1.3.1 Context 39

1.3.2 Proposed applications 40

1.4 Conclusions 41

2 Video object extraction **43**
2.1 State of the art 43
2.1.1 Classification of existing methods 44
2.1.2 Motion-based methods 45
2.1.3 Spatiotemporal methods 52
2.2 Spatiotemporal representation based on graphs 58
2.2.1 Overview of the proposed method 59
2.2.2 Tree and forest representation 61
2.2.3 Region growing 62
2.2.4 Adaptation to the 2D+T domain 64
2.2.5 Temporal region grouping and spatiotemporal consistency 68
2.2.6 Complexity study 73
2.2.7 Experimental results 75
2.3 Objects of interest 80
2.3.1 Joint use of spatiotemporal volumes and Motion information 80
2.3.2 Motion-based grouping 83
2.3.3 Object detection and matching 85
2.3.4 Workflow for long-term sequences 88
2.3.5 Examples 88
2.4 Conclusions 92

3 Semantic interpretation of video shots **93**
3.1 Contribution of multimedia knowledge base 94
3.1.1 Principle and organization 94
3.1.2 Semantic labeling 99
3.2 Spatiotemporal semantic segmentation 102
3.2.1 Introduction 102
3.2.2 Overview of the approach 103
3.2.3 Video shot decomposition 104
3.2.4 Intra-BOF Processing 104
3.2.5 Inter-BOF processing 109
3.2.6 Results 113
3.3 Conclusion 120

4 Indexing and retrieval with the spatiotemporal representation 121
4.1 Indexing and retrieval . . . 122
4.1.1 System architecture . . . 122
4.1.2 State of the art . . . 124
4.2 Video shot analysis with spatiotemporal volumes . . . 131
4.2.1 Representing video shots with the Vector Space Model . . . 131
4.2.2 Evaluation settings . . . 134
4.2.3 Analysis of the Vector Space Model . . . 136
4.2.4 Comparison between keyframe and spatiotemporal representation . . . 142
4.3 Video shot matching . . . 144
4.3.1 Visual and structural similarities . . . 144
4.3.2 Matching algorithm . . . 146
4.3.3 Experiments . . . 146
4.4 Conclusion . . . 147

5 Conclusions and perspectives 149
5.1 Summary . . . 149
5.2 Perspectives . . . 150

A Mpeg-7 Description Example 153
A.1 Classification scheme (CS) . . . 154
A.2 Semantic Description . . . 155
A.3 Visual and Structural Description . . . 160

Bibliography 167

List of figures

1.1 Scope of the MPEG-7 standard. 6
1.2 Classification of MPEG-7 concepts. 8
1.3 Conceptual modeling diagram for the description of video into segments. . . 9
1.4 UML diagram describing the hierarchy of the MPEG-7 description. 10
1.5 Hierarchy of mpeg-7 segment DSs. 11
1.6 Conceptual modeling for the description of semantic aspects. 12
1.7 Illustration of the Collection DS. 13
1.8 Simplified top-level structure for describing AV content. 14
1.9 Simplified structure for video shot description. 16
1.10 Computation of the Color Structure Descriptor. 18
1.11 Examples of shapes for the RegionShape descriptor 20
1.12 SpatioTemporalLocator descriptor for the description of moving regions. . . 21
1.13 Examples of structural relations between segments. 22
1.14 Construction of a formal abstraction. 24
1.15 Content model for MPEG-7 descriptions. 25
1.16 Proposed example. 26
1.17 Semantic descriptions of the narrative worlds for the example. 27
1.18 Temporal decomposition into events. 27
1.19 Spatiotemporal relationships within the temporal decomposition. 28
1.20 Graph, induced subgraph and spanning subgraph 32
1.21 Example of forest 33
1.22 Matching between graphs. 34
1.23 3D pixel grid over a sequence of frames. 35
1.24 Segmentation and volumes 35
1.25 Frame sequence and the corresponding Attributed Relational Graph. 36
1.26 Hierarchical modeling of video shot with ARG. 37
1.27 Example of merging and grouping. 39

1.28 Architecture of multimedia system 40
1.29 Framework of the thesis. 41

2.1 Classification of spatiotemporal segmentation methods. 45
2.2 General scheme for motion-based segmentation. 46
2.3 Scheme of the overall segmentation process. 60
2.4 Disjoint set data structures and related operations. 61
2.5 A bottom-up segmentation procedure. 62
2.6 Merging of components. 63
2.7 The 2D+T graph segmentation problem. 65
2.8 Spatial segmentation. 67
2.9 Space-time grid based merging. 68
2.10 Feature point matches. 70
2.11 Temporal region grouping. 70
2.12 The temporal region grouping scheme. 71
2.13 Neighborhood subgraphs. 73
2.14 Subgraph matching. 74
2.15 Segmentation results with the 3D algorithm. 76
2.16 Segmentation results with the 2D+T algorithm. 77
2.17 Computing time for the foreman sequence. 79
2.18 Computing time for the tennis sequence. 79
2.19 Memory load for the *tennis* sequence. 81
2.20 Video shot and object representation. 82
2.21 Estimation of motion models. 82
2.22 Motion activity maps. 84
2.23 Velocity-based similarity. 84
2.24 Object propagation. 87
2.25 Moving object detection and propagation for long-term sequences. 89
2.26 Motion detection for the *coastguard* Sequence. 90
2.27 Moving object grouping results for the coastguard sequence. 90
2.28 Moving object grouping results for the close-up sequence. 91

3.1 The ontology architecture. 96
3.2 The Knowledge Assisted Analysis system. 98
3.3 Semantic labeling of image regions 101
3.4 Framework for semantic video segmentation. 103
3.5 Spatial and temporal decomposition of a BOF. 104

3.6 Temporal selection of frames. . . . 105
3.7 Semantic Volume Growing. . . . 109
3.8 Selection of matches. . . . 110
3.9 Matching of dominant volumes. . . . 111
3.10 First best visual matches. . . . 112
3.11 Merging of two BOFs. . . . 113
3.12 Video semantic segmentation: Example 1. . . . 114
3.13 Video semantic segmentation: Example 2. . . . 115
3.14 Video semantic segmentation: Example 3. . . . 116
3.15 Repartition of the running time for different BOF sizes. . . . 119

4.1 Indexing and retrieval of video content. . . . 123
4.2 Illustration of the Vector Space Model. . . . 132
4.3 Video indexing and retrieval system using the Vector Space Model. . . . 132
4.4 Vector Space Model for shot indexing. . . . 134
4.5 Examples of annotated shots. . . . 135
4.6 Analysis of the VSM for the texture modality - Example of the video senses. 137
4.7 Analysis of the VSM for the color modality - Example of the video Docon. . 138
4.8 Exact count density for the random model. . . . 139
4.9 Analysis of the VSM for different categories with 50 visual terms - Example of the senses video. . . . 140
4.10 Analysis of the VSM for different categories with 2000 visual terms - Example of the senses video. . . . 141
4.11 Analysis of the VSM for different categories with 50 visual terms - Example of the Docon video. . . . 141
4.12 Retrieval results for different segmentations 143
4.13 Retrieval results for several dictionary sizes. . . . 144
4.14 Matching between video volumes. . . . 145
4.15 Retrieval performance. . . . 147

List of Tables

1.1 MPEG-7 audiovisual principal concepts in a video shot. 15
1.2 MPEG-7 visual features for the description of video segments. 17

4.1 Visual categories. 135
4.2 Segmentation algorithms. 142
4.3 Average number of regions for the segmentation algorithms. 143
4.4 Building of the similarity measure. 146

List of abbreviations

ARG	Attributed Relational Graph
AV	Audiovisual
BOF	Block of Frames
CBIR	Content-Based Indexing and Retrieval
CBVIR	Content-Based Video Indexing and Retrieval
CS	Classification Scheme
DS	Description Scheme
KAA	Knowledge Analysis System
MPEG-7 XM	MPEG-7 Experimentation Model
MST	Minimum Spanning Tree
OG	Object Graph
RAG	Region Adjacency Graph
ST	spatiotemporal
VSM	Vector Space Model

Introduction

Motivation

Nowadays, the average person is confronted to an impressive amount of digital videos and TV programs, which are in turn stored in constantly growing video databases. How do we efficiently structure raw video content, this tremendous group of pixels, when context knowledge is not available? This is definitely a complex task, that even the human brain cannot cope with as soon as the database become large.

Computer vision scientists usually distinguish *low-level* descriptions that can be extracted automatically by a computer program from *high-level* descriptions that are the one achieved by our visual system, or that require its intervention. In order to simplify the interpretation problem, computer vision considers a segmentation which divides the scene in well distinct visual entities. Automatic segmentation is typically a low-level process as it outputs coherent segments with respect to a set of visual attributes.

Examples of automatic segmentation processes in video analysis are *shot* (also known as *temporal*) segmentation which groups frames representing a continuous action in space and time, or *image* segmentation which groups similar pixels into coherent regions. The main challenge remaining is *object* segmentation, where we would like that the extracted segments correspond to the semantic interpretation of the human brain. This is, as we already said, a very difficult task without prior knowledge on the objects that should be recognized in the scene. Fortunately, objects of interest can be decomposed into a set of coherent regions. The problem is then to find a decomposition which remain robust to spatial and temporal variations affecting the object. Compared to object detection in still images, a precious gain is given by considering spatiotemporal representation of objects. Interesting objects own consequent temporal span and motion. As we will see, taking under account the temporal dimension facilitates the extraction of objects as well as their comparison.

As it brings together low-level and high-level descriptions of video content, extraction of video objects is therefore a major point for applications providing content-based functionalities such as manipulations of objects, video browsing or object retrieval in databases. A flexible and widely adopted content-based representation for indexing video objects has been defined by the MPEG-7 standard. Among the possible representation, objects can be described hierarchically by spatiotemporal volumes which are in turn decomposed of frame regions. Each of these levels can be described by a graph with visual or semantic attributes expressing exhaustively the external and internal relationships between objects.

In this dissertation, we will use graph representation and algorithms for building, comparing and indexing video segments at different level of interaction, from raw pixels to complex objects. Benefits are both the powerfulness of the approach and a construction of a visual description that can be directly used by multimedia applications.

Background

Segmentation of video objects is a delicate issue in video analysis. An indispensable prerequisite is to subdivide first the sequence into *shots*, where one shot represents a continuous action in space and time.

Compared to other domains, for instance speech segmentation into phonemes, the mathematical model required for the extraction of a video object is difficult to define. The most accurate methods which have been developed until now necessitates user interaction to select the objects [76]. As manual segmentation becomes impossible when the duration of the shot is important, *semi-automatic* methods have been proposed to alleviate the user's task. In a first approach, the user delimits the object area with respect to the background area, which enables to initialize an object model. This initialization stage is followed by tracking in the following frames, the update of the model and the localization being performed simultaneously. In a second approach, the user interferes to validate or edit the segment boundaries provided by an automatic segmentation algorithm. In this way, the precision of the object boundaries is maintained during the sequence. The semi-automatic approach is highly suitable for extracting a particular object from a shot and gives nowadays excellent results.

In this thesis we address the case where the user does not supervise the segmentation. For that, automatic approaches have been proposed. With the recent development of MPEG-4 coding standard, focus have been put on extracting moving objects on the scene, resulting in a *video object plane* (VOP) decomposition. Current techniques developed are based on spatial segmentation (for an image) associated to motion tracking. Motion estimation is then either performed based on spatial regions or inversely motion vectors are used to extract the support of the objects [129, 85].

According to Gestalt theory on the perceptual organization of objects, the recognition would be performed conjointly in spatial and temporal domains [85], assembling and interpreting *spatiotemporal* structures. Following these organization principles, spatiotemporal methods process the whole set of frames in the shot, requiring to store and process a considerable amount of data. Thanks to huge progress in computational resources, these approaches have nevertheless gained in interest from the past few years. Primitive proposed include video blobs or moving patches. The assembling of extracted structures defining the objects and organization of the shot. Besides these 3D methods, we focus in this dissertation on the spatiotemporal approach and investigate efficient procedures to create spatiotemporal structures.

In the process of automatic video analysis, segmentation and extraction of objects does not always constitute the main target, but rather an intermediary step towards scene understanding and indexation of video shots. Unfortunately, the problems of creation of

spatiotemporal objects and indexing have been treated separately until now. In this thesis we try to fill this gap and propose a framework for segmentation of spatiotemporal segmentation and indexing of video shots and objects. We will see in the next section how this framework is designed and our contributions.

Contributions and Outline

The material in this thesis is organized into 4 chapters. First, the modeling of video shot and the overall framework is introduced. Then, the procedure for extracting objects using spatiotemporal segmentation is detailed. Applications to semantic annotation, and indexing of video shots are described afterwards. Experimental results are presented all along the thesis for the spatiotemporal segmentation and each application.

Following this introduction, the thesis outline is as follows :

Chapter 1 describes a representation of video shot content based on the MPEG-7 standard. In this context, we define a model for the spatiotemporal representation of video objects. The chapter ends by a short introduction of the applications in semantic annotation and video indexing.

Chapter 2 addresses the problem of segmentation into objects. It begins with a state of the art of the existing methods, comparing motion-based and spatiotemporal approaches. Afterwards, the basics of a spatiotemporal segmentation algorithm is introduced and its properties are explained on real video sequences. The construction of high-level moving objects from spatiotemporal regions using motion models is investigated and experimental results are presented.

In chapter 3, we take advantage of spatiotemporal segmentation to elaborate an application targeting semantic annotation of video shots. For this purpose we exploit a multimedia knowledge base developed by ITI [11] which enables semantic labeling of image regions. Spatiotemporal regions allows to propagate semantic information within short video segments. Semantic annotation of the shot is then performed by matching iteratively video segments while reevaluating semantic and visual properties of spatiotemporal regions.

Chapter 4 is dedicated to the problem of video shot retrieval and indexing from the spatiotemporal approach. We show the advantages of spatiotemporal descriptions in shot analysis using an indexing method based on visual signatures. Then we introduce new techniques for search of video objects based on the matching of visual and structural properties of spatiotemporal regions.

Finally, concluding remarks and perspectives are presented in chapter 5.

Chapter 1

Modeling video content

The growth of multimedia applications has led to the increased need to develop tools to access and search information. In particular, video data is among the most complex to represent, because of the diversity of the content. Nowadays the MPEG-7 standard provides great flexibility for the description of multimedia content. This format enables to organize hierarchically video content into scene, shots, objects and regions. Objects are a key element of the representation, linking low-level descriptions to high-level semantic interpretations. MPEG-7 is valuable as well for low-level applications such as edition and manipulation, and for specialized applications, such as event detection in medical or sports domains. The former will require a visual description of objects, while the latter will focus on the visual and semantic relationships between the different objects composing the scene.

MPEG-7 defines a norm of representation of these relationships in form of a graph. Graph-based representations are in the core of the extraction process and a major tool to describe the relationships between video entities inside the shots. A noticeable advantage of these representations is that they enable to analyze relationships independently of the type of properties considered for the object.

In this chapter, we first introduce a description of video shots and objects based on the MPEG-7 standard. Then we propose a framework based on adjacency and attributed graphs for the spatiotemporal representation of video objects and the organization of video shots. Finally, we will mention different possible applications and introduce those we have developed in the thesis.

1.1 Hierarchical modeling in the context of MPEG-7

The number of audiovisual documents available in a digital form has grown exponentially during the last decade. Meanwhile, managing and searching content provided from multiple sources has become more difficult, since each broadcasting organism has its own proprietary way to describe and index the content.

MPEG-7 standard has emerged as a potential solution to this problem, defining a flexible framework to organize multimedia content. In this section, we describe the guiding principles of the standard and show an object representation can be decomposed and exploited

using the tools provided by the standard. We then illustrate the assets of the MPEG-7 framework by constructing a complete description for a given example, dealing with both semantic and structural aspects of the content.

1.1.1 The MPEG-7 framework

Goal and Scope of the Standard

MPEG-7 is a specification of "Multimedia Content Description Interface". This standard for representation of multimedia metadata has been first released in 2001. The first objective of MPEG-7 is to provide interoperability between different multimedia systems and applications, so that different systems can exchange content description. The second target is to help users and applications to search and filter content thanks to a set of standardized tools and hierarchical description of the content [25].

Fig.1.1 shows the scope of the MPEG-7. The standard does not specify the means for content extraction, but defines structures and tools for describing and managing audiovisual (AV) content for a wide range of applications. Main application types are filtering of broadcast media, personalization and interactive television, search in multimedia databases and over the Internet. Application domains are even wider, including biomedical imaging, distant education, e-commerce and many more. For instance, MPEG-7 can provide the descriptions for playing a few notes for the instrument and retrieving the tune, define a set of objects with particular relations, retrieve examples from a library and select those the user is interest in, or simply select and navigate within audiovisual programs from user preferences.

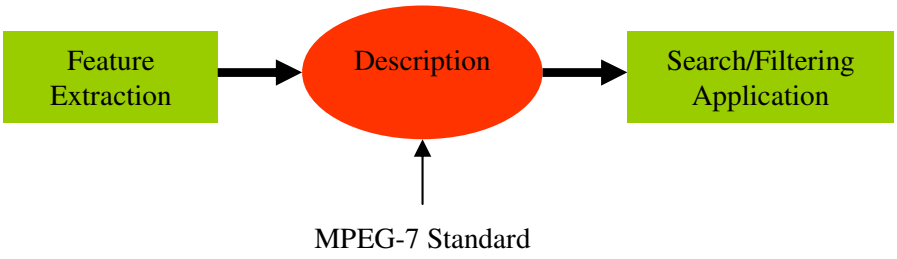


Figure 1.1: Scope of the MPEG-7 standard.

The MPEG-7 conceptual model

To fulfill the requirements of all different applications and scenarios, MPEG-7 has to manage many aspects of the AV content description. One of the main challenge of MPEG-7 was to gather research communities, each one following its own beliefs and technical insights. Indeed, database researchers argue that only a standardized structure and linking mechanisms for high-level descriptions is necessary, whereas signal processing community aims

to define standardized features as for describing and exchanging the representation of AV content [95].

Aspects covered by the standard includes :

- Creation and Production (Title, creator, classification of production).
- Usage (Access, publication rights, ...).
- Media (Storage format, encoding).
- Structural aspects (spatial, temporal or spatiotemporal components of the content with audiovisual and semantic descriptors.)
- Conceptual aspects (objects, events, interaction among objects).
- Collections (sets of related objects).
- User Interaction (register user action and preferences).

Conceptual modeling [117] is in charge of building a high-level model of the audio-visual domain based on content description requirements. This approach uses notion from both Entity-Relationships (ER) and Object-Oriented (O-O) modeling, two methodologies leading an important role for both database and software design. The preliminary step in conceptual modeling is to identify the different elements of the domain, or *principal concepts*. The principal concepts results from the decomposition of audiovisual content into different levels of description, such as structure, semantics, features and meta-data. Examples of concepts from these levels are regions, segments, spatiotemporal organization, objects, events, color, texture, shape, motion, title, author. Figure 1.2 lists the available concepts in function of these categories and their type.

Following E-R modeling, principal concepts are further classified into modeling constructs as follows :

- Entities: A principal object in the AV domain. *Shot*, *objects*, *events* are example of entities.
- Relationships: An association among one or several entities. There are three main relationships:
  - Generalization specifies an *is-a* relationship that partitions a class of entities into mutually exclusive subclasses. For instance, a *key-frame is a frame*, and a *shot is a segment*.
  - Aggregation specifies an specifies a *has-A* assembly-component relationship of entities. As an example, a *video has* multiple *shots*, and a *frame is composed of regions*.
  - Association specifies a relates two or more entities which do not exhibit existence dependency. For instance, a *segment depicts* an *event*, and *regions are associated to* other *regions* in an image.

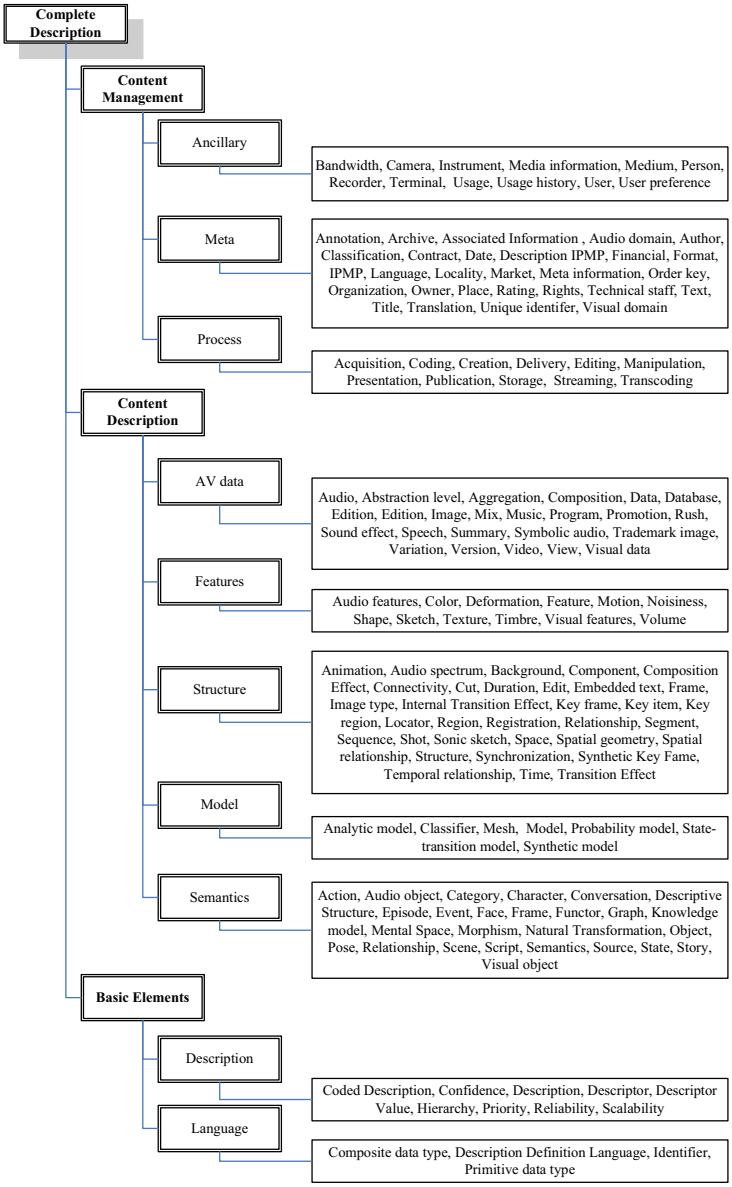


Figure 1.2: Classification of MPEG-7 concepts. Content management regroup information on creation, media coding and storing, and content usage. Content Description describes the structure of the AV content at different levels: AV data, Features, Structure, Model and Semantics. Basic elements covers modeling and construction of MPEG-7 descriptions.

- Attributes: Descriptive information about an entity or relationship. As an example, *color*, *texture* or *motion* are properties of the entity *object*.

An example of these constructs illustrating the decomposition of video into segments is shown fig.1.3.

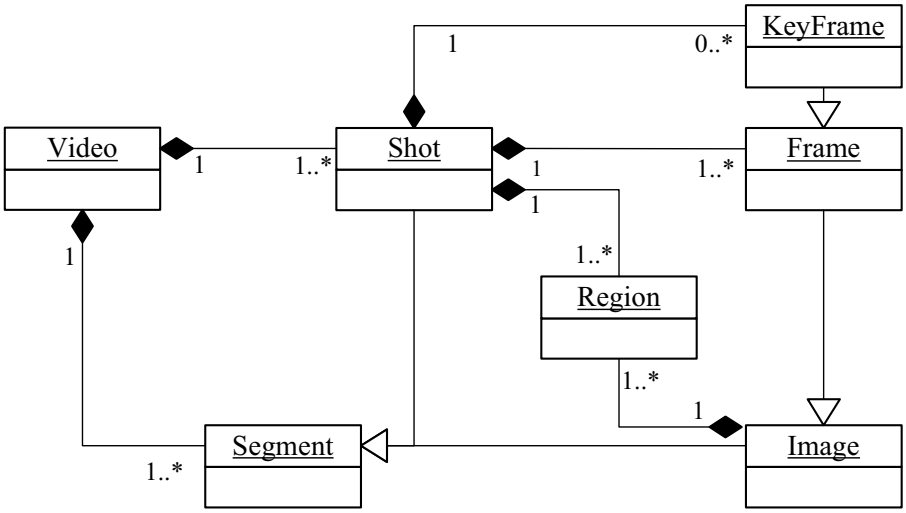


Figure 1.3: Conceptual modeling diagram for the description of video into segments.

Construction of MPEG-7 descriptions

Once established, conceptual models are transformed automatically into appropriate structures for content description, called *Description Schemes* (DSs). The construction of a basic MPEG-7 description is illustrated by the conceptual (UML) diagram of fig.1.4. In MPEG-7 terminology elements of raw audiovisual content are referred as *Data*, regardless of the support or technology. It is sufficiently broad to encompass a large set of media types such as graphic, images, video, music, text, or multimedia source on the web. Data is characterized by distinctive *Features* that signifies something to the end user or the system. Each feature is described by one or several *Descriptors* (Ds) that defines the syntax and the semantics of the feature representation, and enables to compare AV content from their *descriptor value*. For instance, possible descriptors for the color feature are the histogram of the r,g,b components or more simply their mean.

The description of the content is structured from description schemes. A description scheme specifies the structure and the semantics of relationships between its components, which may be both descriptors and description schemes. A descriptor contains only a basic data type, whereas a description scheme can include another DS. Finally, a description scheme and its set of descriptors values forms a *description* of the data.

The *Description Definition Language* (DDL) provides means to structure descriptors into description schemes. It expresses both spatial, temporal, and conceptual relationships

between objects and provides a rich model for links of references between the descriptions and the described data. XML Schema Language as been selected by the MPEG consortium as the MPEG-7 DDL, adding a few specific extensions. The language naturally describes nested structure and allows interoperability. By authorizing modifications and creation of new DSs, DDL ensures extensibility of the standard for different application domains.

In practice MPEG-7 descriptions are in general stored and transported in a binary coded representation (Bim). MPEG-7 Systems specify the functionalities to store, transport the description and its synchronization with the data stream.

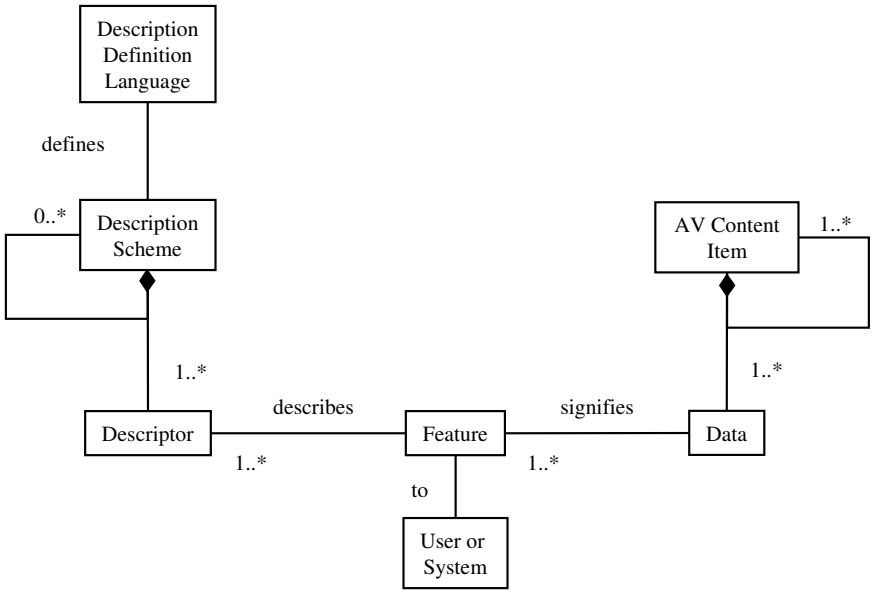


Figure 1.4: UML diagram describing the hierarchy of the MPEG-7 description.

MPEG-7 Description Tools

A large set of predefined descriptors and description schemes using the MPEG-7 DDL have been included in the standard, referred as *Description Tools*. The standardized description tools make up three of the six parts of the MPEG-7 standard, namely part 3 (Visual)[89], part 4 (Audio)[90], and part 5 (Multimedia Description Schemes)[91].

Description tools use generic basic elements extending the XML Schema Language. Schema Tools assists in the packaging and annotation of MPEG-7 descriptions. The tools include numerical and string data types (values, ...), links and locators (time, ...), references to avoid duplicating the description, and textual annotations. These annotations are either flexible (FreeText) or structured (useTerms) by including fields corresponding to a set of questions (Who?, What Object?, What Action?, Where?, When?, Why?, How?). It is further possible to define controlled terms with Classification Schemes (CS), which enables to standardize vocabularies for different application domains.

Audio and Visual description tools are used to describe audiovisual content. The visual descriptors allow to search and filter image and videos based on *features* like *Color*, *Texture*, *Object Shape*, *Object Motion* and *Camera Motion* or *Location*. Similarly, audio tools enables to search and filter spoken and music content based on features like *spectrum*, *harmony*, *timbre* and *melody*.

The structure of an audiovisual document itself is described by *Multimedia Description Schemes* (MDS) tools which define metadata structures for describing and annotating audio-visual (AV) content. An MPEG-7 description can cover both content description and content management aspects. Complex descriptions can be split into different *description units*. For instance a description unit can describe a particular descriptor, object or abstraction.

The core element that describes structural aspects of content is the Segment DS. This description scheme is an abstract base type representing an arbitrary physical or logical section of audiovisual content like audio segments, video segments, audio-visual segments, moving regions, and still regions. Common information to all types of segment relates to creation information, usage information, media information and textual annotation.

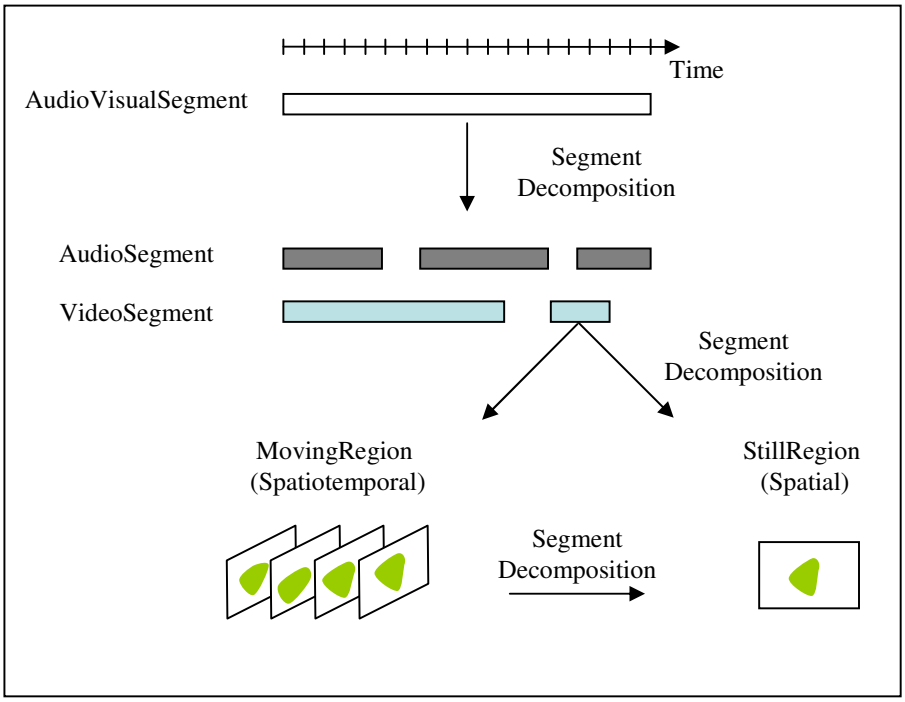


Figure 1.5: Hierarchy of mpeg-7 segment DSs.

A segment may include both spatial and temporal properties. A temporal segment may be a set of samples in an audio sequence, represented by an AudioSegment DS, a set of frames in a video sequence, represented by a VideoSegment DS or a combination of both audio and visual information described by an AudioVisualSegment DS. A spatial segment

may be a region in an image or a frame in a video sequence, represented by a StillRegion DS for 2D regions and a StillRegion3D DS for 3D regions. A spatiotemporal segment can correspond to a moving region in a video sequence, represented by a MovingRegion DS or to a more complex combination of visual and audio content for example represented by an AudioVisualRegion DS.

Segments are not necessarily connected, in both spatial and temporal domains. A temporal segment (VideoSegment, AudioSegment or AudioVisualSegment) is said to be temporally connected if it represents a sequence of continuous video frames or audio samples. A spatial segment (StillRegion) is said spatially connected if it is composed of a group of connected pixels. A spatiotemporal segment (MovingRegion) is said spatially and temporally connected if the temporal segment is temporally connected and if each one of its temporal instantiations in frames is spatially connected.

A recursive or hierarchical decomposition of the AV content into segments that form a segment tree can be described through the SegmentDecomposition DS. The SegmentRelation DS describes additional spatiotemporal relationships among segments. This decomposition of AV content into segments is illustrated fig.1.5.

Besides the description of the content structure, the Segment DS enables to represent conceptual aspects. The Semantic DS provides a description of the narrative world in terms on Events, Objects, Concepts, Places, Time in narrative worlds. Figure 1.6 shows the conceptual modeling for the description of a narrative world.

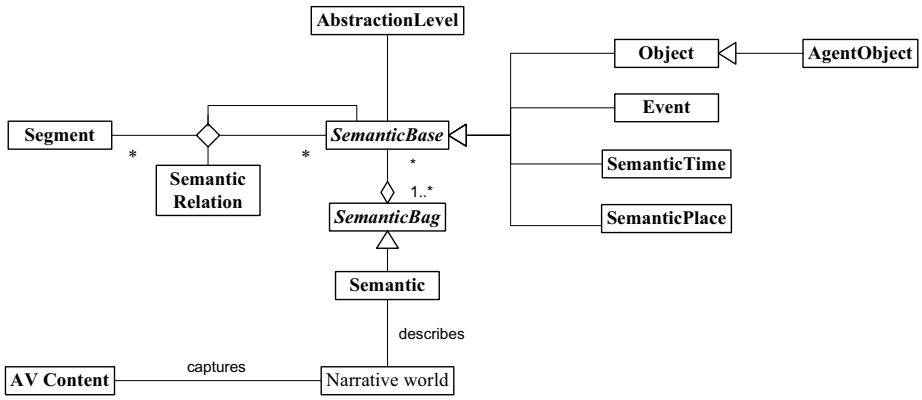


Figure 1.6: Conceptual modeling for the description of semantic aspects captured by the AV content.

Semantics depicts information relating to the underlying meaning or understanding of the audio-visual content. Each segment structuring the AV content can depict semantics. Object, Event, Place and Time are the available semantics. The conceptual aspect of the AV content can be further structured as a graph by specifying the relationships between the notions.

The core description scheme associated to semantics is the SemanticBase DS, a generic abstract type to describe the aforementioned semantic notions. It can be instantiated into

Object, Event, SemanticPlace and SemanticTime DSs. An object which performs an action, like a person, organization, animal, is extended to an AgentObject DS. Finally, SemanticPlace and SemanticTime DSs describe, respectively, a place and a time in a narrative world.

The Semantic DS contains all the semantic descriptions of an audiovisual item. As for segments, semantics are organized in a graph or tree. The SemanticRelation DSs describes the edges between semantics.

Other forms of content organization is provided by collections and models. A collection groups AV of the same type, including segments, descriptor, or semantics. Relationships between collections are expressed with the CollectionStructure DS. An example of collection is given fig.1.7. To link features and content to semantic, the audiovisual content, descriptors and collection are described by a model. ProbabilityModel DS describes different statistical functions and probabilistic structures to describe the AV content data. The AnalyticModel DS is a collection of examples of AV content or clusters of descriptors used to provide a model for a particular semantic class. The Classifier DS describes different types of classifiers that are used to assign the semantic labels to AV content or collections.

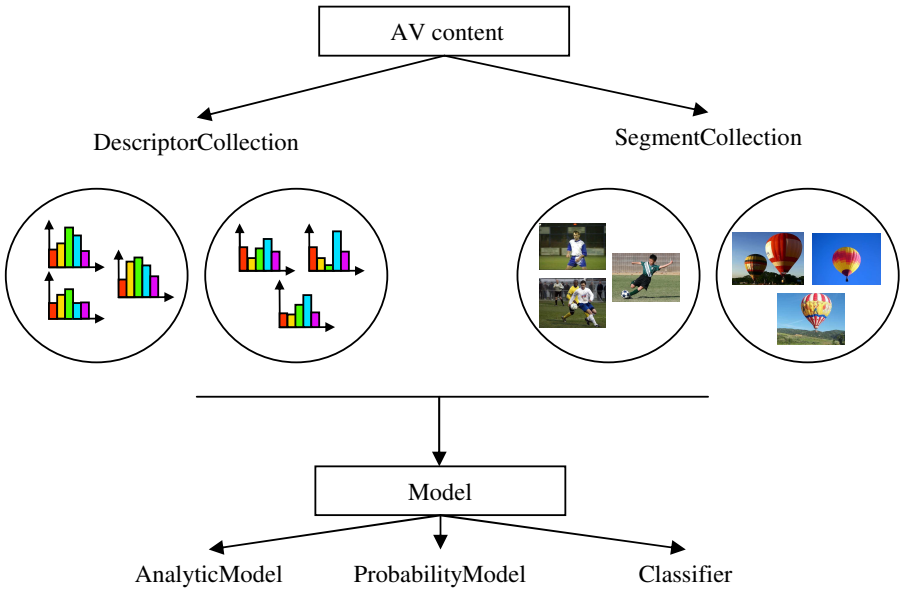


Figure 1.7: The Collection DS enables to organize various types of collections. Here we illustrate collection of descriptors (DescriptorCollection DS) and images (SegmentCollection DS). Relationships within an between collections can also be expressed, for instance a similarity measure between two descriptors or two collections.

The last set of tools are related to the end-user interaction with a multimedia system.The User Interaction DSs describe user preferences and usage history pertaining to the consumption of the AV content. This allows, for example, matching between user preferences and MPEG-7 content descriptions in order to facilitate personalization of audio-visual content

access, presentation and consumption.

1.1.2 Description of video shots

Temporal decomposition of video

In our context, we consider movie or TV program as the audiovisual content. Video is usually decomposed temporally in scenes, shots and keyframes. A shot depicts a continuous action in time and keyframes are representative frames within the shots. Shots can be further grouped into a scene when their semantic content is related. Shots within a scene may or not be visually close to each other. For instance a scene may be composed of a sequence of shots where the action of the character is taken successively from different views.

It should be noticed that MPEG-7 description tools offers a large degree of freedom for structuring AV content, which increases the complexity of understanding and interoperability between applications. To simplify the problem, the MPEG-7 schema has been extended to several *profiles*, which imposes constraints on the elements of the description. The profile with the largest scope for AV content is the Detailed AudioVisual Profile (DAVP) [12]. The top-level structure for AV content description within this profile is shown fig.1.8.

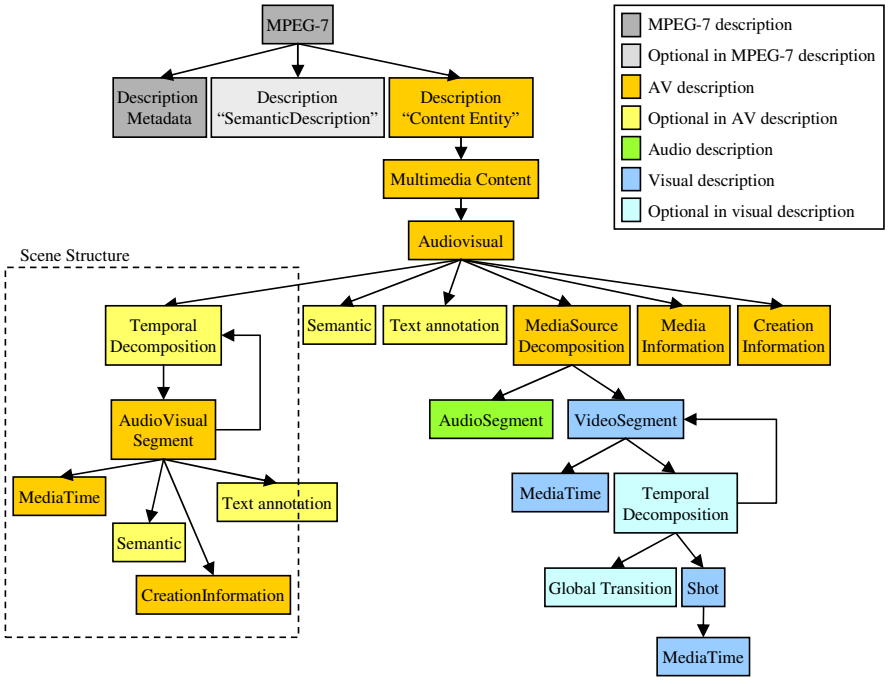


Figure 1.8: Simplified top-level structure for describing AV content.

At a first level, the description includes information about its identification and creation (DescriptionMetadata). Various types of descriptions my be used in MPEG-7. SemanticDe-

Principal Concept	Definition	Type	Description	Related concepts
Scene	An episode or sequence of events representing continuous action in one location	Semantics	Segment DS	Segment, location, event, dialog, shot, story, character, mosaic, camera motion
Shot	The temporal unit of video between cuts	Structure	Segment DS	Cut, segment, acquisition, scene, Bridging shot, long shot, close-up, take
Frame	A single image from a video sequence	Structure	VideoSegment DS	Image, shot, segment Key frame
Region	A connected group of pixels in visual data	Structure	Object Bounding Box, Spatio-temporal Region, StillRegion DS, MovingRegion DS	Segment, image, object. Partition
Object	Descriptions of real objects, composites or abstractions of these objects	Semantics	Object DS, Concept DS, ConceptObject DS	object behavior, semantics, region, source, audio object
Semantics	Information relating to the underlying meaning or understanding of audiovisual content	Semantics	Semantic DS	Knowledge model, object, event, character, story, source

Table 1.1: Elements of MPEG-7 audiovisual principal concepts list intervening in a video shot.

scription is used to describe conceptual aspects appearing (or captured) by the AV content. As several description may share common semantic entities, semanticDescription stands as a separate description that can be referenced in other description of contents. ContentEntity describe the structure of AV content. The media can then decomposed temporally according to audio and visual modalities. The visual content is described by a recursive structure of VideoSegment and TemporalDecomposition DSs. Each video segment is decomposed temporally into shots and optionally transitions. Structures containing both audio and video information, such as a scene structure are described by an AudioVisualSegment DS. Each segment of the temporal decomposition (Audio/Video, AudioVisual, Shot) is localized by a timing structure MediaTime describing its start time and duration, and can be optionally annotated with the TextAnnotation DS or described by its semantics.

Structure of a video shot

A video shot is characterized at different level of understanding. At the semantic level, a shot represents objects in a real world scene, performing some action. At the syntactic level, the shot is composed of AV data structured into segments (spatiotemporal regions, moving regions) and described by some features.

In the MPEG-7 conceptual model, principal concepts include Scene, Shot, Frame, Re-

gion, Object and Semantics. The description of these concepts and their corresponding DS is listed in table 1.1.

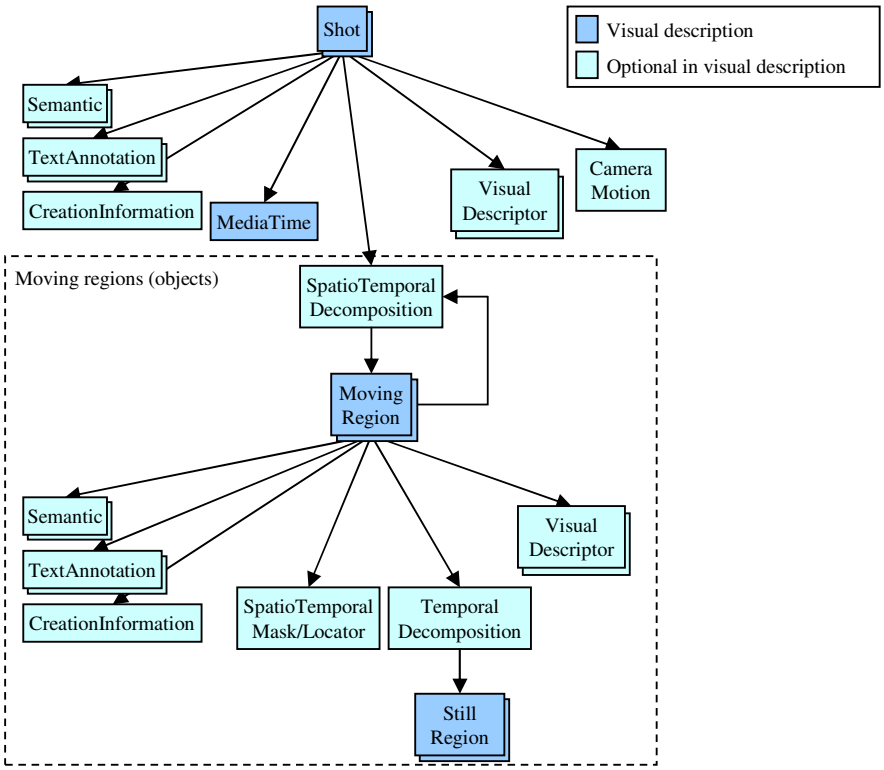


Figure 1.9: Simplified structure for video shot description.

Figure 1.9 illustrates the description of a video shot. The Shot DS extends the Segment DS, conveying TemporalLocation (MediaTime), CreationInformation, TextualAnnotation or Semantics. It can be also depicted by global elements such as camera motion, image or group of pictures (GoP) visual descriptors. The SpatioTemporalDecomposition DS, which implements the abstract SegmentDecomposition DS, decomposes the shot into moving regions. The MovingRegion DS describes a spatiotemporal region of a video and enables to represent visual objects that are detected in the sequence (object, person, face, ...). One moving region is located by a SpatioTemporalLocator DS describing its trajectory and its location (bounding box, polygon). Spatiotemporal decomposition may be recursive, complex objects being further decomposed into several moving region elements. It should also be noticed that a moving region is not necessarily connected in space or time. Objects are described naturally on a per-shot basis. The association between appearances of the same object in different shots can be managed by referencing the entity being represented with a Semantic DS. Inversely, appearances of one object within the video can be referenced within the semantic description with the MediaOccurrence DS, containing location and visual features of the objects at one moment of the sequence. Video segment description includes visual and

Feature	Video Segment	Still Region	Moving Region
Color	X	X	X
Texture		X	
Shape		X	X
Motion	X		X
Camera Motion	X		
Time	X		X

Table 1.2: MPEG-7 visual features for the description of video segments.

semantic aspects. These descriptions are explained in the following sections.

Visual description

Numerous descriptors have been proposed in the past to describe the video content. The selection of descriptors has been for long based on the targeted application and domain [80, 108]. MPEG-7 standard has benefited from this effort to propose a set of basic visual descriptors which can serve for a wide range of applications. Although these descriptors are not necessarily optimum as verified in[37] they have been shown to be reasonably efficient and robust for content analysis tasks [81, 99, 101, 61, 26].

To describe a video segment, MPEG-7 MDS reports 6 feature types, widely used in the context of image/video indexing and retrieval. Table 1.2 lists these features according to the specific segment they apply to. A shot segment can be globally described by the camera motion, and color (GoF/GoP color descriptor). Shape, color and texture features describe (spatiotemporal) regions; and the motion feature is specific to the MovingRegion DS. A brief description of the visual descriptor and their use in context of the thesis is presented below.

Color descriptors

Color is one of the most used features in image and video retrieval systems. The MPEG-7 visual part defines four descriptors to capture color distribution, spatial layout and spatial structure of the color in natural images and videos [81]. The histogram-based descriptors (ScalableColor, ColorStructure) captures global distribution of the colors. DominantColor gives the distribution of the salient colors in the image. ColorLayout captures the spatial layout of the colors in a compact representation. To ensure inter-operability, the descriptors are defined for one or a limited number of color spaces. MPEG-7 supports four color spaces, namely RGB, YCbCr, HSV, HMMD, or Monochrome. The HMMD space is composed of the Hue (H), the maximum (max) and the minimum (min) of RGB values, and the difference between the max and min values. Descriptors are preferentially defined in non-linear spaces (HSV, HMMD) closely related to human perception of color. We focus on the histogram-based descriptors which have been widely used for image and retrieval applications. All descriptors are valuable as they have been extensively selected to have good recall rates, but they are highly redundant [37], and therefore statistically dependent of each other.

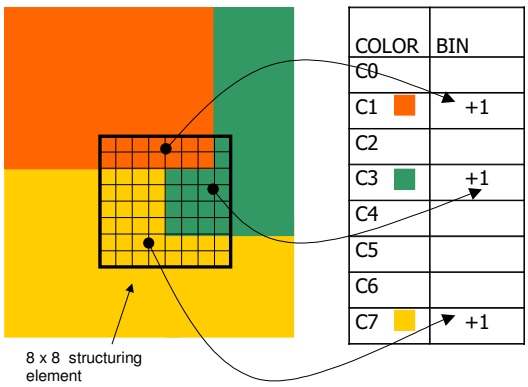


Figure 1.10: Computation of the Color Structure Descriptor. Bin values are incremented by visiting a 8x8 structuring element. *Extracted from [91].*

ScalableColor Descriptor (SCD) performs color histogram in HSV color space at different level of quantization. The Haar transform is used to reduce the number of bins of the original histograms with 256 bins to 16, 32 or 64 bins. The distance between two descriptors can be very efficiently computed in the Haar Coefficient domain by counting the number of bit positions where the coefficient signs are different (Hamming distance). This reduce the computational cost of similarity matching while maintaining a marginal loss precision.

ColorStructure descriptor (CSD) describes both the spatial structure of the colors in an image or region as well as their frequency of occurrence. The descriptor is represented as a 256 bin histogram in a special color space (HMMD) and is calculated by visiting a structuring element (usually a 8x8 square) at each location : the bins corresponding to the colors that appears in the structuring element are incremented. It is finally normalized by the number of positions occupied by the structuring element, which depends of the image/region size, and non-uniformly quantized. The accumulation process is illustrated fig.1.10.

Texture descriptors

Texture is another important feature in image analysis and retrieval which is particularly suited to describe region properties.We have retained two MPEG-7 texture descriptors. The first one, the homogeneous texture descriptor (HTD) provides a quantitative characterization of homogeneous texture regions for similarity retrieval. The second one, the local edge histogram descriptor (EHD) is useful when the region is not homogeneous in texture properties.

HomogeneousTexture is based on the local spatial frequency statistics of the texture.

$$HTD \equiv [avg, std, e_1, \cdots, e_{30}, d_1, \cdots, d_{30}] \tag{1.1}$$

The descriptor is composed on the mean avg and deviation std of the region pixel intensities, followed by the mean energy e_i and energy deviation d_i computed by filtering the image

with a bank of 30 Gabor filters (6 frequency times 5 orientation channels). A distance function is also defined by a normalized weighted L_1 norm over the frequency channels :

$$D_{HT}(HT_1, HT_2) = \sum_{i=1}^{N_{HT}=62} \left| \frac{HT_1(i) - HT_2(i)}{\sigma_i} \right| \tag{1.2}$$

σ_i is a normalization value for the i$^{\text{th}}$ component of the descriptor, which can be computed by the standard deviation of $HT(i)$ over the complete database [81].

Edge histogram describes the spatial distribution of 4 directional edges (vertical, horizontal, diagonal with direction 45 and 135 degrees) and one non-directional edge. The image is partitioned into 16 sub-images. The number of blocks in each sub-image is kept constant by adapting the block size for the sub-image. For each block and each type of edge, an edge strength is computed by applying a 2×2 edge filter on the average intensity values in 4 sub-blocks. A block is considered to contain a type of edge if the edge strength exceed a minimum threshold value. In total, a local histogram of 80 bins B^l is formed from the edge decomposition of the sub-images (16 sub-images, 5 types of edges). This descriptor can be improved for retrieval tasks by adding global and semi-global levels of localization of an image [101]. The global edge histogram B^g summarizes the distribution of the edges in the whole image and is computed by cumulating the local histogram for each type of edge (5 bins). The semi-global edge histogram B^{sg} is obtained by cumulating the local histogram on 13 subsets of sub-images : rows (4), columns (4), corners (4) and center (1). Bin counts are further normalized by the total number of image blocks in the sub-image and non-linearly quantized. The resulting vector with 150 coefficients can be used after reconstruction for similarity retrieval with the following distance measure.

$$D_{EHD}(B_1, B_2) = \sum_{i=0}^{79} |B_1^l(i) - B_2^l(i)| + 5 \times \sum_{i=0}^{4} |B_1^g(i) - B_2^g(i)| + \sum_{i=0}^{64} |B_1^{sg}(i) - B_2^{sg}(i)| \tag{1.3}$$

Since the number of bins of the global histogram is relatively smaller than that of local and semi-global histograms, a weighting factor of 5 is applied in eq.1.3.

Shape descriptors

Shape as played an important role in searching for image objects. However, shape features are less developed than their color and texture counterparts because of the inherent complexity of representing shapes and depends on the quality of the shape extraction process. MPEG-7 supports both region-based and contour-based shape descriptors. Here, we retain the RegionShape descriptor, which has been demonstrated to perform well on many type of media content [37]. This descriptor expresses the pixel distribution of the shape of a region or a complex object using the Angular Radial Transform (ART) [89]. A set of 36 basis function is used with various angular and radial frequencies. The feature vector is made of the ART coefficients, normalized by the magnitude of the largest coefficient and non-uniformly quantized to 4 bits. The descriptor can be applied to any arbitrary region, is compact (35 bins) and has been shown robust to noise. Examples of shapes that the descriptor can distinguish are given in fig 1.11.

Figure 1.11: Examples of shapes that can be described with the RegionShape descriptor. *Extracted from [91].*

Motion descriptors

Four descriptors characterize various aspects of motion: camera motion, motion trajectory, parametric motion, and motion activity in MPEG-7, which characterize 3D camera motion parameters, temporal evolution of key points, the motion of regions, and the intensity or pace of motion, respectively.

The CameraMotion descriptor characterizes 3-D camera motion parameters within a temporal segment. The camera operation described includes Panning, Tilting, Rolling, Zooming and Static.

The ParametricMotion descriptor depicts the motion of objects in video sequences, as well as global motion, with a parametric motion model. A powerful and simple model is given by the affine motion model.

$$\begin{bmatrix} v_x(x,y) \\ v_y(x,y) \end{bmatrix} = \begin{bmatrix} a_0 \\ a_3 \end{bmatrix} + \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{1.4}$$

The 2D *apparent* motion is obtained by orthographic projection of a 3D planar surface undergoing an affine motion. Compared to more complex parametric models, the affine parametric model achieved a low motion estimation error (on the total region area) in most situations, notably when the region area is small [68]. To estimate global motion in a frame, a robust method based on iterative dominant motion estimation is used [98]. Concerning motion models of regions, we use the method of [120], which has been used in the MPEG-7 XM experiments. The approach consists in estimating roughly the global displacement of the region (a_0,a_3), and then compute a local optimum for the remaining rotation parameters (a_1, a_2, a_4, a_5). In chapter 2, global motion is estimated to compensate camera motion in a sequence of frames. We prefer using ParametricMotion descriptor instead of CameraMotion descriptor as the compensation can be done directly from the parametric motion model.

Localization

Locating efficiently a region or object in also important in search and retrieval. The RegionLocator descriptor gives a brief description of the localization of a spatial region with a bounding box or a scalable polygon. SpatioTemporalLocator extends this descriptor to moving regions in multiple frames by one or several sets of a reference region and its motion (fig.1.12). Rigid moving regions are in general described by the ParameterTrajectory

descriptor referencing a set of reference regions and their motion model parameter along time. When the motion is non-rigid, FigureTrajectory is used, describing the trajectories of a set of points (e.g. a polygon) of the region. These descriptors are particularly efficient for hypermedia applications.

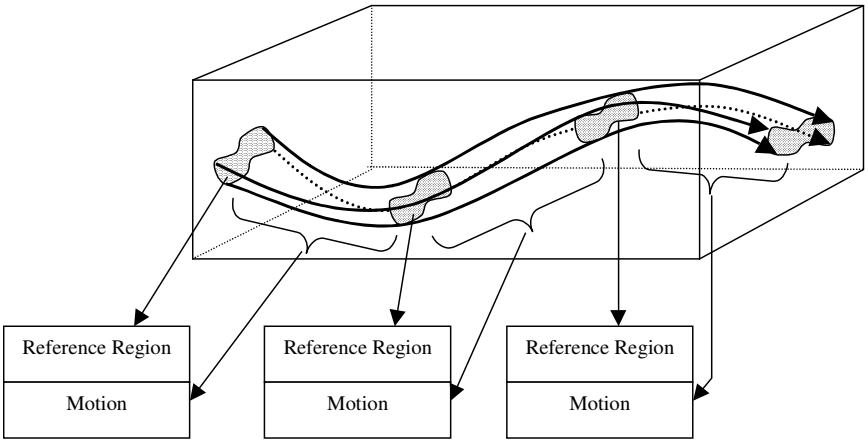


Figure 1.12: The SpatiotemporalLocator descriptor localizes a moving region with a set of reference regions and their motion. *Extracted from [91].*

A very useful and concise information on the localization and geometry of a spatiotemporal region is given by the Center Of Mass descriptor (CM)

$$CM = \{p, c_x, c_y, c_t\} \tag{1.5}$$

where p is the percentage of occupation of the region in the shot, (c_x, c_y, c_t) its coordinates relative to the frame size and shot duration.

Finally, the description of the complete region mask is not included in the standard, but a reference to a labeled image can be added in the StillRegion description [12].

Structural relations

The structural description of a video segment can be characterized in a hierarchical structure with SpatialDecomposition and TemporalDecomposition DS, as we mentioned previously in the section on the overview of the description tools. However, a tree does not depict the spatiotemporal relationships between the different objects. The Graph DS and SegmentRelation DS are used to describe a graph structure between segments. Nodes correspond to segments (e.g. MovingRegion) and the edges to relationships between segments. A relation is either quantitative, using a specific vocabulary or qualitative by specifying the strength of the relation (a value between 0 and 1). Two classification schemes are included in the standard, describing temporal and spatial relations, the TemporalRelation and SpatialRelation CS, respectively.

Temporal relations describe the relationship between two multimedia objects with respect to time. The topic of relations between temporal intervals has been first addressed in [5], which contains a definition of a complete set of temporal relations between two actions. These relations are: *before*, *during*, *overlaps*, *starts*, *ends*, *equal* and their inverses (except for *equal*). Some of these relations are illustrated in fig.1.13 (a).

When defining a spatial relation between regions, each argument is represented by the minimum bounding rectangle (MBR) of the region, as shown fig.1.13(b). Spatial relation can be grouped into: a directional relation (*south*,
relnorth, *west*, or *east*), a mixed directional relation (*northwest*, *northeast*, *southwest*, or *southeast*), or a positional relation: *left*, *right*, *above*, *below*, or *under*. The above relations are obtained by combining Allen's interval temporal logic on (x, y) spatial coordinates. MPEG-7 defines the very basic spatial relations; a refined vocabulary for spatial relation between objects has been proposed in [70], defining *overlap*, *contains*, *touch*, *disjoint* for spatial intervals.

In the general case, when the relations are insufficient to describe the interaction between two objects in the application domain, specific relations are created in a new classification scheme.

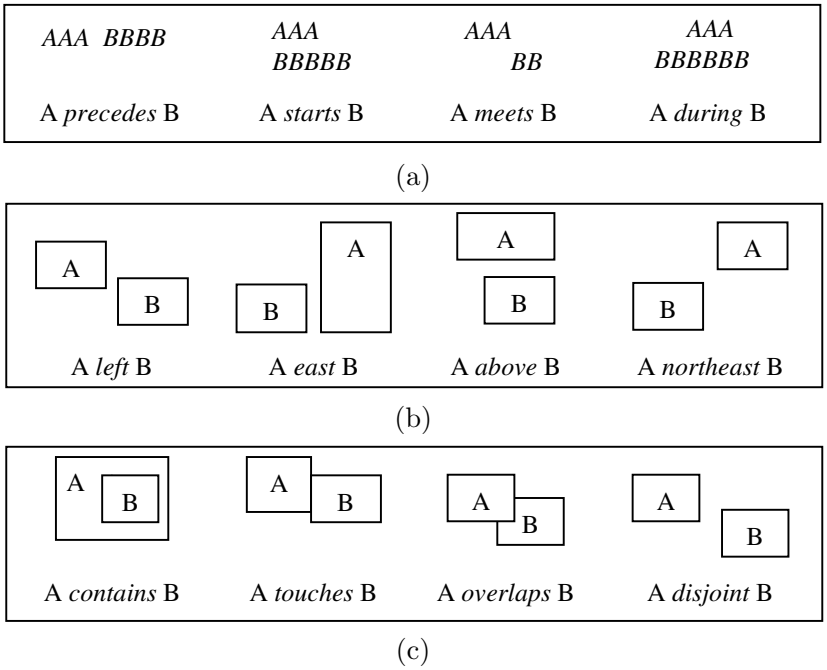


Figure 1.13: Examples of structural relations between segments. a) Temporal relations. b) Spatial Relations. c) Other possible spatial relations not defined in the SpatialRelation CS.

Semantic description

In some applications, the graph structure is of no real use, and the user can be more interesting in the semantic meaning of the content. In multimedia databases, a conventional way to model domain knowledge is given by ontologies. An ontology is a data model formalizing the concepts within the domain and their relationships. Expressing domain-specific ontologies in MPEG-7 has been early reported as a problematic issue for two main reasons [94]. Firstly, the MPEG-7 schemata have been inspired by the domain of broadcasting and audiovisual-based entertainment, and have been shown not efficient for describing highly structured domain such as medicine or sports [136]. The second problem was the integration of current semantic descriptions, such as the RDF (language) or ontology-based technologies (DAML+OWL) into the MDS (based on XML syntactic data schemes). Among the solution proposed, Tsinaraki et al. in [130, 131] express the MDS as an ontology with OWL, and defines domain-specific ontologies by extending the MDS core ontology. Garcia et al. in [7] follow in this direction and provides a complete description of the standard in OWL. Domain specific concepts are then mapped to the one provided by the MPEG-7. To have a better description of semantic aspects, [128] proposes to reserve MPEG-7 XML schemata for describing the structural relations of AV content, and use OWL to model the semantic aspects using domain-specific vocabulary. Vembu et al [136] go further and express a new approach to design multimedia ontology based on the MPEG-7 standard, replacing the Semantic Tools.

However, MPEG-7 defines structure for describing graph of relations based on a triplet (source *relation* target) as in the RDF. Basic ontologies can be constructed using the semantic part of MPEG-7, with the use of classification schemes (CS) to define a controlled vocabulary and its relationships. MPEG-7 has a predefined set of semantic relation, expressed by the SemanticRelation CS. SemanticRelation can depict the interaction between entities in the narrative world. For instance, New York is *part of* Manhattan, a cat *is a specialization of* animal, or banana *property* concept ripeness.

MPEG-7 distinguishes two main levels of semantic entities, individual instances of AV content (object A in shot B) and *formal abstractions* representing a generic class of entities ("Person","Football game",...). Fig.1.14 illustrates the segment-semantics relation and the abstraction procedure. At the top level, a formal abstraction is filled with a concrete instance using the relation *exemplifies*. Semantic entities perceivable in the AV content can be located in the media stream with the MediaOccurrence element and inversely a segment can reference any semantic description with the semanticRef element.

Most of time segment-semantic relations cannot be assigned with certainty to a video segment, when the relation has been machine-generated using a content analysis procedure. The strength of the relation can be specified between the formal abstraction and its instantiation (object A *exemplifies* "Person" with degree 0.8) or directly in the segment description (shot B shows "Person" with degree 0.8). Semantic relationships can be also weighted: comparison of two objects with the *similarTo* relation, or causality between two events with the *resultOf* relation. Finally, two objects representing the same entity in the real world are related with the *identify* relation e.g. "Person A" identifies "Person B" means that A and B represent the same person. Such weights can be assigned through the content

extraction procedure to segment-semantics relations, or assigned manually by a domain expert(e.g. object-event and object-concepts relations).

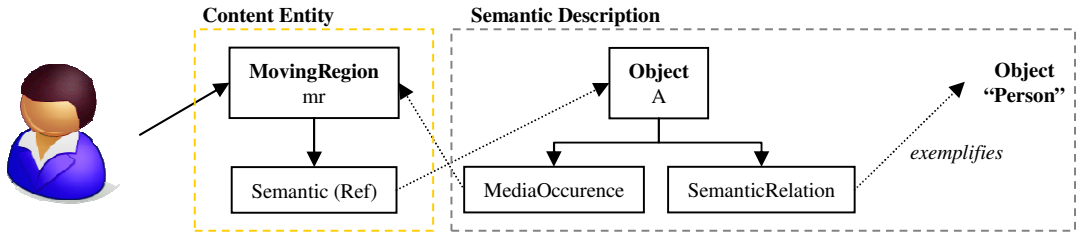


Figure 1.14: Construction of a formal abstraction.

1.1.3 Example of representation with the MPEG-7

Definition of a content model

The flexibility and genericity of the MPEG-7 standard can lead to generate quite complex and intricate descriptions, even for describing a simple scene. Thus, the covered aspect and the expression of the relations between them must be defined with attention. Attempts to formalize models for content description with the MPEG-7 have been proposed in [2, 8]. [2] proposes a generic "hanging basket model" to organize mpeg-7 metadata and apply this model on user preference metadata. [8] address the problem of organizing the low-level feature and high-level semantic aspects of multimedia content within a single framework. In the same spirit, we describe here content description model to organize the content of video shots. This model is illustrated fig.1.15.

Visual description and spatiotemporal relationships are grouped within contentEntity descriptions. Visual objects within a shot are described using the spatioTemporalDecomposition and movingRegion DS (fig.1.15(a)). The movingRegion DS defines location, visual descriptors, motion of the object and can reference its semantics. Recursive decomposition is permitted for describing complex objects or reflect object changes within the shot.

The TemporalDecomposition DS is used to group and reference events occurring in the shot (fig.1.15(b)). Each level of the decomposition (shot, object, region) can reference semantics perceived in the shot. Global semantics (narrative worlds, abstraction) related to the shot are referenced within the shot segment(not shown in the figure). The objects intervening in each event along with their spatiotemporal relationships are indicated within the related event segment.

Semantic entities referenced in the visual description and their semantic relationships are defined with a semanticDescription DS (fig.1.15(c)). Objects, events and narrative worlds are grouped in distinctive descriptions. To relate occurrences of semantic entities into specific temporal segments, the MediaOccurrence elements together with TemporalMask is used. Events are related to object's description and inversely with the *agentOf* and *agent* relations. Similarly semantic entities are related to a narrative world with the *partOf* relation. Objects are related to each other using the graph and semanticRelation DSs. More specifically,

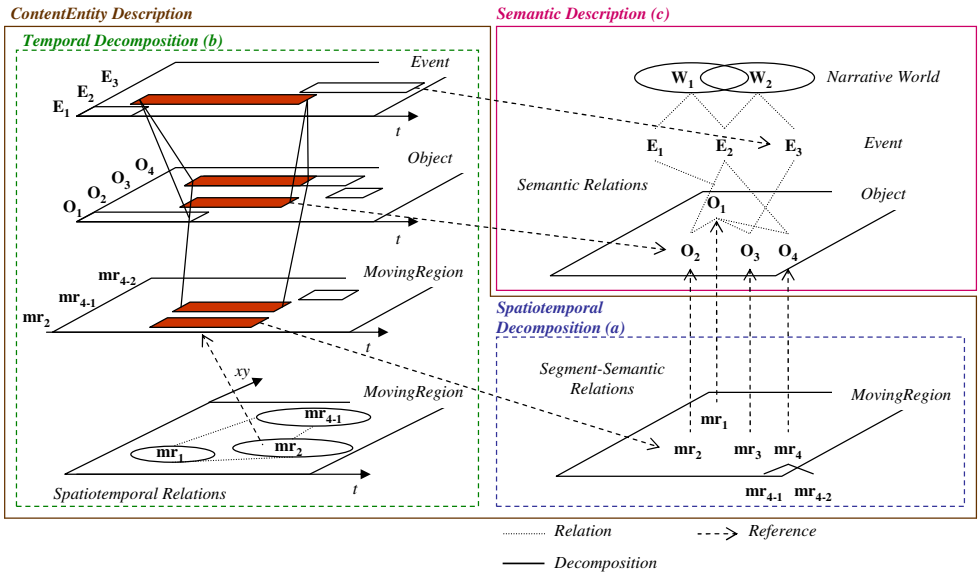


Figure 1.15: Content model for MPEG-7 descriptions.

the hierarchy between objects is represented with the *specializes* or *generalizes* relation.

Formal abstraction of semantic descriptions are defined separately, representing a variety of multimedia content. For instance an abstraction for a narrative world is specified in a particular semantic description. To avoid dispersion and duplication of the elements composing different narrative world and enable interoperability with other applications, semantic terms depicting the narrative worlds are defined and structured with the ClassificationScheme DSs. With this scheme, a particular object in a concrete semantic description can be linked to real-world semantic terms by the means of instantiating a formal abstraction.

In the following section we represent a scene with the guidance of this model.

Example of video shot content description

To illustrate the construction of AV content description with the MPEG-7, we propose the example of fig.1.16 showing two consecutive and semantically related shots of a video sequence.

The semantics of the sequence can be described as follows:

> *A truck is driving on the road. the truck stops at a military roadblock. A military walks in direction of the truck and verify the interior.*

A simplified classification scheme that encompasses the semantics of the sequence is shown in appendix A.1. The CS enables experts to define the vocabulary needed for modeling a particular domain and the hierarchy between terms. The scene can be decomposed as part of two narrative worlds : “Road Driving” and “Military Roadblock”. A formal abstraction

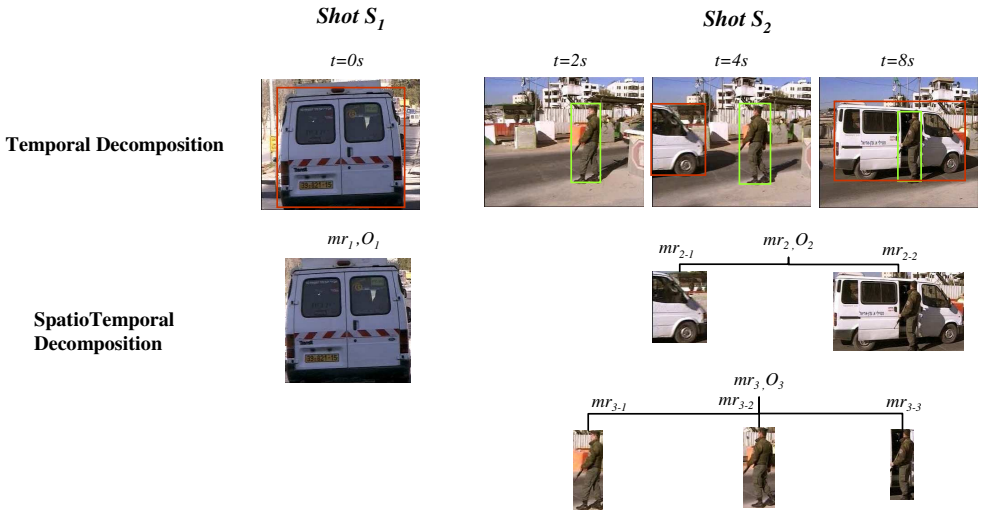


Figure 1.16: Proposed example. Temporal decomposition is based on events and spatiotemporal decomposition on objects.

modeling the semantics of the sequence is depicted fig.1.17. The corresponding MPEG-7 description is given in appendix A.2. In this abstraction, the "Road Driving" narrative world is composed of three entities : object "Vehicle", the Semantic Place "Road", and the event "Drive". The narrative world is limited to a single statement : "A vehicle is driving on the road". Similarly, the "Military Roadblock" narrative world encompasses the event sequence "Stop"-"Walk"-"Verify", the objects "Vehicle" and "Military" and the SemanticPlace "RoadBlock". Narrative world is composed of the statements "A vehicle stops at a military roadblock. A military walks in direction of this vehicle and verify the interior." Additionally, the context of the "Military Roadblock" is interpreted with concept "war".

A video sequence can be temporally decomposed into visual shots by searching for transitions and cuts. In general, a video shot shows a semantically related sequence of actions involving one or several objects. Thus shot semantics can be fit into a single narrative world. In the example fig.1.16, the shot has been automatically cut into two shots S_1 and S_2. The shots are related to the "Road Driving" and "Military Roadblock" semantics. The temporal decomposition describes multimedia content in terms of events. The three events composing the "Military Roadblock" semantic are depicted at times $t = 2s$, $t = 4s$, $t = 8s$.

The complementary spatiotemporal decomposition describes the visual objects in the shots : truck mr_1 for S_1, truck mr_2 and military mr_3 for S_2. The temporal decomposition and the interaction between events an objects is detailed in fig.1.18. As the military object mr_3 and moving object mr_2 overlaps different events, these segments are further decomposed temporally into sub-objects related to a single event.

Structural relations between the different video segments are expressed within the temporal decomposition. These relations are illustrated fig.1.19. At the shot level, temporal relations consist of S_1 containing mr_1 and S_2 containing mr_2 and mr_3. A spatiotemporal

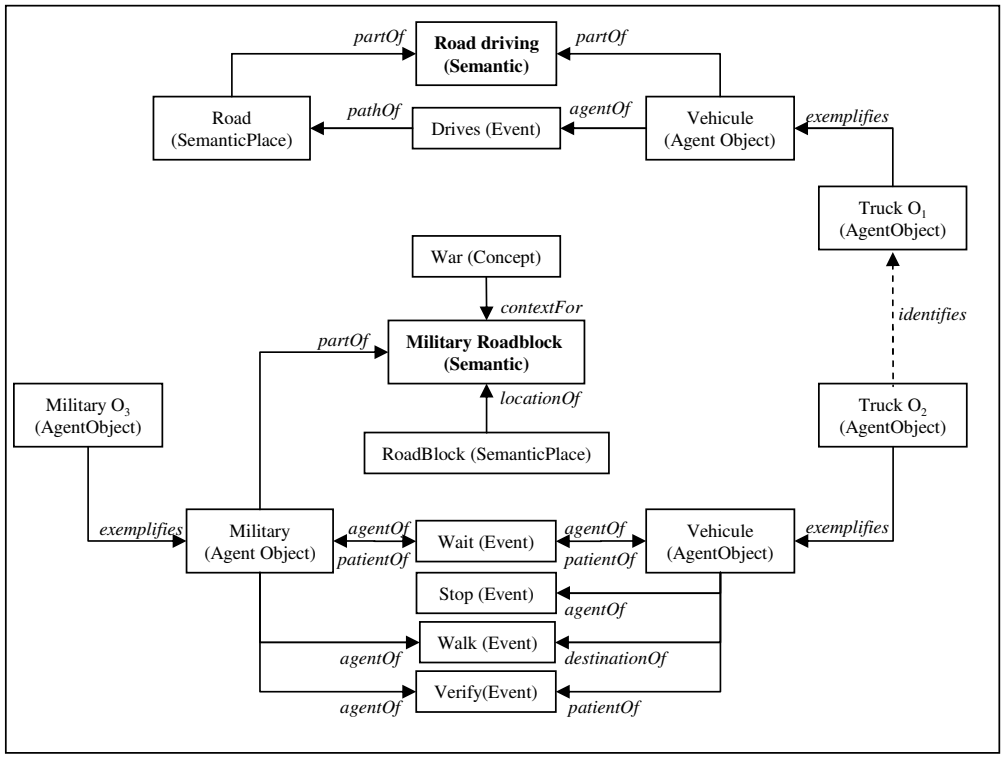


Figure 1.17: Semantic descriptions of the narrative worlds for the example.

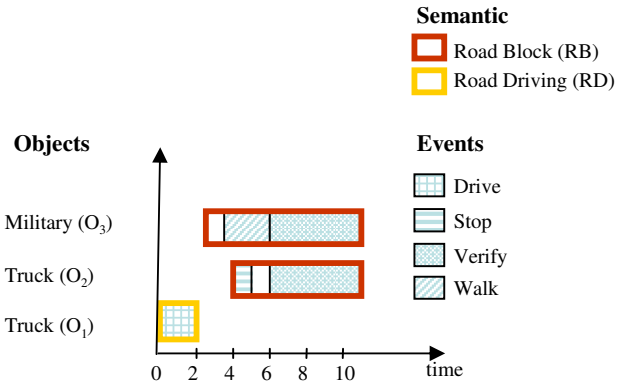


Figure 1.18: Temporal decomposition into events.

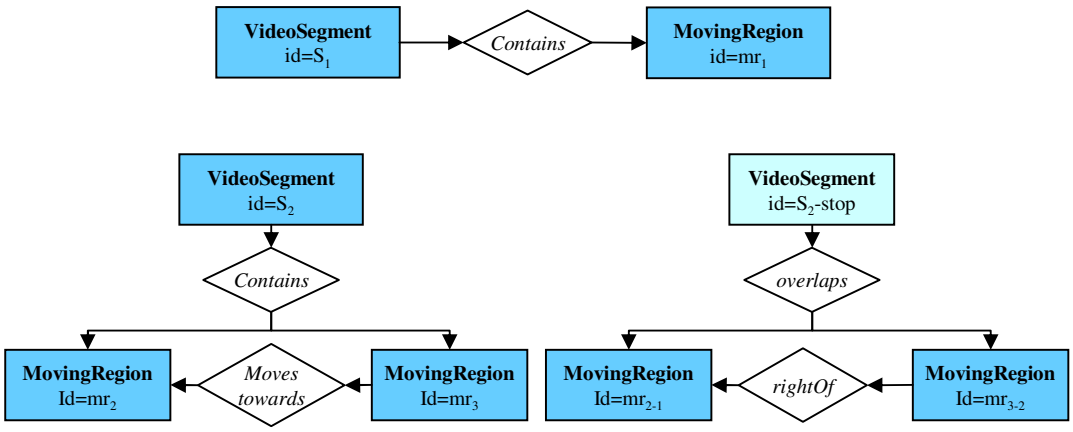


Figure 1.19: Spatiotemporal relationships within the temporal decomposition.

relation *moves toward* can be established between mr_2 and mr_3 in shot S_2. At the event level, the spatiotemporal relation can be specified more accurately. For instance in the second event of S_2 (stop) mr_3 *is right of* mr_2. Temporal and spatial decomposition, along with visual descriptors and structural relations are grouped in a single content description which can be found in appendix A.3.

Multimedia entities in the visual description are related to the real-world semantics by a semantic description, reported in appendix A.2. Each spatiotemporal segment mr_i, $i = 1 \cdots 3$ is associated to a media object O_i which exemplify a class of objects. O_1 exemplifies "truck" and "vehicle", O_2 "vehicle" and O_3 "military". O_2 and O_3 are reported as identical (relation *identifies*). Similarly the temporal segments are associated chronologically in the semantic description to events (E_1 to E_4).

We have seen in this example how to handle various aspect of video content description with the MPEG-7, in particular for object representation. A method for the construction of a low-level visual description of video shot will be exposed in chap.2 and the integration of semantic information from domain knowledge will be further introduced in chap.3. Of course, every system will have its own description model, but the descriptions can be read and understood easily by other applications if the specific schemata are provided. Among emerging systems that are based on MPEG-7 content description, we can cite tools aiming at authoring and video editing such as COSMOS [8], Ricoh Movie tool [106], Frameline 47 [47] or CBVR systems such as Vizir [38], IBM Marve[6], SVAT [105], ifinder [78]. Generic applications never covers all aspects of the description. Authoring tools will try to complete the semantic description, whereas CBVR tools will focus on search and retrieval of video segments (shots, objects) using audiovisual description and textual semantic annotation of the multimedia content.

Recognition of object semantics and event detection requires deep domain knowledge or user interaction and depends of the application. Attempts to build complete descriptions of the scene with the MPEG-7 have been proposed in specific domains. These efforts have

been mostly related to sports and medicine [132], and motivate to enlarge the use of the standard.

1.2 Spatiotemporal representation of video objects

MPEG-7 defines the structure for exhaustive description of visual content, and can adapt to many environments. We have seen that objects are an essential aspect of the description that links the low-level visual descriptions to the understanding of semantics and events of the shot. However, extraction of content will never be part of the standard. Thus, efficient models for object representation and extraction procedure are still needed. In this section we introduce graph-based representation for describing a video shot, structuring pixels into complex objects.

1.2.1 Visual processing

A video sequence contains huge amount of data that consist in a 3-dimensional array of pixel color intensity. A single frame in an MPEG-1 video will contain around 100000 data points, a sequence of a few seconds millions of points. Making sense of the world represented from this volume of data is very difficult to achieve. A major target of video analysis consists in the detection and interpretation of the physical objects that are projected on the sequence of frames. The most advanced system known so far is the human vision; a person can locate and catalog all objects interfering in its field of view in a fraction of a second.

Understanding the mechanism of perception has been widely studied by cognitive sciences. Among the different approaches, Gestalt theory, early introduced by psychologists [143, 65] states that our vision is based on successive groupings, in a cognitive process lying both on inborn characteristics and experience. The main problematic was to describe the global nature of perceptual experience.

Whereas digital computer solve problems by isolating simple steps and assembling their results, Gestalt theorists state that for understanding the perceived world, its elements must not be taken isolated but together as a global construct. This suggests that the understanding of an object depends both on its content and of its environment [75]. The theory was comforted by the neuroscience research [65] which shows that biological mechanism of human perception was found fundamentally different of those known so far. A “gestalt” or (“whole”) is a structured form which has a meaning for us. Every elements of a gestalt influences each other, and the gestalt itself depends of its surrounding environment.

The equilibrium of Gestalt systems are governed dynamically by the properties of *emergence*, *reification*, *multistability* and *invariance*. Emergence refers to the fact that a phenomenon is not interpreted by just assembling of its components but by seeing it at a whole. Reification is the capacity of perceiving a complete whole even when the actual sensory stimuli are ambiguous or uncomplete. One example is that perception can fill the missing parts of an object with the pattern in the visible parts. Multistability refers to the fact that our perception can alternate between multiple interpretations (optical illusions). Finally, invariance property states that objects can be recognized independently of

geometrical transformations and deformations.

These properties describe how we perceive a scene, but does not explain the underlying mechanisms. According to Gestalt theorists, the fundamental principles of perception are based on figure-background distinction and "natural" laws of grouping. Perception groups sensory stimuli into "good forms", which are either simple geometric and regular structures, or a known pattern which is significant for us. As instance a "good form" may be a head, and arm or a hand constituting a person. A "good form" can be also associated to a particular visual primitive as a blue color suggests sky or sea and the red color blood. In this way the stimuli are grouped into elements that can be distinguished from the background, which is perceived more irregular and less structured. This distinction enables us to focus on different elements on the scene, for instance a particular face in a crowd. Several grouping mechanisms interferes to unify partial perceptions into a whole (the gestalt) and are referenced as the laws of Gestalt (also known as laws of perceptual organization):

- *Proximity* : First elements grouped are spatially or temporally closest to each other.
- *Similarity*: Elements that appear similar to us (in form, color, brightness, ...) are grouped into collective entities.
- *Closure* : A closed form is more easily identified as a figure than an open form.
- *Continuity*: Close points are linked together when they can be prolongated the ones to each other.
- *Common fate*. Moving parts with similar trajectories are perceived as a single form.

According to these laws, stimuli are strongly encouraged to be organized into a gestalt system with strong, regular or known forms, or secondly to a structure as closed as possible of a gestalt. All that can be unified is unified, the only elements separated are those presenting features incompatible with fusion.

Gestalt theory have been criticized on the fact that it does not translate abilities of human perception into models and algorithms, i.e. explain how the biological information is transformed into an interpretation. The computational approach (also referred as computer or cognitive vision), initiated by the work of [82] aims to tackle this difficult task. First developments were to understand the geometry and physics involved in vision, then to apply to more complex problems. Complete computational models of vision have been proposed by [133], but restricted to the application domain. Most recent considered approach, or connexionism, models the cognitive process as a result of a parallel computation operated by a neural network structure, by analogy to the processing of information in the human brain. Although criticized for its complexity and opacity [133], it has been successful for some pattern recognition problems and other imaging applications. Therefore problems of vision and video processing is an open area, and the frontiers between different approaches are not fixed.

Gestalt scientists believed that one of the fundamental problem in vision was to distinguish objects and background, characters and their environment. This belief has been translated in the computational approach, leading research on segmentation methods that

decompose the scene into regions with semantic meaning. Image and video segmentation and object recognition have made wide use of grouping techniques. Generic grouping algorithm can be based on some probabilistic framework. Gestalt rules of organization have played important role in the domain of generic grouping algorithms. Most important and widely used ones are based on proximity and similarity. Region Adjacency Graphs (RAG) encompass these two notions, grouping connected pixels based on their similarity. Adjacent regions can be further grouped with other principles of the Gestalt. In [149], a distance is computed that takes into account the closure, similarity and continuity. Continuity and closure constraints have been also modeled using Markov Random Fields [83]. In the case of video, motion grouping are used in conjunction with these rules [129]. Besides these generic algorithms, [149] argued that grouping process must be controlled by the domain knowledge about objects composing the scene. Semantic objects are organized at different levels, grouping being performed at each level. Scene composition leads to the idea that grouping should be processed in an iterative way, involving different mechanisms and properties to be considered according to the level of grouping.

Grouping techniques can reach high complexity in function of the level of semantic interpretation to be reached. Computational cost is driven by the complexity of the model and the range and structure of interaction between visual aggregates. In global techniques that considers data information at once (graph-cut, or MRF for labeled graphs), the solution-space is large and the problem generally belongs to the class of NP-hard problems, thus approximations and costly optimization procedure are required [115], though progress have been achieved recently in [18].

In this thesis, we focus on efficient techniques for grouping of video regions. To this aim we propose to follow an approach based on adjacency graphs. We explore structural properties and constraints to be modeled in the graph in order to represent different structural and semantic levels of the video.

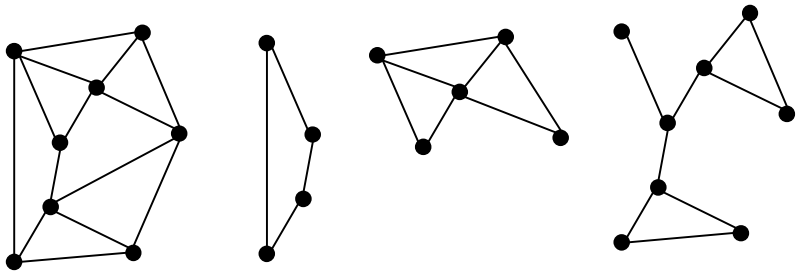
1.2.2 Basic graph terminology

As we make extensive use of graph terminology and algorithms for building a structured representation of the video content, we introduce in this section the required basic notions of graph theory.

A graph is a pair $G = (V, E)$ of disjoint sets, such that $E \subseteq V \times V$. V is the set of vertices (also called *nodes*) and E is the set of edges. By convention the vertex set of a graph G will be noted $V(G)$ and its edge set $E(G)$. A common abuse will be to declare a vertex v in G instead of $V(G)$ and an edge $e \in G$ instead of $E(G)$.

The number of nodes of a graph G is denoted by $|G|$ and is called the *order* of the graph; the number of edges is written as $||G||$. Therefore, we have $|G| = |V|$ and $||G|| = |E|$.

Two vertices a, b of G connected by an edge $e \in E$ are *adjacent* or *neighbors*. This is denoted by $e = (a, b)$. a and b are said to be *incident* to e and e is an edge *at* a and b. Two edges are said to be adjacent if they have a common endpoint. Pairwise non-adjacent vertices or edges are also called *independent*; a set of vertex (or edges) is then independent if none of its elements are adjacent. When edges (a, b) and (b, a) should not be distinguished, the graph is called *undirected*. Otherwise the graph is *directed* and (a, b) is said to be and



a) b) c) d)

Figure 1.20: a) Graph. b-c) Example of induced subgraph. d) Example of spanning subgraph.

outcoming edge for a and an *incoming* edge for b. In the case of a directed graph, the term *arc* is preferably used instead of *edge*. The number of edges incident to a vertex a in G is the degree of a and is denoted $deg(a)$. If the graph is directed the *indegree* $deg^-(a)$ is number of incoming edges and the *outdegree* $deg^+(a)$ the number of outcoming edges.

Considering two sets A, B of V, (a, b) is an $A - B$ edge if $a \in A$ and $b \in B$. The set of all $A - B$ edges is denoted $E(A, B)$. When A is limited to a singleton a, $E(a)$ is the set of all edges at vertex a. In addition, the set of neighbors in G for vertex a is denoted by $N^G(a)$ or more simply $N(a)$ when no confusion is possible. By extension the neighbors in V/A for the vertex set A are called *neighbors* of A; the set is denoted by $N(A)$.

Considering two graph $G = (V, E)$ and $G' = (V', E')$, G' is a subgraph of G if $V' \in V$ and $E \in E'$. When $V = V'$, G' is said to be a *spanning subgraph* of G. If all edges in E with both ends in V' are in E', G' is an induced subgraph of G : $G' \equiv G[V']$. An illustration of these subgraph properties is shown fig.1.20.

To impose constraints on the graph structure, many specialized types of graphs can be defined and are particularly useful for visiting a graph or defining a subgraph on a graph. A *path* between any two vertices a, $b \in V$ is a subgraph $P = (V, E) \subset G$ such that $V = \{a_0, a_1, \cdots, a_k\}$ is a non-empty sequence of vertices where all $\{a_i\}_{i=1:k}$, are distinct and $E = \{(a_0, a_1), \cdots, (a_{k-1}, a_k)\}$ connects each vertex to the next one in the sequence.

A *cycle* is a path where the first and last vertex are the same, and a graph with no cycle is said to be *acyclic*. A *connected* graph is a graph where any two vertices are connected by a path. A connected subgraph of a graph G with a maximal edge set is a *component* of G. A *tree* is an acyclic connected graph, i.e. there is a unique path between every two vertices. A *forest* is a graph whose components are trees. Thus a connected forest is also a tree.

Another example of connected graph is the *grid*. A $M \times N$ grid is a graph on $\{1, \cdots, M\} \times \{1, \cdots, N\}$ with an edge set E defined as:

$$E = \{(i, j), (i', j') : |i' - i| + |j' - j| = 1\} \tag{1.6}$$

Another mean to express connectivity in graph is given by the notion of r-partite graphs. An r-partite graph partitions its vertex V into r classes such that all vertices of the same class are independent. A 2-partite graph is called *bipartite*.

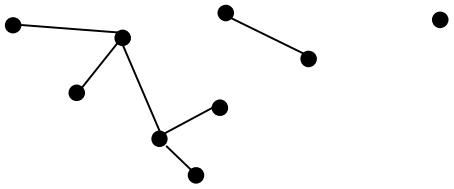


Figure 1.21: Example of forest. Each component is a tree.

A set of independent edges M in a graph G is called a *matching*. A vertex $a \in G$ is matched by M if it is incident to an edge in M. Otherwise the vertex is unmatched. When every vertex in G is incident to M the matching is said to be *complete* or *perfect*. The matching problem is often related to bipartite graphs, trying to find one to one correspondences between two vertex sets, with maximum edge cardinality or weight (fig.1.22(a)). In general, it can be extended to any pair of graph by considering graph homomorphisms. A graph homomorphism f from graph $G_P = (V_P, E_P)$ to $G_Q = (V_Q, E_Q)$ is a mapping $f : V_P \rightarrow V_Q$ such that $(a, b) \in E_P$ if $(f(a), f(b)) \in E_Q$. When $V_P = V_Q$ and f is bijective f is a graph isomorphism and the problem of finding f is referred as *exact graph matching* (fig.1.22(b)). When the two graphs G_P and G_Q do not have the same number of edges or vertices, the problem is referred as *inexact graph matching* (fig.1.22(c)). Letting $|V_P| > |V_Q|$ the class of *subgraph matching* problems consists in finding a subgraph $|V'_P|$ isomorphic to V_Q. More specifically, the maximum common subgraph problem searches an induced subgraph $|V'_P|$ which order is maximal. Most difficult inexact graph matching problems allows multiple correspondences (fig.1.22(d)) and can be expressed as finding an homomorphism $f : V_P \rightarrow W \subset \mathcal{P}(V_Q)$ where $\mathcal{P}(V_Q)$ is the set of all parts of V_Q. The search space for the solution is gigantic and its resolution is highly dependent of the particular problem considered.

A graph can also conveys information on its nodes and edges. When this information consist in a simple label added to the nodes, the graph is called *labeled graph*. Otherwise an edge e often relates to a numerical value $w(e)$ indicating a distance or a flow between between nodes; this graph is called a weighted graph. In the general case, a graph which contains both vertex and edges attributes is called an *attributed graph*. An attributed graph is therefore a tuple $G = (V, E, \nu, \xi)$ where :

- $\nu : V \rightarrow A_V$ is the set of functions generating node attributes.
- $\xi : E \rightarrow A_E$ is the set of functions generating edge attributes.

Attributed graphs have been particularly useful for a large set of visions problems. Indeed many real world entities can be modeled as an ARG. An entity can represent an object in a visual scene, such as a face or a character, a landmark in a remote sensing image, a segment of character symbol, and any set of pixels that shows a certain visual cohesion in general. In our study, the elementary entities are defined as spatiotemporal volumes that compose the video shot. We defined this decomposition in the following section.

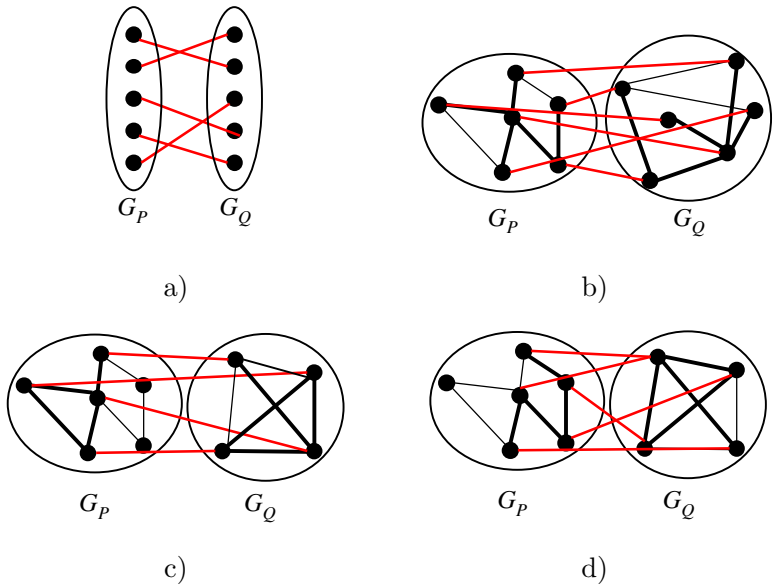


Figure 1.22: Matching between graphs. The matching is shown in red, and matched subgraphs with thick lines. a) Bipartite matching. b) Exact graph matching. c) Inexact graph matching with $V_P > |V_Q|$. d) Matching with multiple correspondences.

1.2.3 Spatiotemporal volumes

At the raw level of information, a video shot is a sequence of frames that can be represented by a 3D grid of pixels, as shown fig.1.23. Each node in the graph $G = (V, E)$ corresponds to a pixel in one frame (or voxel in the shot), and the edges E connects neighboring pixels of the grid. We consider a traditional rectangular grid where each pixel is connected to 6 neighbors (4 spatially and 2 temporally).

The problem of extracting objects in the shot can be formally represented as achieving a set of disjoint subsets (or *partition*) S of the vertex set V :

$$S = \{V_1, \cdots, V_N\}, \bigcup_{i \in [|1,N|]} V_i = V. \tag{1.7}$$

S is said to be a *segmentation* of the video shot. For a subset V_i to represent an object, a reasonable constraint is that V_i corresponds to a component C_i in the 3D grid, i.e. that all vertices of V_i are connected by at least one edge in E. The segmentation S is then called *spatiotemporal* and subsets V_i are referred as *volumes*. This representation is illustrated in fig.1.24.

A complementary view of the partitioning problem is expressed by the notion of *cuts*. A cut separates two disjoint vertex sets V_1 and V_2 (and their respective components, as we are considering volumes). Formally the cut between V_1 and V_2 is defined as $E(V_1, V_2)$ i.e. the

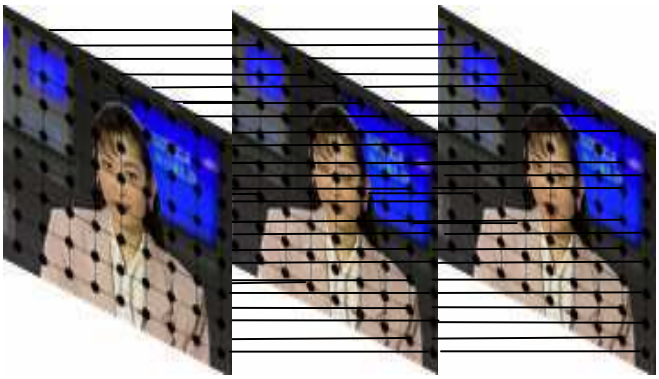


Figure 1.23: 3D pixel grid over a sequence of frames.

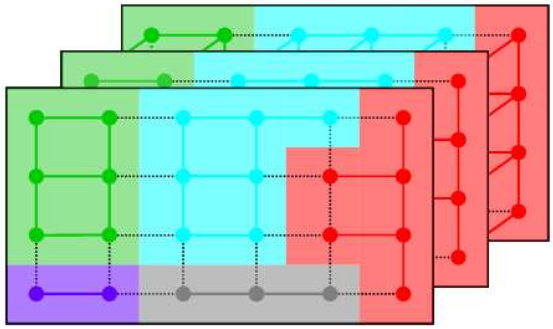


Figure 1.24: The grid is decomposed into volumes (colored areas). Components associated to volumes are shown with dots (vertex) and lines (edges) of the same color.

set of all $V_1 - V_2$ edges :

$$cut(V_1, V_2) = \{(p_1, p_2) \in (V_1 \times V_2) \cap E\} \tag{1.8}$$

The segmentation S is then obtained by removing all cuts in the graph. These different representations leads to two approaches for the partitioning problem. The first approach (bottom-up) groups pixels into grid components, the second one removes edges present in the cuts to delimit the partition.

To group pixels into volumes, visual properties are considered. The grid edges are weighted by a local visual similarity measure between pixels. In that sense, a volume can be constructed by gathering locally edges with high visual similarity, representing a connected homogeneous area in the sequence.

1.2.4 Adjacency graphs

The grid graph only depicts local structure and organization of the volumes. The relationships between volumes provide a very flexible representation for the global perceptual organization of a shot. Indeed, the relational structure of objects has been a powerful tool in computer vision and pattern recognition. Within the pattern recognition domain, the relational paradigm states that *patterns are formed not only by the nature of the objects which make the pattern, but also by their relationships to each other* [13].

The most basic relational graph is the *Adjacency Graph*. *Region Adjacency Graphs* (RAG) are usually considered for describing the neighboring relationships between components (regions) of an image. Each node in the graph corresponds to a region, and an edge corresponds to two spatially connected regions. The concept of RAG can be easily applied to video shots, volumes replacing regions and edges connected volume that are spatiotemporally connected. Two volumes A and B are spatiotemporally adjacent if there exists at least an $A - B$ edge in the 3D grid. The resulting graph, representing the spatiotemporal structure is called a Volume Adjacency Graph. As volumes reach a significant size and meaning, more significant properties can be extracted from the volume. The structure of objects along with their properties are represented by an ARG. When only adjacency relations to be considered, the object structure is represented by a VAG. The link between an ARG $G = (V, E, \nu, \xi)$ and its VAG G_r is expressed as $G_r = (V, E)$.

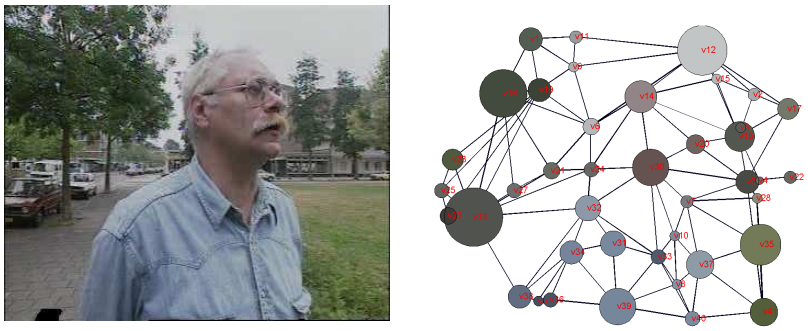


Figure 1.25: a) Frame sequence. b) Attributed relational graph between volumes. Volume properties are modeled with disks indicating their mean color and size.

1.2.5 Hierarchical representation

We have seen how to to depict volumes and their organization with different graph structures. Segmentation as a single step is difficult to handle as not enough relevant information can be found with small structures. Solutions for global partitioning have been proposed in context of spectral graph theory [115], but the solutions for an unknown number of objects and approximation effects are not well understood. In contrast, a bottom-up approach enables to group different type of information in function of the considered level of description.

The content of a video shot is modeled hierarchically at several levels from the following principles:

- Volumes are constructed by grouping components of a grid graph.
- Relationships between volumes are analyzed within an ARG to form objects.
- Objects subgraphs can be compared for matching and recognition purposes.

These different levels forms a pyramidal decomposition (fig.1.26). When considering small structures in the beginning, only little relevant information is noticeable at the pixel level. Adjacent points in the spatiotemporal domain are grouped by criterion of visual similarity to form volumes. These volumes are considered as elementary objects which attributes are composed of visual descriptors mentioned in section 1.1.2 and potentially of semantics when domain knowledge is available. At the next level, the organization of the shot into volumes is represented by an ARG. Relationships in the ARG can include proximity, visual and semantic similarity. The decomposition of the scene into ARG enables different shots to be compared (graph matching problem) or finding similar structures (patterns) in the shot. At the highest level the ARG is partitioned into object subgraphs. An object is described both by its volume properties and their structure, i.e. the organization of its subparts in the shot. Different objects can be compared both with their global properties (semantic, visual) and with their subgraph structure.

This representation is conform to the MPEG-7 approach and can be fully mapped with the visual description tools and the content model we detailed in section 1.1. In this way, raw video data can be transformed into an understandable structural description.

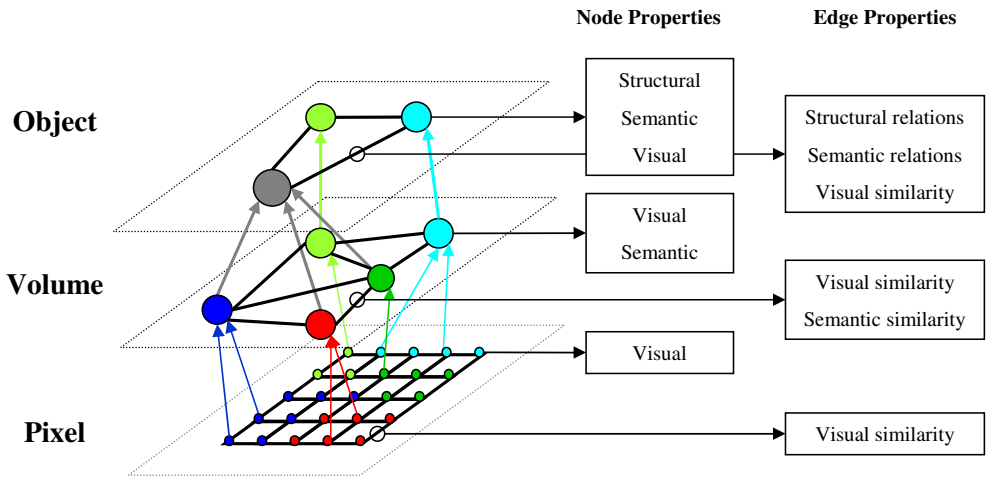


Figure 1.26: Hierarchical modeling of video shot with graphs. The complexity of the description and the graph attributes are function of the structural level : pixel, volumes or objects.

1.2.6 Merging and grouping

When considering a large graph, the interpretation of its intrinsic structure is often hindered by the huge number of vertices and relations. Two principal mechanisms are employed to extract the structure of graph, *merging* and *grouping*. Let $G = (V, E, \nu, \xi)$ an ARG with vertices $a, b \in V$ joined by an edge, $(a, b) \in E$. A merge $M(a, b, G)$ is an operator that constructs a new graph $G' = (V', E')$ such that G' is similar to G except that nodes (a, b) have been collapsed to obtain a new vertex $a \vee b$, so that edges (a, c), $(b, c) \in E$ are mapped into one edge $(a \vee b, c) \in E'$. A merge is also referred as *vertex contraction* in graph theory. The resulting vertex attribute, is computed by a merging function that combines local information from vertex a and b, $\nu(a \vee b) = f_\nu(a, b)$. Considering that ν is a function generating numerical attributes in R^N, common merging functions are chosen between:

- Average operator. The contribution of each attribute is weighted by a factor w:

$$f_\nu^{ave}(a, b) = \frac{w(a)\nu(a) + w(b)\nu(b)}{w(a) + w(b)} \tag{1.9}$$

- *min* and *max* operators.

- Median operator can also be considered when merging several nodes.

The affected edges are also updated by combining local information of the edges at a or b, i.e. $\xi(a \vee b, c) = f_\xi(a, b, c, \xi(a, b), \xi(a, c), \xi(b, c))$. In practice, edge attributes often represents a certain similarity between vertices. In such case $\xi(a \vee b, c)$ is reevaluated from computation of the new vertex attributes of $a \vee b$ and attributes of c, i.e. $\xi(a \vee b, c) = f(a \vee b, c)$.

The second mechanism refers to grouping. In graph theoretic formulation, the grouping problem is to decompose the set of vertices V of a graph into a partition $S = \{V_1, V_2, \cdots, V_N\}$. A set of attributed vertices is called a *group*. Thus for $V_i \in V$, the group associated to V_i is a tuple $H(V_i) = (V_i, \nu')$. Group attributes ν' represent global properties; they may be different from the ones of the related graph as further information can be deduced from a group than of a single node. When the considered attributes remains the same, aforementioned aggregation operators can be used to compute the group properties. Structural properties are from their part represented by the subgraphs spanned by the groups, i.e. the components $C_i = (V_i, E_i)$.

Fig.1.27 illustrates these notions. The initial graph G is given in fig.1.27(a). Fig.1.27(b) shows the graph obtained from different merges. On the contrary, fig.1.27(c) cannot be obtained from G by merging, as there is no $a - b$, $a - f$ or $a - g$ edge in G . The grouping resulting from the merge operations of fig.1.27(b) is shown in fig.1.27(d).

The set of groups associated to a partition $S \in \mathcal{P}(V)$, $H(S) = \{H(V_1), \cdots, H(V_N)\}$ forms the set of attributed vertices of a new graph $G' = (V', E', \nu', \xi')$ with $V' = S$ and $E' = S \times S$. Edge attributes ξ' are reconsidered from the vertex properties ν' and subgraph structure related to each group. In this way grouping mechanism enables to go through the different levels of representation mentioned in section 1.2.5.

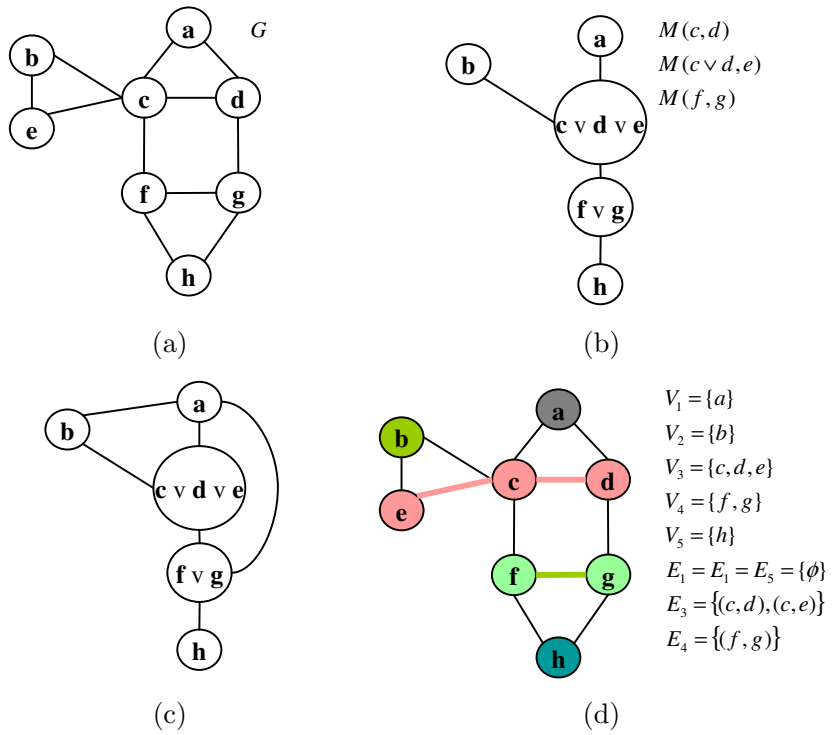


Figure 1.27: Example of merging and grouping. a) The initial graph G. b) The graph after merge operations. c) This graph cannot be obtained by merging of G. d) The partition of G resulting from the merging. Groups are composed of the nodes V_i and their properties. Associated subgraphs components (V_i, E_i) are marked with the same color.

1.3 Applications of the proposed representation

Applications involving content-based video manipulation and information retrieval have been widely increasing for the past years. Development of e-services has driven the development of video databases, among with transmission and delivery technologies. Consequently we have to support large amount of data. Good representations are needed for analyzing video content. Examples include browsing through main video parts, searching for a particular scene, indexing and retrieval with different modalities ... In this context, we detail the domains for which our spatiotemporal representation is of interest.

1.3.1 Context

Object-based representation plays a central role in video analysis, as illustrated fig.1.28. A general architecture for leading video applications is depicted. At the lowest level, the system stores the raw-video data. The extraction process provides description of the scene with objects and features. This is core of the model as it should support several services. Possible functionalities include compression and coding (generating MPEG-4 VOP or MPEG-7 con-

tent description), broadcasting according to the content. Another important service is the retrieval. The query could be textual, depicting semantics of the shot in term of events and objects, visual (an object or a video shot), or interactive where user sketches the behavior of objects composing the scene.

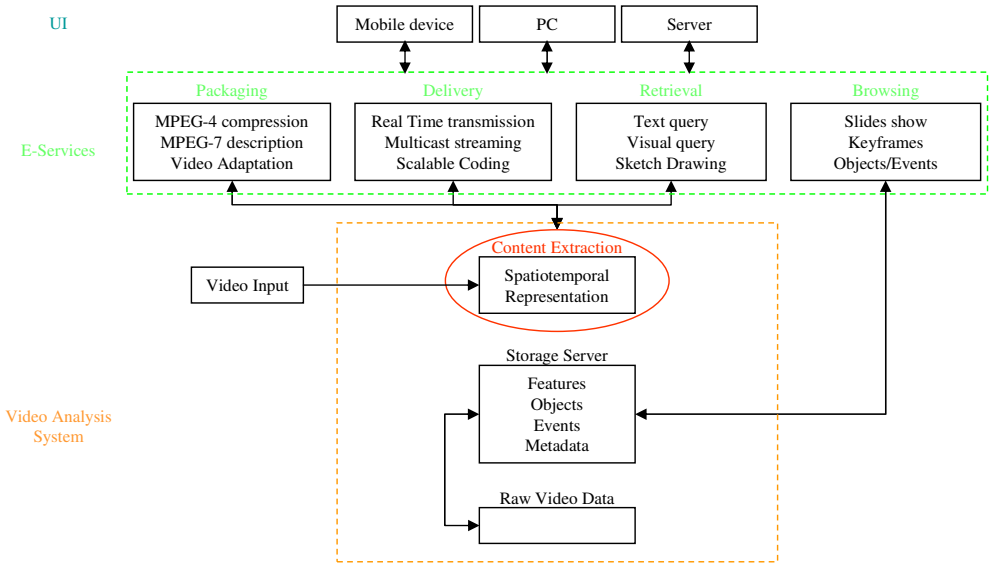


Figure 1.28: Architecture of multimedia system, beyond the aspects covered in the thesis. Spatiotemporal representation is the core of many content-based functionalities.

1.3.2 Proposed applications

The developed ARG representation can be adapted to different types of applications. Figure 1.29 shows the different content-based functionalities we investigated. The first targeted task concerns the construction of a spatiotemporal segmentation and the use of motion models for the extraction and indexation of moving objects (chap.2).

When domain knowledge is available, semantics can be inferred from the proposed representation. We use knowledge database to label object regions semantically. The semantic information is propagated between shots by matching objects based on both visual and semantic properties (chap:semantic).

The last functionality studied in the thesis concerns automatic indexing of video shots. We first investigate an indexing model based on the construction of a dictionary of visual words. A compact and efficient index of the shot is generated from volume visual features. Secondly we construct a structural and visual similarity measures to match ARG representation of video objects and a strategy for searching objects is proposed (chap.4).

These three problems attests that spatiotemporal segmentation and model of representation appears crucial for main applications. In the next chapter we will investigate the extraction of video objects and show how the graph representation is used in practice.

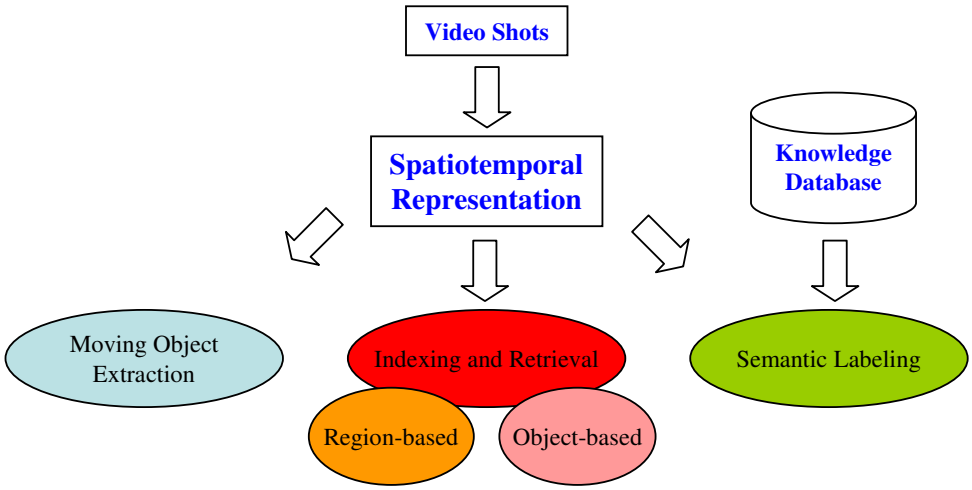


Figure 1.29: Framework of the thesis. The targeted applications are based on spatiotemporal representation of video shots.

1.4 Conclusions

This first chapter was dedicated to introduce video shot representation in video analysis systems. The MPEG-7 standard provides a global framework for building representations for multimedia applications. We detailed the use of MPEG-7 multimedia tools for content description of video shots, both in terms of visual and semantic structure. From this study we deduced a possible content model for MPEG-7 shot descriptions emphasizing the description of visual objects and their relation to high-level semantics of the shot. In particular we have seen a small real-world example with a full description that illustrates the difficulty of the content analysis task, a simple scene is represented with a complex description. We propose a spatiotemporal representation of the visual content of video shots. Attributed Graphs enables to represent efficiently the information at different levels : pixel, volume, object. Finally, we show the context of applications of our representation in a large view and introduced the ones we will examine in this thesis.

Chapter 2

Video object extraction

In the previous chapter, we have detailed the organization of a video shot in the context of the MPEG-7 standard. However, the standard only defines the structure, but does not provide methods to extract video content. Nowadays, there is no basic methods which builds high-level descriptions in general, since the process is highly dependent of the application domain. However, it remains possible to organize video content to facilitate the set up of high-level applications requiring search or indexing. Automatic segmentation methods have been widely investigated for the last 20 years. Most techniques are based on image segmentation followed by a tracking based on motion estimation. The issue is that their initialization remains difficult to manage and handling of dynamic objects is rarely taken into account.

On the contrary, we propose to investigate a spatiotemporal approach for the automatic extraction of objects. The target is to enhance temporal consistency of the regions by decomposing the video sequence into volumes and to reconstitute the structures directly within the shot.

In this chapter, we first establish a state of the art and a classification of the methods for automatic segmentation of video objects. Then, we propose an efficient algorithm for spatiotemporal segmentation in our graph framework. Finally we investigate grouping spatiotemporal regions using motion models and deduce a method for building complex objects within the shot.

2.1 State of the art

The human visual system decomposes a scene into distinct entities. These entities corresponds to semantic objects that are recognized by the person. Due to its subjectivity, the decomposition of a scene into objects may differ from one person to another. Moreover, the appearance of objects may vary considerably in function of the object but also in function of is projection in the image. The segmentation of video objects is therefore an ill-posed problem, whose solution depends on the user requirements or is defined for a particular domain. First, we will introduce the different existing schemes for the video object segmentation problem. Then, we will detail the spatiotemporal methods and the approach developed in

this thesis.

2.1.1 Classification of existing methods

Video is naturally decomposed into objects. Human vision is able to detect the different objects in a scene automatically and effortlessly. However, this is not the case for a segmentation method which should be defined cautiously. Research developments on video object segmentation has driven the emergence of semi-automatic and automatic methods. The former utilizes the user input to overcome the difficulty of extracting objects at the semantic level. For instance, semantic information is given at the beginning by specifying the location, contours or motion of the objects. In the second case, automatic methods restrict the object type to a specific domain: face, human, vehicle, ... The criteria for detection and localization of the objects are then established from a specific model.

In a more global context, methods are based on low-level features that can be extracted automatically, such as color, texture, contours, ... These features are generally insufficient to define a full semantic object, the partitioning into homogeneous regions does not usually reaches the granularity desired by the end-user. Considering a video sequence enables to add the temporal dimension and the motion information, which have been mainly considered for grouping regions into objects. These developments have given rise to the so-called *spatiotemporal* methods. It is possible to build a classification of these methods according to the type of information they use preferentially.

Two main families of approaches can be distinguished. Spatiotemporal approaches refer to:

- In a large sense, approaches which perform spatial segmentation of the frame, followed by tracking of frame regions, which we qualify as *motion-based* approaches. Methods favoring temporal information first segment the image into homogeneous region with respect to the optical flow, the ones favoring spatial information exploit different spatial features such as color or motion. Then, the motion between two images is used to link temporally frame regions, ensuring the coherence between the segmentation of each image. The obtained moving regions are considered as objects.

- In a more limited sense, approaches that use the temporal and spatial information simultaneously to build spatiotemporal structures [85], which we refer simply as *spatiotemporal* or *3D* methods. Within this family, we can distinguish "pure" approaches which considers a video sequence as single volume. Spatiotemporal structures are extracted directly from a block of frames, spatial and temporal grouping are performed at the same time. This approach is in direct relation with human vision and the Gestalt theory. The eye has been shown as interpreting structures dynamically in space and time [52]. A second type of approach constructs volumes considering a $2D + T$ space: spatial structures temporally connected are grouped using spatial and temporal information conjointly. Main techniques following this approach are based on RAG labeling and matching.

Figure 2.1 shows in a simplified view the aforementioned approaches. Our approach rely on

the 2D+T framework. This perspective turned out to be simpler and efficient with respect to the computational complexity.

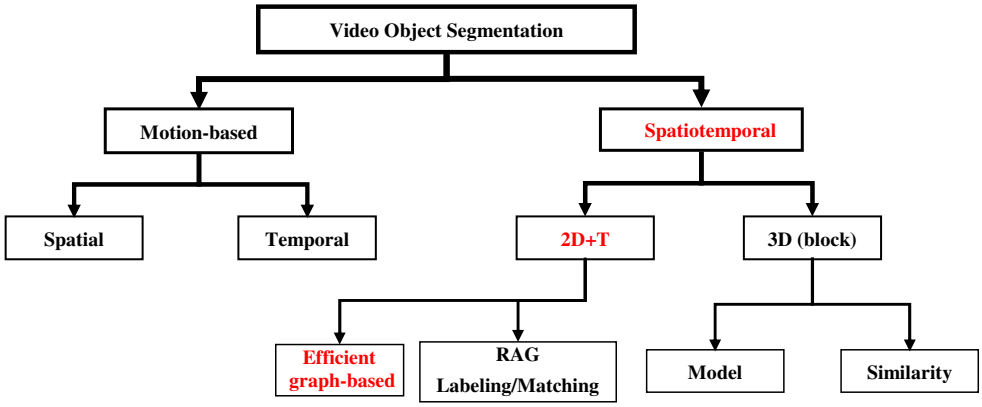


Figure 2.1: Classification of spatiotemporal segmentation methods.

2.1.2 Motion-based methods

Motion based segmentation consists in decomposing the objects undergoing different motions. Objects are supposed to have coherent motions that are distinct from the background. Motion is then a leading source of information in a video sequence. To model motion, camera parameters and the displacement of objects should ideally be known to compute the transformation between the scene and the acquired image. Unfortunately, this information is rarely known, so that both camera and object motion must be evaluated. Estimating the 2D projected motion, called *apparent motion* or *optical flow* is a difficult and ill-posed problem as itself [59]. To represent the apparent motion from the 3D scene, a common assumption concerns the rigidity of objects. If a rigid object is also constrained to move on a plane, the apparent motion of the object can be described by an affine model (6 parameters). Affine model estimation has been widely used in practice [141, 42, 88, 135].

Motion information can be treated in two ways. The first approach is to estimate a dense motion field in the image ; each pixel is assigned to a motion vector. Block matching methods are often preferred for their simplicity. In the second approach, a region undergoing coherent motion is depicted by a parametric model. The support of motion is determined either from the estimation of the motion field or on the basis of homogeneous regions with respect to the spatial features. The methods related to the first approach are referred to as favoring temporal information. Otherwise, methods favoring spatial information start from segmented regions to estimate the motion between frames. Finally, most advanced methods performs joint estimation of motion and support. Once the spatial segmentation is achieved the resulting segmentation or object model is tracked over the sequence. The basic scheme for motion-based method is shown fig.2.2.

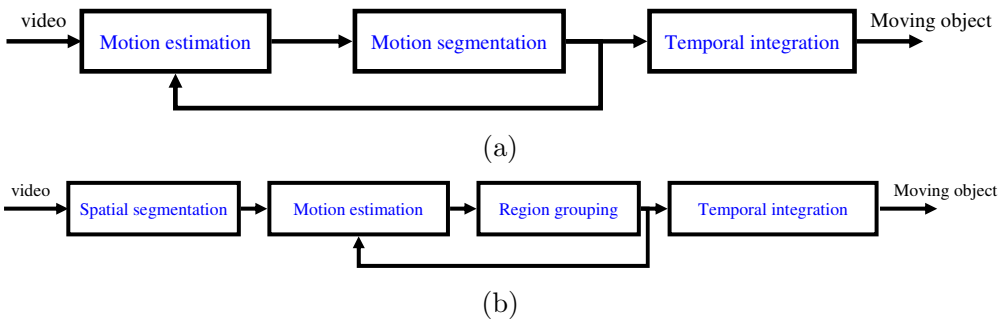


Figure 2.2: General scheme for motion-based segmentation. (a) Methods favoring temporal information. (b) Methods favoring spatial information.

Segmentation favoring temporal information

A traditional approach in motion segmentation is to estimate first a dense motion field, followed by a segmentation based on motion vectors. We distinguish in this section three characteristic types of approaches: segmentation by dominant motion estimation, Bayesian motion segmentation and multiple robust motion estimation.

Segmentation by dominant motion estimation

A reference method for motion estimation and segmentation is the work of Wang and Adelson [141]. The iterative scheme for motion estimation and classification is typical of motion-based segmentation methods. The principle is to extract an object with dominant motion once at a time. The decomposition of remaining objects is then performed on the image parts that are not associated to the dominant motion. In this method, each motion is associated to a layer related to the plan of an object in the real 3D scene. Starting from local motion estimation, an affine model is computed for every block of the image. For this purpose, motion segmentation consists in finding planes in the motion model parameter space. A clustering of affine models enables to group regions in the image by minimizing the distortion between the locally estimated motion and the predicted motion model, while still maintaining a low number of clusters. The support for each motion is therefore obtained by classification of blocks. In order to obtain coherent regions, the groups that contains disjoint supports are split. Similarly, regions below a certain size are pruned. This procedure is reiterated until new assignments occurs. For the following frames, the procedure restarts at the region classification stage.

The difficult part in this procedure remains the construction of the support of motion from the blocks. This can fail if there is no dominant motion in the image. For instance the objects may be small or undergo non-rigid motion. A second problem lies in the use of motion parameter space for clustering. Indeed these parameters do not have the same dimensionality. Zero order parameters have uniform sensibility to spatial coordinates whereas first order parameters are most sensitive a the border's area i.e. while moving away from the motion model origin. Moreover, a given optical flow may be described by several different

parametric representations [1]. This implies that two regions with similar motion may turn out to have very different motion parameters and, thus, may end up in different clusters. More recently, Ke et al. [62] proposes to reduce the disparity of the clustering by projecting the motion parameter space in a subspace. The reason was that motion of rigid objects lies in a 3-dimensional space.

An improvement to this technique was to consider pyramidal decomposition of the motion field. Irani, in [60], segments each level of the pyramid into foreground or stationary regions, starting from the lowest resolution. The segmentation of the current level is based on the labels (class) of the previous level and a local motion measure defined for each pixel. This type of approach can improve the classification of regions [60]. These top-down approaches are characterized by simplicity and low computational complexity. However, the process of successively determining the characteristics of the remaining dominant motion imposes an artificial hierarchy among the objects in the scene.

Bayesian motion segmentation

Other techniques have also been developed to enhance coherence of the extracted regions. Bayesian estimation methods provide an elegant framework, modeling the statistical distribution of motion or intensity of the image. They have been widely used in practice for solving the segmentation problem. The general idea is to estimate a segmentation S, considered as a label field, given some set of observations O. In this context, Markov Random Fields (MRF) models have been widely adopted. The general idea is to maximize both likelihood of the observations given the labeling while maintaining spatial coherence with a prior model on the distribution of the labels. This prior model is given by a Gibbs probability distribution function defined by a neighborhood system and a potential function reflecting prior knowledge and constraints on the segmentation process. Thus, the segmentation problem is transformed into an optimization problem: finding the segmentation which maximizes the MAP (maximum a posteriori) of the label field, given the observations.

Chang and Tekalp [23] take into account both intensity and motion field as observations. The potential function of their MRF is composed of a spatial coherence term and a temporal term which encourages spatiotemporal connectivity of the regions. For the conditional probability, the difference between a pixel intensity value and the mean value for the region as well as the difference between the motion vector and the displacement obtained from the motion model are Gaussian distributions. The influence of these two terms is locally related to the displaced frame difference (DFD): the motion term is dominant when the motion vector is reliable (low DFD value), and the intensity term is used in areas with unreliable flow (high DFD values). Iterative Conditional Modes (ICM) algorithm is used to solve the optimization problem.

This type of technique partitions the scene into small region homogeneous in terms of spatial and motion features. For the semantic foreground/background segmentation problem, a single object should be obtained. In this case, observations are often given by change detection masks. Change detection algorithms starts by computing the gray value difference image between two consecutive frames, followed by post processing operations and a decision rule to classify pixels as foreground or background. In the work of Paragios

and Tziritas [100], the likelihoods are modeled with a zero-mean Gaussian distribution, and the prior consists both of a spatial and temporal continuity term. The labeling problem is then solved by using ICM algorithm over multiple scales. Change masks are easy to compute, but the interior of the object will be not detected if the object is insufficiently textured.

Robust estimation of multiple motion

More recently, Xu et al. [144] tackle the problem of multiple and complex motions. The principle is to perform estimation of multiple models in competition to each other. Their method aims to overcome the deficiencies of those estimating recursively the dominant motion, in particular due to the imperfect detection of the outliers. For this purpose, they use a robust estimation technique, the Mean Square Robust Estimator (MSRE) and propose an adaptation for multiple motions. Moving regions are represented by a quad-tree. This particular structure decomposes hierarchically the images into blocs. The root node represent the image, and each node can be divided into 4 equal-sized blocs in the next level. Each block is associated to a parametric motion model. A criterion for motion coherence is defined by the squared difference between the dense motion field and the model-generated motion vector. A block is decomposed until it verifies the coherence criterion. When the quad-tree is completed, adjacent blocks are grouped if they undergo similar motion. A statistical test based on the distribution of the residuals of motion estimation error is performed. The Kolmogorov-Smirnov hypothesis test is used to check the correspondences between the two distributions. After this step, the number of motions in the image is known. The likelihood of the observed motion field given the parameters of the models and supports is maximized using a M-estimator. This family of robust estimators enables to limit the influence from important errors. Iterative Recursive Least Squares (IRLS) is used to simulate the robust norm with a weighted least square estimation method. The residual error over all regions is computed adding the residual error on each region, selecting the most coherent model that has the lowest residual error for this region. The weights are recomputed from the residual error and the least square problem is solved to provide new motion parameters. The procedure is reiterated until there is no more change in the assignments.

Segmentation favoring spatial information

The main problems related to the estimation of the optical flow are *occlusion* and *aperture*. The occlusion problem arises when the motion is estimated in covered or uncovered areas where no correspondence between the two images exists. The aperture problem is that an area can be considered locally as stationary whereas motion is actually present. This arises in flat or untextured regions, where the intensity gradient is low. Because of the limited accuracy of the optical flow, spatial features such as color, texture and edges have been also introduced for segmentation. Considering spatial regions reduces noticeably the complexity of motion estimation and is shown more robust than the pixel level [122].

Methods based on spatial segmentation intend to subdivide the image into homogeneous

regions. In general, coherent moving regions occupies a larger scale than color regions. When this assumption is not respected, an oversegmentation of the image is produced. The watershed technique became very popular for this purpose [140]. This morphological approach combines region growing and edge detection. The magnitude of the image intensity can be considered as a topological surface. Local minima (or seeds) are represented as holes punctured in the surface. The surface is then immersed in a lake. Water progressively forms catchment basins. When the water floods over two basins a dam is erected, separating two minima. At the end of the flooding process, each minima is surrounded by dams, forming a partition of the image. A practical implementation of watershed is given by the immersion process of Vincent and Soille [138]. The number of regions obtained at the end of the algorithm (granularity of the segmentation) can be monitored by a region grouping criterion. Haris [58] proposes a criterion favoring the grouping of small-sized regions with similar colors.

In [36], spatial partition is obtained using a open-close per reconstruction morphological operator, enabling the to produce flat zones while preserving contour information. A K-means clustering on luminance values complete the procedure. Affine motion models are computed from these affine regions. In order to take into account a possible failure of segmentation, a robust estimation is performed. Regions that do not have coherent motions are split using a K-means on their luminance values, while regions with similar motions are grouped by clustering in the parameter space. K-medoid is preferred for this task, being less sensitive to outliers.

Chang et al. [24] use a comparable technique for initializing spatial segmentation in the first image. Color quantification is performed in the perceptually uniform HSV color space, followed by median filtering to prune insignificant details in the image while preserving contours in the image. In their grouping technique, edge map is used in order not to merge regions that are clearly separated by a contour.

Deng and Manjunath, in [33], uncouples color quantization and spatial segmentation. Pixels values are replaced by their color class labels, forming a class-map of the image. Then, they define a cost function J which indicates the proximity of region pixels to their contours. A criterion for "good" segmentation is therefore obtained by minimizing J over the image. In practice, a region growing technique is implemented, considering the local minima of J as seeds in the image. The method also enables to compute motion explicitly, by expanding the regions on a spatiotemporal neighborhood. For this purpose, the J measure is then employed to detect reliable areas for propagating regions, and avoid confusion at the neighborhood of region borders.

Region and object tracking

Once the motion models and support regions are found in the first images, the temporal correspondence between images should be established.

Adopted tracking strategies fall in different categories:

- Matching of spatial segmentations at time t and $t+1$.
- Temporal projection of the spatial segmentation from frame at t to $t+1$.

- Tracking of an object model.

Region matching

When the spatial segmentation of two images is run independently, temporal tracking is done by matching the partition in consecutive images. In the method of Deng et al. [33] mentioned before, seeds in the first frame are assigned as the initial objects. Then, seeds in the current image are projected in the previous image. The temporal matching is based on a simple overlap measure. Seeds that falls in one object of the previous frame are assigned to this object. If more than one object intersect, these objects are grouped. Then, stable objects that have a minimum life span are retained. Del Bimbo [30] searches for similar regions in terms of location and color. Location similarity combines position, size and overlap of the regions. Color similarity is done with a customized distance in $CIE\ LUV$ space. Regions are tracked by finding the match that maximizes the overall similarity between image regions. A minimum search path technique is employed to solve the problem.

When a region is segmented at a fine scale, variations between successive images may occur frequently. To overcome the temporal stability problem, Gomila et al. [53] proposes to construct a hierarchy of nested partitions. The decomposition of each image is obtained from a morphological technique. To establish the correspondences between two images, they first compute the intersection between partitions. The regions that do not find any satisfying correspondences are further divided according to the hierarchy, until similar parts are found. The partitions of the two images are therefore modified so that they are closer to each other.

Temporal projection

Segmenting each image remains costfull and increase computation time, especially when requiring both spatial region growing or accurate motion estimation. Methods by temporal projection initially segment the first image. Spatial segmentation techniques mentioned in the previous section can be used for this purpose. This approach assumes that a motion model is available for every region of the image. Next image is then roughly segmented by compensating the motion of each region of the previous image. Areas that are uncovered or covered by several regions are then resegmented locally. The method of Wang [140] fits in this framework. Starting from spatial segmentation into moving objects, a projected partition is obtained by displacing each object following a linear motion model. To take into account projection errors and superposition, reliable parts of the object obtained by morphological erosion are extracted as markers. A modified watershed procedure enables to track the previous regions while authorizing the creation of new objects. To this aim, the marker's area are imposed as initial catchment basins. The flooding procedure starts by extending the initial basins to a predefined level. At this stage the basins that have not been labeled are considered as new regions. Indeed, new objects are likely to be flat zones surrounded with high watersheds (gradient value). Finally, the flooding terminates by extending both types of basin.

In [24], interframe projection or regions is also followed by a local intra-frame segmentation. They use an iterative clustering algorithm to merge regions with the smallest color distance inside the uncertain areas. These new small regions are then merged to their neighbors by morphologic open-close algorithm. In this way, existing homogeneous color regions are tracked frame to frame.

Bonnaud [17] introduces a method for tracking multiple objects based on boundary adjustment. A boundary-based structure is used to represent the segmentation. In this structure, a node connects three or more neighboring regions and an open boundary connects each couple of nodes. The partition is created in the first image using a spatial segmentation technique based on a Minimum Description Length (MDL) criterion. The boundaries are compensated and adjusted during the sequence in several steps. First, motion model is estimated for each region and its boundaries are motion compensated. The displaced boundaries are then connected with geometrical criteria, and new nodes are created for the image. Then, they are correctly placed in the current frame by minimizing an energy function that measures the deformation along the boundary area. The approach shows good localization and temporal coherence for significant boundaries but can be hampered with block artifacts in case of low-contrasted boundaries and too small regions.

In [45], the partition is projected using a block-matching approach. Blocks that belongs to the partition borders are used to determine uncertainty areas. The segmentation of these local uncertainty area is facilitated by the use of a spatial irregular pyramid. At the first level, existing regions and pixels in the uncertainty area are gathered to form an adjacency graph. A decimation procedure is performed in the graph to attach progressively uncertain pixels to existing regions using color similarity.

Model tracking

In the case of object segmentation, region tracking is generally improved (more accurate boundaries) with matching of object models instead of regions. Complete object masks for an image can be obtained by merging regions obtained from spatial segmentation. A technique for merging moving regions is to define a similarity measure based on motion compensation error or increment of motion compensation error. In the first case, the criterion is governed by a global threshold. In the second case adjacent regions are merged if the compensation error using the motion model computed from the union of the two regions is significantly lower than the compensation error on each region [140].

In the work of Moscheni [88], a spatiotemporal similarity measure between regions is constructed from spatial and temporal similarities. Two distinct hypothesis tests are performed: the spatial test verifies if the luminance in the neighborhood of the border of the two regions is similar, i.e. if their contrast is sufficiently low. The temporal test checks if the distribution of the residuals obtained from compensated motion-models are similar. A modified Kolmogorov-Smirnov (KS) test is used for this purpose, reducing the influence of outliers when comparing distributions. Quasi similar measures are used by Xu et al. [144] but moving regions are first grouped before using the KS test and the spatial test which groups low-contrasted regions is performed afterwards.

Besides region merging techniques, object mask can be obtained from other techniques.

The most common one is to compute a change detection mask. After compensation of camera motion, object mask is constructed by pixel or block classification into foreground or background. A technique based on morphological operations is proposed by Meier et al. [86]. They apply a morphological motion filter on the flow field, keeping only components that are not following the dominant motion. The object masks, or independently moving components (IMC), are obtained by the residue of the difference between the original and motion filtered image.

Once the partition into objects is obtained of the first frame, an object model is computed. In the method of Meier et al. [86], object is represented by a binary edge model. Tracking is based on matching the model edges with the edges of the next image. The comparison is performed efficiently using the generalized Hausdorff distance. To handle rotation or changes of shape of the object, the model is updated in two steps. First, the rigid part of the object (slow motion) is represented by selecting directly the edges in the new image that matches the previous model. Second, edges that belong to the non-rigid part (fast motion) of the object are selected in the new image from the IMC mask. Finally, accurate video object plane (VOP) and boundaries are obtained from filling-in and line processing techniques. However, the selection of the edge model must be done carefully, being sensitive to clutter and altered in occlusion areas.

Xu et al. [144] use a similar binary edge model to track the object. However, the model update procedure is different and can reduce the tracking errors substantially. After detection of moving objects in the first frame, a change detection mask is computed for each object, the image difference being compensated with the object motion (translation). This enables to place object and background markers on the image, along with an uncertainty area. The object contours are finally obtained with the watershed transform, which is applied on the morphological gradient of the image within the uncertain area.

The advantage of these object tracking methods is the efficiency and good accuracy of object boundaries, thanks to the efficient matching and dynamic update mechanism. However, if part of the background is included in the model in one frame, it will be also retained for all successive frames of the sequence.

2.1.3 Spatiotemporal methods

2D+T Segmentation

Previously we reviewed motion-based methods that predict segmentation of the current image by projection of the previous segmentation [110, 119, 140]. However, the segmentation may move away from the optimal solution as new frames are being segmented, errors being incorporated in the model. Segmentation in 2D+T space consists in extracting elementary regions on frames, and group them into spatiotemporal volumes according to a given criterion. In this way 2D+T methods aim to enforce temporal constraints on the long-term to maintain consistency of the labeling between frames and to enhance coherence of the segmentation.

As mentioned before, region information helps to the robustness of the motion estimation and to reduce the complexity. At the region level, adjacency graphs are used to represent

various relationships between regions. In this context, methods have benefited from MRF framework for temporal consistency of the segmentations. The MRF neighborhood system is naturally represented by the region graph, each region being a single site. Patras et al. [102] the labeling of a new image regions is based both on motion hypotheses, image intensity and the estimate of the labeled field of the previous frame. An explicit term in the Gibbs potential indicates the correspondence of the label field of the previous frame, the motion hypotheses and the current labels. This term can be simply expressed by counting the number of pixels whose label is different from their projected counterpart. MAP estimation of the label field and motion models is performed with a Expectation-Maximization (EM) procedure. In a first stage the region labeling is optimized given the motion parameters. In a second stage, motion estimation is performed considering the labeling of regions. Extensions of the method will be to relax the temporal constraint to handle partial occlusion. The method offers several advantages compared to other region-based methods.

In the work of Fablet et al. [40], a region is considered as an object if its motion is not conform to the dominant motion. The approach is therefore dependent of the estimation of dominant motion. When multiple independent motion patterns are present in the scene (as noticed before), outliers of the model may not belong to a real object [88]. Second limitation is that static part may be not compensated [42]. To handle this problem, they have to further separate the static parts which is not embraced by the dominant motion from moving objects [42].

In the method of [51] the terms of the energy potential express similarity between motion models, number of regions and degree of adjacency between regions. In contrast, motion models are estimated separately on each initial region, assuming that each region has sufficient texture. This is not crucial in [102], as the motion models are estimated globally. A second difference is that in [51], the MRF model is used to initialize the prediction of a new label field, and not during the whole procedure as in [102].

To overcome the problem of static/moving regions, region labeling methods have also used memory that register the previous states of the pixels or the regions. In the foreground/background segmentation method of Tsaig [129], the segmentation is initialized by watershed regions, followed by a spatiotemporal merging stage which merges adjacent regions with similar color or motion. Segmentation is followed by an initial classification stage based on the change detection mask and a dynamic memory. As a simple memory that counts how long a pixel has been moving is prone to label erroneously background as foreground in case of fast motion, they propose a dynamic tracking memory whose values are updated according to the displacement of the regions. As a consequence, the memory is incremented in areas that are detected as moving regions, based on the previous tracking score. On the contrary, the memory slowly diminishes the detected background area. The classification of regions is then seen as a MRF labeling problem. In the Gibbs potential, the memory is used as temporal continuity term, preventing moving regions to be labeled as background even if they stop for a long period. In this way, the effect of incorrect labeling at one time is attenuated, achieving better temporal coherence.

Another type of methods exploits spatiotemporal neighborhood to maintain consistency of the segmentation. An early approach was the work of Deng [32] design of object-based

retrieval system. The approach is related to the MPEG-2 coding. A basic temporal unit is composed of a group of one intra (I) frame and 6 surrounding predicted (P) frames. The I frame spatially segmented using color and texture information, and the regions of P frames are obtained by affine motion estimation. Long-term spatiotemporal region segmentation is then achieved by region tracking between central I frames, as they are segmented accurately. Spatial layout, affine motion compensation and local features are used to perform one to one match of these regions. A region is not matched if the distance between one type of feature exceeds a predefined threshold.

Tao [126] formulates explicitly temporal constraints on motion, shape and appearance of the objects. Motion layers [141] are used to represent independently moving objects and are tracked over time. A layer is defined at each time by a a shape prior, a motion model and its appearance. The shape prior represents a global constraint used to regularize the estimated object shape. An affine motion model is used and the appearance model governs the distribution of pixel intensity in the shape prior referential. Temporal changes are reflected by introducing uncertainty in the motion and appearance models, in the form of a Gaussian noise. The layer model is updated for each frame by EM estimation procedure taking into account the aforementioned constraints that penalizes changes in the object motion, location and shape between frames. A major advantage of the method is that the ground layer and the objects compete with each other in the layer estimation using motion cues. This improves the robustness of the tracker against the background clutter and makes the tracking more resilient to distraction from other nearby objects.

A larger spatiotemporal neighborhood is defined in [146]. In their object extraction scheme, they first divide the video sequence into temporal non-overlapping sections. Each frame of the section is segmented and a RAG is constructed for each frame. A whole 2D+T graph is constructed based on the spatiotemporal adjacency between frame regions. Each graph is then partitioned into objects by minimizing a global cost function derived from the normalized cut, considering an unknown number of objects. As one graph contains few nodes (about 100), a genetic algorithm is used to find a solution to the partitioning problem. Finally, the section graphs are merged temporally considering the trajectory of the merged subgraphs: the region in the last frame of one subgraph is projected in the region in front of a subgraph of the next section; merging occurs when the two regions coincide.

Segmentation of the 3D pixel block (pure methods)

Methods that requires exact object motions suffer from different problems. In traditional approaches based on motion estimation, the motion vector field is often erroneous due to camera noise and single affine model is often insufficient to depict big objects. Concerning region-based methods, occlusion and disocclusion areas can also introduce errors in the estimation. Furthermore, the interdependence of the motion model and its support make the optimization procedure complex. For scenes containing multiple objects or objects with non-rigid motion, these problems become very difficult to overcome in practice.

Pure spatiotemporal approaches stands as an alternative to the motion estimation problems by considering the video frames as a single 3D pixel volume. This category of methods extract visual structures simultaneously in space and time, forming continuous volumes

that represent meaningful objects of the sequence. In consequence, no tracking is needed to achieve temporal consistency. Viewing the scene as a whole has been inspired by Gestalt psychology (section 1.2.1). Taking as input the whole video volume and extract spatial and temporal descriptors for every pixel generates important needs in computational power and memory. As a consequence, these methods have only raised interest recently. However, it still remains difficult to apply them to large video databases. The techniques developed until now can be grouped into 3 categories. Morphological approaches extends 2D image morphological segmentation to the spatiotemporal domain [39]. Another approach is to associate pixel structures with space-time models obtained from clustering of spatiotemporal features [31, 55]. Finally, global approaches using graph-based segmentation in the spatiotemporal domain have been developed. A global partitioning criterion is defined to decompose the video into coherent volumes [112, 46, 76].

Morphologic segmentation

The first category relies on 3D morphological operations and aims to detect object edges in the spatiotemporal domain. Object boundaries form a spatiotemporal surface in the xyt space. Instead of using complex mathematical functions, Korimilli [71] depicts these surfaces, called temporal envelopes, with a collection of planar patches. To construct these planar patches, edge detection is first performed by extending Canny detector to xyt space. Each edge point is represented as a planar surface with the Hough transform ; planar patches corresponds to local maxima in Hough space. The envelopes are then detected by grouping similar planes using Gestalt principles. With this framework they achieve good motion segmentation in presence of noise, illumination changes and occlusion from perceptual organization principles without computing motion estimation.

In the work of Saban et al. [39], objects are constructed inversely by watersheds of a spatiotemporal edge gradient function. To avoid oversegmentation of the sequence, flooding is performed starting from a set of markers manually placed in the initial frame of the sequence. However, the method is sensitive to the location of these markers, which need to be approximately known. Main work remains to deal with long sequences, including occlusions, entry of new objects and fast motion.

Space-time models

A second type of approach aims to simplify the description of the video sequence by fitting a global space-time model applicable to the video block. At the most general level, objects are represented by a set of spatiotemporally connected pixels (volumes). Kompatsiaris et al. [67] modified the k-means clustering technique to impose spatiotemporal connectivity and adapted it for the extraction of spatiotemporal volumes. An object is characterized by its mean intensity over the sequence among with its mean differential motion and position in each frame. The distance from one pixel to an object is defined by combining these three features. K-means is then used to minimize the total distance between pixels and the object clusters. The algorithm includes a connected component labeling to ensure spatiotemporal connectivity and limits automatically the number of objects. A different clustering approach

is presented in [31]. Objects are locally represented as color patches with linear motion, forming a video tube or strand in the *xyt* space. A tube can be represented as a single point in a 7D feature space. These space-time features based on color, position and optical flow are extracted for each pixel. A global and compact model for the video sequence is then obtained by a mean-shift algorithm that performs hierarchical clustering of the pixels. This type of representation is useful for comparing efficiently the shots and give cues on the type of activity the shot may refer to.

Global models are rarely valid along a whole frame sequence. Greenspan et al. [55] uses a probabilistic model for video content representation. Each pixel is assumed to be drawn from a Gaussian distribution in the *Labxyt* feature space. Thus, homogeneous volumes are represented as blobs and the whole set of volumes as a GMM in this feature space. They further extend the spatiotemporal representation for long-term sequences. To handle complex motion patterns, a succession of spatial GMMs are estimated in overlapping temporal segments (BoF). GMMs are initialized in the first frame via an EM clustering algorithm, then are updated in subsequent frames. The initial number of classes can be determined from the Minimum Description Length (MDL) principle, but requires the estimation for different number of models. Appearance of new objects is handled by projecting the blobs in the image and grouping pixels that do not fit to existing blobs. Long-term Temporal coherence is ensured by matching the blobs in the common frames of two successive BoFs. Such representation is interesting for event modeling and detection.

Spatiotemporal graph segmentation

It is difficult to prove the superiority of one space-time model to another, the choice of the model depending on the initial hypothesis on the underlying structure of the objects and on the application domain. Graph-based segmentation does not make hypothesis of a particular representation and avoids the computation of space time models. Pairwise similarity between pixels is used as the cue for grouping.

Graph-partitioning techniques have been employed for segmenting video stack. In graph-cut approaches, the grouping is achieved by minimizing the energy to partition the graph, or *cut value*. To balance the size of the groups and their homogeneity, [113] proposes the normalized cut criterion (Ncuts) to search for cuts that maximize the similarity within groups while minimizing the similarity between groups. To solve this partitioning problem in large graphs, a spectral clustering approach is proposed. This consists in the resolution of a generalized eigenvector system involving a global affinity matrix which registers the similarity between every couple of pixels. A bipartition is then obtained from the second smallest eigenvector. Each partition can then be further subdivided by running the algorithm in the obtained parts. The procedure stops when an unstable eigenvector is obtained, i.e. no significant break can be made from the eigenvector. To overcome the problem of instability and discretization of the continuous eigenvectors, [46] takes the largest eigenvectors and clusters points by their respective components in these eigenvectors. A rigorous formalization of a problem can be found in [145], deducing an iterative algorithm to transform eigenvectors into binary indicators.

A drawback of these approaches is that considering the whole affinity matrix is not

tractable for an image and even less video sequences, considering $N \times N$ nodes, N being the total number of pixels. To reduce the complexity of the algorithm, Shi [112] restricts the spatiotemporal neighborhood to a local area and diminishes the density of neighbors by sub-sampling. However, increasing the sparsity of the affinity matrix affects the final segmentation results, diverging from the solution provided by the optimal criterion. To further consider long sequences, the affinity matrix is decomposed into blocks corresponding to pairwise frame relationships inside a temporal sliding window. The eigenvector system is then defined for each frame by updating partially the affinity matrix blocks involving the new frame, and solved efficiently by utilizing the previous eigenvectors for initialization. Fowlkes [46] overcomes the complexity problem by computing an approximate solution for the eigenvalue problem. The numbers of points in the sequence is decreased by sub-sampling, and the system is solved for the sampled points. The eigenvector components for the remaining points are then estimated using the solution for the sampled points using the Nystrom approximation.

Graph-based segmentation enables to handle different type of segmentations, using the same generic procedure. Shi [112] performs motion segmentation by considering local distribution of the optical flow components (or motion profiles) over a spatial window centered at the pixel location. Fowlkes [46] builds a spatiotemporal segmentation for various grouping cues, including color, texture, and optical flow within a Mahalanobis distance. As the spectral clustering performs clustering in the similarity space, the resulting groups tolerates smooth spatiotemporal variations, propagated through neighboring nodes of the graph.

More recently, graph-cuts have been used to solve more general graph labeling problems involving the minimization of a global energy function composed of a prior term and a likelihood term. Under certain restrictions of the energy potential, it is shown that the bipartite labeling problem equivalent to the minimum cut problem on an particular graph [66]. This framework has started to be employed for different computer vision problems, among which object segmentation [73]. A graph labeling method for foreground/background segmentation is introduced in [76]. The graph is constructed by watershed image segmentation and connected regions within a spatiotemporal neighborhood. Color models for object and background colors are represented with a GMM. The likelihood term corresponds to the distance of a region to the color model (background or object) and the prior term penalizes regions that are assigned different labels but shares the same color. A drawback of this approach is that it is not fully automatic and requires labeling of one or several keyframes to build the color models. Graph segmentation errors are located by the user. A 2D graph-cut is then used to refine the segmentation within the specified window.

Compared to methods based on explicit computation of a space time model, graph-based segmentation only uses spatiotemporal proximity and a defined similarity measurement to group regions. This avoid estimation of parametric models [55] and problems related to model convergence (local minima and dependence of initialization). Motion is taken into account implicitly considering edges within a spatiotemporal neighborhood, or directly within the similarity measure [112]. In particular, the graph-cut formulation enables to provide a global optimal solution for the segmentation problem, and is capable to handle local spatiotemporal variations within objects, or at least the object parts. However, the

partitioning problem is NP-hard and the estimated global solution for the whole sequence does not ensure that segmentation is locally adapted in practice because of simplification of the problem and complexity of the scene. Predicting the optimal number of cuts is also difficult, and to our knowledge, no good criterion has been proposed. Recursive 2-way cut is a potential solution to this problem but does not generally provide stable solutions [113]. Interest can be placed in pyramidal approaches, but only few work have been performed in this direction until know [50]. Thus most reliable methods for graph-based segmentation use a prior on the object model that constrain the solutions. Such type of models includes simple color models [76] and pictorial structures. However, this type of approaches are supervised and models are learned for specific objects.

The approach we present in this thesis is related to the graph-based spatiotemporal segmentation and spatiotemporal region merging approaches [88]. Spatiotemporal methods generate an important complexity, which make their use difficult for video databases and content indexing problems. We face this problem considering a bottom-up segmentation based on similarity between groups of pixels and regions. We also avoid considering estimation of motion field for the construction of low-level spatiotemporal volumes by considering spatiotemporal neighborhood in a 2D+T graph. Temporal coherence is maintained through the use of spatiotemporal contraints and a region matching process. To further group volumes into objects, we estimate simple motion models for the volumes and find volume patterns that undergo coherent motion, following the Gestalt principle of common fate.

Compared to the spatiotemporal 3D graph segmentation and motion-based region labeling approaches, complexity of the approach is kept at a lower level through the use of appropriate structures at different levels of representation. The complexity of grouping and cues used for grouping increases when considering pixel, region and finally object structures.

2.2 Spatiotemporal representation based on graphs

In this section, we describe a method for spatiotemporal segmentation of video shots. The shot is defined as a continuous actions in space and time. In order to construct the spatiotemporal representation of a video sequence (section 1.2.4), it is necessary to take under account different aspects simultaneously. The first aspect concerns the use of efficient structures for describing elements at each level of representation. The second aspect is to define a scheme for organizing the grouping task. As far as the last aspect is concerned, we need to elaborate appropriate algorithms for grouping and merging of coherent volumes.

With respect to these considerations, the main features of our approach are defined as follows:

- **Representation with adjacency graphs**: Different adjacency graphs are used according to the grouping level. At the pixel level, we consider local interactions with a simple 3D grid graph, which is represented as a forest for fast merging operations. At the region level, we can consider broader interaction using region adjacency graphs of the last frame pair.

- **2D+T domain**: Instead of processing the entire video volume at once, we append single frames to the video volume. The spatiotemporal grouping is updated using pixel data from the last frame pair.

- **Low-complexity graph algorithms**: The complexity of the merging algorithms is adapted to the grouping level. Pixel graph algorithms are derived from minimum spanning tree algorithm. The complexity of region graph analysis is theoretically higher, but far less region nodes have to be considered.

First, we give a short insight about the method strategy. Then we analyse the low-level representation and explain a region growing algorithm and its extension to the 2D+T domain, focusing on the necessary spatiotemporal merging constraints and matching techniques proposed. The overall theoretical complexity of the approach is examined and finally the properties of the method are analyzed experimentally through different segmentation examples.

2.2.1 Overview of the proposed method

The method intents to various to segment various type of scenes, including both static and dynamic contents that are commonly found within a video sequence. Unlike most of space-time segmentation approaches, the proposed workflow does not need computationally intensive clustering or global optimization algorithms. They are implicitly replaced by decomposing and simplifying the segmentation process into several stages.

A global view of the method is shown fig.2.3. In the following, we denote by $S_{0\to t}$ the spatiotemporal segmentation from times 0 to t and S_t the segmentation of a single frame F_t. The segmentation is first initialized on the first frame of the shot F_0. This step sets approximately the level of spatial details for the final segmentation. An efficient graph segmentation algorithm is introduced in section 2.2.3 and is used for this purpose. Once initialized, the segmentation S_t is obtained from the previous segmentation $S_{0\to t-1}$. We create a set of over-segmented spatial regions for each new frame F_t using an edge constraint described in section 2.2.4. Ideally, they correspond to a partition of the final regions S_t, except some new regions induced by motion. The grouping between $S_{0\to t-1}$ and S_t is done in three steps. To take into account region motion, we create new temporal edges linking regions in S_{t-1} to S_t by feature point tracking between frame pairs (section 2.2.5). Considering statistical properties of feature points within region couples, we can group most of dynamic regions. The linkage of remaining static regions is done with the spatiotemporal merging rules described in section 2.2.4. The merging is performed on a pixel neighborhood, considering both local and global properties of the current segmentation $S_{0\to t}$. As the projected segmentations S_{t-1} and S_t become close after this stage, we finally compare their region adjacency graphs to check the validity of new regions. In this way, we achieve incremental grouping of space-time regions by considering different levels of interaction between pixels, frame regions and spatiotemporal regions.

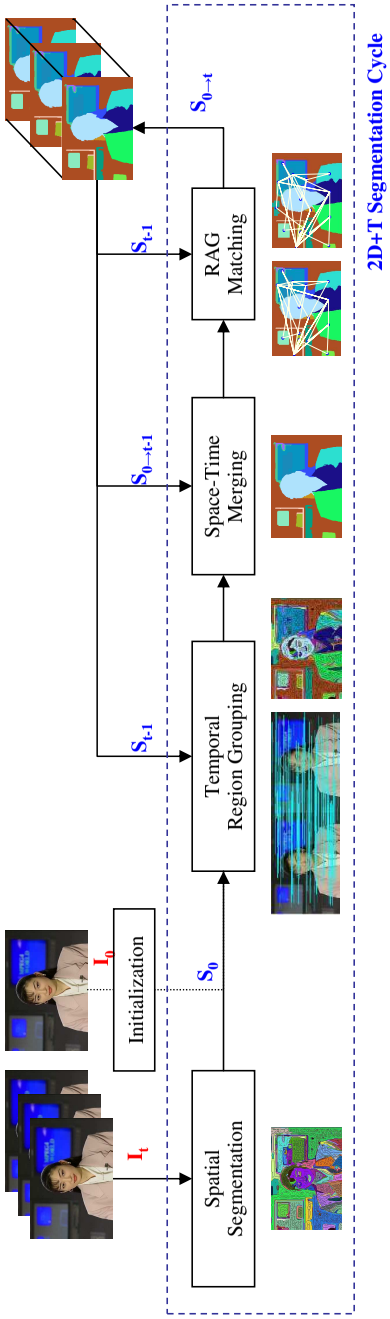


Figure 2.3: Scheme of the overall segmentation process.

2.2.2 Tree and forest representation

At the lowest level the graph is composed of pixel grid, each pixel corresponding to a node. When effecting groupings on the graph, there is usually two ways of representation. The first one relates to merging of the graph, performing vertex contraction that changes the graph structure. The second one consists in partitioning the graph into disjoint sets. Disjoint set data structures have been introduced by [48] for keeping track dynamically of the partitioning.

This structure has the following properties. Each set is identified by a representative, which is a member of the set. Given two nodes x,y, three operations are available:

- *Make-Set*(x): makes a singleton set with x.
- *Union*(x, y): takes two disjoint sets with representatives x and y respectively and creates their union. The representative is either x or y.
- *Find-set*(x): returns the representative of the unique set containing x.

In practice disjoint set data structures are naturally represented as forests, as illustrated fig.2.4.

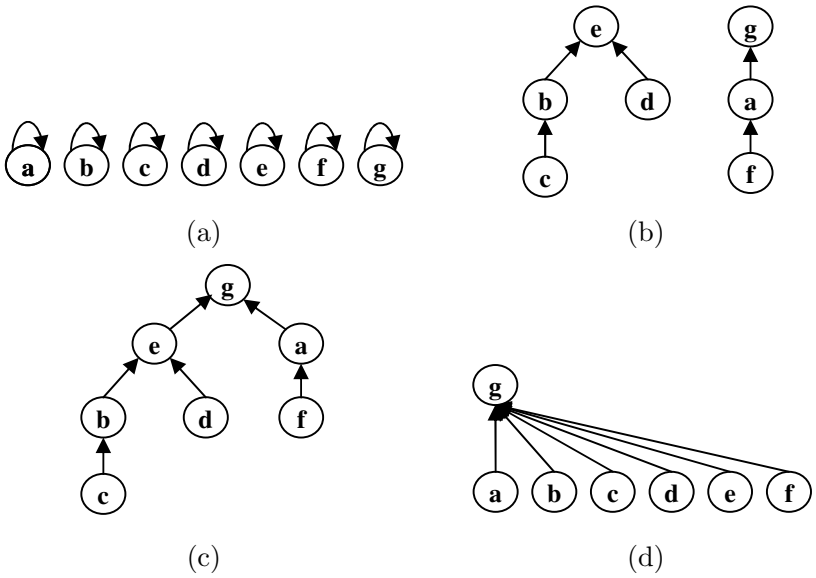


Figure 2.4: Disjoint set data structures and related operations. (a) Initial forest. Each node is a tree. (b) Forest after some grouping operations. (c) Union of sets represented by g and e. (d) Path compression.

Considering a pixel graph structure, the find-set operation becomes less efficient as the trees grow, visiting all parent nodes corresponding to successive groupings. The complexity is reduced by attaching the visited nodes to the representative (root node), as shown

fig.2.4(d). This mechanism is called path compression. Complexity for these structures will be detailed in section 2.2.6.

Now let's consider a bottom-up partitioning procedure. A grouping algorithm will start by creating the elements as singletons with the *make-set* operation. Any two elements will be compared by looking at their representative with the *find-set* operation. These elements can represent an edge connecting two groups. Two sets are finally merged with the *union* operation. The merging criterion can involve both edge or group attributes, represented as map. This scheme is illustrated fig.2.5 for the segmentation of a local grid. The advantage is that merging and accessibility of groups is fast. Properties resulting from the union of two groups are calculated with common merging operators defined in section 1.2.6. If properties have to be computed considering the support of the group, region representation can be constructed by scanning all the nodes in the graph with the *find-set* operations. As mentioned in section 1.2.4 this is only useful when the groups reach a consequent size.

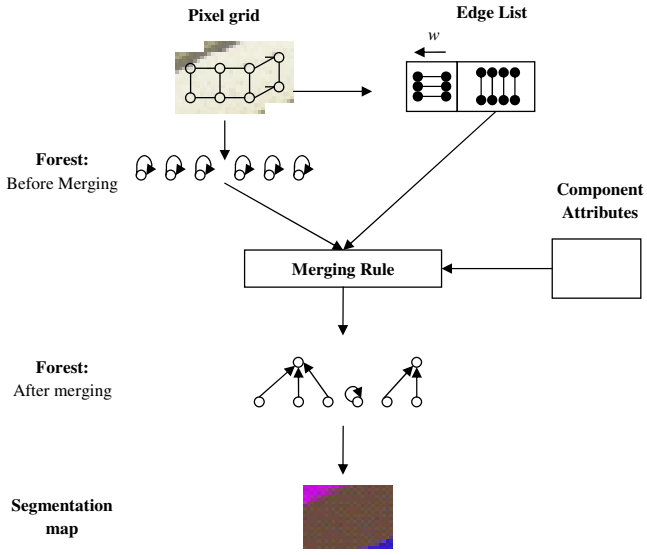


Figure 2.5: A bottom-up segmentation procedure for graph with a disjoint set data structure.

2.2.3 Region growing

A fast graph segmentation algorithm has been introduced in [43]. In opposition to spectral global clustering methods or local single linkage methods, the algorithm groups region based on both local and global properties. We consider the graph of pixel $G(V, E)$ introduced in section 1.2.4. Each edge e between pixel is attributed a weight $w(e)$ that represents their dissimilarity.

The algorithm aims to decompose G into a set of components $C_1, C_2, \ldots, C_k$, where each component $C_i, i = 1 \ldots k$ is built from a minimum spanning tree (MST). A MST is

a tree which connects every node of the graph and has minimum total edge weight. The key of the method consists in single linkage and adaptive thresholding. The former aspect enables fast processing with the disjoint data set structure while the latter considers both local and global aspects of the components. Single linkage merges two adjacent components by bridging them with one edge. A typical method for single linkage is to construct first a minimum spanning tree of the graph considering all edges, and to form components afterwards by pruning weakly connected edges. In contrast to other methods which assume that the regions are piecewise constant (with respect to color or texture, for instance), or are closed together in some feature space, the method is based on measuring the variability (or dissimilarity) of a boundary between two components and the variability between adjacent nodes within each component. The segmentation is obtained on the basis of the well-known Kruskal algorithm [72]. At the beginning the components are disconnected, i.e. that each component corresponds to a pixel. Similarly to the construction of the MST of the graph, two components are connected by an edge, which are taken in increasing order of their weight. Thus, the variability of each component grows little by little. To limit the expansion of the components a merging criterion has to be defined. Each component C_i is described by the internal variability $Int(C_i)$, which represents a characteristic value of the edges grouped in the MST. Two adjacent components C_i and C_j are compared by analyzing their common boundary, which is described also by a representative value $Ext(C_i, C_j)$. A schematic representation of the merging is shown fig.2.6.

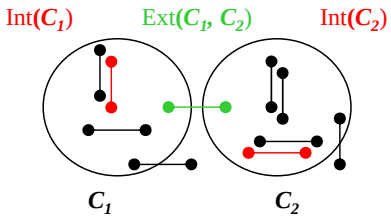


Figure 2.6: Merging of components. Characteristic values for the decision are based on their edge weights.

The decision to merge C_i and C_j is governed by the criterion $M(C_i, C_j)$ which takes into account both local and global properties of the components:

$$Ext(C_i, C_j) < \min(Int(C_i) + \tau(C_i), Int(C_j) + \tau(C_j)) \tag{2.1}$$

Equation 2.1 allows to merge components if the dissimilarity in the boundary of C_i and C_j is lower than the dissimilarity within each component, plus a tolerance τ which decreases with the component size. Due to the tolerance, fewer merge are done when the region size increases. With this adaptive rule, the segmentation is sensitive in areas of low variability whereas it remains stable in areas of high variability. This criterion is computed very efficiently by following the MST construction. The external and internal values are given

by:

$$Int(C_i) = \max_{(a,b)\in MST(C_i,E)} w_{ab} \tag{2.2}$$

$$Ext(C_i, C_j) = \min_{(a\in C_i, b\in C_j)} w_{ab} \tag{2.3}$$

The tolerance is adjusted with respect to the component size.

$$\tau(C_i) = \frac{K}{|C_i|} \tag{2.4}$$

Each component can be characterized by the largest weight in the minimum spanning tree, and the components are compared with the smallest edge weight linking the two components. Components with a single element have value of $Int(C_i) = 0$. Practically, $Int(C_i)$ corresponds to the last edge added to C_i and $Ext(C_i, C_j)$ is the weight of the current edge being processed.

The main property of the algorithm is to respect a criterion of *good* segmentation. A segmentation S is said to be *too fine* when the merging criterion M is false between a pair of regions, or inversely too *coarse* if there exists a component that can be partitioned in a segmentation which is not *too fine*. A good segmentation is then neither *too coarse* nor *too fine*. A second advantage of the graph algorithm remains the genericity of the framework. The algorithm is applied for fast image and spatiotemporal segmentation considering 2D and 3D grid graphs. Frames are first preprocessed by smoothing with a Gaussian filter of scale σ, for noise reduction within the image. The edge weights are built using color distance between pixels.

Intuitively, the Ext measure (eq.2.3) for two components could be enhanced by considering the distribution of the weights rather than the minimum weight in the cut between C_i and C_j. A common measure in robust statistics is given by the p-quantile function, which returns the value x such that the cumulative probability that the weight value is at most x is p. However, the modification of the merging criterion with the quantile based measure makes the segmentation an NP-hard problem. Indeed the minimum ratio-cut problem, which is known to be NP-hard, can be transformed to the problem of finding a good segmentation [43].

2.2.4 Adaptation to the 2D+T domain

As mentioned in section 2.1.3, segmentation of the whole frame block generates important computational cost and memory requirement. The region growing algorithm requires that the data and graph structures (nodes and edges) are available. When the length of the sequence grows, the efficiency is either hampered by the increasing access time to elements (list), or by the required memory which increases dramatically. A $2D + T$ procedure constructing a segmentation $S_{0\to t}$ from $S_{0\to t-1}$ will have the following advantages:

- Reduce the memory load considering data at times t and $t-1$, keeping the same order of complexity.

- The algorithm becomes causal, a new segmentation being rendered at each time t.

This adaptation is not straightforward. The property of "good segmentation" is lost if the image F_t is segmented without considering F_{t-1}, or more generally if the segmentation is built by parts. If the components at time $t-1$ and t are built separately, we cannot merge directly two spatial components C_i and C_j using the rule of eq.(2.1). Indeed, the rule remains effective on the condition that the components remain minimum spanning trees. The temporal edge weights may be lower than the internal values of each component. In such case the merged components are no more minimum spanning trees. This problem is illustrated in fig.2.7.

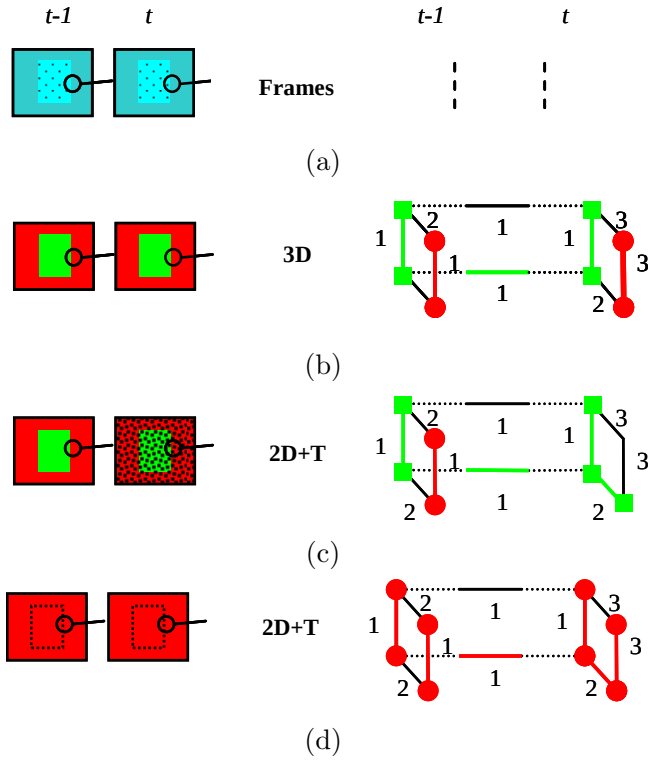


Figure 2.7: The 2D+T graph segmentation problem. (a) Input frames at times t and $t-1$. (b) 3D merging. (c) 2D+T merging. (d) End of the 2D+T merging.

In the example, frames F_t and F_{t-1} are composed of two regions: the biggest one in dark blue, the smallest one in light blue (fig.2.7(a)). Corresponding segmentation is shown (fig.2.7(b-c-d)), a local area at the boundary of the two regions being represented by the grid at the right. The grid is composed of 4 connected nodes in each frame, and 2 temporal edges are represented. The numbers further indicate the edge weights. In the 3D algorithm (fig.2.7(b)), spatial and temporal are taken in increasing order, thus the edges with higher weights that links the two regions are only considered at the end of the process. The size of each region is then important, so that the internal value is low and the merging criterion

is not verified. Thus two distinct regions are obtained. In fig.2.7(c), a 2D+T merging is considered. $S_{0\to t-1}$ is constructed from F_{t-1} and then $S_{0\to t}$ from the spatial edges in F_t and temporal edges from F_{t-1} to F_t, i.e. that region growing in S_t and temporal merging between F_t and F_{t-1} are achieved simultaneously. When the merging between components in S_t and $S_{0\to t-1}$ is tested, the internal value of the latter will be higher than in the 3D case (more spatial edges have been added to the group). As a consequence, an edge can separate two regions in $S_{0\to t-1}$ and group them in S_t (components with green squares in fig.2.7(c)). In other terms, the segmentation becomes too coarse. In fig.2.7(d), the two regions have fusioned before the end of the merging process (the red component in $S_{0\to t-1}$ is merged temporally with S_t within the dark blue region as in the 3D case), so the structure found in the 3D algorithm does not appear anymore.

As most bottom-up algorithm, the MST is a greedy algorithm ; once the decision to merge two components is taken, it cannot be reconsidered later. From an other side, it is the basis for simplicity and computational efficiency. To circumvent the problem of temporal stability of the segmentation, between frames (segmentation variations in different images), the linkage can be performed considering an over-segmentation of the new image, i.e. imposing that S_t is too fine. The fineness of S_t is then adapted considering that $S_{0\to t-1}$ is a good segmentation.

Edge constrained segmentation

Instead of stopping arbitrarily the grouping process in the new image, we impose a boundary constraint that limits the growth of the component in the new image. For this purpose, a contour map $\mathscr{C}_t$ obtained from Canny edge detection is used. For a graph edge $e = (a, b)$, this constraint $L(a, b)$ is expressed as:

$$L(a,b) \;:\quad \forall e = (a,b),\, w(e) = \infty \text{ if } a \text{ or } b \in \mathscr{C}_t \tag{2.5}$$

Thus, the propagation is done in areas of low-variability, resulting in more homogeneous local components. This is particularly useful at the end of the procedure, adding only most strongest edges to MST of the components. Another view of this heuristic is that we retain only the lower part of the edge weight distribution in the computation of the internal value of components. Edge-constrained segmentation and original image segmentation can be compared in figure 2.8. We can see that the constrained method (c) results in a decomposition, or over-segmentation of the unconstrained one (b). In addition, since we use Canny detection, edges with local intensity variations are also pruned so that the components are more homogeneous.

Spatiotemporal merging rules

With this boundary constraint, we adapt the merging criterion of eq.2.1 based on the spatiotemporal continuity of the regions. Temporal edges between F_t and F_{t-1} are listed to decide for the grouping of the components they belong to. The internal value of one

(a) (b) (c)

Figure 2.8: (a) Input Image. (b) Unconstrained image segmentation used as initialisation. (c) Edge-constrained initialization of the new regions.

component is represented by the mean value of the MST μ_i:

$$\mu_i = \frac{1}{|C_i| - 1} \sum_{(a,b) \in MST(C_i,E)} w_{ab} \tag{2.6}$$

Ideally, it should represent a certain quantile of the distribution, but this would lead in increasing considerably the complexity.

τ_L and τ_G are local and global adaptive thresholds. The criterion first relies on a local neighborhood W_L (fig.2.9). The considered neighborhood is used to group common region parts in F_t and F_{t-1} with sufficient overlap and similar features, when the distance between connected node pairs is low. The global part of the criterion is also used to check if the variability of the common regions is the same. This takes into account regions which has slightly moved or whose illumination has locally changed. Given these considerations, the space-time merging constraint is expressed as:

$$\max(W_L) < \tau_L(C_i, C_j) \quad \text{and } |\mu_i - \mu_j\| < \tau_G(C_i, C_j) \tag{2.7}$$

where

$$\tau_G(C_i, C_j) = \max(T_G, p_G \min(\mu_i, \mu_j)) \tag{2.8}$$

$$\tau_L(C_i, C_j) = \min(T_L, \mu_i) \tag{2.9}$$

τ_L and τ_G are local and global adaptive thresholds that controls the degree of variations authorized on the components and are thus defined on every couple of components (C_i, C_j). For local properties, we define a four edge neighborhood W_L (fig.2.9(a)). The neighborhood is considered as homogeneous if the maximum weight is weak compared to the variability μ_i and T_L (fig.2.9(b)). Small values of T_L limit grouping in inhomogeneous areas. In this way, we do not merge component from edges with high variability. For global properties, we check if the components have similar homogeneity. For regions with strong homogeneity (H.R in fig 2.9(c)), we consider directly the distance between μ_i and μ_j. For more variable components, a tolerance p_g is accepted on the relative error between μ_i and μ_j. Small values of T_G and p_g limit the temporal variation of the components. Thus, by combining these two aspects, the merging occurs in space-time areas of low local variability on globally coherent components.

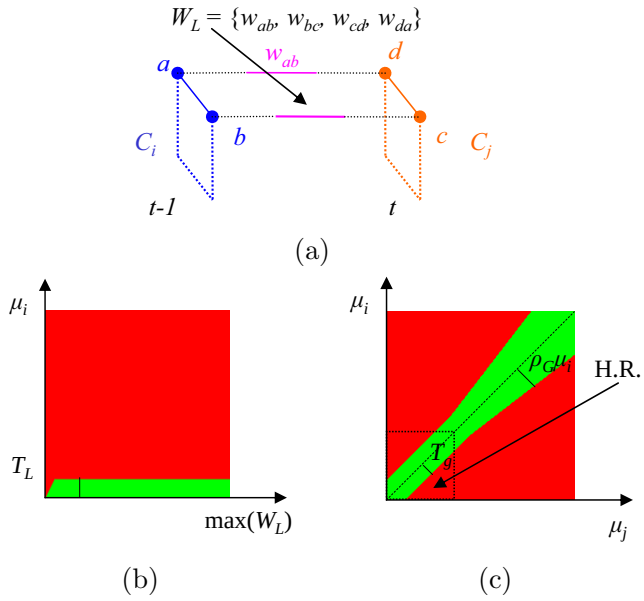


Figure 2.9: Space-time grid based merging. (a) Space-time local neighborhood W_L. (b) Local acceptance zone. (c) Global acceptance zone distinguishing between homogeneous regions (H.R.) and variable regions.

2.2.5 Temporal region grouping and spatiotemporal consistency

In this section, we first describe the temporal grouping of regions based on dense feature points and space-time constraints. Then, we show how RAGs are employed to check efficiently the stability of the regions.

Creation of temporal dynamic edges

The fixed structure of the 3D graph can be problematic for the displacement of the regions, the spatiotemporal tree formed may be not follow the actual color flow, seeing the volume as a tube.

To tackle the issue of region matching, previous approaches have considered different techniques: optimizing the fitness of the overall match [30], using a relaxation technique [53], or performing one to one match after motion compensation [32] of the regions. The regions obtained at the previous time S_{t-1} are assumed to be of various shape, size and with possible non-rigid motion. Therefore, motion compensation could not be used over entire regions. In addition, regions might be partially occluded in the new frame, so that one region can have several matches in the next frame. Given that the new regions are oversegmented, our solution is to track partially these regions by spreading a population of feature point trackers $\mathbf{P}$. in this way, we extend the graph by creating dynamic temporal edges, which enables a broader connectivity between segmented volumes and regions in the new image. Regions linked with these dynamic edges are then temporally grouped

considering space-time constraints. In this way, no hypothesis is made on motion models and we avoid optical flow computation on full regions.

Feature point detection

Feature point trackers have been proposed by Tomasi et al. [114], in the aim to recover structure from motion. Good feature points are extracted from corners or textured regions. Given one image I and a local window B candidates points are found by analyzing a 2 by 2 matrix of intensity changes, given by:

$$\mathbf{M} = \sum_{(x,y)\in B} b(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{2.10}$$

where $b(x,y)$ is the window mask, e.g. Gaussian-like for noise reduction. Good features are then located by examining the eigenvalues λ_1 and λ_2 of M. In [114], corners, textured regions are considered as most valuable to estimate their motion.

However, these points are likely to correspond to region borders, thus hampering the matching between regions. Therefore, we rather consider flat points that we can expect to lie reliably inside regions, at the expense of motion precision. The flat point detection is given by:

$$min(\lambda_1, \lambda_2) < \lambda \tag{2.11}$$

where λ is a predefined threshold. The threshold map is then reduced to a set of disjoint points through non-maxima suppression. Feature points are then tracked using a simple block matching algorithm. Figure 2.10 shows typical feature point detection and tracking.

We can see that feature points are concentrated in homogeneous areas (fig.2.10(a)). Even if some tracked points are inaccurate (fig.2.10(b)), they can be considered as outliers in the statistical distribution of the points.

We explain how we use these points for region grouping in the next section.

Construction of the matching graph

The feature point matches are considered as potential edges between components. For grouping regions temporally, we consider the subgraph in the graph G that contains the components present in frame F_t and F_{t-1}, and the dynamic edges linking the components.

At this stage, a region-based representation is more appropriate than the grid to handle larger interactions. We construct a graph $G_T = (V_T, E_T)$ that corresponds to the components between two consecutive frames. The node set V_T contains two subsets (V_t and V_{t-1}) and the temporal edge set E_T is constructed from the feature points.

The vertex representing a region a is $v_a = (a, \mu_P(a))$ where the attributes $\mu_P(a)$ characterize the population of feature points P_a in a. A relationship between two regions a and b, $\xi_P(a,b)$ contains the list of matched feature points between a and b, $P_{a,b}$. The construction of the graph is illustrated fig.2.11.

Each feature point is described by its velocity $\mathrm{v} = [\mathrm{v}_r, \mathrm{v}_\theta]$ where v_r is the displacement norm and v_θ is the motion orientation. In case there is substantial background or camera

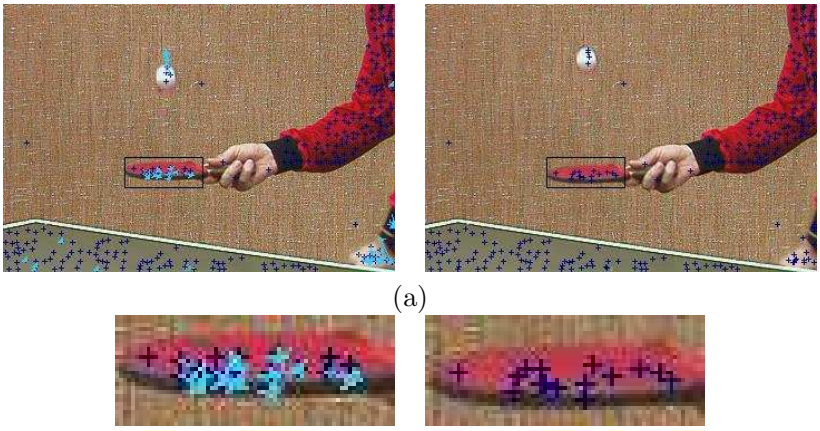


Figure 2.10: (a) Distribution of feature point matches. (b) Feature points inside the racket. Arrows represent the estimated displacement.

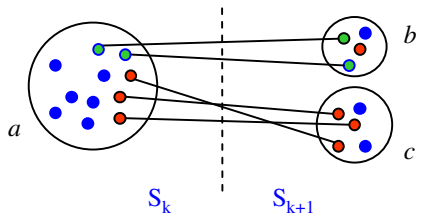


Figure 2.11: Temporal region grouping. Each region (circle) contains a set of feature point samples. Feature point matches establish new temporal edges between components.

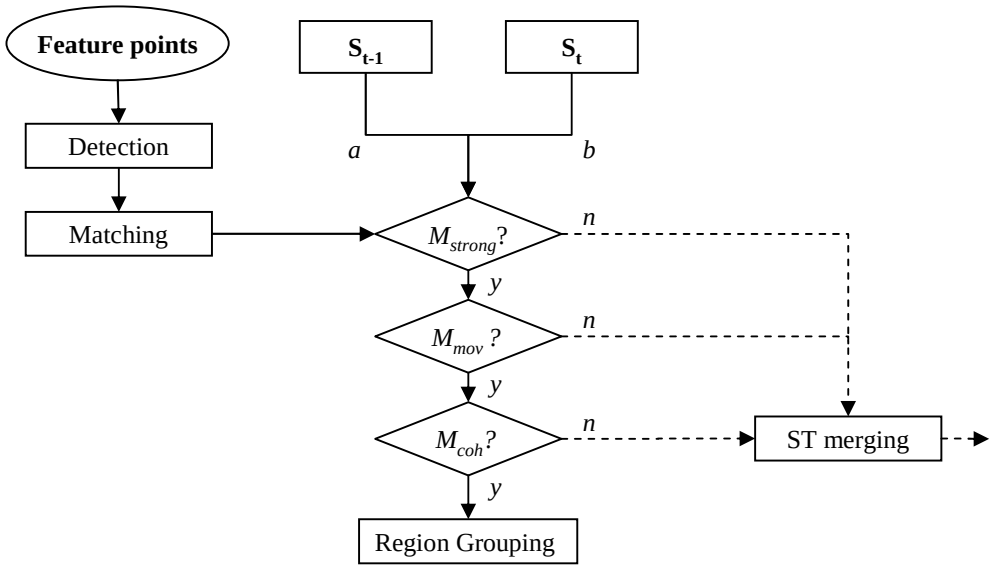


Figure 2.12: The temporal region grouping scheme. Feature points are first detected and matched. Then, the temporal merging of strongly connected regions is performed using a hierarchy of tests. Ungrouped regions are processed using the grid graph and the spatiotemporal merging rules (section 2.2.4).

motion, the displacements are compensated with the mean velocity of the complete set of points.

To describe a population P_a, the vertex attributes $\nu_P(a)$ obtained by analyzing statistical properties of the distribution:

- The population size $|P_a|$, which is also the degree of vertex v_a.
- The mean displacement of the population $d(a) = [\mathrm{v}_r(a), \mathrm{v}_\theta(a)]$.
- The covariance matrix of velocity points of a $\Sigma_\mathrm{v}(a)$, which indicates the coherence of the motion.

Temporal grouping of regions is considered as a bipartite matching problem between subsets V_t and V_{t-1}. Starting from the initial relationships established from the feature points, we keep only relevant connections between regions. The rules for grouping dynamic regions are explained in the following section.

Dynamic rules

The procedure for grouping regions temporally is illustrated fig.2.12. The graph G_T is constructed from the regions in S_{t-1} and S_t and the matched feature points in P. Let

$a \in V_{t-1}$ and $b \in V_t$ denote two candidate regions for grouping that are connected in G_T. The relationships between the two regions are analyzed by elaborating a sequence of tests. First, the graph is simplified and weak edges between a and b are pruned with the M_{strong} test. Second and third tests (M_{mov} and M_{coh}) verify if region couples undergo significant and similar motion. This helps to detect potential splitting regions. These tests operate as follows:

$\mathbf{M_{strong}}$: Two regions a and b are strongly connected if there is a significant proportion of points linking a to b. The test is accepted if:

$$\max\left(\frac{|P_{a,b}|}{|P_a|}, \frac{|P_{a,b}|}{|P_b|}\right) > \alpha \tag{2.12}$$

Formally, eq.2.12 is analogous to the cut ratio between a and b, considering edges with unit weight. In other words, if edges are given equal weights, the test is verified when a minimum proportion of edges at a or b connects the two regions. Once all regions have been tested, weak edges that do not satisfy the condition are pruned.

$\mathbf{M_{mov}}$: From the displacement of feature points, we deduce information on region motion. The test separates moving regions from static regions. The moving condition is given by:

$$\mathrm{v}_r(a) > T_{mov} \tag{2.13}$$

where T_{mov} is a minimum substantial displacement.

$\mathbf{M_{coh}}$: If a and b are moving regions, they must undergo coherent motion to be grouped. A simple measure is to compare the variance of the velocity distributions of a, b and $a \cup b$. The test $M_{coh}(a, b)$ is given by

$$tr(\nu_{a\cup b}) < \gamma(\nu(C_a) + \nu(C_b)) \tag{2.14}$$

where ν_a denotes the covariance matrix of the velocity points of a. The test encourages grouping of regions with similar motion and favors splitting of segmented regions when they are likely to have multiple motions. In this way, we handle apparition of new moving regions.

At the end of the process, temporal grouping has been performed reliably on homogeneous moving regions. To group more textured areas on the sequence, the population of seed points will be increased inside regions created in S_t, that have not been linked to S_{t-1}. In this way, the tracked points will focus progressively on the regions of interest.

New regions with no significant motion in S_t are grouped with the 2D+T merging criterion of eq.2.7 which checks if the regions are globally and locally coherent.

Subgraph matching

Among the regions in S_t, some may have received no correspondences. To valid the creation of new regions, we analyze the neighborhoods through spatial adjacency graphs $G_{t-1} = (V_{t-1}, E_{t-1})$ from S_{t-1} and $G_t = (V_t, E_t)$ from S_t. Graph matching is known to be difficult and computationally intensive. In our method, the matching task is simplified through the algorithms we detailed previously, and we can compare the RAGs of two consecutive frames

more easily. In our graph matching stage, we check whether new segmented regions in V_t can be grouped with those in V_{t-1}. The procedure is represented fig.2.13. For each node v in V_t, we define its neighborhood subgraph $N_t(v)$ as the smallest subgraph containing all its adjacent nodes $u \in V_t$. Search is done as follows. For one new region node v in S_{k+1}, take one of its neighbors $u \in N_t(v)$ that is connected to a node $u' \in V_{t-1}$. Candidate nodes l belong to the neighborhood of u in the previous frame, $N_{t-1}(u')$. The condition for grouping is that the distance between v and u' is minimal with respect to the neighborhood of the candidate:

$$d(u', v) < \min_{z \in N_{t-1}(u')} d(u', z) \tag{2.15}$$

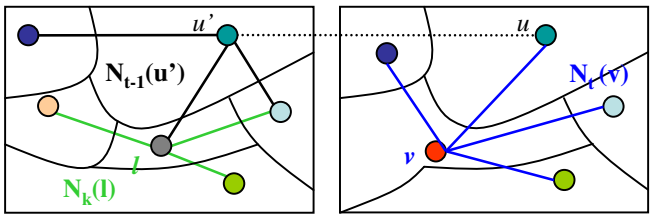


Figure 2.13: Neighborhood subgraphs for matching new nodes v_n. For each node $u \in N_t(v)$, the neighborhood of u in V_{t-1}, $N_{t-1}(u')$ is examined. Lost nodes l are then retrieved by comparing v to adjacent nodes of u' in $N_{t-1}(u')$.

Equation 2.15 checks if an untracked node in V_{t-1} can be matched with a new node in V_t in the proximate neighborhood. In this way, lost objects can be recovered in case of fast motion or homogeneity changes. For the distance measure, the node attributes represent dominant color (c) and size (s) of the regions. For two nodes u and v, the distance is given by

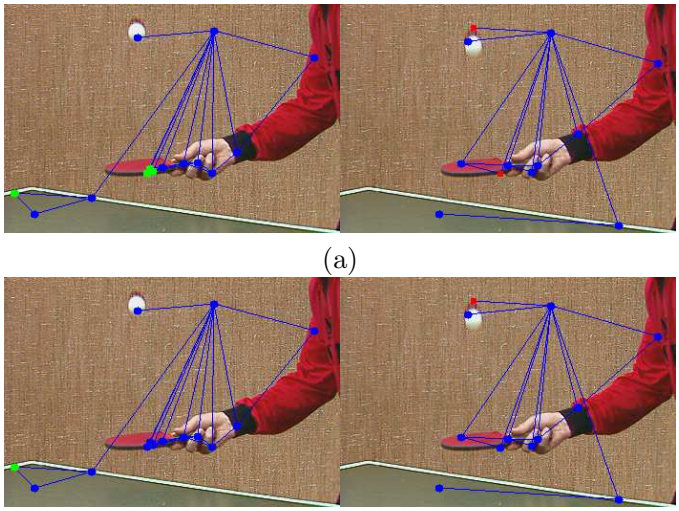
$$d(u, v) = |c_u - c_v|^2 \frac{s_u s_v}{s_u + s_v} \tag{2.16}$$

Thus, we favor the grouping of smaller regions with similar attributes.

An example of matching is shown fig.2.14 on the *tennis* sequence. Before matching (fig.2.14(a)), untracked regions are located in the racket and the table left corner. The new regions are located above the ball and inside the racket border. After matching (fig.2.14(b)), the nodes at the racket border have been grouped as they have close similarity, whereas the table left corner is not linked to any new node and thus cannot be reliably tracked.

2.2.6 Complexity study

The computational complexity of the method is relatively low compared to other spatiotemporal segmentation algorithms. As discussed above, the proposed algorithm consists of pixel and region-based operations. First, we consider the segmentation of the pixel grid, represented by a forest of n elements, with N elements per frame. The grid contains $m = O(n)$ edges. The complexity of the algorithm is dominated by the sorting of edges of the graph,



(a)

(b)

Figure 2.14: Subgraph matching. Untracked nodes are shown as green (clear) rounds, tracked nodes as dark (blue) rounds and new nodes as (red) squares. (a) RAGs before matching. (b) RAGs after matching.

with complexity in $m \log(m)$. In the growing stage, initializing the forest with the *make-set* operation is performed in $O(n)$. Checking the representative of one component has complexity of $\alpha(m)$ where $\alpha(m)$ is a very slowly growing function, and the merging of two components is done in constant time. Thus, the growing stage takes $O(n\alpha(n))$ time.

The complexity order is reduced for the 2D+T adaptation. Sorting is performed in $O(N \log(N))$ for each image and therefore in $O(n \log(N))$ for the whole sequence. Considering the frame size as fixed, the complexity becomes $O(n)$.

Region growing with the spatiotemporal rules keeps the same complexity of the original growing stage, as merging operation is still performed in constant time. Low-level image processing operations includes Canny edge detector and feature point detection. The optimal edge detector is approximated by computing the gradient of a smoothed image and identifying local maxima of the gradient magnitude as edges. The gradient image is also used for feature point detection, where a 5x5 window is used for the intensity change matrix. All these operations are linear in time in $O(N)$.

For operations on regions, let $|P|$ the number of feature points, $|V|$ the number of regions, and $|E|$ the number of spatiotemporal edges in the graph. The construction of the matching graph G_T from a forest requires $O(N \log(|V|))$ time for building subsets V_t. The construction of the list of feature points for all regions is negligible in $O(p)$ time. Temporal matching is then performed in $O(|E|)$ time. For the subgraph matching procedure, spatial adjacency graphs G_t and G_{t-1} are constructed in $O(N \log(|V|))$ time, and the complexity is higher in $O(|V||E|^2)$. This may seem high but only a few nodes are processed, and the out-edges for each region is usually far less than $|E|$.

All considered, computational complexity is driven by the diverse operations at the pixel level, whose complexity is in $n \log(N)$. This can be considered as low compared to the graph-cut methods [113] of order $O(n^2)$, and comparable mean-shift clustering, which can reach $O(n \log(n))$ using efficient range search algorithms [31].

2.2.7 Experimental results

In order to highlight the method properties, we have tested it for diverse video sequences. Segmentation is in general context difficult to evaluate, depending much upon final application. Objective criteria for evaluation of segmentation have been defined in [148], by defining a reference segmentation. Two criteria evaluate spatial accuracy of the results, whereas the third measures stability of the segmented form over time. However, such measures are intended for foreground/background segmentation, and do not reflect the segmentation of object parts. Direct observation of results, on various content still remain a relevant method to analyze the segmentation quality and deduce strengths and weaknesses of our method.

Segmentation of various sequences

We analyze the segmentation results on test sequences of the MPEG standard. These includes the *tennis*, *foreman*, *akiyo* and the *walking* sequence. In the *tennis* sequence, the camera performs a strong zooming out motion, and the player progressively appears in the sequence. In the *foreman* video, the camera motion is instable and the character object shows important deformations and uncovered area across time. The *akiyo* sequence also shows a character, where most deformations affect the anchorwoman's face. Finally, the *walking* sequence shows a walking man tracked by the camera in a corridor. The man limbs shows deformations, and the background moves fast due to the camera. The shots lasts a few seconds, about 100 frames each.

Properties of the initial algorithm (3D) are shown on the beginning of the *foreman* and *tennis* sequence (fig.2.15). Firstly, main areas that are similar spatially and temporally can be found in the same component, such as the foreman face, the background of the tennis sequence. Secondly, spatiotemporal propagation of the components enables to extract regions at a fine scale. This could be object parts, such as the pink collar (fig.2.15(a)) but also transitions in low contrast boundaries, for instance between the player arm and the background (fig.2.15(b)). These parts persist throughout the sequence. Thirdly, the segmentation is limited by connectivity of the pixel graph. Two regions without common overlap are not grouped. This can happen for small objects moving fast, such as the ball (fig.2.15(b)). Finally, two crossing objects can be grouped together if they show similar characteristics, such as the hand and the table tennis border (fig.2.15(b)).

Our proposed 2D+T method result is shown on a longer sequence fig.2.16. The figures show the final spatiotemporal segmentation, i.e. when all the frames have been processed. In contrast to the direct segmentation of the video stack, segmentation is initialized on the first frame, resulting in more proper region.

In the *akiyo* sequence (fig.2.16(a)), the video is composed of stationary background and slow head motion. We see that the main regions are the woman and the TV screens which

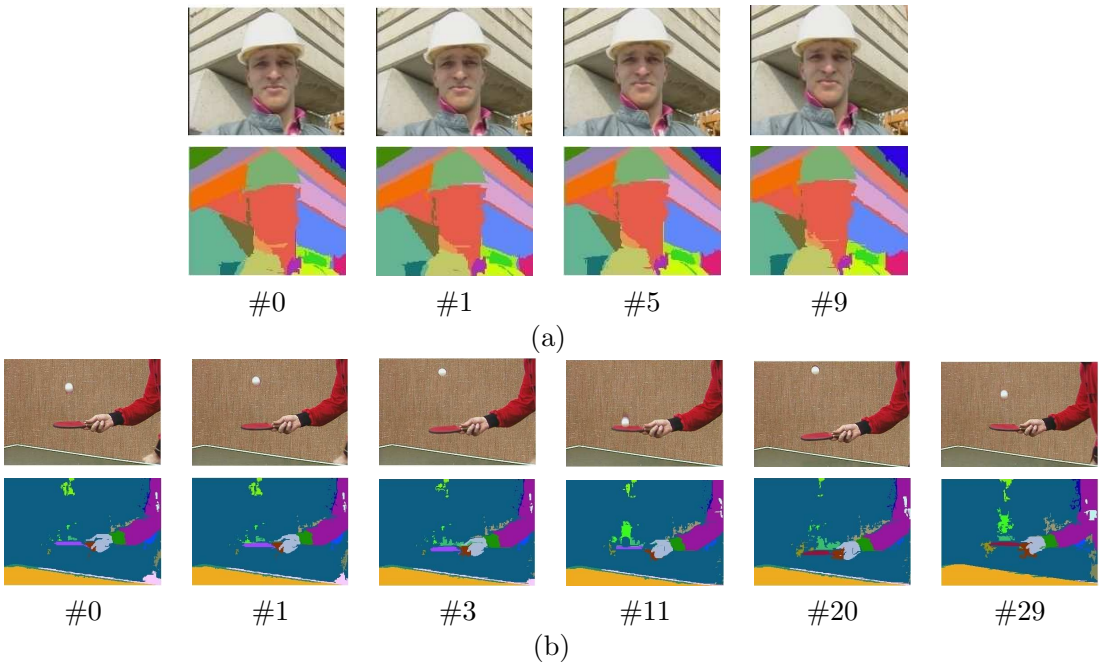


Figure 2.15: Segmentation results with the 3D algorithm. (a) *foreman* sequence. (b) *tennis* sequence.

have smooth spatial variations whereas tiny varying components such as face elements are not kept. In the next images, face moving elements are detected, but they are too tiny to be extracted from the sequence. In consequence, these elements are incorporated into the face.

The conclusions for the *foreman* sequence (fig.2.16(b)) are similar; the segmentation following the deformations of the character. However, the sequence shows less contrast on some object parts and the background. As a consequence, the helmet is attached to the background when initializing the sequence.

For the *tennis* sequence (fig.2.16(c)), the video is composed of several motions. The ball and the racket undergo rigid motion whereas the player undergoes non rigid-motion. Besides theses motions, the camera is zooming out during the entire sequence. Large and stable regions, such as the table, the racket and the hand are correctly segmented during the whole sequence. Finally, we can see that a strong scale change happens gradually between frame #31 and frame #60. While the player is appearing progressively at the left of the image, the corresponding regions are split until fitting the body of the player. In this way, the segmentation follows the temporal changes of the video. The ball region is more difficult to tackle, with high speed motion and important shadow. It remains until it hits the racket in frame #26 where a new region is created for the ball. The ball is tracked successfully until frame #31. From this moment on, the camera quickly zooms out and the ball becomes

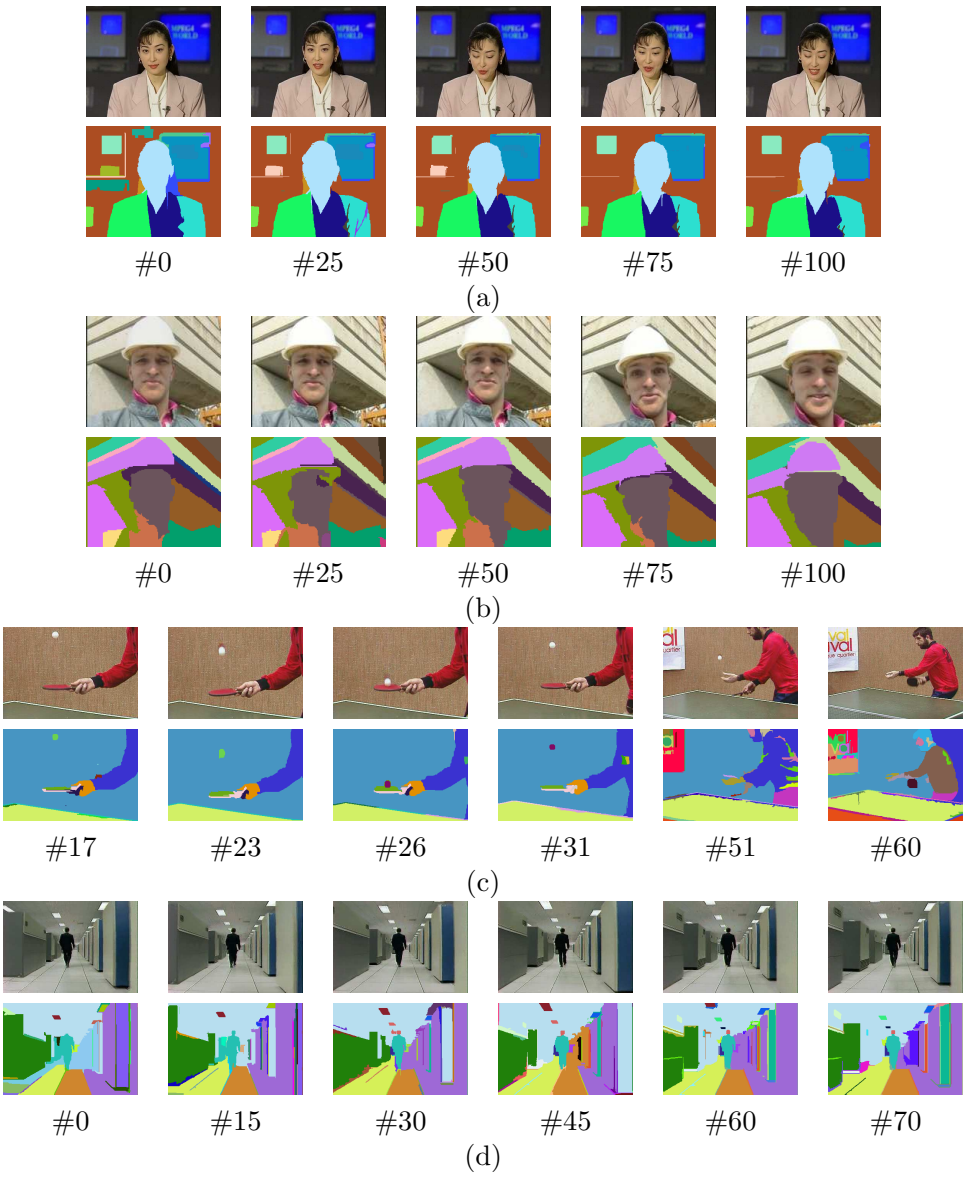


Figure 2.16: Segmentation results with the 2D+T algorithm. a) *akiyo* sequence. b) *foreman* sequence. c) *tennis* sequence. d) *walking* sequence.

smaller and less homogeneous. As a result, the ball sometimes does not appear after the spatial merging stage.

In the *walking* sequence (fig.2.16(d)), the camera is tracking the walking man so that the walls surrounding him are moving toward the foreground and exiting the frame progressively. In the first frame #0, the regions are composed of the man, the tiled floor, the walls, the

ceiling and the lamps. The man region remains consistent along the sequence, just as the different parts of the walls and the lights until they exit the frame. We can further notice that apparent static regions such as the floor and the ceiling are coherent in the entire sequence.

These results illustrate the potential of the proposed method to extract coherent volumes from video shots. Given a level of details, both moving and static elements can be tracked thanks to our hierarchical matching stage. Besides, we handle dynamic temporal changes by favoring the creation of new regions when some regions cannot be reliably matched between frame pairs. In this way, we achieve good compromise between the span and the consistency of the regions. Therefore, the method can help higher level grouping tasks considerably.

Computational ressources

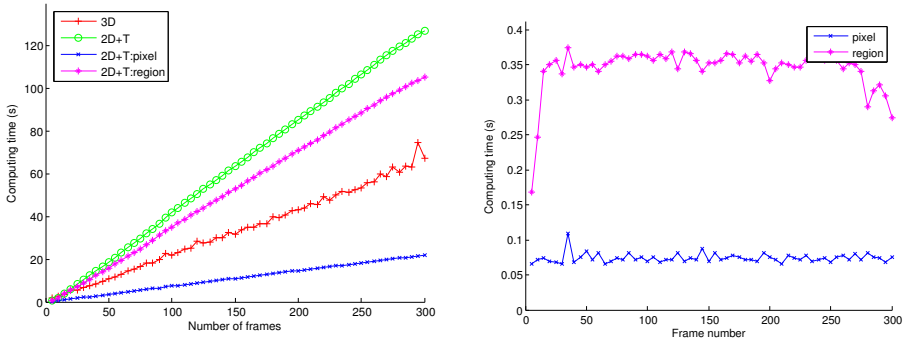
Computing time

To show the low computational complexity of our method, we examined the computing time required to segment the video stack. Computing times give an order of magnitude, as no optimization were performed on the algorithms; the measures were achieved in a 2.8 GHz Pentium.

Figures 2.18 and 2.17 show computing time analysis for different video sequences. We compare the segmentation time with respect to the duration of the sequence fig.(2.17-2.18)(a). The 3D method refers to the segmentation of the whole stack at once, and the 2D+T method implements the scheme depicted in fig.2.3. As they show different properties, the computing time is further decomposed in pixel (2D+T:pixel) and region-based (2D+T:region) operations. Pixel-based operations include spatial segmentation and spatiotemporal merging on the grid while region-based operations includes feature-point tracking and subgraph matching. Their respective times for processing frames are indicated in fig.(2.17-2.18)(b).

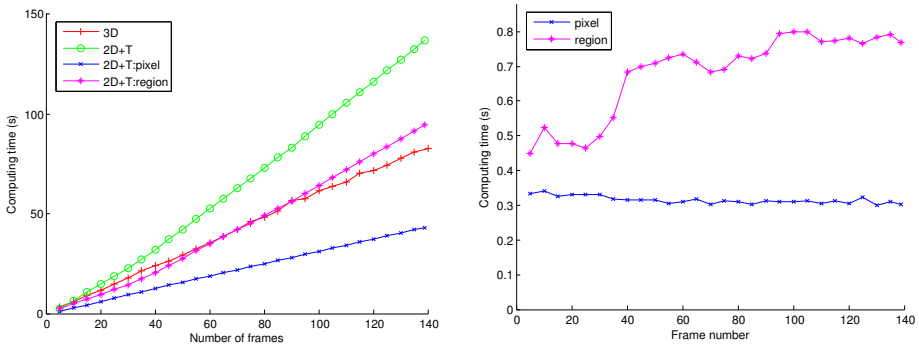
Let observe the results for the $foreman$ sequence (fig.2.17). The sequence is in the QCIF format (176x144). The computing time for the 3D method is quasi-linear; it takes about $0.22s$ per frame, which is twice as fast as the 2D+T method (fig.2.17(a)). Variations of computing time (fig.2.17(b)) depend mostly of the region grouping stage. Factors intervening are the population of feature points and the number of regions. In the beginning of the sequence, population grows as new feature points are detected in the frames. At the end of the sequence ($t > 250$) the character comes out the image at the left so the accompanying points do not survive and computation time drops a little. In comparison the pixel-based operations are done in constant time because of the fixed grid structure and the cost of forest structure operations. These are also performed efficiently (less than 0.1s) which is at least twice better than for the 3D method.

In the *tennis* sequence (fig.2.18), the image is in CIF format, so 4 times larger. As a consequence, the computation times are higher: about 3 times for the 3D method (0.6s per frame) and from 2 to 3 times for the 2D+T method (from 0.7 to 1s). We can notice fig.2.18(b) that the computation time is more balanced between pixel and region-based operations; pixel operations are performed in 0.3 s and region grouping takes 0.6s in the



(a) (b)

Figure 2.17: Computing time for the foreman sequence. a) Comparison of the 3D and 2D+T method. b) Comparison of pixel and region based operations in the 2D+T segmentation.



(a) (b)

Figure 2.18: Computing time for the tennis sequence. a) Comparison of the 3D and 2D+T method. b) Comparison of pixel and region based operations in the 2D+T segmentation.

average, which is comparable to the 3D-method. Region computation follows the variations of the content. Computation time is low at the beginning of the sequence, as there is only a few regions (arm, racket, ball, table). While zooming out more regions are found. In consequence, new points are also detected and more time is required for temporal grouping of regions. Putting these observations aside with the theoretical complexity studied in section 2.2.6, we deduce the following properties:

- The complexity of 2D+T method is comparable to the one of the 3D method.
- Grouping on the grid is significantly more efficient in the 2D+T scheme than on the full 3D grid.
- Complexity of region grouping adapts to the granularity of the content (number of regions and population of feature points).

Computational requirements

Another advantage of the 2D+T framework is to consume far less memory than 3D methods. Memory load curves are shown in figure 2.19. Both methods keeps a full forest for the whole sequence, so that memory increases with the number of frames.

For the segmentation of the whole grid (3D), memory load is much more important since grid edges are loaded from the full sequence, and there are four edges for one node. Memory management also plays a role in the loss of performance of this method for long sequences as the volume stack increases.

In the 2D+T method the memory slowly increases with the number of frames. Forest allocation is the most important part of the memory load as we do not need the whole edge grid set anymore. Furthermore region graph structures are defined for frame pairs and do not vary much. As a consequence, for the full tennis sequence, twice less memory is consumed for the 2D+T case than for the 3D case. For short term sequences, the memory load of 2D+T and 3D approaches are nearly the same.

If the memory is required to be kept constant or severely limited, one efficient solution is to allocate a forest corresponding for a Block of Frames (BOF) which number of frames is fixed. Once these frames are processed, store the segmentation and make list of volume representatives. Then, the last processed frame can be considered as the first frame for the new BOF. Once the second BOF is processed, find the representative that corresponds to the previous volumes and merge the previous and new BOF volume representative list. This corresponds to achieve path compression within each BOF only and keep the merging tree for the whole sequence. Once the full sequence is processed, the actual segmentation can be obtained through the store pixel labels and the representative lists. However, these limits have appeared unnecessary for the sequences we experimented with.

2.3 Objects of interest

Semantic objects are usually composed of several volumes. Spatiotemporal segmentation results in volumes that can be grouped into complex objects. However, the problem is different to traditional object detection and recognition as we do not focus on particular object categories. In this context, motion remains the most appropriate common property to objects. In this section we examine how we can group volumes into complex objects from their motions. We split the problem in two steps. We first detect moving volumes and associate these volumes together to propagate object. Secondly we try to propagate detected objects through the sequence using a matching procedure taking into account visual appearance and motion of the object's volumes.

2.3.1 Joint use of spatiotemporal volumes and Motion information

Object graph

In our framework, a video shot is represented at highest level with an ARG $G = (V, E, \nu, \xi)$ depicting relationships between spatiotemporal volumes (section 1.2.5). An object subgraph

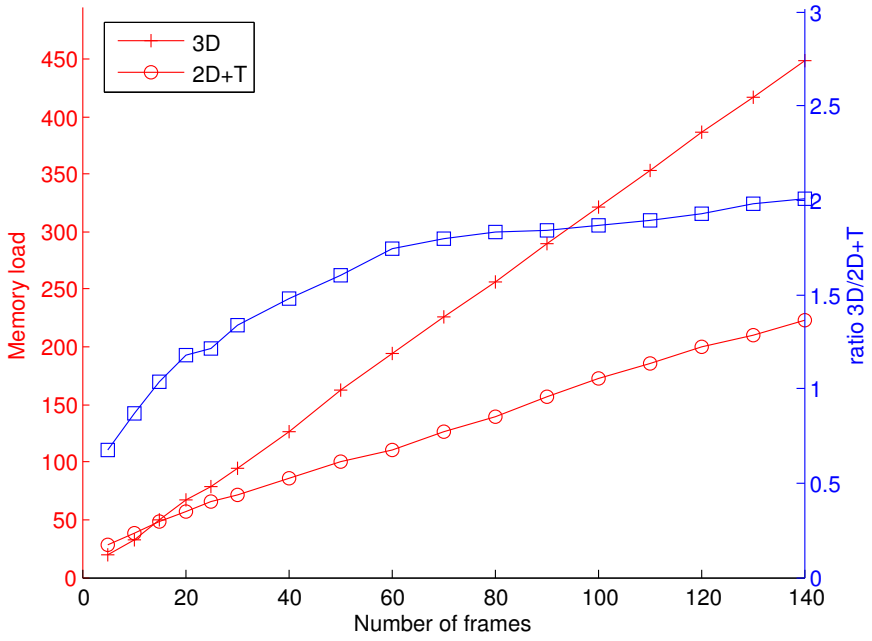


Figure 2.19: Memory load for the *tennis* sequence.

$OG = (V', E')$ is defined as an induced subgraph of G: $OG = G[V']$, where V' is the set of volumes composing the object. To build object subgraph, we consider two types of useful volume attributes:

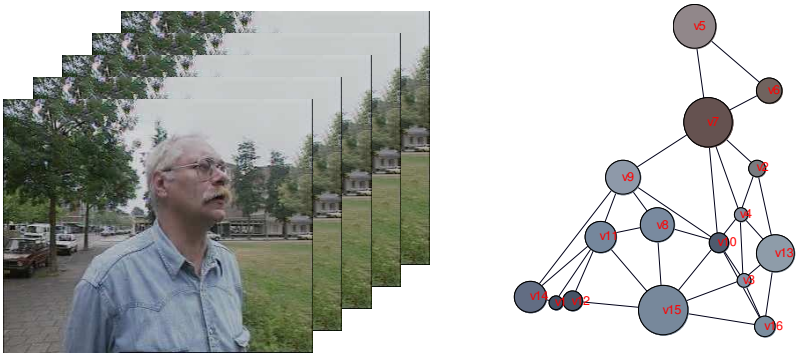
- Visual region features (ν_S).
- Motion features (ν_T).

Edge attributes $\xi = \{\xi_S, \xi_T\}$ are built from associated distance measures between two spatiotemporal volumes. Visual region features incorporate MPEG-7 region descriptors (color, texture, location) described in section 1.1.2 and the motion features used will be detailed in section 2.3.2.

The structure of the object can be complex, involving several spatiotemporal relationships between volumes. This turns out to be particularly important in close-up shots involving large and complex objects. Fig.2.20 illustrates the problem of finding object subgraph in such a scene. The different parts are composed of major parts (head, hair, chest, arm) and segmentation details (small nodes).

Estimating Motion Models on volumes

Motion information can be extracted and represented in many different ways (section 2.1.2). Motion of object parts can be complex, and the volume support can undergo deformations. Rather than estimating a complex unique model for the volume in the sequence, we prefer



(a) (b)

Figure 2.20: Video shot and object representation. (a) Example of close-up sequence. (b) Corresponding object graph.

decomposing simple models on a short-time period. Thus, we decide to use flexible representation for motion, being adaptive to temporal changes. A volume a lying in the temporal interval T_a is described with a series of parametric models $\theta_a(t), t \in T_a$, which support is defined on the intersected frame regions $R_a(t)$. As shown in the illustration fig.2.21, volume motion models are available for each region. It can be identical to the one of the previous frame or different if changes are important enough. Practically, the set of models can be embedded within the MPEG-7 Parameter trajectory descriptor. Simple affine model is used for simplicity and computational efficiency.

To estimate parametric motion models, we follow the technique proposed in the early MPEG-7 XM [120]. The estimation is performed in two steps. A block matching technique enables to estimate the rough displacement of the support. The model is refined afterwards by performing local optimization of the rotation parameters of the model.

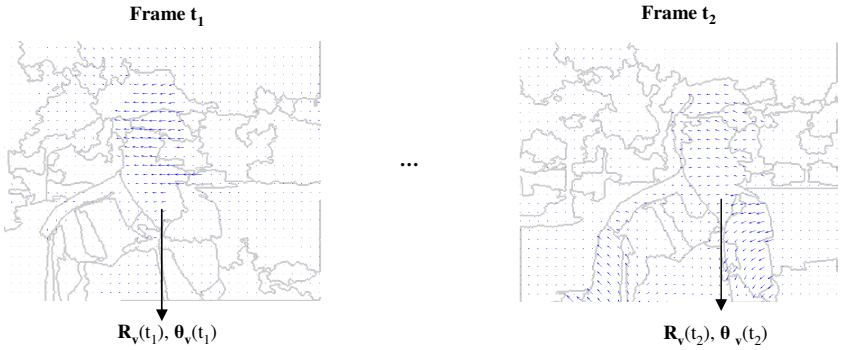


Figure 2.21: Estimation of motion models. Each model is built based on a region support. The resulting motion field for the image is superposed to the segmentation map.

2.3.2 Motion-based grouping

We have already mentioned the difficulties linked to motion in section 2.1.2. Motion is usually hampered by aperture problems for low textured regions, small size of the support and deformations. Fig.2.22 illustrates the problem. In frame #13 the object is nearly static and only some parts are detected as moving whereas in frame #82 the whole body is turning and shows reliable motion. Thus sophisticated methods are needed in general. We benefit from the spatiotemporal volumes for good use of motion information. Once motion estimation performed, it is possible to form primary objects from grouping of spatiotemporal volumes. For this purpose, we consider grouping on volumes whose motion remains significant and stable along time. To compare the motions of regions or bigger object several approaches are possible: trajectories, statistical distributions, and parametric motion models [147]. We select the last approach as it owns these suitable properties:

- Accuracy of the comparison of motion fields.
- Independence with respect to the type of model used.

However, comparing motion fields has an important computation cost as the operation is performed on region masks.

Denote by $\mathrm{v}(p;\theta)$ the velocity for the pixel p using a motion model θ. Zaharia et al. [147] expresses the distance between models θ_1 and θ_2 with respective support R_1 and R_2 as:

$$d_{\mathrm{v}}(\theta_1,\theta_2) = \sum_{p\in R_1\cup R_2'} d(\mathrm{v}(p;\theta_1),\mathrm{v}(p;\theta_2)) \tag{2.17}$$

where R_2' is the alignment of support R_2 to R_1. For simplicity R_2' can be obtained by translating the center c_2 of R_2 to the center c_1 of R_1. Several distances d have been proposed; we select the L_1 distance for accurate comparison of the velocity vectors. The distance can be easily extended to the spatiotemporal domain. For two volumes a, b the distance is computed over their common temporal support:

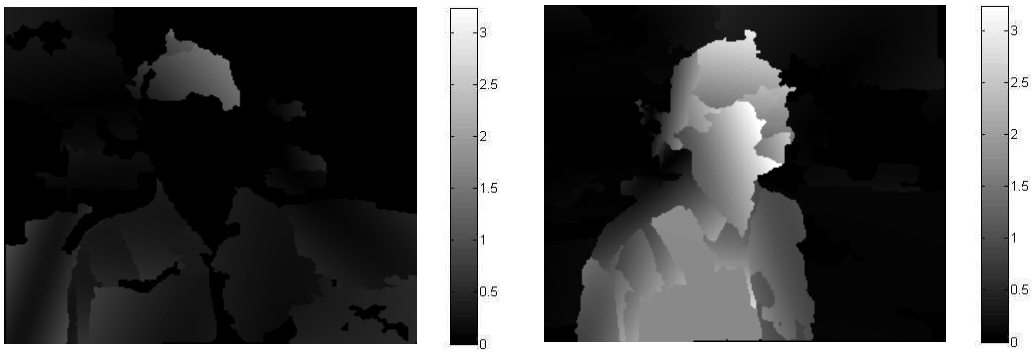
$$\xi_T(\theta_a,\theta_b) = \frac{1}{T_a\cap T_b}\sum_{t\in T_a\cap T_b} D(M_a(t),M_b(t)) \tag{2.18}$$

From the motion models and the defined velocity distance we derive measures to depict volume motion features. The first measure is the motion activity over the volume, which indicates strength of motion. The activity of a volume a is given by:

$$Act(a) = \frac{1}{|a|}\sum_{t\in T_a}\sum_{p\in R_a(t)} \mathrm{v}(p,\theta_a(t)) \tag{2.19}$$

A stability measure is based on the evolution of the motion model across time. Estimation is assumed to be correct if the stability of the motion models is important. The stability of volume a is defined as:

$$Stab(a) = \frac{1}{T_a}\sum_{\substack{t\in T_a\\ t-1\in T_a}} exp\left(-\frac{D(\theta_a(t),\theta_a(t-1))}{\sigma}\right) \tag{2.20}$$



(a) (b)

Figure 2.22: Motion activity maps. (a) Frame #13. (b) Frame #82.

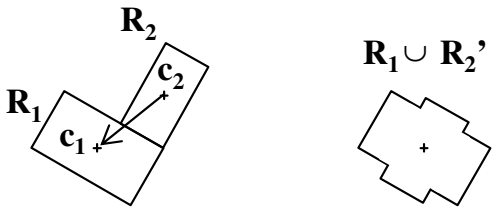


Figure 2.23: Velocity-based similarity is obtained by applying motion models on R_1 and R_2 in the compensated union of the two regions. Model origin of R_2 and its support are translated by $c_1 - c_2$.

Thus, the measure expresses the confidence on the correctness of the motion estimation for the full volume. To group moving volumes into OGs with the proposed measures, we suggest a bottom-up procedure. As far as the number of objects is unknown and the number of graph nodes is not high (typically a hundred of nodes), global graph partitioning techniques in the similarity domain are not appropriate. On the contrary, a merging procedure can cope with an arbitrary number of objects.

Grouping is then performed on volumes constructing the spanning tree of the objects, i.e. finding the minimum path linking all object volumes. The technique slightly differs from the one in section 2.2.3 as we use activity and stability as strong cues to initialize and guide the grouping. We first detect the background volumes along with those that are static or where motion is uncorrectly estimated. A volume a is considered as background if $Stab(a) < S_{min}$ or $Act(a) < A_{min}$. We denote BG the graph of background volumes.

Important objects are then partially localized looking at zones of high motion stability. Stability is generally also better in region with large support [88]. In this sense, we assume that the object undergoes coherent motion during a sufficient period, which is reasonable for most sequences. In the beginning these volumes form markers for the OGs. Hopefully, each object has received a marker (or cluster). We denote k the number of markers. Volumes

are linked to markers using a merging procedure. We consider node attributes $\nu_T(a) = \{Stab(a), Act(a)\}$ and edge attributes $\xi_T(a,b)$. Typical merging on the tree requires to set a maximum edge distance. It is rather difficult to set such threshold automatically, and the tolerance in adding edges to existing components depends on the object itself.

To estimate statistics of the different object parts corresponding to the OGs, we first perform a simple k-clustering technique of the graph G/BG from the markers. In the merging process, edges that bridge two OGs are not connected.

We then assume that each object part has smooth variations on the edge weight and attribute distribution. For each cluster we detect breaks within the stability, activity and edge weights distribution. OG vertices on both sides of each break are split, forming new OGs that are homogeneous. In the second step of the procedure, merging is performed on adjacent groups with close statistics. OG_1 and OG_2 are merged if:

$$\min_{\substack{a\in OG_1\\ b\in OG_2}} \xi_T(a,b) \quad < \quad \min\left(\max_{e_1\in E[OG_1]} \xi_T(e_1)+\tau(OG_1), \max_{e_2\in E[OG_2]} \xi_T(e_2)+\tau(OG_2)\right) \tag{2.21}$$

τ is a function that penalizes grouping of an OG in function of its volume attributes. More tolerance is accepted on the velocity distances as τ increases. In this grouping criterion, zones with important activity and good stability have to be privileged. Indeed, a fast moving object is prone to have more variations between each volume, especially if its motion is non-rigid. Stability indicates the accuracy of the motion model. In consequence the estimated velocity distance between models ξ_T is more certain in zones of high stability. For these zones the tolerance for a component can be solely based on its activity. For components with lower stability the tolerance is reduced to take into account the possible errors in the estimation of ξ_T. Following these considerations we define the function τ as:

$$\tau(OG) = \begin{cases} d_{min} + \frac{d_{max}-d_{min}}{A_{max}-A_{min}} Stab(OG)|Act(OG)-A_{min}| & \text{if} \quad A_{min} \le A(OG) \le A_{max} \\ d_{max} & \text{else} \end{cases} \tag{2.22}$$

d_{min} is the minimum tolerance for every region and d_{max} is the maximum tolerance allowed, reached for $Act(OG) = A_{max}$. The criterion is used to group only adjacent objects which motion are strongly related, until all connections have been tested.

After the grouping stage, unmerged volumes either correspond to single objects where motion estimation was not successful, object detail, or cluttered zones (mixed object and background). We will see on the next section how to attach these volumes to the main OGs.

2.3.3 Object detection and matching

Once groups have been obtained, we search for salient objects through the sequence. Good candidates are groups that shows strong connectivity and stable motion. For measuring saliency of OGs, we choose a simple measure that finds the largest OG with important activity and good stability:

$$Sal_{OG} = \sum_{a\in OG} |a| Stab(a) Act(a) \tag{2.23}$$

Detected objects can be then extended in the whole sequence by by establishing correspondences with other OGs in the graph. Typically, visual appearance of objects changes slightly during a sequence. As mentioned in the previous section, and illustrated in fig.2.22, the whole object may be either moving coherently or some object parts remain nearly static relatively to the other moving parts. Matching between OGs takes under consideration visual region-based properties. Thus we use visual features as the edge attributes ξ_S between volumes of two OGs.

We consider the graph G_1 and G_2 that corresponds to the volumes in two different temporal interval T_1 and T_2. Let $P \in G_1$ the graph of a detected object, $OG_2^i \in G_2$ denote an object graph in G_2 and BG_2^i one of the remaining background nodes. We aim to find the corresponding object for P in G_2, Q, by assembling the object graphs OG_2^i and background volumes BG_2^i.

To reduce the complexity of the matching, the procedure is decomposed into two steps. Firstly, we search for OGs in G_2 that reliably match part of the detected object graph P. Secondly the matched object Q is progressively completed with the background nodes BG_2^i by comparing the structure of the two OGs.

Let examine the matching between OG volumes, which is illustrated in fig.2.24(a). Given a set $A \in G_1$, we note as M_A the subgraph of G_2 that is matched by A; and $M_A[B]$ the restriction of M_A to a set of volumes $B \in G_2$. An object OG_2^i in G_2 is matched to P if a majority of its volumes are matched to P. To establish a match from $a \in P$ to G_2 we proceed as follows. We find the volume b in G_2, which distance with a, $\xi_S(a, b)$ is minimal. If b is also the closest volume to a and the distance does not exceed a predefined threshold, the volumes are matched, i.e. $b = M_a$.

In the example of fig.2.24(a), only OG_2^2 and OG_2^3 are matched to P. Remaining OGs in P are not matched as no matches have been established.

After this first stage, some nodes in G_1 are not matched, either due to low similarity with nodes in G_2 or ambiguities when the similarities with several nodes is comparable. For the latter case, we rely on the neighborhood context to propose new matches. The idea is that the strength of volume matches between two subgraphs reflects the similarity of the object structures. Consider an unmatched volume $a \in P$ and its neighborhood N_a. We search volume $b \in G_2$, with neighborhood N_b that maximizes:

$$mc(a, b) = |M_{N_a}[N_b]| \tag{2.24}$$

which is the count of matches from N_a to N_b.

The completion of Q is illustrated fig.2.24(b-c-d). In fig.2.24(b) $mc(a, b) = 3$ and b is added to M_P. In fig.2.24(b) there are two possible configurations of value $mc(a, b) = 2$ for matching the remaining node in P: one from the original matches (fig.2.24(c)), the other with the new match (fig.2.24(d)).

Such example shows that several candidates may own same count values: the proportion of matches will be low at the beginning of the procedure if the object undergoes important changes from T_1 to T_2. Thus we analyze the neighborhoods in detail. Edit distance have been proposed for comparing two graph structures, expressing the cost of transforming object nodes to another. Definition and computation is in general tricky; the cost for

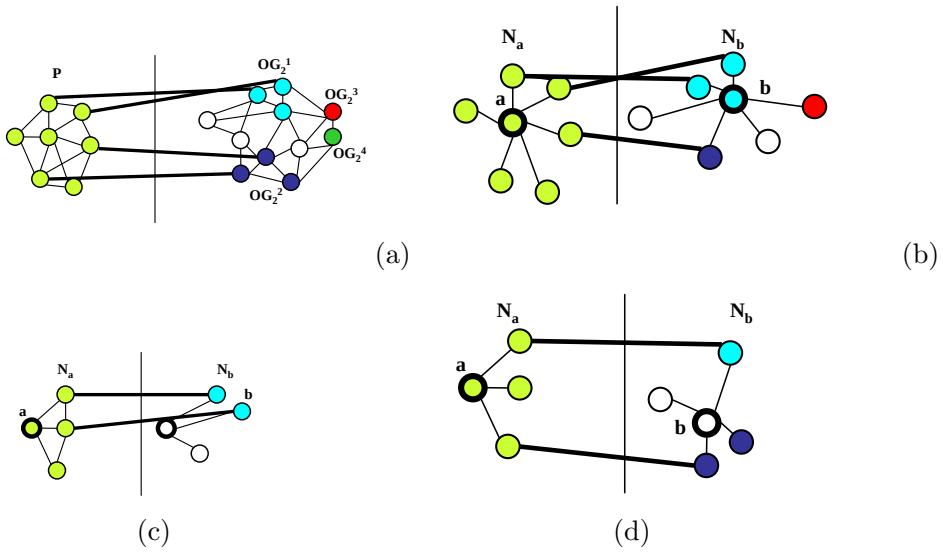


Figure 2.24: Object propagation. Matchings are represented as bold lines, and the volumes in the match under test as bold circles. (a) Construction of volume matches between OGs.(b) (c) (d) Recursive completion of OG using the neighborhood context.

adding, deleting or transforming a node depends of the targeted application [21]. The guidance of the count measure proposed above and the reduction of the graph to their immediate neighborhoods greatly simplify the problem and the search for optimal solution in state space. Consider a and b as roots for N_a and N_b. We adapt the edit distance proposed by Shasha et al. [111] between unordered trees:

$$ED(N_a, N_b) = \min \begin{cases} \min_{u \in N_a, v \in N_b} ED\,(N_a/u, N_b/v) & + & \xi_S(u,v) \\ \min_{u \in N_a} ED\,(N_a/u, N_b) & + & \gamma(u,\phi) \\ \min_{v \in N_b} ED\,(N_a, N_b/v) & + & \gamma(\phi,v) \end{cases} \tag{2.25}$$

where $\gamma(u,\phi)$ and $\gamma(\phi,v)$ expresses the cost to delete and insertion of node u and v. Deletion of $u \in N_a$ occurs when a volume in N_a has no match in N_b, we consider that u is in fact grouped with another node in its neighborhood:

$$\gamma(u,\phi) = min_{v \in N_u}(\xi_S(u,v)) \tag{2.26}$$

Insertion is defined identically.

Eq.2.25 is defined recursively and the problem of finding the best edit sequence is known to be NP hard. Dynamic algorithms such as $A*$ have been proposed [96], but it is more computationally efficient to constrain the problem. As best matches between A and B have been already found, we only compute the edit distance for unmatched graphs $N_a^U = N_a/M_a[N_b]$ and $N_b^U = N_b/M_b[N_a]$. As the degrees of these graphs is low and the tree has only one level, ED is equivalent to a string edit distance if an ordering is fixed for these two graphs. The complexity of such operation is reduced to the product of the string sizes.

Multiple orderings for the neighborhood with the smallest degree are considered for a robust solution.

Adjacency graphs highlight the presence of relationships between parts but do not qualify the relative deformation of the object structure. We limit search of volumes in G_2, by computation the location distances (center of mass) between object volumes P and Q, which gives information of the displacement of the object parts between the temporal intervals T_1 and T_2. Proposed matches for which location distance does not fit their neighboring counterparts are pruned. Alternatively, this could be further enhanced considering semantic temporal spatiotemporal relationships (*above*, *before*,...) to depict the inherent structure of the object.

We complete the OGs P and Q using the count of matches mc and edit distance ED as follows. For each unmatched node $a \in P$, we find among the possible matches the ones which maximizes mc. Largest volumes are prone to have important match count values as they are adjacent to many other volumes. We then select the match that minimizes the edit distance. These two steps enable to find the matches efficiently, favoring similar neighborhoods with a few unmatched nodes. Both structural similarity (here the degree of a and b) and the overall visual similarity between N_a and N_b are considered.

The match from P to G_2 with the best match count and edit distance is added to M_P and M_Q. Match count and edit distances are reevaluated in the neighborhood and a new match is proposed. The procedure ends when no new matches are found, i.e. when all nodes in P are matched or the visual distance for a match is below a threshold.

In comparison, finding directly the match with the minimum edit distance between their neighborhoods is not appropriate for two reasons. First, it will favor matching from smallest to biggest nodes of small nodes with low degrees, since only a few edit operations has to be done. Secondly, it will demand an important and unnecessary computational cost.

2.3.4 Workflow for long-term sequences

We studied the different mechanisms for grouping and matching objects through a sequence. For long sequences, we organize these tools with an efficient scheme. The method workflow is shown fig.2.25. First, we cut the sequence into short temporal interval or block of frames (BOF) where the object motion does not vary much temporally. A spatiotemporal segmentation is computed and the volume ARGs are obtained from each BOF (section 2.3.1). Motion models are then estimated for each volume, and the motion attributes of the ARG are computed for the model. Grouping can then be performed within areas with significant and stable motion. Once grouping is performed for the whole sequence, we detect most important objects in the BOFs (section 2.3.2). The detected objects serve as object graph models and are propagated by the object matching procedure (section 2.3.3) in the surrounding BOFs.

2.3.5 Examples

We carried out experiments for real video sequences using the proposed framework. Most salient objects are detected and the object graphs are propagated through the sequence.

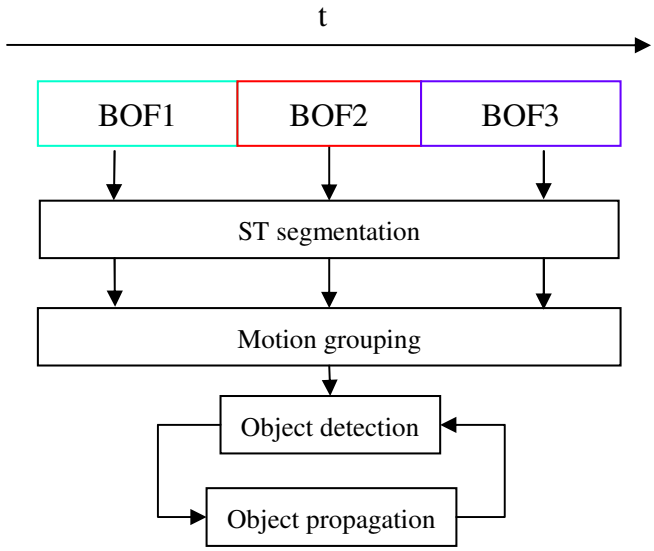


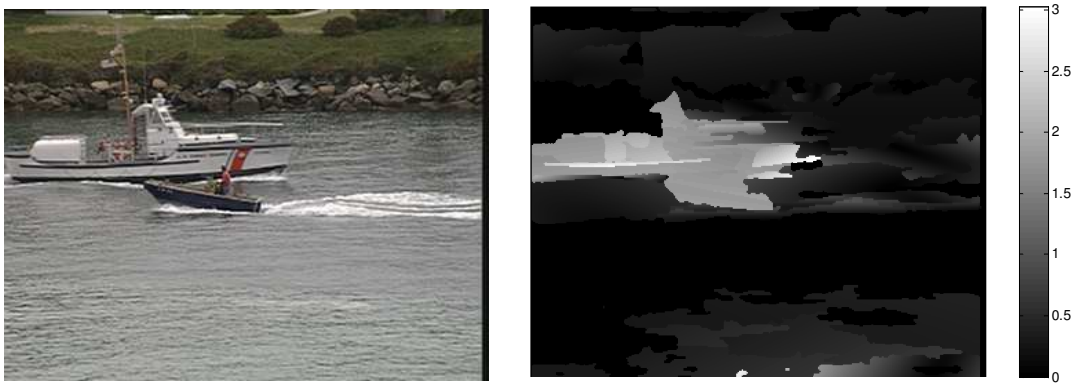
Figure 2.25: Moving object detection and propagation for long-term sequences.

The first example (fig.2.27) is the *coastguard* sequence from the MPEG standard, which contains two moving rigid objects. Each BOF has a duration of 50 frames. A small boat (OG_1) crosses and sails in front of the speedboat (OG_2) in BOF 1 and BOF 2. From BOF 2, the camera tilts up then approximately track the second object moving to the right.

The detection of objects is facilitated by the important rigid motion (fig.2.26). Water areas can be also detected as moving because of the waves and of the slipstream left behind the boat, but the motion in these areas is less important and stable so they could be filtered out. OG_1 is composed of two volumes: the boat itself and the driver (BOF 1-2). The object slightly differs from BOF 1 to BOF 2 as the segment of sea confined between the two boats (BOF 2-1 and BOF 2-2) has been attached to the OG. The detected parts of the second object (OG_2) include homogeneous areas within the boat profile (notably in the body) and the rod on the bridge, which is segmented as a sharp horizontal tip from BOF 2-1 to BOF 3-1. As the bottom body of the boat is often visually close to the sea parts and the contrast between these two areas is low; some parts of the boat are not detected as they were attached to the sea regions in the segmentation stage.

The second example shows a close-up shot where a man is speaking in front of a camera fig.2.28. The BOF length is shorter (10 frames), as there are fast variations in camera and object motion. The camera is tilting quite unstably all along the sequence, being probably hand-held. In consequence it cannot be compensated entirely. In addition the man area is subjected to important deformation. His head is turning fast from left to right (BOF 1 to 6), The body following partially the rotation of the head with a short delay.

In the end of the sequence (BOF 9 to 11), his arm abruptly moves up to point towards something. In this context, object is detected in BOF 6 while all the parts revolve coherently



(a) (b)

Figure 2.26: Motion detection for the *Coastguard* sequence. a) Input image #50. b) Corresponding motion activity map.

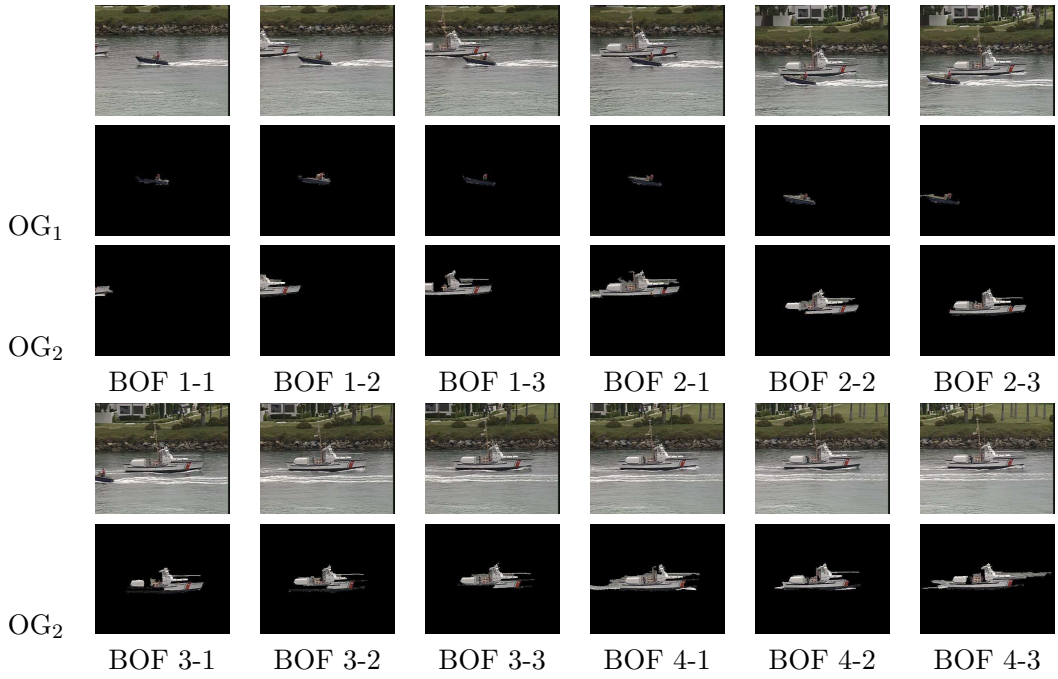


Figure 2.27: Moving object grouping results for the coastguard sequence.

to the right. Both head and body part of the man are propagated before that point using the visual matching. Missing details of the body correspond to background areas in the spatiotemporal graph of which motion is not significant. The visual similarity for these

Figure 2.28: Moving object grouping results for the close-up sequence.

unmatched volumes was not important enough to one of the object volumes and a better match was likely to be found in the neighborhood of these areas. For the remaining sequence after the detected object, most areas are detected as the visual similarity between object parts is sufficiently high. In BOF 10 the appearing arm is correctly attached to the object even if the corresponding volume includes some background area. However, in BOF 11 this is not the case anymore as the motion is really different from the rest of the body and no corresponding volumes are found in the model. In the previous BOF the arm and the right part were part of the same volume. A potential solution for this problem is to propagate the object temporally from one BOF to another, as BOF 11 is visually closer to BOF 10 than BOF 6. The drawback of such approach is that the matched object can progressively drift from the actual object. Finally, the last BOF is correctly propagated, being close to the original model.

These results illustrate that motion and visual information can be used for grouping of spatiotemporal volumes in different contexts. An advantage of the spatiotemporal approach is to propagate objects even if all parts cannot be retrieved at some points of the sequence, alleviating the occlusion problems of causal methods which only consider the immediate temporal neighborhood which is restricted to a pair of frames in general. A possible improvement for the presented approach would be to use both the detected and propagated object graphs in the matching of a new BOF to handle hard deformations of the object parts that are not part of the object originally detected.

2.4 Conclusions

In this chapter, we have introduced a novel spatiotemporal approach for the segmentation of the video stack, based on an adjacency graph representation. Direct segmentation of large pixel graphs becomes inappropriate and generates important computational cost when the size of the video increases. Using a restricted connectivity on a grid does not enable to handle correctly the temporal evolution of the regions. To tackle this issue, we analyze spatiotemporal relationships at different levels and limit the complexity with a 2D+T approach. Pixel-based graph enables for the construction of low-level stable volumes and ARGs enforce consistency of dynamic regions. Moreover, we show that the 2D+T approach keeps good performance in the both pixel and region-based representations. The proposed spatiotemporal representation based on ARGs is shown valuable for the detection and propagation of objects of interest. We develop a grouping procedure that considers motion of volumes as the essential attribute for the formation of coherent objects. To handle the lack of consistent motion information that occurs in real sequences, we further propose a scheme that first detect and then propagate significant moving objects using graph matching with visual attributes. First results show that this framework is promising for making object-based description of visual shots even if progress can still be achieved in the segmentation and object matching stage.

Chapter 3

Semantic interpretation of video shots

Compared to image analysis, adding temporal dimension allows to solve partially the object segmentation problem with the help of motion information. However, final users and search engines are still more interested in the semantic content of the video, essential to the understanding of the scene. This semantic information is usually estimated by comparing the visual information that was extracting from the scene to available models in a knowledge base. Every region or object does not lead to a unique semantic interpretation. Without taking into account the general context of a scene, different concepts can be assigned. Then, it is preferable to associate regions and concepts in a fuzzy manner, attributing a certain degree of confidence for each concept.

A system able to achieve fuzzy semantic labeling of image regions has been made available to us [11]. Semantic information is related to the material composing the regions and is constructed from the matching of the visual description to a knowledge base. This new resource for representation of domain knowledge along with the spatiotemporal segmentation are put to good use to develop a method to perform jointly segmentation and annotation of video shots. From one side, semantic labeling brings additional information from domain of knowledge and can be used to group visually different elements sharing the same concept. From the other side, spatiotemporal segmentation propagates semantic labels inside the shot.

In this chapter, we first introduce the infrastructure of the knowledge base used to represent the semantic content. An internal ARG structure is adopted to describe both visual and semantic properties of the regions. The structure is intended for labeling of the regions and their semantic grouping. Secondly we address the problem of spatiotemporal semantic segmentation, focusing on the complexity of semantic labeling and the visual quality of the segmented regions. The proposed method is then based on two phases, the segmentation and the semantic labeling of block of frames, or intra-BOF processing, followed by the grouping of volumes from their semantic and visual characteristics, referred as inter-BOF processing. Finally concrete examples are presented to show the suitability of this new approach.

3.1 Contribution of multimedia knowledge base

The segmentation and annotation of semantic objects within a video sequence is becoming increasingly important in multimedia applications. Indeed the MPEG-7 provides facilities for manipulation of objects and its semantic metadata (section 1.1.2), but not for extracting these descriptions. As the semantic attributes of an object usually depend on the targeted application, semi-automatic methods have been mostly considered for this purpose [56]. However, this type of approach requires the intervention of the user to define the semantic video objects in one or more frames within the video sequence. Comparatively, automatic segmentation produces visual description at a lower level. In order to help structuring the low-level content into human understandable semantics, a representation of knowledge has to be considered. Traditionally, domain knowledge is incorporated only for semantic annotation of regions after the segmentation stage. Extracted visual features are used as input for classification techniques [10]. However, segmentation can suffer for the lack of prior knowledge, and classification need accurate segmentation of regions to obtain good results.

The framework of [11] tightly couples segmentation and labeling of multimedia content. An ontological infrastructure expresses multimedia content semantics and relationships within the domain knowledge. Low-level features are linked to high-level semantics through the use of the ontology, bridging the semantic gap. A multimedia Knowledge Assisted Analysis tool (KAA) is in charge of this task, by achieving fuzzy labeling of regions and semantic region growing techniques.

First, we describe the adopted knowledge base and its associated framework for object segmentation and labeling. Secondly, we explain in detail the semantic classification stage that enables automatic annotation of regions.

3.1.1 Principle and organization

The segmentation and object labeling system utilizes a knowledge base that links low-level features to semantics. It is composed of two main entities, an ontology which describes the semantics of the domain in a formal representation and the KAA experimentation platform that couples image analysis techniques and the ontological representation for semantic interpretation of the content. In this section we first introduce the multimedia knowledge infrastructure based on ontologies. Then, we describe the formal representation of multimedia content with ontologies for the creation of semantic descriptions. Once these two aspects are covered, we provide a concise description of the existing experimentation platform.

Design of a knowledge base

Petridis et al. [103] have described the basic requirements for a knowledge infrastructure to represent structural and semantic properties of multimedia content. The main issue is to gather the media content representation, notably described by its structural and visual properties, and the semantics contained in the media that are meaningful to the user.

The first point in designing the knowledge base is to manage multimedia content itself. The representation of low-level visual information necessitates:

- Low-level description representation to depict visual properties of concepts.
- Support for multiple visual descriptions, as a semantic object can appear in different views and aspects.
- Spatiotemporal relations to describe the arrangement of objects in the scene, which can be appropriate to characterize a particular domain.
- Structural representation adapted to the type of content, to capture content information and relations at different levels.

These requirements for multimedia representation are covered through the MPEG-7 standard, of which we show all fullness in chapter 1. MPEG-7 content entity tools (section 1.1.2) provide support for the visual descriptors and the hierarchical decomposition of the multimedia content into segments. Structural relations between segments are usually expressed with the SpatialRelation and TemporalRelation CS.

Besides these multimedia description aspects, the infrastructure has also to support specification of the knowledge domain. Authoring of domain ontologies requires:

- A way to associate visual features with concept descriptions.
- Modularization of the knowledge infrastructure. It is important to separate the definition of the ontological concepts from the visual descriptors.
- An annotation framework to train domain ontologies. Correspondences between visual features and semantic concepts has to be automatically generated.

The first and second point has been debated in the Semantic Web community. In MPEG-7, this can be handled by the abstraction mechanism provided by the semantic description tools. Concepts are defined in a separated formal abstraction of the semantic description as we proposed in the example of representation (section 1.1.3). The visual features can then be associated to any instance of the concept with the MediaOccurence DS in a semantic description scheme (semanticBase DS). As we will see in the next section, the authors propose a comparable mechanism with a *prototype* approach.

Ontology infrastructure

To manage these two different knowledge representation requirements, the authors define an ontology infrastructure. The framework includes ontologies for the description of low-level features (multimedia concepts) and for linking these descriptions to the domain knowledge concepts, considering prototype instances of these concepts. As discussed in section 1.1.2, expressing domain specific ontologies with the MPEG-7 is possible but not efficient in practice. The authors adopt instead the RDFS (Ressource Description Framework Scheme) modeling language which is simple to use and offer good trade-off between the capability

of the language and its complexity. Thus domain specific ontology are easily depicted with the RDFS while necessary MPEG-7 entities have to be adapted with the representations available in the RDFS.

The ontology infrastructure relies on a core ontology (DOLCE) that enables to represent generic knowledge and functionalities. A specialized multimedia ontology is built over the core ontology and implements MPEG-7 specifications. More precisely, a visual descriptor ontology (VDO) handles MPEG-7 visual descriptors, and multimedia structure ontology (MSO) represents multimedia entity and their relations. Last, domain ontology are added to model semantic aspects of the real-word domain ; the concepts are in general meaningful to the end-user. Concepts of the domain ontology are linked to the multimedia low-level content by referring a set of *prototypes* or instances for the concept. Each prototype points to its associated visual descriptors.

A user friendly annotation tool (M-OntoMat annotizer [11]) enables to import domain-specific knowledge inside the ontology structure. Region descriptors are associated to semantic concepts, instantiating domain concept prototype descriptors. Spatiotemporal relations inside or between two concepts are obtained considering couple of adjacent regions.

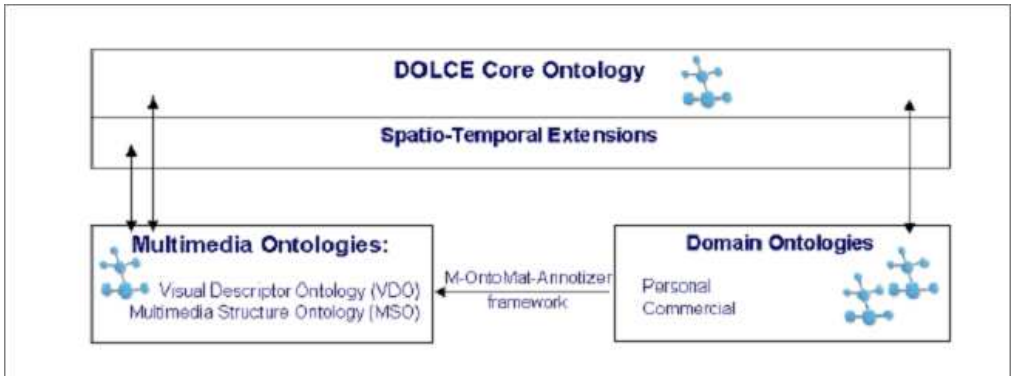


Figure 3.1: The ontology architecture. Extracted from [11].

Knowledge representation with ontologies

Ontologies attempt to model the real world by describing entities and their relationships in a formal machine processable representation. An ontology model $O = \{C, R\}$ is usually composed of a set of concepts C and a set of semantic relations. Semantic relations are functions that either relates to concepts or not :

$$\begin{aligned} &R = R_{pq}, (p, q) \in C \times C \\ &\text{where} \\ &R_{pq} \colon C \times C \rightarrow \{0, 1\} \end{aligned} \tag{3.1}$$

Different type of relationships can be defined between concepts, thereof the MPEG-7 defines more than 40 relations. However only *taxonomic* relations are efficient in practice

for the determination of the context [10]. Taxonomic relations enable to organize concepts hierarchically thanks to their properties of transitivity and antisymmetry [4], i.e. :

$$\begin{array}{lllll} R_{pq}=1 & \wedge & R_{qp}=1 & \Rightarrow & p=q \quad \text{antisymmetry} \\ R_{pq}=1 & \wedge & R_{qr}=1 & \Rightarrow & R_{pr}=1 \quad \text{transitivity} \end{array} \tag{3.2}$$

The considered semantic relations include *specialization* (Sp), *part* (P) and *property* (Pr) from the MPEG-7 standard. As we notice in section 1.1.2, concept and semantic relationships presence are governed by fuzzyness. Uncertainty comes either from the extraction procedure or from the nature of the relationships in the real world domain. A detection algorithm may give a confidence value for a detected object, e.g. "this is a person with a confidence of 0.8". Likewise, two semantically related concepts may not be instantiated every time. Stating that "Sky is part of Beach scene" will imply that sky is always present in a beach scene while this is not automatically the case.

Uncertainty in the semantic description of a video segment is then introduced by means of a fuzzy set F over the set of concepts C:

$$F = \sum_{i=1}^{|C|} c_i / \mu_F(c_i) \tag{3.3}$$

$\mu_F \colon C \rightarrow [0,1]$ is the membership function of F and $c_i \in C$. Along with the concepts, semantic relations are fuzzified considering fuzzy relations r_{pq}

$$r_{pq} \colon C \times C \rightarrow [0,1] \tag{3.4}$$

To simplify the relationships between concepts, a taxonomic relation T is built considering relations Sp, P, Pr :

$$T = trans(Sp \cup P^{-1} \cup Pr^{-1}) \tag{3.5}$$

$trans(R)$ denotes the transitive closure of R which is the smallest transitive relation on C that includes relation R. In consequence T is also a taxonomic relation. The value of the relation $T(p,q)$ indicates how much the meaning of q approaches the meaning of p. If q is detected, high values of $T(p,q)$ will indicate that p is likely to be present as well. In that sense, the value can also be interpreted as conditional probability of p given q. For instance, consider p="Beach Scene" and q="sky". Thus $P^{-1}(p,q)$ means that sky is part of Beach scene. Similarly if p="Vegetation" and q="tree", relation $Sp(p,q)$ that tree is specialization of vegetation, i.e. the meaning of p include the one of q (generalization), which is of relevant semantic meaning.

The ontology proposed is rather generic and can be applied in different domains. The ontology was constructed for different concepts for natural scene domain (beach, sand, sky, sea, person, boat, ...). In general, concepts are defined by domain experts, whereas the relationships values are trained using statistical properties obtained from the manual annotation. To enrich the domain of knowledge, the annotation tool (OntoMat-annotizer) can be used.

Knowledge Assisted Analysis (KAA) system

We have explained the knowledge representation using an ontology infrastructure in the previous section. Once the later is created, the KAA module enables to achieve semantic annotation of images. A global view of the platform architecture is shown fig.3.2. Image analysis begins with the segmentation of the image into regions based on a region merging algorithm that makes use of color region similarities and shape regularity. These regions are used to initialize an ARG which is the core of the architecture. The ARG stores both low-level descriptors and the degree of memberships for each concept in the knowledge base. The latter semantic information is obtained by matching each region to the sets of concepts. Classification algorithms are used for this purpose, taking into account the example prototypes in the knowledge base.

Region labeling is a difficult task and notably depends on the initial regions, along with the accuracy of the concept description in the knowledge base. To improve both segmentation and labeling, image segmentation algorithms that integrate domain knowledge and uncertainty of the labeling are proposed. It has been shown that traditional region merging algorithms such as Recursive Shortest Spanning Tree (RSST) or watershed can be adapted to operate on the fuzzy label sets of the ARG regions [10]. Finally, the degrees of memberships of the labels can be readjusted according to their overall relevance to the domain, using contextual information stored in the ontology [93].

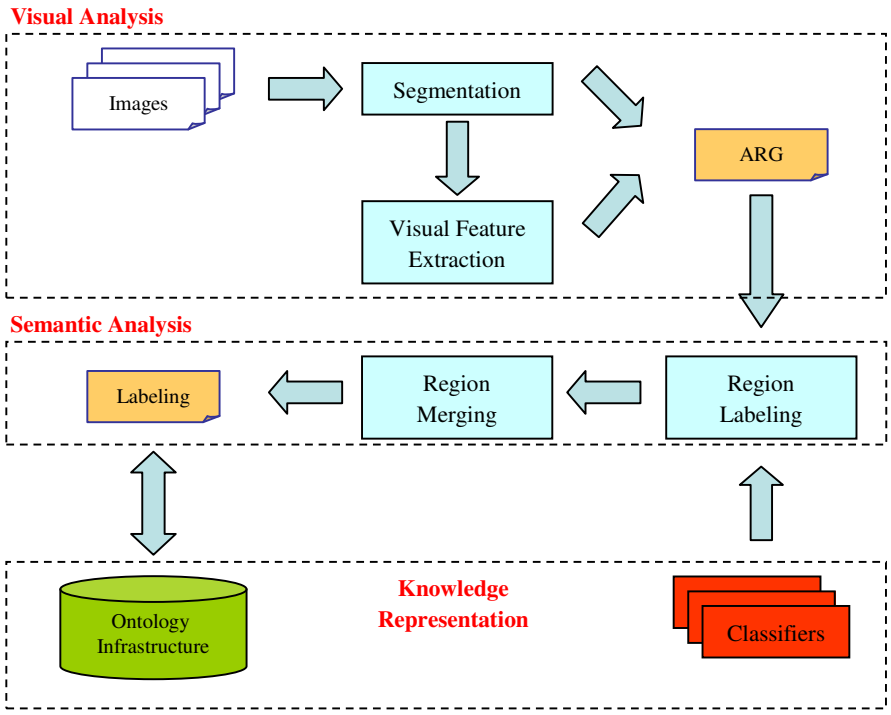


Figure 3.2: The Knowledge Assisted Analysis system.

3.1.2 Semantic labeling

The labeling of regions is an important step in the KAA system that links the low-level description to the semantic level. As a consequence, it conditions the quality of the final image annotation. Fuzzy sets are used to represent the couple of the label and its associated degrees of confidence for a region. This provides solid theoretical background for reflecting operations on the ARG structure on the semantic labels, with fuzzy operators such as aggregations, t-conorms and so on.

ARG representation

Image and video data can represent a structured set of objects and is naturally described by an ARG (section 1.2.4). We construct an ARG $G = (V, E, \nu, \xi)$ from a segmentation S as follows. A region $a \in S$ of the image is represented in the graph by vertex $v_a \in V$, where $v_a \equiv (a, \nu_D(a), \nu_L(a))$. Vertex attributes $\nu_D(a)$ represent the ordered set $\mathcal{D}$ of MPEG-7 visual descriptors characterizing the regions in terms of low-level features, while

$$\nu_L(a) = \sum_{i=1}^{|C|} c_i / \mu_a(c_i), \quad c_i \in C \tag{3.6}$$

is the fuzzy set of labels for the region defined over the crisp set of labels C and μ_a is the membership function of the fuzzy set of region a.

The adjacency relation between two neighbor regions $a, b \in S$ is represented by the graph edge $e_{ab} \equiv ((v_a, v_b), \xi_D(a,b), \xi_L(a,b))$. $\xi_D(a,b)$ is the visual similarity between a and b, calculated from their set of MPEG-7 descriptors $\nu_D(a)$ and $\nu_D(b)$ respectively. These descriptors come from different type of measurements, and several distance functions can be used. We further calculate the maximum and the minimum for the distance function in each descriptor and normalize with a linear scaling to unit range. Considering the set of descriptors $\mathcal{D}$, with normalized distance functions $\{n_d \colon d \in \mathcal{D}\}$ the final visual similarity of e_{ab} is computed by linear combination:

$$\begin{array}{l} \xi_D(a,b) = 1 - \sum_{d \in \mathcal{D}} \alpha_d n_d \\ \text{where} \\ n_d = \frac{d - d_{min}}{d_{max} - d_{min}}, \quad \sum_{d \in \mathcal{D}} \alpha_d = 1 \end{array} \tag{3.7}$$

where α_d is the importance given to feature d and d_{min}, d_{max} are minimum and maximum of the two distance functions, respectively.

The second similarity $\xi_L(a,b)$ is calculated based on the semantic similarity of the two regions a and b described by their fuzzy sets of labels $\nu_L(a)$ and $\nu_L(b)$:

$$\xi_L(a,b) = \sup_{c_i \in C} \left(t(\mu_a(c_i), \mu_b(c_i)) \right), \quad a \in S, b \in N_a \tag{3.8}$$

where N_a is the set of neighbor regions of a. The above formula states that the similarity of two regions is the least upper bound of all common concepts of the t-norm t of the degrees of confidence $\mu_a(c_i)$ and $\mu_b(c_i)$ for the specific concept c_i of the two regions a and b. Intuitively, eq.3.8 states that the semantic similarity $\xi_L(a,b)$ is the highest degree, implied by our knowledge, that a pair of neighboring regions a and b share the same concept.

Creation of visual and semantic attributes

The construction of the ARG G requires that low-level attributes are extracted for every segmented region $a \in G$. Visual descriptors considered include Scalable Color (SCD), ColorStructure (CSD), Homogeneous Texture (HTD), Edge Histogram(EHD), and Region Shape which where previously described in section 1.1.2.

Region labeling is based on the matching of the region descriptor to the prototype instances of all concepts in the ontological knowledge base [93]. Let $P(c)$ denote the set of prototypes for a concept $c \in C$. The matching of the region a to a prototype $p \in P(c)$ is performed by computing the visual similarity $s_{ap}^{\mathcal{D}}$. Feature weights α_d of eq.3.7 are adapted to reflect prior knowledge on the concept c. To estimate the degree of membership of a for the concept c, $\mu_a(c)$, exhaustive matching is performed from between region a and all prototype instances. $\mu(c)$ is calculated as :

$$\mu_a(c) = \max_{p \in P(c)} \xi_D(a, p) \tag{3.9}$$

Repeating this procedure for all concepts $c \in C$ gives the fuzzy labeling $\nu_L(a)$ of region a.

Efficiency of the matching depends directly of the size of the database, the ontology domain, and the relevance of the domain concept prototype instances. Such matching approach is related to nearest-neighbor classification, looking for the closest prototype for each concept. As the number of training examples increases, the quality of labeling is more robust but at the expense of computational cost.

To boost the efficiency of the labeling stage, region classifiers have been proposed. Among existing classifiers, support vector machines (SVM) have been widely adopted as they provide good efficiency for a large range of classification problems. They have particularly the ability for solving high-dimensionality problems and show good generalization performance [63]. With concern of the extensibility of the knowledge domain, a binary classifier is introduced for detecting each concept of the knowledge base. Then, each SVM was trained independently using a "one against all" approach.

SVM classifier estimates the optimal separating hyperplane between two classes in a certain feature space. For a sample $x \in \mathbb{R}^N$, the binary decision function $h(x)$ can be written as :

$$\begin{array}{l} h(x) = sign(f(x)) \\ \text{where} \\ f(x) = \sum_{i=1}^{L} \alpha_i K(x_i, x) + b \end{array} \tag{3.10}$$

$\{x_i\}_{i=1...L}$ is a set of samples called support vectors, K is a kernel function that transforms the original feature space, and b is a constant factor. The function $f(x)$ indicates the distance to the hyperplane in the transformed space. When the classes are not separable, support vector x_i stands within a margin ϵ_i of the optimal hyperplane, imposing $|f(x_i) - 1| < \epsilon_i$.

The distance to the hyperplane $f(x)$ can be used to return a degree of confidence in $[0, 1]$ for the binary decision $h(x)$. A sigmoid function is employed for this purpose :

$$P(h(x) = 1 | f(x)) = \frac{1}{1 + exp(-mf(x))} \tag{3.11}$$

This degree of confidence is used to estimate the membership functions $\mu_a(c_i)$ for every region $a \in S$ and concept $c_i \in C$. The slope parameter is experimentally set but fixed for all computations. However the estimation of this parameter can be done automatically by considering the maximum likelihood estimator for probability distribution of the training data [104].

Typically, a range of 500 images has been used to train each concept. Preliminary evaluation results show that polynomial kernel was most appropriate. Potentially, many descriptors can be considered in the training by use of the MPEG-7 XM software [92] or the AceMedia Toolbox [97], but can lead to increase the dimensionality of the problem worthlessly.

Material	Membership
Person	0.50
Sailing-Boat	0.49
Building	**0.63**
…	

Figure 3.3: Semantic labeling of image regions.

An example of labeling is shown fig.3.3. The image is segmented into regions. For each region each concept classifier outputs a score that is interpreted as the degree of membership for the concept. A score about 0.5 means that there is no indication for the concept ($f(x) = 0$), while high values indicates that the region is likely to belong to this concept. Thanks to the fuzzy labeling no hard decision is made, and all possible concepts are retained. Most robust classifiers are obtained from concepts associated to a certain type of material that can be characterized well by the visual features. In the example the building concept is predominant compared to the other concepts for the region at the center of the image.

3.2 Spatiotemporal semantic segmentation

We have introduced a knowledge representation infrastructure and a knowledge assisted analysis platform for semantic segmentation and annotation of images. An asset of this platform is to offer potential openings in both knowledge domain and different types of media content. In particular, we believe that the presented knowledge representation can be united with the spatiotemporal approach we developed previously in the frame of video content analysis. In this section, we will examine how coupling of visual information extracted from spatiotemporal segmentation with the knowledge analysis system helps the segmentation and the interpretation of video shots.

3.2.1 Introduction

We have seen in section 2.2 that spatiotemporal segmentation extracts continuous volumes from a video sequence. Algorithms developed until now consider that these volumes are homogeneous with respect to a set of visual features [49, 55, 31]. Important visual variations that can be found within a video sequence makes maintaining an object as a unique volume difficult.

For its part, semantic information would be able to link different part and views of an object, based on a set of prototypes that instantiated the object. More precisely, knowledge base recognizes region materials characterized by the visual indexes: shape, color, texture. However, when considering large range of multimedia documents, especially videos, it becomes a point where domain knowledge cannot cover all variations that may occur, and cannot give accurate interpretation of video content. Even if we cannot match closely part of real-world content to any concept prototypes, we can gain significant information by comparing this content to other contents that instantiates known concepts within the domain.

We believe that combining both knowledge-based representation and available visual information can help semantic interpretation of video shots. The ARG representation conveys these two aspects, expressing visual and semantic relationships between object volumes simultaneously. Grouping and matching procedures can bring visual and semantic information closer together and make them benefit from each other. In other words, we would like to appraise to which extent grouping and matching techniques can help for better semantic interpretation.

To fulfill this idea, we propose a framework where spatiotemporal segmentation and object labeling are coupled to achieve semantic annotation of video shots. On the one hand, spatiotemporal segmentation utilizes region merging and matching techniques to group visual content. On the other hand, semantic labeling of regions is obtained by computing and matching a set of visual descriptors to a set of concepts. The integration of semantic information from the existing knowledge base within the spatiotemporal grouping process sets two major challenges. Firstly, the cost of visual descriptor extraction and region labeling is important. For this reason, we cannot reevaluate descriptors or labels at each merging step unless only a few regions remain. Secondly, the relevance of the semantic description depends on the accuracy of the region descriptors, which means that the regions should have

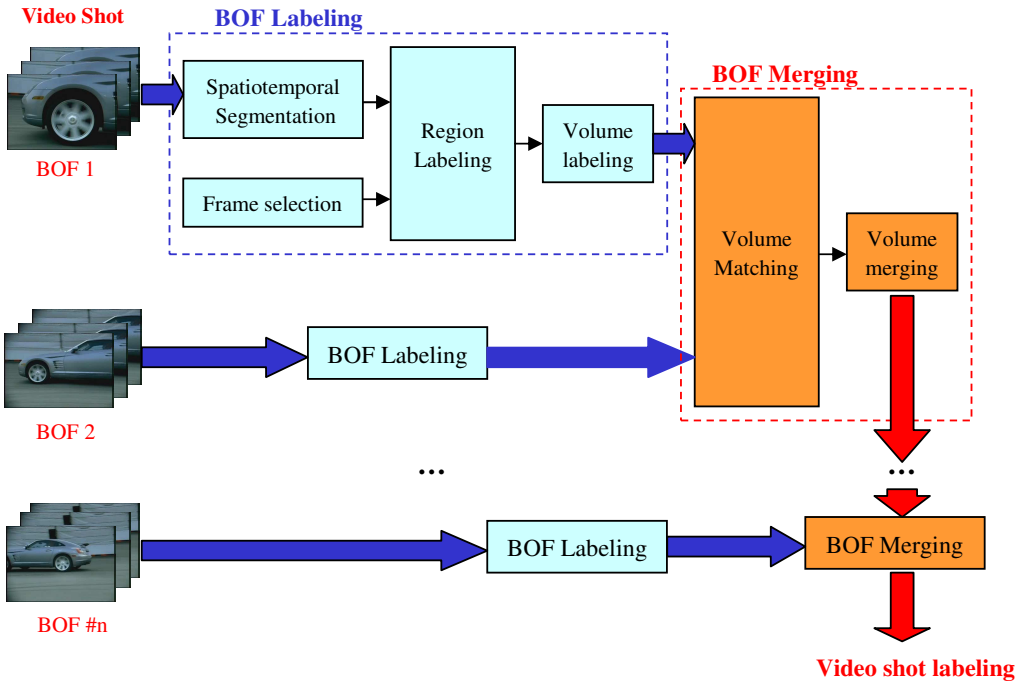


Figure 3.4: Framework for semantic video segmentation.

at least a sufficient area. These considerations suggest that use of semantic information during the early stages of the segmentation algorithm would be highly inefficient and ineffective if not misleading. Therefore, we add semantic information when the segmentation has produced a relatively small number of volumes. Taking into account these factors, we introduce a method to group semantically spatiotemporal regions within video shots.

3.2.2 Overview of the approach

We decompose the video shot into a sequence of Block of Frames (BOF). The advantage of such piecewise representation is that it naturally covers the different aspects (views, variation) of the semantic objects in the sequence.

Semantic video shot segmentation is achieved by an iterative procedure on the BOFs. It operates in two steps, respectively the labeling of volumes within the BOF and the merging with the previous BOF, which we will refer to as *intra-BOF* and *inter-BOF* processing. The procedure scheme is depicted in fig.3.4.

In a first stage of intra-BOF processing, a spatiotemporal segmentation decomposes each BOF into a set of volumes. The resulting frame segmentation maps are sampled temporally to obtain frame regions. These regions are semantically labeled and the result is propagated within the volumes. A semantic region growing algorithm is further applied to group adjacent volumes with strong semantic similarity.

In the second stage of inter-BOF processing, we perform joint propagation and re-estimation of the semantic labels between video segments. The volumes within each BOF are matched by means of the semantic labels and the visual features. This allow to extend the volumes through the whole sequence and not just within a short BOF. The semantic labels of the matched regions are re-evaluated and changes are propagated within each segment. Finally both BOFs are merged and the process is repeated on the next BOF.

3.2.3 Video shot decomposition

In section 1.2.4, we have seen different structures to decompose the video shot in the spatiotemporal domain (object, volume and pixel structures). In order to interact with the knowledge analysis system (section 3.1.1) we need to precise the structure of video segment in both spatial and temporal domain.

A shot is divided temporally into M Block of Frames $B_i, i \in [1, M]$. A block B_i is itself composed of a set of successive frames $F_t, t \in [1, |B_i|]$. Spatiotemporal segmentation decomposes each block B_i into a set of video regions (or volumes) S_{B_i}. Each volume $a \in S_{B_i}$ is subdivided temporally into frame regions $R_a(t)$, $F_t \in B_i$. Finally, the frame segmentation at time t is defined as the union of frame regions of all volumes intersecting frame F_t: $S_t = \bigcup_{a \cap F_t \neq \emptyset} R_a(t)$. Fig.3.5 illustrates the temporal and spatial decomposition of a BOF.

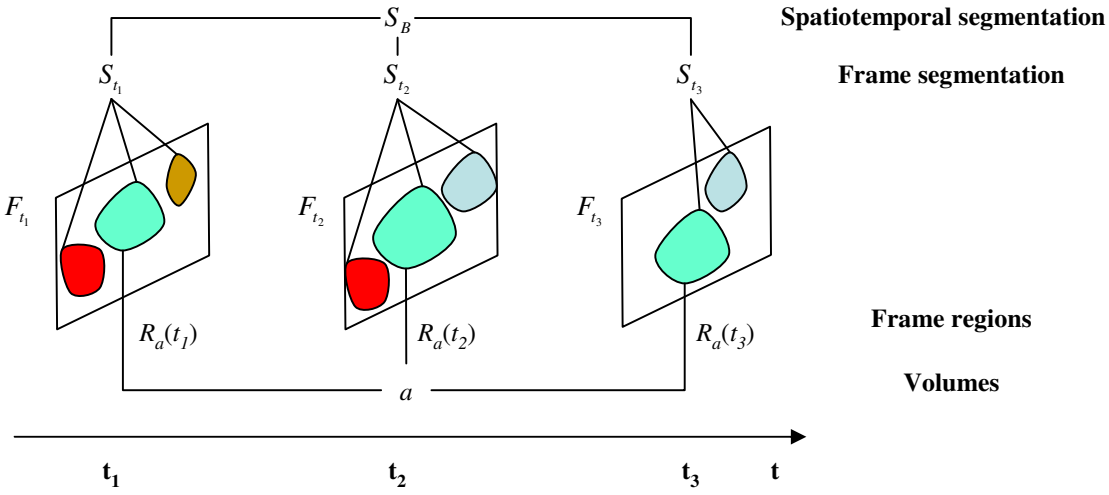


Figure 3.5: Spatial and temporal decomposition of a BOF.

3.2.4 Intra-BOF Processing

The most straightforward way to label one video segment would be to build volumes, extract spatiotemporal visual descriptors and use a knowledge base to select most relevant concepts for this region. Unfortunately, this is not directly possible for two reasons. Firstly, visual descriptor extraction has an high computational cost when considering complex descriptors,

and it will not be reasonable to apply it on all the frames of the video sequence. Secondly, the semantic labeling module (section 3.1.2) only performs labeling of image regions using spatial descriptors. In consequence, we exploit the spatiotemporal segmentation to build visual and semantic description efficiently, using only a few frames. Both a criterion for selecting these frames and semantic merging operations of volumes are presented below.

Selection of seed images

Once the segmentation masks are obtained for the whole BOF, region descriptor extraction and labeling tasks are substantially reduced by selecting temporally a set of frames within each video segment. This choice is important as it determines the span of the semantic labels within the BOF. Choosing an important number of frames will lead to a complete description of the BOF but will requires more time to process. On the contrary, using a single frame will be very efficient but important volumes may receive no labels. One possibility is to consider a set of frames T and its corresponding segmentations $S_T = \{S_t\}$, $t \in T$ and measure the total span of the intersected volume a. Given a fixed size for T we choose the set T_{sel} that maximizes the span of the labeled volumes :

$$T_{sel} = \underset{T}{\operatorname{argmax}} \sum_{a \cap S_T \neq \phi} |a| \tag{3.12}$$

The selection process is illustrated in fig.3.6. Estimation of the best set T_{sel} is performed as follows. First frame selected (t_2) is the one that intersects the biggest volumes in total. The selection of the second frame (t_1) is performed identically, volumes found in t_2 being not taken into account anymore. The procedure is repeated for a fixed number of frames or until the selected volumes occupy a sufficient proportion of the BOF.

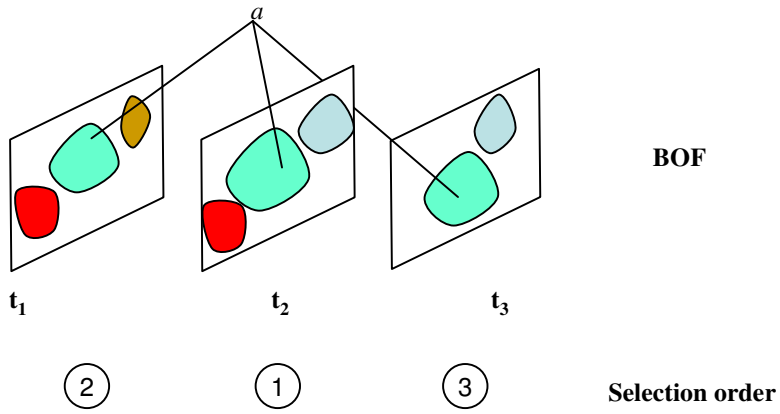


Figure 3.6: Temporal selection of frames. The frames which intersect most important volumes are selected.

The first advantage of this criterion is its independence to the descriptor type. Compared to fixed sampling, a second benefit is that it offers scalability for the extracted descriptors

in function of the desired total volume span for the shot. Indeed the span increases with the number of frames selected.

Spatiotemporal Propagation on a block of frames

We use the semantic annotation module to perform the region labeling in the individual frames selected in the previous step. Four visual descriptors, namely *Region Shape*, *Scalable Color*, *Edge Histogram* and *Homogeneous Texture* that characterizes different modalities of the regions are used to construct the SVM classifiers.

Similarly to frame regions, semantic attributes of volumes in BOF are described by a fuzzy set of labels ν_L over the set of concepts C. To compute the degree of memberships $\mu_a(c)$ of a volume a in a BOF for each concept $c \in C$, we gather the contributions from the set of labeled frames. In case that we consider a single frame, the spatiotemporal labeling is obvious as we just need to assign frame region labels to their corresponding volumes. Otherwise, when the volume is labeled at different instants, a unique set of labels is computed with fuzzy operators. Possible operators include :

- Default fuzzy union
$$\mu_a(c) = \max_{t \in T_{sel}} \mu_{R_a(t)}(c) \tag{3.13}$$
- Aggregation
$$\mu_{R_a(t)}(c) = \frac{\sum_{t \in T_{sel}} \mathcal{A}(R_a(t)) \mu_{R_a(t)}(c)}{\sum_{t \in T_{sel}} \mathcal{A}(R_a(t))} \tag{3.14}$$

Fuzzy union operator states that the volume a is assigned the concepts with their best confidence values. In other terms, detected concepts on frame regions are propagated on the whole volume. Other fuzzy union operators can also be used, such a dombi co-norm which softens the confidence degrees. The aggregation operator weights the confidence degrees with the importance we place in the frame regions. $\mathcal{A}(R_a(t))$, are obtained by a measure of temporal consistency of frame regions:

$$\mathcal{A}(R_a(t)) = \min_{u \in N_t} (R_a(t)) \tag{3.15}$$

where N_t represents the neighboring frames at time t in the BOF. These operators are selected based on the temporal consistency measures. In case of high consistency values, detected concept can be propagated reliably with the default fuzzy union operator. Otherwise, the importance of frame regions can be taken into account with the aggregation operator.

Besides its semantic labeling, volumes are also characterized by the set of visual descriptors $\mathcal{D}$ which describes its low-level content. MPEG-7 visual descriptors are originally considered for frame regions. However, we can extend some of them to volumes by considering aggregation operators, mentioned in section 1.2.6.

For histogram-based descriptors, MPEG-7 has proposed the GoF/GoP color for joining multiple image frames of a video segment by computing the *mean*, *median* or *intersection*

of histograms bins. As we consider homogeneous short-length volumes we prefer averaging the spatial descriptors with the *mean* operator. This type of aggregation can be applied for *Edge Histogram*, *Color Structure*, *Color Layout*, *Region Shape* and *Scalable Color* (after reconstruction in the HSV domain).

On the other end, the *Homogeneous Texture* descriptor (HTD) is composed of the average and the standard deviation of the region intensities, followed by the average and the standard deviation of the energy of 30 Gabor filters within the region (eq.1.1). Thus the corresponding volume descriptor (HTD) can be obtained by computing the average intensities and energies inside the volume, along with the standard deviations. Average energy is given by :

$$e_a = \frac{\sum_{t \in T_{sel}} |R_a(t)|\, e_{R_a(t)}}{\sum_{t \in T_{sel}} |R_a(t)|} \tag{3.16}$$

$$d_a^2 = \frac{\sum\limits_{t \in T_{sel}} |R_a(t)|\, d^2_{R_a(t)}}{\sum_{t \in T_{sel}} |R_a(t)|} + \frac{\sum\limits_{\substack{(t,u) \in T_{sel} \times T_{sel} \\ t \neq u}} |R_a(t)|\, |R_a(u)| \left(e_{R_a(t)} - e_{R_a(u)}\right)^2}{\left(\sum_{t \in T_{sel}} |R_a(t)|\right)^2} \tag{3.17}$$

Average and standard intensity can be computed in a similar fashion. For other descriptors, averaging is still possible with the *median* operator as long as a distance metric is defined. However it is only meaningful with a large number of descriptors.

In addition to spatial descriptors, we also store the sizes and location of the volumes. This last feature can be used to qualify spatiotemporal relationships. More precisely, each volume is located by its center of mass (eq.1.5) and its spatiotemporal bounding box expressed within the RegionLocator descriptor.

Semantic Volume Growing

Spatiotemporal segmentation usually creates more volumes than the actual number of objects present in the BOF. We examine how a variation of a traditional segmentation technique, the Recursive Shortest Spanning Tree (RSST) can be used to create more coherent volumes within a BOF.

RSST [87] is a bottom-up segmentation algorithm which iteratively merges similar neighbor regions until certain termination criteria are satisfied. RSST is based on the ARG representation of image regions, where each edge indicates the dissimilarity between two adjacent regions, e.g. color dissimilarity using Euclidean distance of the color components. In the beginning, all edges of the graph are sorted according to the edge weights. The edge with the least weight is found and the two regions connected by that edge are merged. After each step, the attributes of the merged regions (e.g. mean color) are re-calculated. Traditional RSST also re-calculate weights of related edges as well and sort them again, so that in every step the edge with the least weight will be selected. This process goes on recursively until termination criteria are met. Such criteria may vary, but usually these are either the number of regions or a threshold on the distance. The method can be put in relation to the

segmentation algorithm we describe in section 2.2.3, which is also based on the Minimum Spanning Tree of the graph. However, the latter algorithm neither recomputes weight of edges, nor re-sort edges, but use an internal difference criterion to evaluate if two regions can be merged.

The idea behind using RSST for semantic segmentation is that neighbor volumes, sharing the same concepts, as expressed by the labels assigned to them, should be merged, since they define a single object. As the graph contains reasonable number of volumes and edges, the cost of the ARG update operation is not a penalizing factor. The RSST is modified to operate on the fuzzy sets of labels of the volumes in a similar way as if it worked on low-level features (such as color, texture) [10]. Let G an ARG for a BOF. The modification of the traditional algorithm to its semantic equivalent lies on the re-definition of the two criteria. First the dissimilarity between two neighbor volumes a and b, $v_a, v_b \in G$ is expressed as :

$$w(e_{ab}) = 1 - \xi_L(a, b) \tag{3.18}$$

For one iteration of the semantic RSST, the process of volume merging decomposes in the following steps: First the edge e_{ab} with the least weight is selected. Then, vertices v_a and v_b are merged. Vertex v_b is removed completely from the ARG, whereas v_a is updated appropriately. This update procedure consists of the following two actions:

- (i) Re-evaluation of the degrees of membership of the labels in a weighted average fashion from the union of the two volumes:

$$\mu_a(c) \leftarrow \frac{|a|\mu_a(c) + |b|\mu_b(c)}{|a| + |b|} \tag{3.19}$$

- (ii) Re-adjustment of the ARG edges by removing edge e_{ab} and re-evaluating the weights of the affected edges, incident to a or b : $e \in E(a) \cup E(b)$.

This procedure terminates when the edge e^* with minimum weight in the ARG is above a threshold: $w(e^*) > T_w$, which is calculated in the beginning of the algorithm, based on the histogram of all weights of the set of all edges E.

The example of fig.3.7 illustrates the application of semantic volume growing. Semantic segmentation (fig.3.7(c)) reduces the number of volumes from the spatiotemporal segmentation (fig.3.7(b)), especially those with similar material type, the sea in occurrence for the example. Semantic segmentation helps therefore to overcome the oversegmentation problem mainly from illumination variations and color variances of the volumes. The volumes corresponding to the boat are partly merged. The "boat" concept prevails on its body where the degree of membership μ_{boat} is estimated to 0.61 and "foliage" for the rear buckle where μ_{boat} varies from 0.56 to 0.58. The rear buckle parts are merged together as they refer to the same type of material. Semantic information of the body also includes the "foliage" concept with degree of membership $\mu_{\text{foliage}} = 0.58$, which is comparable to the buckle labeling. This is due to the green color and the texture of the body, and the fact that the basis of the buckle is included in the volume. In consequence these parts are merged together (in pink in fig.3.7), resulting in a volume where the "foliage" concept prevails ahead

of the "boat" concept. At the front of the boat the buckle is unfortunately grouped with the sea, as the volume includes mixed material where the "sea" concept prevails ($\mu_{\text{sea}} = 0.56$, $\mu_{\text{foliage}} = 0.54$). Thus, this demonstrates that semantic segmentation incorporates compromise about the common semantic information between volumes and the predominance of concepts in each volume. Different interpretations can often be proposed for a volume, the best one actually depending on the global context.

(a) (b) (c)

Figure 3.7: Semantic Volume Growing. (a) Input sequence. (b) Spatiotemporal segmentation. (c) Segmentation after semantic volume growing.

3.2.5 Inter-BOF processing

We have defined the procedure for labeling volumes within each single BOF. Now we examine techniques to extend volumes over consecutive BOFs and improve labeling of the video segments processed individually. For this purpose we develop techniques for visual and semantic volume matching. First, semantic grouping is performed on volumes with strong concepts, then concepts are propagated temporally and spatially with the use of both semantic and visual similarity.

Selection of matches

We consider the merging of two successive BOFs represented by their ARG G_1 and G_2. It is not worth computing all volume matches between the two ARGs. As we consider continuous sequences, semantic objects are coherent spatially and temporally. In consequence, numerous matches can be pruned by exploiting spatiotemporal location of the volumes.

We establish temporal connections between G_1 and G_2 by selecting candidate matches from G_1 to G_2 and G_2 to G_1. Let G the merged graph of G_1 and G_2. At the beginning $G = G_1 \cup G_2$. Given volume $a \in G_1$ and $b \in G_2$, a is connected to b in G if the bounding box of b intersects a truncated pyramid that represents the possible locations of a as shown fig.3.8. The pyramid top base is defined by the bounding box of a, the bottom base is enlarged by a factor $D_s = v_{max}T_{max}$ where v_{max} is the maximum displacement between two frames and T_{max} is the height of the pyramid along the temporal axis. Thus the

condition is expressed as :

$$\begin{cases} min_t(b) - max_t(a) & < & T_{max} \\ min_u(b) - max_u(a) & < & v_{max}(min_t(b) - max_t(a)) \qquad u = x, y \\ min_u(a) - max_u(b) & < & v_{max}(min_t(b) - max_t(a)) \qquad u = x, y \end{cases} \tag{3.20}$$

where $[min_u\, max_u]$, $u = \{x, y, t\}$ denotes the bounding box coordinates along dimension u. The connections are established in both forward and backward temporal directions. As a result v_a has a set of candidate matches $E_a^C = E_a[G_2] = \{e_{ab} \colon b \in G_2\}$. The list of candidate volumes for b, $E_b^C = E_b[G_1]$ is defined and created symmetrically.

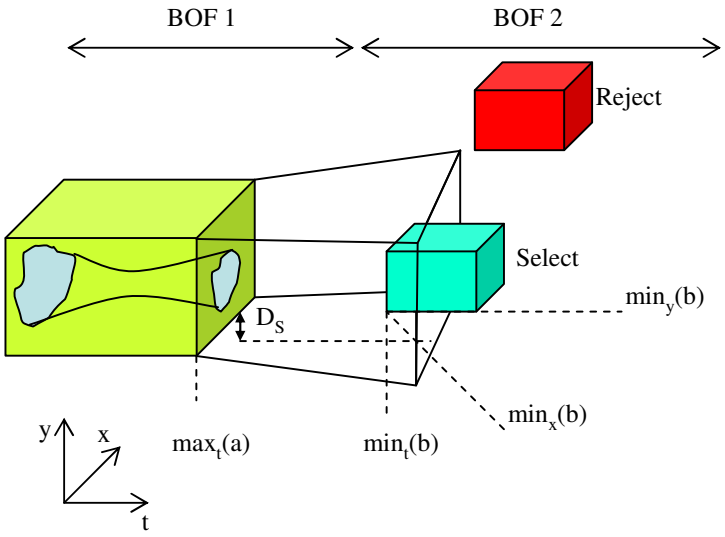


Figure 3.8: Selection of matches is based on the spatiotemporal location of their bounding boxes.

Matching of dominant volumes

After creating the list of candidate matches, we first consider the matching of volumes with reliable or *dominant* concept. A concept $c^* \in C$ is considered *dominant* for a volume $a \in G$ if :

$$\begin{cases} \mu_a(c^*) > T_{dom} \\ \mu_a(c^*) > T_{sec}\mu_a(c) \qquad \forall c \in C - \{c^*\} \end{cases} \tag{3.21}$$

A dominant concept has a degree of memberships above T_{dom} and is more important than all the other concepts, with minimum ratio of T_{sec}. Practically, there is usually a few dominant volumes after the labeling process in a given sequence. Therefore, values of T_{dom} may be reconsidered in case that the selection is empty.

The best match for one dominant volume may not be dominant because its visual appearance changes during the sequence. For this reason, we match either dominant volumes that have sufficient visual similarity or one dominant volume to any volume in case they have perfect visual matches. The criterion to match a dominant volume a to a volume b, $e_{ab} \in E_A^C$, is based on both semantic and visual attributes. Let c_a^* and c_b^* be the dominant concepts of $\nu_L(a)$ and $\nu_L(b)$. If b is dominant but $c_a^* \neq c_b^*$ then no matching is done. In case c_b^* is empty, then e_{ab} has to be the best visual match from a, otherwise we compute the normalized rank of the visual similarity ξ_D in decreasing order, which values do not depend of the descriptors used. Formally the criterion is validated if:

$$\begin{cases} rank\left(\xi_D(a,b)\right) = 1 & \text{if} \quad c_b^* = \emptyset \\ \begin{cases} c_a^* = c_b^* \\ \frac{|E_a^C| - rank(\xi_D(a,b))}{|E_a^C|-1} > T_s \end{cases} & \text{otherwise} \end{cases} \tag{3.22}$$

T_s indicates the tolerance allowed on visual attributes. When T_s is close to 1, only the best visual match is considered. If T_s is set to 0.5, half of the matches are kept.

The procedure is illustrated fig.3.9. In the example v_a is linked to v_c as it shares the same concept "foliage" and the semantic similarity is the second best ($\xi_D(a,c) = 0.8$). v_a is also linked to v_b since the similarity between a and b is the best one ($\xi_D(a,b) = 0.9$). v_d is not matched even if it shares the same dominant concept as they are visually different from v_a. Indeed only dominant matches with good similarity are kept.

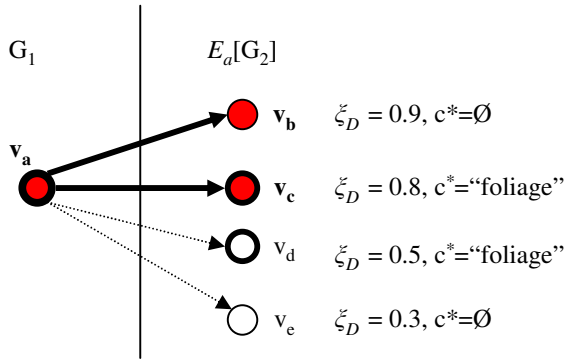


Figure 3.9: Matching of dominant volumes. Dominant volumes are represented with thick circles.

Since region and volume labeling are processes with a certain degree of uncertainty, reliable semantic concepts do not emerge from every volume, either due to the limited domain of the knowledge base, the imperfections of the segmentation or the material itself. We introduce also volume matching considering low-level visual attributes, expecting semantic of these volumes to be recognized with more certainty in a subsequent part of the sequence. To avoid propagating matching errors and hamper the accuracy of the volumes, we only

consider the matches with the strongest similarities and we are most confident in. We introduce the notion of *first best* match for an edge $e_{ab} \in G_1 \times G_2$. We denote e_a^* and e_b^* the edges in lists E_a^C and E_b^C which have maximum visual weights. e_{ab} is a first best match if $e_a^* = e_b^*$. This is illustrated in fig.3.10. An arrow from v_i to v_j indicates that $e_{v_i v_j}$ has been selected in the list of edge candidates $E_{v_i}^C$. In the example, couples (v_1, v_5) are matched as $e_{v_1 v_5}$ is the best match for both v_1 and v_5.

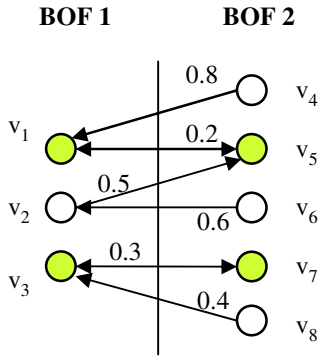


Figure 3.10: First best visual matches. Matched nodes are in green, and the numbers indicate the visual weights.

Update and Propagation of labels

After the semantic and visual matching process, volumes are merged and their semantic and visual properties are calculated using the aggregation operators we defined previously.

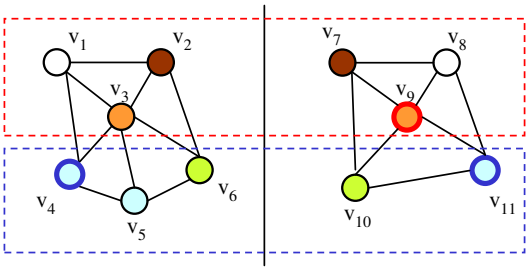
For this reason, new evidence for semantic similarity can be found in the merged graph as new dominant volumes are likely to be found. We do not merge further these volumes at this stage, to keep the accuracy of the visual description as they may correspond to different materials belonging to the same concept. Instead of this, the concepts of dominant volumes are propagated in the merged graph G. Let a be a non-dominant volume, $v_a \in G$; we define a set of candidate dominant concepts $C_a = \{c \in C | \mu_a(c) > T_c\}$. For a concept $c \in C_a$ we compute the degrees of membership $\mu'_a(c)$ resulting from the aggregation of v_a and its neighbor vertices in G with dominant concept c:

$$\mu'_a(c) = \frac{\sum_{b \in N_a^c} |b| \mu_b(c)}{\sum_{b \in N_a^c} |b|} \tag{3.23}$$

where $N_a^c = a \cup \{b \in N_a | c_b^* = c\}$ is the aforementioned neighborhood and $|b|$ is the current size of volume b. The concept $c^* \in C_a$, maximizing $\mu'_a(c)$, is selected and all degrees of membership of $\nu_L(a)$ and the size $|a|$ are updated by the aggregation of volumes in $N_a^{c^*}$. This propagation is performed in the whole graph G recursively. Let G^D be the subgraph of G containing only the dominant volumes of G and their incident edges. Once non-dominant volumes in G are processed, new dominant volumes may emerge in the subgraph

$G' = G - G^D$. The update procedure is repeated considering G' as the whole graph until no more dominant volumes are found, i.e. $G^D = \emptyset$. As a consequence, the degrees of membership of non-dominant volumes tend to increase using the neighborhood context, correcting the values from the initial labeling.

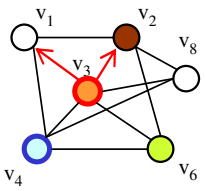
Fig.3.11 gives an example of the relationships before and after the merging. Only two concepts c_1 and c_2 are considered for the sake of clarity. The ideal semantic segmentation would be composed of two objects with dominant concepts c_1 and c_2. Before merging (fig.3.11(a)), a few dominant volumes are detected (v_4, v_9, v_{11}) in the two BOFs. After merging (fig.3.11(b)) the degrees of membership are re-evaluated according to eq. 3.19 and semantic weights are computed on the new edges. New evidence for semantic similarity is found between volumes (v_3, v_1) and (v_3, v_2), since v_3 has been matched with dominant volume v_9. Thus, due to propagation of concept c_1, v_1 and v_2 are linked to the dominant volume v_3 and their degrees of membership are increased according to eq. 3.23.



	$\mu(c_1)$	$\mu(c_2)$
v_1	0.7	0.5
v_2	0.65	0.45
v_3	0.7	0.6
v_4	0.55	**0.75**
v_5	0.5	0.7
v_6	0.5	0.55

	$\mu(c_1)$	$\mu(c_2)$
v_7	0.65	0.5
v_8	0.6	0.5
v_9	**0.85**	0.6
v_{10}	0.5	0.6
v_{11}	0.6	**0.85**

a)



	$\mu(c_1)$	$\mu(c_2)$
v_1	0.7	0.5
v_2	0.65	0.48
v_3	**0.78**	0.6
v_4	0.56	**0.79**
v_6	0.5	0.58
v_8	0.6	0.5

⇨

	$\mu(c_1)$	$\mu(c_2)$
v_1	0.74	0.5
v_2	0.72	0.48
v_3	**0.78**	0.6
v_4	0.56	**0.79**
v_6	0.5	0.58
v_8	0.6	0.5

b)

Figure 3.11: Merging of two BOFs. a) Matching between two BOF. b) Merging of a BOF and update of semantic labels. Ideal semantic segmentation is represented by the dashed boxes. Matched volumes are marked with similar colors, and dominant volumes are indicated with thick circles. Here, $T_{dom} = 0.75$ and $T_{sec} = 1.25$.

3.2.6 Results

Segmentation examples

We illustrate the potential of the method on a set of examples. The knowledge domain encompasses various elements encountered in a natural scene, such as “sea”, “sky”, “foliage” or “people”.

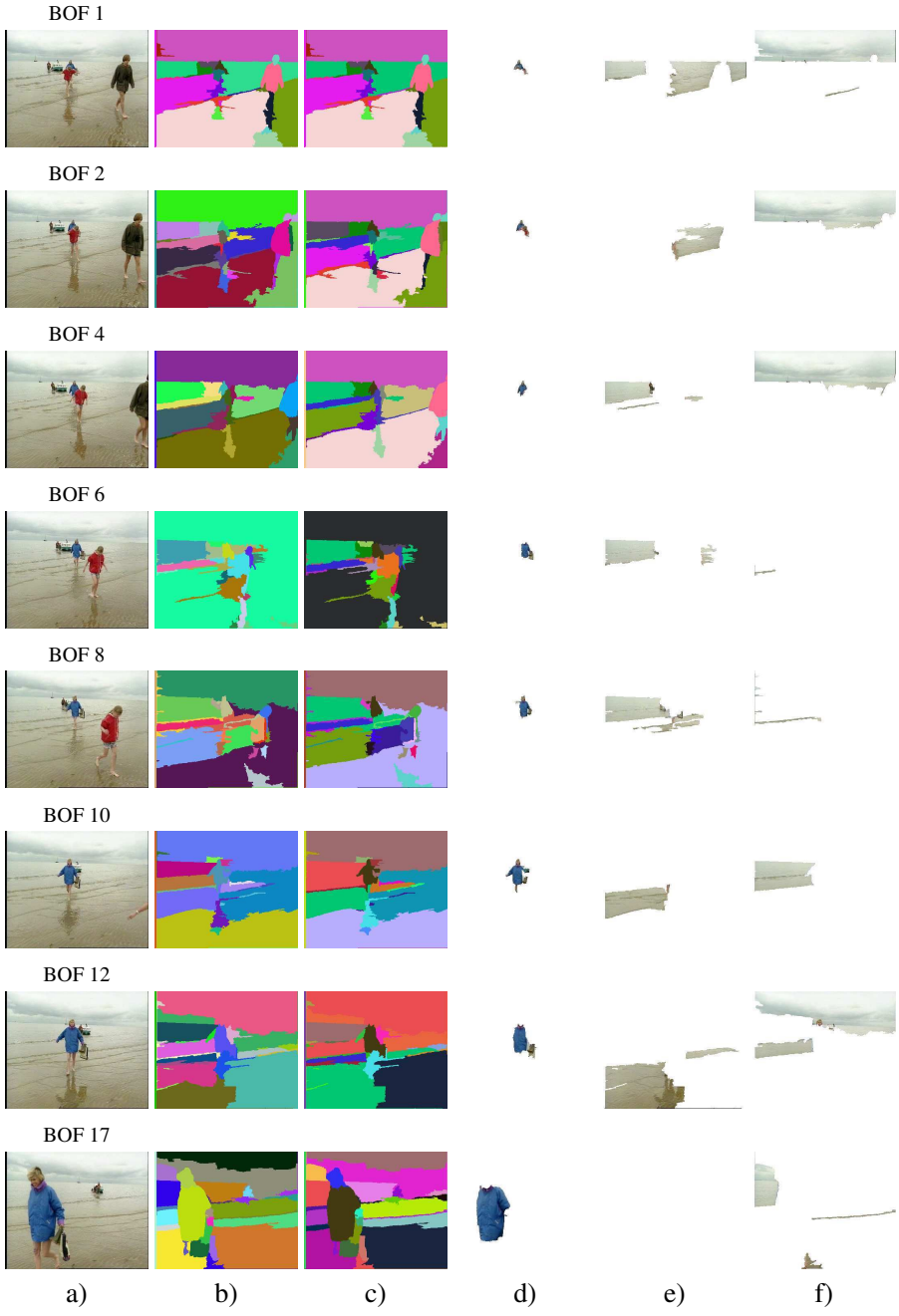


Figure 3.12: Video semantic segmentation. a) Frames in various BOFs. b) Spatiotemporal segmentation. c) Merged segmentation. d) Concept “person”. e) Concept “sea”. f) Concept “sky”.

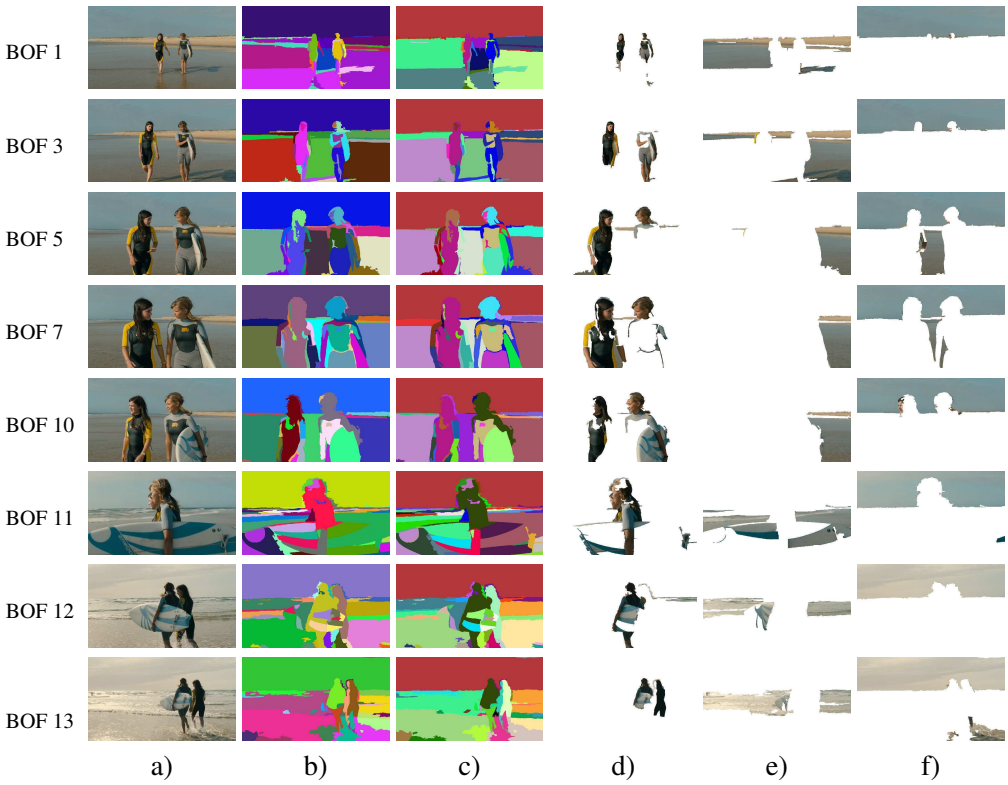


Figure 3.13: Video semantic segmentation. a) Frames in various BOF. b) Spatiotemporal segmentation. c) Semantic segmentation and inter-BOF matching. Volumes are extended throughout the shot (note the consistency in coloring). d) Concept "person". e) Concept "sea". f) Concept "sky".

For each example (fig.3.12,3.13,3.14) we show the original frames (a), the output of ST segmentation algorithm for the individual BOF (b), the merged segmentation for the whole sequence (c) and the volumes corresponding to the detected concepts (d-e-f). In the results shown, segmentation parameters are the same for all examples and give slightly over-segmented low-level volumes, outputting volumes with homogeneous color. The proposed example sequences of fig.3.12, 3.13 and 3.14 respectively lasts $7s$, $26s$ and $4s$. The BOF duration in the first and third sequence is of $|B| = 10$ frames while for the second sequence we increase the duration to $|B| = 50$ to show the behavior of the method at a larger scale while maintaining reduced computational costs. For the temporal sampling of the BOF, only two frames are selected for labeling using the criterion of eq.3.12.

The first example is a beach sequence where different persons are disembarking from a small boat in the seashore (fig.3.12). During the labeling of the different BOFs, relevant concept "people", "sky", "sea" have been detected, but only in a few volumes, as the woman with a blue coat sea, the upper part of the sea, and the clouds. One difficulty in the sequence is that the sky and sea regions are close visually and not well separated (BOF 6, 12, 17),

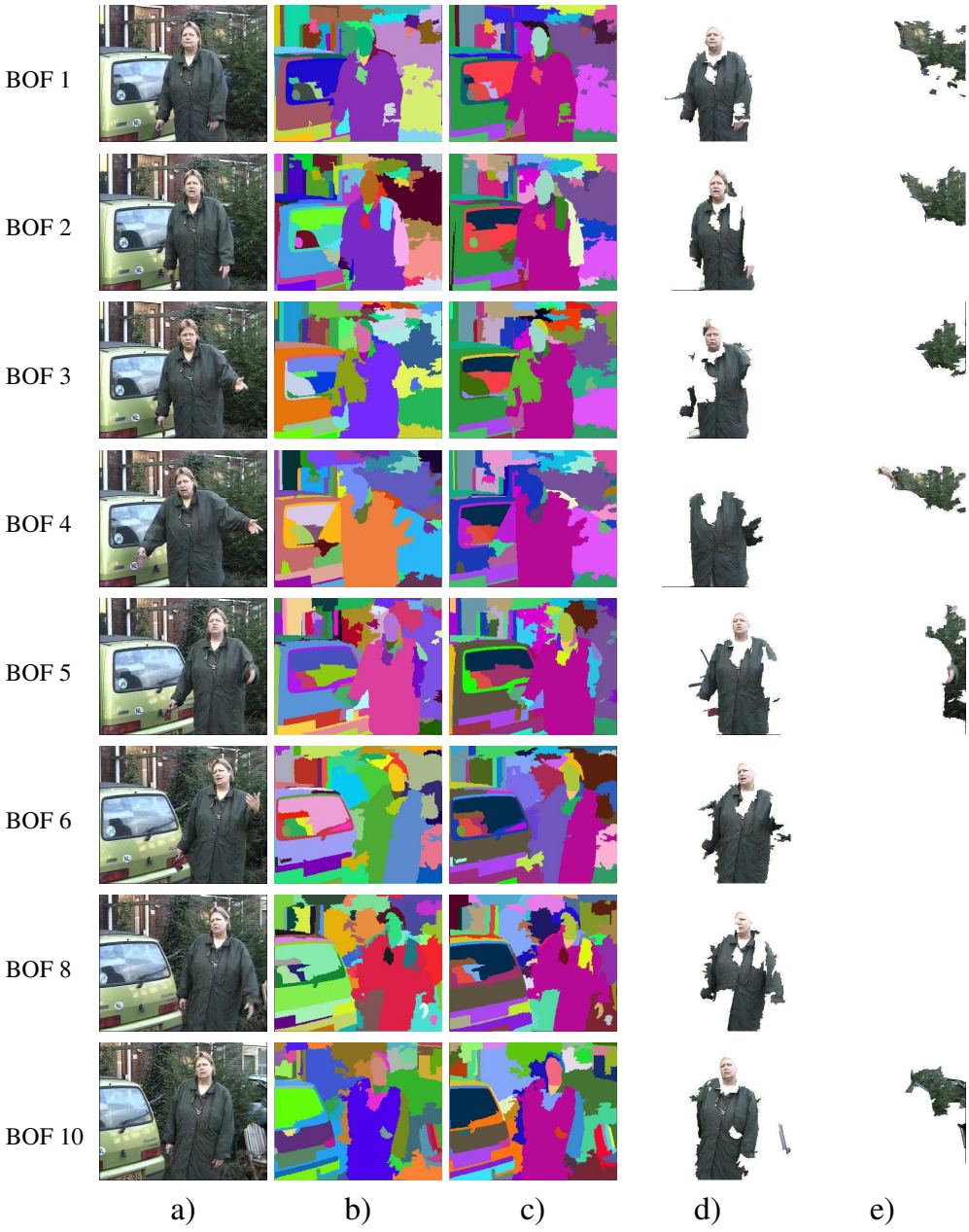


Figure 3.14: Video semantic segmentation. a) Frames in various BOF. b) Spatiotemporal segmentation. c) Merged segmentation. d) Concept “person”. e) Concept “foliage”.

hampering the low-level spatiotemporal segmentation process. For the same reason, the degrees of confidence for sea and sky concept are quite close in the concerned volumes in these BOFs. When matches between dominant concepts are not found, visual matching links regions which are visually close. This performed well on the person at the center, whose coat is composed of a single volume through the sequence from BOF 1 to 17, and on the girl on the right from BOF 1 to 4 as the layout changes smoothly. For the sea and sky regions the matching is only partial. Indeed they are subdivided in volumes with the same material, but with different layout. In such case these volumes should have been grouped after the segmentation of each individual BOF, in the semantic volume merging process. When no match is significantly good enough or stronger than the others, we prefer not to merge the volumes. For instance most volumes are not matched from BOF 4 to 6 and BOF 6 to 8 since sea and sky areas have been grouped in a single volume in BOF 6. Thus the error is kept within the BOF rather than being propagated subsequently in the sequence.

The second example shows two girls walking on the beach (fig.3.13). Firstly, the girls are approaching the camera (BOF 1-5). Then, they are observed in a close-up view (BOF 6-10). Finally the camera rotates quickly by 180 degrees to shoot them backside. Relevant concepts “people”, “sky” and “sea” are detected within the shot. First we can see that the sky area is recognized all along the sequence. Although its aspect slightly changes at the end, it is still detected as dominant in the labeling stage and thus merged as a single volume. We can notice that isolated areas are also labeled “sky”, as their material is visually close to this concept (BOF 5, 13). For the same reason, only part of the sea is identified at the right. In contrast, the left part is not dominant but is correctly grouped with the visual matching from BOF 3 to 10. After that point, the sea areas are easily detected as they are viewed frontally.The detection of “people” is more challenging since the related object includes different materials. In BOF 1 each silhouette is identified correctly standing as a single volume. The left girl’s area is propagated from BOF 3 to 10. After that point she is completely occluded by the other girl in BOF 11 and the concept is re-detected within a new volume in BOF 13. For the girl on the right the labeling is more uncertain as part of her suit and head have been confused with the background area (BOF 5, 7, 11). However, the upper part is still detected and propagated from BOF 5 to 9 and from 10 to 12 while the view is changing.

In the third example, a woman talking in front of her car (fig.3.14). The detected concepts include “person”, and “foliage”. Concept “building” and “car” are also present in the sequence, but are fragmented in too many volumes to be detected. The head and the coat both belong to the same semantic concept and can be viewed as a single object, but are still separated in the merged segmentation (c), which is an advantage as they are visually different. In BOF 4 only the coat is recognized (d). The reason is that the head has been partly confused with the background in the spatiotemporal segmentation. In such case the volume is not matched, as its visual properties are different from the other volumes in the previous and subsequent BOF. In the right part of the sequence, the upper branches are well identified as “foliage” and are merged in a single volume from BOF 1 to 4 (c). From BOF 6 to 8, the branches are occluded by the woman. In consequence the volumes are more fragmented and less homogeneous, so they are not linked to the previous part of the

sequence. In BOF 10, the volume material in this area is homogeneous and the branches are correctly identified again.

Computing time

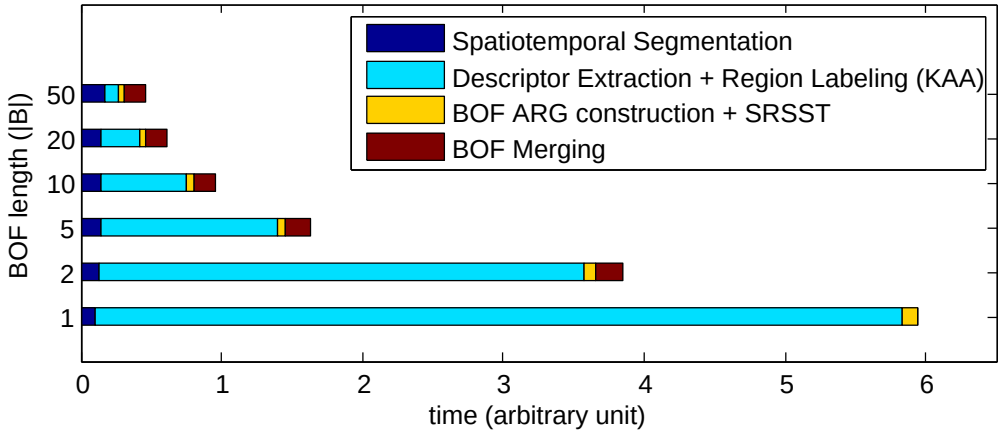
When considering a sequence of images, semantic labeling become very costly using the KAA system for every image. The spatiotemporal scheme we proposed enables to reduce significantly the complexity of semantic labeling by propagating the labels between regions. We compare the respective computing time of the different modules. We decompose the whole process in four steps:

1. The spatiotemporal segmentation stage.
2. Visual descriptor extraction and region labeling for the KAA system for the selected frames.
3. The construction of the ARGs for each BOF, including the semantic RSST.
4. The inter-BOF processing stage that merges consecutive BOFs.

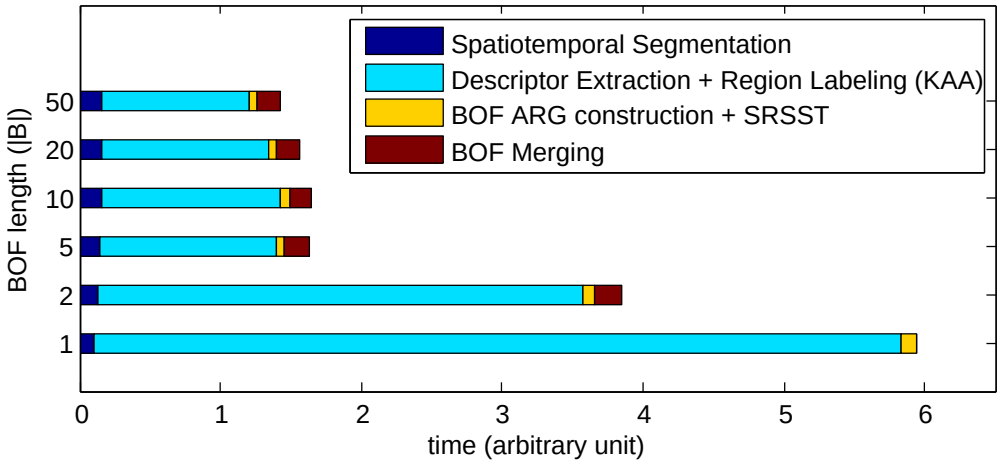
To show the impact of splitting the sequence into video blocks, we consider different BOF lengths, from $|B| = 1$ to $|B| = 50$ frames. The evaluation is performed on the beach sequence of fig.3.13, which is composed of 650 frames. Fig.3.15(a) and 3.15(b) show the repartition of the overall running time selecting respectively 1 frame per BOF and 20% of the BOF length for labeling. The former configuration gives the minimum time for processing a sequence, while the second one utilizes an important number of frames. Processing frames independently ($|B| = 1$) leads to consider the KAA system alone. In this case, only the segmentation (1), the region labeling (2) and the semantic region growing stage (3) are included in the computation time.

Fig.3.15(a) clearly demonstrates that the complexity of processing individual images is dominated by the descriptor extraction and region labeling stage (96% percent of total time). When considering the spatiotemporal approach, these costly stages are reduced up to 30% of the total time considering a BOF with 50 frames.

The remaining computing time is shared by the other modules. Spatiotemporal segmentation on a large BOF does not imply a significant increase in the processing time. Even if the segmentation of individual frames has a lower complexity, the increase in the computational cost is very low compared to the region labeling stage. For the construction of the BOF ARGs the ratio is inverted as constructing an ARG for each frame is less efficient than processing a single ARG on a block. In the final BOF merging stage, the processing time does not decrease significantly as the I/O processing required to load and update the segmentation maps from the ARGs is more important than the merging of the ARGs itself. When the latter merging stage is considered solely, the complexity of the procedure is directly linked to the number of blocks used for the sequence. Increasing the block length from 2 to 50 frames results in reducing the required time by a factor of 15. All modules considered, the gain with the spatiotemporal approach can reach a factor up to 12.



(a)



(b)

Figure 3.15: Repartition of the running time for different BOF sizes. a) A single frame is selected as input to the KAA system. b) 20% of the BOF frames are selected.

In fig.3.15(b) a minimum ratio of one to five frames is selected (only 1 frame is selected for block sizes of 1 and 2 frames). The results corroborate the fact that the processing time is led by the descriptor extraction and region labeling task. Indeed, the processing time from block sizes of 5 to 50 are comparable and the relative importance of these two steps in the total processing time is about 80%. The minor differences between each experiment are due to the different sets of selected frames and the number of labeled regions. Compared to fig.3.15(a) the cost for building the ARGs also increases with the number of selected frames as the visual and semantic attributes from different frame regions are merged.

The gain in computing time for other sequences also complies to the example we have described. Thus, running time analysis shows that the proposed framework enables to

extend single image annotation to continuous sequences efficiently. The cost of labeling is reduced to a level comparable to the other components of the system. Although the presented results are already significant and the ARG matching and merging procedure are efficient, the cost of the procedure can still be enhanced by reducing extra computation due to the I/O processing between the different modules.

3.3 Conclusion

In this chapter, we have developed a method for semantic spatiotemporal segmentation which makes use of a-priori knowledge to help for segmenting objects with semantic meaning in a video sequence. To this aim we introduced a framework based on an existing multimedia knowledge-assisted analysis system which enables semantic annotation of image regions. The extension of this system to the spatiotemporal domain faces two major challenges, respectively the temporal variations within a video sequence which affect image segmentation and labeling, and the computational cost of region labeling. The method we proposed propagates semantic labeling along low-level spatiotemporal volumes. In this sense, we compensate semantic information with visual information when the former is missing. In addition, propagation of semantic information is then extended to a broader area thanks to a matching procedure between video segments. Our approach groups volumes with relevant concepts together while providing a spatiotemporal segmentation for the entire sequence. In this way, the volumes we segmented previously can still be linked to a concept at any subsequent point in the sequence. Experiments on various sequences show that the method is promising for semantic interpretation of video shots an can reduce significantly the cost of region labeling in video sequences.

Further challenge will be to consider structured objects instead of simple individual objects, exploiting ontological knowledge for proposing global interpretation of graph structures from their fuzzy labeling. Such approach will be a step forward to the detection of complex events.

Chapter 4

Indexing and retrieval with the spatiotemporal representation

In the preceding chapters, we structured the content of video shots with a decomposition into segments. These fragments can either represent visual information under the form of spatiotemporal volumes or a semantic information in the form of object and materials. Now, we address the usage of spatiotemporal representation for the indexation of video shots.

The indexation of a document consists in the extraction of a structured information with the aim to retrieve it within a collection of documents. This information has to be organized and described concisely in order to make the search efficient and limit the cost of storage and transport.

Until now numerous image indexing and retrieval systems have been developed, leaning upon textual annotation, global or local visual features. Most operational indexing techniques make usage of textual content of the document, as attested by the massive usage of Internet search engines, facilitating the access and consultation of pertinent information among billions of available documents.

Popular image and video databases are still indexed using textual terms, which is made possible by user annotation. As a rule keyword-based indexing has proved to be insufficient in context of visual databases, and is not able to represent the perceptual properties of the visual data. Visual content is not adequately described by a few keywords and the annotation itself depends on the user interest. Furthermore, manual interaction for annotating and querying the database requires a huge effort. For these reasons, searching particular shots and objects in an unknown document are problems which require to analyze and compare the visual content of the documents.

Comparing with textual or global image features, indexing from spatiotemporal volumes is more complex but enables to build an accurate representation of objects and their relationships, operating in a similar way as the visual system. In addition, indexing and retrieval experiments give a feedback on the representativity of the spatiotemporal description which is difficult to evaluate by other means.

This chapter presents the indexing of the shot from spatiotemporal volumes. First section reviews the problem of search in image and video databases. We draw up a state of the art

of the indexation models and techniques for the description of the video content.Then, we introduce the first approach that enables a compact representation for image and video shots, called Vector Space Model (VSM) and discussed from its properties. Using this model, we compare different image segmentation methods with spatiotemporal segmentation and show the advantages of the latter. Finally, we address the problem of retrieving complex objects using directly the graph representation introduced in chapter 1. To this aim we define a new similarity measure using visual and structural properties to compare two ARGs. We discuss the benefits of the measure from evaluation with a retrieval experiment.

4.1 Indexing and retrieval

The growth of visual information available in video has spurred the development of efficient techniques to represent, organize and store the video data in a coherent way. Content-based indexing address this problem, setting up a structured representation of the content (the index) which makes access and comparison easier. The indexing process involves to define a model for representing content, which will be exploited to identify the documents readily in the search phase. Possible forms of representation encompass manifold aspects of the document, according to the desired level of abstraction for the data. The index keys can be based on the low-level description of the content at the signal level, e.g. the characteristic colors or textures within an image. At a superior level, description of content includes structured objects and their interaction within the scene. At the highest level, the representation can include semantic properties of the objects and of the scene in general. The higher the level of abstraction is, the more complex the extraction and reasoning process becomes.

Comparatively, the problem of retrieval consists in returning a list of documents which conforms to the request of a user. The query is constructed to represent the user needs, and compared to the index that contains the representation of all documents. The retrieved image and video are then presented to the user in order of their relevance to the query. A request can be formulated into different forms, given the capabilities offered by the model used in the indexing phase. Textual queries are the easiest to manage as the request is usually limited to a defined set of keywords. On the contrary, image and video requests are more difficult to manage as they cannot be formulated in the numerical form of representation that is stored by the index. An appropriate solution is to adopt a *query by example* scheme. An example can include an image, a shot, an object, or a sketch drawn by the user. Content properties featured in the indexing model are then evaluated to construct the query and perform the search in the database.

4.1.1 System architecture

The problems of indexing a document and of the retrieval of a query are tightly coupled, the choice of a content model leading to the design of an efficient search strategy. Generally speaking, it is referred to as content based indexing and retrieval system (CBIR). An architecture for a generic CBIR is shown in fig.4.1. In the indexation phase, the structured

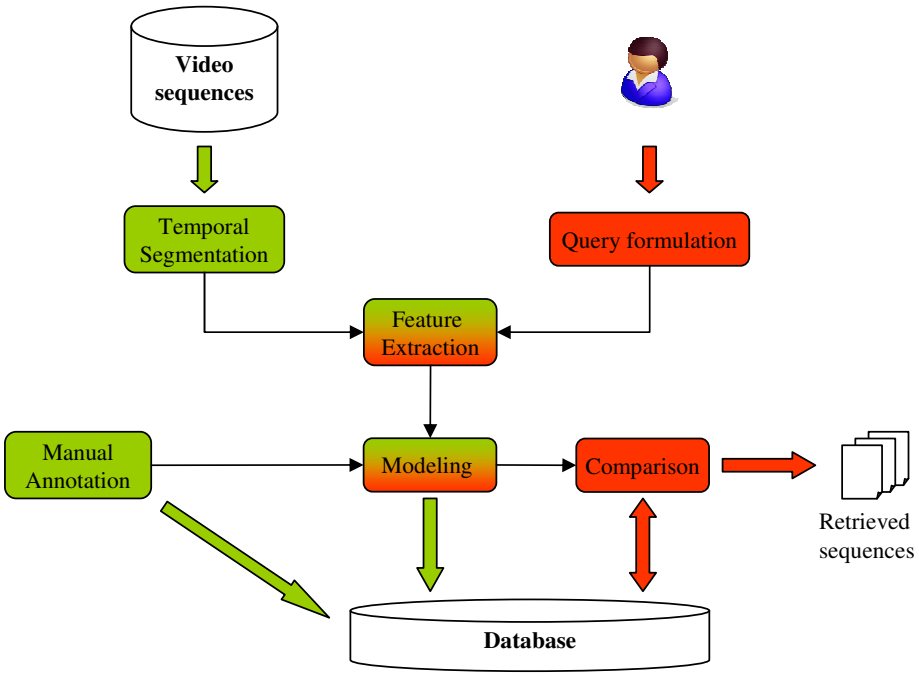


Figure 4.1: Indexing and retrieval of video content. The indexing phase is highlighted in green, the retrieval phase in red.

organization of the database is computed in several steps. The first step is to discover the temporal structure of the video. The process of decompositing a raw video sequence (video stream) into shots is called temporal decomposition. To achieve this task, efficient techniques for the detection of shot transitions have been proposed, either in the uncompressed or compressed domain [69].

Once the temporal decomposition of the sequence is appropriately achieved, features are extracted from the shot or its subparts (regions, objects). Dynamic features that capture temporal aspects of data are represented in the shot, while spatial features can be extracted from one or more representative keyframes.

A model of video content is then constructed from the low-level feature representation. The model can be directly described by the whole set of features, to which a structure that depicts the spatiotemporal arrangement of the regions is added. This type of content modeling scheme is well adapted in the query by example context, where a query is formulated from visual data (image, video clip, sketch). Retrieval based on semantic properties (keywords) requires in addition high-level indexing of semantic terms related to the content, which impose to model the domain of knowledge. Inference is usually based on machine learning systems trained from user annotations.

Content information is stored in a database that contains all representations of the

extracted segments. Basically, the index is a table of couples (identifier, descriptor). The identifier contains information such as temporal and physical location of the segment, which enables to find and present the content easily. A descriptor contains the associated content, according to the content model representation. This descriptor is used for comparison of the documents. Extracted indices have to be organized to optimize the search in the database. To reduce the cost of processing and storage, the descriptor can be compacted into a single feature vector (signature). When the content model is composed of multiple attributes, hierarchical structures such as R*-trees, quad-trees have been demonstrated to be appropriate for multi-dimensional indexing [3].

4.1.2 State of the art

Efforts have been made into developing methods for content-based indexing and retrieval of videos (CBVIR). Our interest is placed on CBVIR systems based on shot and object representations. The sequences are supposed to be preliminarily parsed into smaller meaningful temporal units (a.k.a. the shots). In these systems, an instance of the shot or of the object is given as a query, and the similar objects need to be found. The problem of unsupervised indexing system is difficult in itself, as objects and shot content can take different appearance in different shots and different videos. In this section, we review the approaches for indexing and retrieval that are based on the visual content description. Prominent issues include indexing of high-dimensionality descriptors, definition of good similarity matching measures for comparing accurately visual information from two units, or efficient techniques for compact storage and search. In the following section, we review the approaches that have been proposed through the literature to tackle them.

Video indexing methods have primarily used image-based methods, reducing the extraction of information in keyframes. Common approaches leans upon global properties of the image, or local description provided by the segmentation into regions.

Global Description

Different features have been considered to depict the image content. Usually three categories of features are considered : color, texture and shape. Color and texture have been widely adopted for image retrieval, while shape descriptors have been most efficient for specialized domains (e.g. object recognition problem). Typically the color of an image is represented by an histogram in an existing color space. Histogram descriptors are then compared to the one of a query image by means of similarity metric. Measures used for search are histogram intersection [123], euclidean distance. To reduce the quantization effects involved by histogram representation, Smith et al. [118] proposed to divide the color feature space into color sets. Color sets are obtained from the quantization of the color space. Robust comparison of histogram is then performed using weighted distance which computes the cross-correlation between the color histograms [44]. Another solution to the binning problem is to adapt the length of the descriptor according to the perceptual complexity of the content. Rubner et al. [107] propose to use a similarity metric called Earth Mover's Distance (EMD) to compare signatures which are composed of a set of representative feature clusters of the

image. However, the increase in complexity is important. Color histograms have been shown to support invariance to common image transformations, such as translations, rotations, scaling. However, remaining problems from the use of color similarities is that the feature can be affected by global illumination variations, also known as the color constancy problem.

Texture is also an important feature to depict the patterns present in an image. In the early work of Jain [134], an histogram of edge directions is added to color information for retrieval of trademark images. Other statistical approaches have been developed. Among them co-occurrences matrices and the derived Haralick indices [57], Tamura features that characterize coarseness, contrast and directionality [125] are worth mentioning. The feature is also effectively represented in the spectral domain, as patterns are composed of repetitive basic elements. Fourier Transform is used to detect global periodicity of an image [54]. A more precise decomposition in the spectral domain is introduced through the Gabor wavelet decomposition on which is based the MPEG-7 Homogeneous Texture descriptor.

An early platform that integrates all features of CBIR systems is the IBM's Query By Image Content (QBIC) [44]. Images are described by multiple visual properties. Color distribution is represented in a color space with 256 representative colors obtained from the clustering of an RGB histogram. Texture is described using Tamura's features and shape is expressed by a combination of geometric features including area, eccentricity and major axis orientation. The queries are based on one or several visual properties from an image or object drawn by the user. A simple query may consist in identifying images that have a similar color distribution from the example image. More complex queries can be formulated, searching into different features in parallel or conducting a new search based on the result of an initial search. In the retrieval phase, nearest neighbor search using multiple features is performed in the feature space efficiently by storing the multidimensional indexes in a R*-tree.

The QBIC components have been reused for newer CBVIR systems. The more advanced IBM's CueVideo system achieves indexing of multiple modalities (audio, video, text) in target of video summarization and event recognition. The QBIC system is integrated to perform image based search by computing the similarity between frames.

Other well-known image systems based on multiple features of an image includes Virage Photobook, VisualSeek and Netra. These systems are reviewed in [137]. In these CBIR engines, the use of multiple descriptors is let to the choice of the user. The weighting of each feature is refined by the user itself or in function of the relevance of the displaced results (relevance feedback).

To simplify the indexing process, these descriptors can be fused in a unique single vector. A static approach by concatenation is straightforward but leads to high dimensional descriptors. The problem of increasing the feature space dimensionality is also known as the "Curse of dimensionality problem" [15]. As the feature set increases the ability to describe optimally the data diminishes. Dimensionality reduction is also suitable to enhance the index structure performance and speed distance computations. Common techniques consist in mapping the original data set into a lower representative space. The most known transform is the KLT (Karhunen-Loeve transform) that performs an optimal linear mapping of the data, in a way that the variance of the data in the low-dimensional representation is

maximized. In practice, the problem is solved by computing the eigenvectors of the data covariance matrix with the SVD decomposition. Other interesting transform includes transform to the spectral domain, such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT). In comparison to the SVD, these techniques are highly computationally efficient, but the efficiency of representation is not optimal and depends on the domain considered.

Region-based description

A human user naturally organizes the scene into objects, and would like to find similar elements in other shots. To achieve this goal, the modeling of image and video demands, in function of the application, that the content is analyzed locally at the region and object level. Such representation is obtained by segmentation, that subdivides the scene into objects.

Blobworld, proposed by Carson et al. [22], is an example of a region-based indexing system. Image regions are represented by the color, texture, shape and spatial characteristics of the segment. These "blobs" are obtained by a segmentation technique that produces coherent regions in texture and color space using an EM algorithm. Each blob is indexed separately in the database, one entire image being constituted of a small number of blobs. In the retrieval process, one or more blobs are selected in the query image and blob comparison is achieved. The quality of the approach depends heavily on the ability of the segmentation to cover meaningful objects.

Netra [79] is another retrieval system that uses color, texture, shape and spatial location to perform region-based search. A major feature of the system is the image segmentation algorithm that is shown to improve quality of image retrieval when the image is composed of multiple complex objects. Other improvements reside in an efficient color representation and in the indexing of color, texture, and shape features for fast search and retrieval. Thanks to this internal representation, the system enables to elaborate more accurate queries that specify both characteristics and spatial relationships. Characteristics can be also formulated in terms of "texture keywords" to select and query, such as "blue sky", "flower", "mountains", while a number of pre-classified image categories of image collections are used, such as "glacier and mountains", "pacific coasts", "autumn". In the systems described above, regions are retrieved independently of the scene they belong to.

Indexing and retrieval operations can also consider the whole set of regions at a glance. In the SIMPLicity system [142], a measure for the overall similarity between images is developed using a region matching scheme that integrates properties of all the regions in the images. Image categorization into general purpose classes (Indoor/Outdoor, textured/non textured) enables to adapt and reduce the range of search within the database. Compared with systems that consider individual regions, using an overall similarity reduces the effect of inaccurate segmentation and helps to clarify the semantics of a particular region.

Other representations have been proposed for-object oriented approaches. A segmented image can be converted in a symbolic picture that depicts the relationships between the objects in the image. 2D strings have been proved efficient for retrieval of symbolic and pictorial images. The problem of retrieval consists in a 2D sequence matching. To represent relationships in the image, each object is labeled and attributed a symbol. A 2D string is

then constructed by projecting the symbolic pictures onto the x and y axis. A symbolic projection is described by the relationships between objects, such as *below-above* or *left-right*. Since the original 2D-string theory was developed by Chang et al. [27], diverse extensions have been proposed to enhance the descriptions of spatial relationships. For instance, the 2D G string can be used to describe accurate relation between sub-objects, allowing a larger set of relationships such as *meets*, *overlap*, *contain*. On the other hand, partitioning objects increases significantly the storage requirements and the complexity of matching. 2D-strings have been employed in the VisualSeek system [118] for performing search of a sketched query composed of multiple regions. To reduce complexity of the matching, candidate regions in the target image are first listed from the response to individual query region. Then, the 2D string of the target image is created and compared to the query to obtain an overall matching score. Using spatial and color similarity is shown to perform both than a simple color histogram approach. However, representing a complex image with the 2D-string can lead to ambiguities, thus it should be used in conjunction with other attributes.

Representation of content can also take advantage of a dictionary of visual keywords. In [77], these keywords are extracted from statistical learning techniques and depict typical entities of the visual content domain. The image is then spatially described by referencing the visual terms assigned to the regions. The polysemy of the visual words can be further reduced by applying latent semantic indexing. For this purpose the vector is encoded via singular value decomposition before similarity matching.

Finally, structural properties of content can be represented directly as a graph of regions [84]. In [28], an object representation with Directed Acyclic Graphs (DAGs) is adopted. These objects are roughly extracted on I-frames considering the motion vectors of the macroblock in the compressed domain. As the structure of similar objects can differ at different instants of the video (size, shape, viewpoint), inexact matching is considered to build similarity between two DAGs. Each vertex is depicted by a set of metrics that depicts the graph structure (degree of nodes, subgraph size, Strahler number measuring the complexity of a tree). To find sub-matches in the DAGs, the nodes are classified so that nodes in the two DAGs that have similar structure receive the same labels. Matching is completed by comparing both topology of the DAG trees and extrinsic features of the nodes (mean color). This structural object-oriented approach is shown to perform well for retrieval of small objects (protagonist) in different scenes of a video or in a collection of shots in general, and the use of structure tends to improve the result of a region-based CBIR system. However, the limitations of the object extraction method prevents its use for generic videos.

Integration of spatiotemporal cues

Extension of image retrieval systems to video has been firstly achieved using keyframe representation. These methods that rely on the extraction of spatial features [124, 32] only give limited description of the sequence since the temporal context is not considered. Adding motion information enables to inform on the dynamic content of the scene and the activity of its objects.

Motion-based description

Most video retrieval systems are either based on the mentioned image-based approach or provide good support for describing temporal structure of the video. Among the latter category, camera motion has been used to bring semantic information such as panning, tilting, zooming and so on. Motion indexing can be based also on an estimation of the optical flow. A shot activity histogram is presented in [139], where the level of activity is estimated in each frame from the optical flow. Ardizzone et al. [9] propose to index directly the optical flow computing attributes from the optical flow distribution in four equally-sized regions. Other representations of the optical flow are based on the Fourier Transform [19] or on the Wavelet Transform [20]. Furthermore rough motion vectors are already provided in MPEG-1 and MPEG-2 encoding, so that these features can be processed very efficiently. Thus, the motion vectors from the macroblocks can be directly used for indexing [64]. Otherwise simple methods can be used [127], but the accuracy cannot be guaranteed in complex situations.

To avoid the computation of dense optical flow fields which remains a difficult task in general, Fablet et al. [41] proposes an original approach. They exploit spatiotemporal image derivatives to build local motion measurements. Statistical modeling of motion activity allows to a define similarity measure between two shots. The similarity is further used to index shots hierarchically into a binary tree. The framework has the advantage to handle both classification of motion activity and query by example using the similarity measure between shots.

Trajectory-based description

Description of the dynamic content of the scene has been also based on trajectories. A system for indexation and retrieval of object trajectories is VideoQ [24]. The indexing is performed as follows. Objects are extracted and tracked with a motion segmentation technique, and the trajectory is computed from the object centroids. Motion features such as velocity acceleration, arc length are derived and indexed. Furthermore complex object trajectories are segmented into sub-trajectories to recover partial motions. In the retrieval phase the object trajectories are matched either by projection of the x-y plane for time invariance or directly by computing the euclidean distance from the spatiotemporal domain. Motion trails can be used in conjunction with spatial image features for retrieval with query by sketch. The VideoQ system has been proved successful in retrieving video clips in the sports domains, such as soccer players, high jumpers, and skiers, where the object trajectory can be used directly for indexing.

Other trajectory-based approaches were designed without the use of an explicit object segmentation. One of the first approach extracts trajectories from forward and backward motion vectors of macroblocks in the MPEG stream [34]. The extracted trajectories are then grouped to extract object motion, of which final trajectory is obtained by averaging the trajectories of the macroblocks. Object trajectories can be also obtained considering a sparse representation of the sequence. Different features have been considered. In [116], viewpoint invariant regions are detected and represented with SIFT descriptor to be matched through

the frames of the sequence. Similarly, in [14], trajectories are built from SIFT interest points. They further group motion coherent trajectories by a clustering technique that takes into account both motion similarity and spatial proximity of the trajectories.

Comparison of trajectories faces from the problem of spatial and temporal invariance. To solve this issue, efficient models and robust search techniques are proposed in [29].

Spatiotemporal modeling

Methods that segment the video shot in the spatiotemporal domain (as a single block) have recently raised interest for indexing and retrieval. An advantage is that compact representations used in the segmentation phase can directly serve for indexing. A representation with colors flows is adopted in [30], in particular for indexing TV advertisements in which color dynamics are important. A related approach is presented in [31], in the context of retrieval of near duplicates and action recognition for surveillance systems. In this framework, linearly moving color patches are indexed in the database as a point in a 7D feature space. Then, the retrieval of a video clip consists in finding the k-nearest neighbors for each strand of the query and voting for the clip which maximizes the count.

In [55], color features and spatiotemporal locations within a shot are represented with a GMM. Each component of GMM forms a blob corresponding to an elementary object of the sequence. Velocity and direction of the blobs can be derived from their covariance matrix and use in addition to color feature for efficient indexing and retrieval. This representation has been shown useful in context of medical image and video, in particular for detecting lesions in a sequence of radiographic images.

In the approach of [14], an hybrid representation of the volumes is used. First trajectories of interest points are used to generate spatiotemporal volumes. The volume supports are defined as the bounding boxes that encompass the trajectory groups. Each volume is represented with sparse features that include motion direction and the SIFT descriptors from the points within the volume. Complementary dense features such as color and texture are also extracted from the volumes. To reduce the dimensionality for indexing, each feature is represented with a small number of clusters. Similarity measure between volumes is then derived from the EMD distance. This measure is employed for the retrieval of the video shots. For the comparison of two volume sets, a minimum matching technique is adopted (Hungarian algorithm). The approach for motion indexing of scenes with rigid moving objects such as cars, planes, where the interest points of the object can be associated to the object correctly. However, volume extraction from trajectories remains problematic in case of deformable objects.

Indexing with graphs

Structural organization of objects within the scene has high potential for modeling semantically rich and complicated multimedia data. Temporal and spatial relationships between objects have rarely been considered simultaneously until recently. Typically the sequence is represented as a series of Region Adjacency Graph (RAG) that describes the regions of each frame. Video object is then defined on each frame as a subgraph of a RAG [146, 74].

Lee et al. [74] uses a spatiotemporal indexing structure where the shot is decomposed into object region graphs (ORG) and background graphs (BG). Temporally connected ORGs are merged into object graphs (OG) that represents the object in the whole shot. Similarly, overlapping BG nodes of each frame are merged since they are mostly redundant in the segment. The indexing of object graphs is performed in two steps. A graph edit distance is introduced to define the similarity between objects and is used to perform the clustering of the OGs in the database. Secondly, the edit distance is extended to a metric for the indexing of OGs. To manage efficiently these different structures, the index tree is decomposed in three levels. The first level contains the BGs associated to a segment. The second level stores the centroid objects obtains from the clustering and the leaf nodes are OGs which are indexed with the distance to the cluster they belong to. Using the index structure, a query object is processed by finding most similar cluster node and return the k-nearest neighbors OGs within this cluster. The query can be also formulated at the shot level; the search is then limited to the segment with most similar BG.

Summary

First, we have identified most representative image features for indexing and their use in different contexts. Considering spatial information from region-based representations offer several advantages for indexing and retrieval tasks. Locally, a decomposition into regions allows to describe accurately parts of objects composing the scene. Thereby one can search a particular object or region in an image. At a global level, the information gathered from all regions or from their spatial organization can be used for interpreting visual content or search for similar image content. In this way, such systems come close to the visual system, which has the ability to analyze content at a variety of levels. Besides spatial properties, dynamic content of the video is generally represented with the level of motion activity and the motion models estimated in the video. For its part, the use of trajectories appears useful for recognition of action features and event detection in a controlled context, but is in not always discriminative for visual search. The extension of region-based systems to video sequences also sets new challenges. In general the temporal evolution of the video is lost and there is no consideration to link information from different keyframes. Secondly the representativeness of the regions is sensitive to the quality of image segmentation. Spatial variations of the segmented regions affect the ability of the content model to represent and compare accurately related visual content. More solid evidence for the region structure and motion can be obtained by considering information from the whole video. Spatiotemporal primitives that decomposes the sequence into visually coherent elements have revealed promising results for event detection and recognition, or object manipulation. These primitives remain limited to a short term duration and are associated essentially to homogeneous volumes or rigid moving objects. Considering the graph structure of objects allows for fine indexing of complex objects in the sequence, but indexing and matching procedures are more complex in return.

4.2 Video shot analysis with spatiotemporal volumes

In this section, we examine how spatiotemporal representation can improve video shot description for indexing and retrieval. We first introduce the Vector Space Model (VSM) used in information retrieval and its adaptation for spatiotemporal representation of shots. Then, we analyze experimentally the indexing properties of the model, and finally compare the keyframe and spatiotemporal representations for video shot retrieval task.

4.2.1 Representing video shots with the Vector Space Model

A complete low-level description of a shot is given by the graph of spatiotemporal regions and their descriptors. These descriptions represent huge amount of data, which is difficult to index and compare efficiently. The Vector Space model [109] enables to gather and compact this information to a single feature vector. This model was first introduced for text retrieval systems. Statistical distribution of the keywords appearing in a text document is modeled with a simple vector. Each term of the vector corresponds to a keyword and the term value is related to the frequency of the keyword in the document. A keyword can be a single word, or more complex phrases defined in a dictionary. The reason behind this representation is that the distribution of visual terms gives information on the latent semantics of the shot. Although simple, the model shows great efficiency for keyword-based search engines.

Formally, we consider a collection of documents D and the set of terms $C = \{c_1, \ldots, c_{|C|}\}$ representing the keywords. A document model $d_i \in D$ is represented as a vector of terms :

$$d_i = \{w_{d_i,c_1}, \ldots, w_{d_i,c_{|C|}}\} \tag{4.1}$$

In the simplest scheme, the term weights w_{i,c_j} represent the number of term occurrences in the document. Seldom terms can be given higher weights by considering in addition the frequency of the term in the corpus, which is known as the tf-idf measure [109]. A graphical representation of the VSM is given in fig.4.2. To compare two document models d_i and d_j, an intuitive measure of similarity is given by the angle α between the two vectors, capturing the relative proportion of common terms in the two models. Cosine measure is used for this purpose:

$$sim(d_i, d_j) = \frac{d_i \cdot d_j}{\|d_i\| \, \|d_j\|} \tag{4.2}$$

By analogy, visual content can be described with visual terms relative to the regions features. Ideally, the dictionary should highlight visual entities of a particular content domain [77]. In the general case visual terms can be obtained from unsupervised clustering of the descriptors in the database [121].

A CBVIR system for video shot indexing and retrieval with the VSM model is described fig.4.3. The system directly inherits of the generic CBIR architecture of fig.4.1. First, the video database is organized as a collection of visual shots obtained by temporal segmentation algorithms. These shots are further manually annotated. The next task is to build a visual dictionary that is composed of representative elements of the shots (the keywords). Shots are segmented into homogeneous volumes with the method we described in chapter 2. At

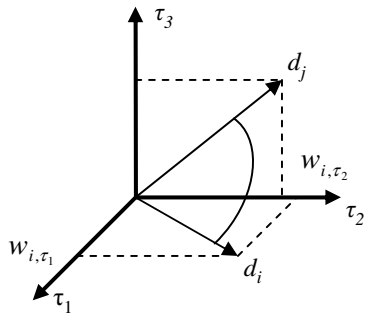


Figure 4.2: Illustration of the Vector Space Model.

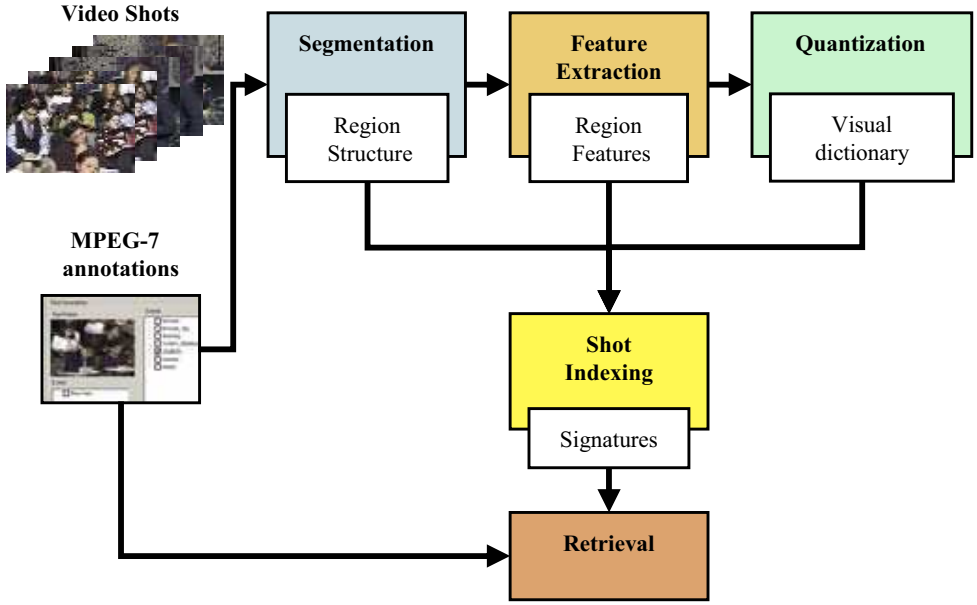


Figure 4.3: Video indexing and retrieval system using the Vector Space Model.

the same time, we store the low-level descriptors of the extracted regions. The overall set of descriptors is then clustered to obtain the visual keywords. Within this framework, indexing of a new shot is performed in two steps. The shot is segmented into regions and region descriptors are extracted. Then, these descriptors are quantized to the nearest visual terms to obtain compact shot signatures from the vector counts. Search and retrieval can be performed as shot signatures can be compared very efficiently.

Visual Dictionary Construction

Representation of visual keywords is based on the visual attributes of the spatiotemporal volumes. Various attributes can be considered for this purpose, for instance among those proposed in section 1.1.2. In a general context, it is preferable to separate different information sources and construct a dictionary for each type of attributes. Fusion of information is then achieved at the search stage, being potentially guided by the context or the feedback of a end user.

Region-based descriptors can be extracted from spatiotemporal volumes in two ways:

- Compute a single descriptor for each spatiotemporal region. This can be seen as averaging the descriptors through time (S-extraction).
- Compute a descriptor for each projected frame region, which depicts the evolution and variations of the attributes but more storage is required. (F-extraction).

Once the descriptors have been extracted for the whole database, the visual keywords are constructed through an automatic clustering algorithm. An efficient and simple method is the K-mean algorithm. The visual keywords corresponds to a predefined number of cluster centers. An advantage of the strategy is to consider an important number of regions, as long as the number of clusters is kept reasonable (typically up to 2000 clusters).

Clustering gives a further reason to split modalities into different dictionaries, as the algorithm can be more easily trapped in local minimum with high dimensional data. In the case that the feature space is non-metric, algorithms such as K-medoids can be used. A shortcoming of this approach is that all pairs of points have to be compared, increasing notably computing time.

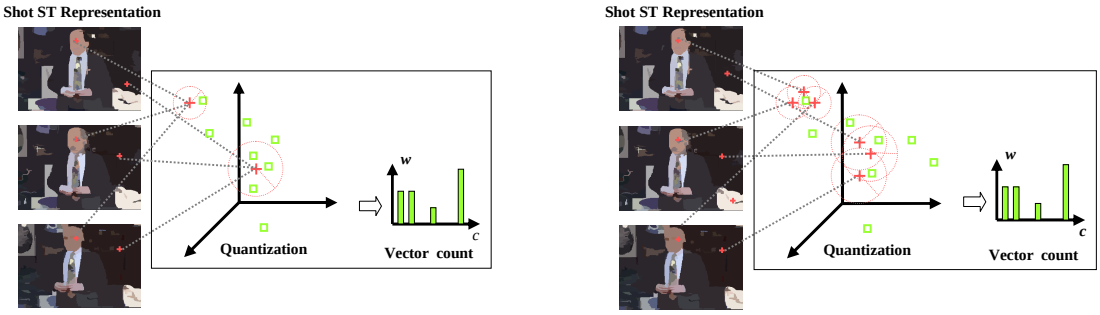
Shot Indexing and Retrieval

Based on the visual dictionary, indexing of a video shot can be processed very efficiently. The principle of the indexing model is to alleviate the problem of the variations of the visual content by considering its statistical structure. First spatiotemporal regions are extracted for each shot using the spatiotemporal segmentation algorithm. Then, volumes are assigned to visual keywords for each dictionary.

Fig.4.4 illustrates the indexing procedure for the two extraction methods. In case of S-extraction, each volume descriptor is quantized to its nearest terms in the dictionary. For the second case of F-extraction, quantization is performed for each frame region descriptor and the result is accumulated in the count vector. The effect is similar to a voting process for the volume representative keywords, more evidence on the clusters is found as the number of votes (frames) increases.

For a given descriptor, the assigned number of neighbors depends on the quantization error. If the descriptor is very close to its nearest term, only the nearest term is counted. Otherwise, when the descriptor is approximately distant of several terms, all these terms should be counted. Let f_r the a region descriptor of one shot, and $(f_i^C)_{i=i_1 \ldots i_K}$ the ordered K nearest visual terms of f_r in the dictionary. The set of counted terms $nearest(f_r)$ is :

$$nearest(f_r) = \left\{ i_k | d(f_r, f_{i_k}^C) < \alpha d(f_r, f_{i_1}^C) \right\} \tag{4.3}$$



(a) (b)

Figure 4.4: Vector Space Model for shot indexing. (a) S-extraction with a single descriptor. (b) F-extraction with multiple descriptors.

where d is the distance measure for comparing two descriptors. This soft quantization enables to deal with spatial and temporal variations of the visual content. Firstly, the closest visual term can change between different instants. Secondly, spatial parts of one region may be better described with different terms. When mapping to multiple terms, the maximum number of neighbors K should be kept small with respect to the dictionary size, such that the count vectors still remain discriminative. In our observations, up to 10 clusters can be chosen for less than 2000 visual terms.

For the search phase, the Cosine similarity (eq.4.2) between the count vectors is employed. To obtain a unique similarity measure, similarities from different modalities are fused with a weighted sum. For simplicity, each feature is given equal weight.

4.2.2 Evaluation settings

Experimental data

The studies in this chapter are conducted on two videos. The docon's cartoon of the mpeg-7 dataset and the lecture video "Senses" from the open video project[1]. Each video has its own challenges for segmentation and indexing. The presence of homogeneous colors in the cartoon video enables accurate segmentation, but different objects can share the same environment or interact with each other (turtle, dolphin, shark), which complicates the indexing task. In the second video, we can find scenes with static content (students, screen, drawing). The lecturer is moving within the scene and is seen from close-up or wide-angle shots (lect1 and lect2). An illustration of typical annotated shots is shown fig.4.5. To obtain enough occurrences of each element and variable visual content, each video is subsampled into nearly 1000 shots. Table 4.1 show the number of occurrences for each type of visual category.

[1] http://www.open-video.org/

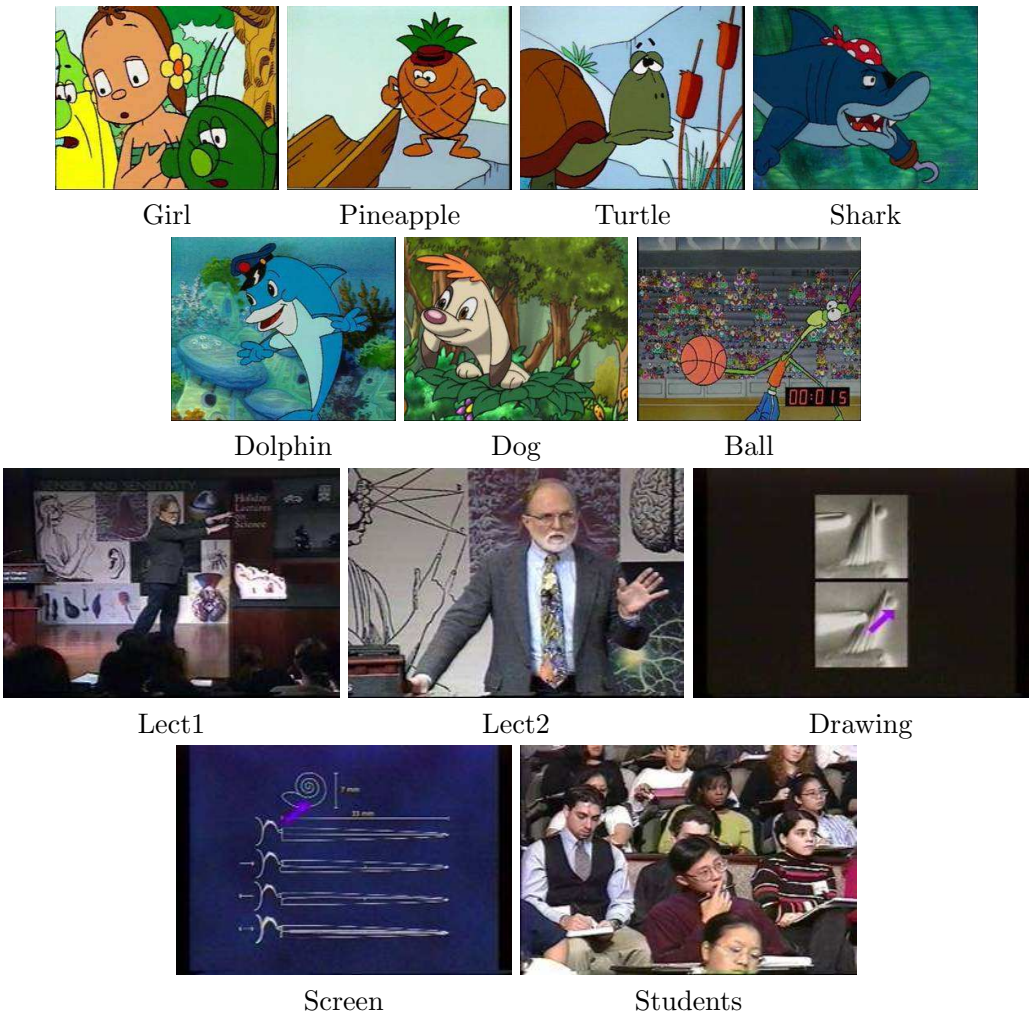


Figure 4.5: Examples of annotated shots. (a) Docon cartoon. (b) Senses video.

Docon	
Girl	70
Pineapple	67
Turtle	116
Shark	87
Dolphin	21
Dog	80
Ball	74

Senses	
Lect1	136
Lect2	346
drawing	81
Screen-displays	200
Students	69

Table 4.1: Visual categories. a) Video Docon. b) Video Senses.

Evaluation measures

Many different measures for evaluating the performance of information retrieval systems have been proposed. The system is evaluated from processing a query within a collection of documents. A retrieved document is considered as correct (or relevant) if it responds to the query. Let n the number of documents of the collection, n_r the number or document relevant to a query and n_c the number of correctly retrieved documents. Basic performance measure is given by precision (P) and recall (R) :

$$P = \frac{n_c}{n} \qquad R = \frac{n_c}{n_r} \tag{4.4}$$

A usual way to represent the performance graphically is given by precision-recall curves. This is achieved by computing precision and recall for different numbers of document retrieved. It is also useful to have a unique measure for comparing performances of different systems. Average Precision considers the precision after each change in recall on average:

$$AvP = \frac{\sum_{r=1}^{n} p(r)\Delta(r)}{n_r} \tag{4.5}$$

where $\Delta(r)$ is a binary function which takes as values 1 if a new relevant document is retrieved at rank r. The measure emphasizes the fact that relevant documents should be returned earlier. To handle the variability of the queries in each category, each performance measure is averaged on all queries of a category.

4.2.3 Analysis of the Vector Space Model

For a deep understanding of the VSM and achieve a pertinent evaluation of the model, it is important to know the model properties and the different factors that impacts on it. Numerous factors deserved to be studied in this model. Here, in this section we first focus on the influence of the segmentation properties (granularity, homogeneity) among with the number of visual keywords in the dictionary. Then, we analyze the behavior of the model and its particularities with respect to the proposed different visual categories.

Measurements

Factors impacting the construction of the visual dictionary.

Segmentation and clustering quality are key elements of the Vector Space Model. Extracted regions should be representative of the shot, while the signatures should be characteristic of the visual categories. For this reason, we study the influence of these elements on the VSM performance. More precisely, we consider two factors:

- The granularity and the homogeneity of the regions.
- The visual dictionary representativeness.

The former factor rules the segmentation quality. Segmentation algorithms usually aim to obtain homogeneous regions while limiting their number, but each method has its own way to fuse different sources of information to obtain perceptually coherent regions. Then, it is generally difficult to specify the degree of homogeneity of the regions and the granularity of the segmentation at the same time. To have more control on this factor, we choose to fix the segmentation layout utilizing a grid. In this way, we indirectly set the region homogeneity with respect to each feature, as refining the grid will lead to more homogeneous content on the average. In total, 7 scales, including 4 to 512 regions are considered.

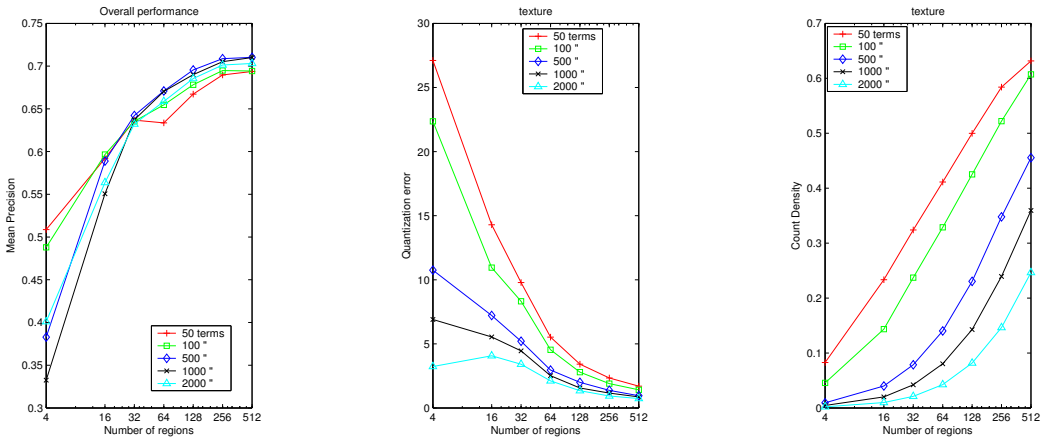
Besides segmentation properties, visual dictionary can be depicted by the capacity of visual terms to fit each visual category. To examine the underlying dictionary properties, we propose different measurements:

- The distance of one region to its nearest clusters, which we denote as *quantization error*. When indexing shots, the error depends on the distribution of the visual terms and of the number of clusters.

- The proportion of occupied bins in the count vector, which we call as *count density*, which summarizes the distribution of shot regions over visual terms.

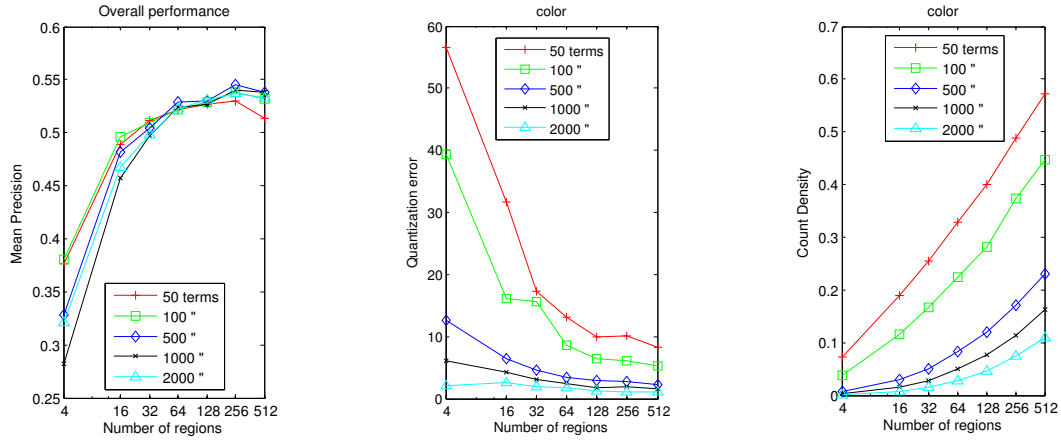
These measures are computed on the whole video dataset and averaged on each semantic category.

Fig.4.6-4.7 illustrate the overall analysis results considering different dictionary sizes. Fig.4.6(b)-4.7(b) show that increasing the number of regions and the number of terms diminish the quantization error. First, this is inherent to the clustering process : more terms results in more compact clusters. Secondly, decreasing the grid scale leads to more homogeneous regions and therefore accurate descriptors.



(a) (b) (c)

Figure 4.6: Analysis of the VSM for the texture modality - Example of the video senses. (a) Overall retrieval performance. (b) Overall quantization error. (c) Overall count density.



(a) (b) (c)

Figure 4.7: Analysis of the VSM for the color modality - Example of the video Docon. (a) Overall retrieval performance. (b) Overall quantization error. (c) Overall count density.

Besides the quantization error, the analysis of count vector statistics gives an indication about the usefulness of the information conveyed by the signatures. When each region is randomly indexed to a term in the dictionary, count density value can be represented as a stochastic process. Let X_n a random variable that represents the number of occupied bins after adding n counts. The filling procedure is modeled with a Markov Chain $\{X_1, \cdots, X_n\}$ defined as follows. The set of possible states is $S = 0, \ldots, |C|$, where $|C|$ is the dictionary size. The Markov chain is initialized by its initial value $X_0 = 0$ and its transition matrix P is given by :

$$P_{ij} = Pr(X_{n+1} = j | X_n = i) = \begin{cases} \frac{i}{|C|} & \text{if } j = i \\ 1 - \frac{i}{|C|} & \text{if } j = i + 1 \\ 0 & \text{else} \end{cases} \tag{4.6}$$

The probability $Pr(X_n = j)$ of populating j counts in n steps is then :

$$Pr(X_n = j) = \sum_{i=0}^{|C|} P_{ij}^{(n)} Pr(X_0 = i) = P_{0j}^{(n)} \tag{4.7}$$

To evaluate the relevance of the quantification process using a visual dictionary, we compare the experimental density counts (fig.4.6(c)-4.7(c)) with those expected for the random model (fig.4.8) as a reference. For the dictionary of fig.4.6(c), the count density is at least 25 percent lower than for the random model up to 500 terms. On the contrary, when the dictionary size is more than 1000 clusters, the curves are very close to each other. This means that for small dictionary sizes, we obtain numerous spatial cooccurrences of the same visual terms, so that the number of these cooccurrences become important when comparing vector counts. For a large number of terms, regions are typically assigned to different clusters. In consequence, the added counts are unrelated and the density is close to the one of

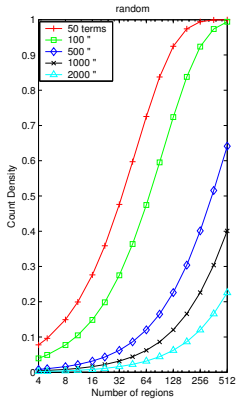


Figure 4.8: Exact count density for the random model.

the Markov process. For the docon video fig.4.7(c), the population of filled counts increases more slowly with the number of regions than for the random model for large dictionary sizes (more than 500 terms). In fig.4.7(b), we can notice that the quantification error remains stable between 32 and 256 regions, which shows that the color region descriptors does not vary significantly when refining the grid. As a consequence, descriptors in a local area are prone to be quantified to the same terms.

The impact of the granularity and the quantification process on the retrieval performance is shown fig.4.6(a)-4.7(a). When considering few and inhomogeneous regions, the quantization process is quite unstable leading to low and variable precision rates. Reducing the quantization error by augmenting the number of clusters does not help in this case as the count vector becomes very sparse, altering the comparison between shots. When more regions are available, the performance is less sensitive to the dictionary size. If the dictionary is small, the high number of cooccurrences gives dense but distinctive signatures. Otherwise, the signatures remain discriminative in spite of the reduced cooccurrences as we use accurate visual terms.

We can draw several lessons from this general analysis. Firstly, the clustering and indexing tasks appear more robust when considering numerous and relevant regions, boosting the accuracy of the visual terms in the quantization stage, and leading to more discriminative shot signatures thanks to the term redundancies. Secondly, the dictionary size can be chosen to have intermediate count density values, balancing spatial redundancies and visual term accuracy.

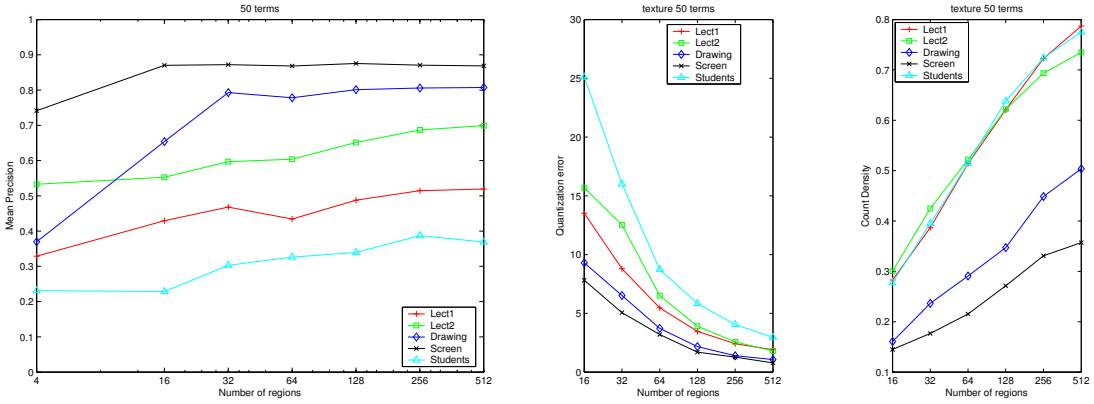
Behavior of the model on different categories

After studying of the general properties of the VSM, we now examine its behavior with respect to different visual categories. Figures 4.9 and 4.10 show that the scenes with stable and specific visual content, such as screens, drawings are easily integrated with the VSM. Indeed, these categories have both small error rates (fig.4.9(b)-4.10(b)) and sparse vector counts (fig.4.9(c)-4.10(c)). Good retrieval results are observed for both small and large

dictionary sizes, which reveals that the shots are indexed efficiently with a few category-specific terms in the dictionary. In some categories, such as students, the regions are rather inhomogeneous. As shown by the high quantization error and density of their vector counts, they are finally improperly represented in the visual dictionary. As noticed in fig.4.6, augmenting the dictionary size does not lead to significant enhancement while requiring more regions.

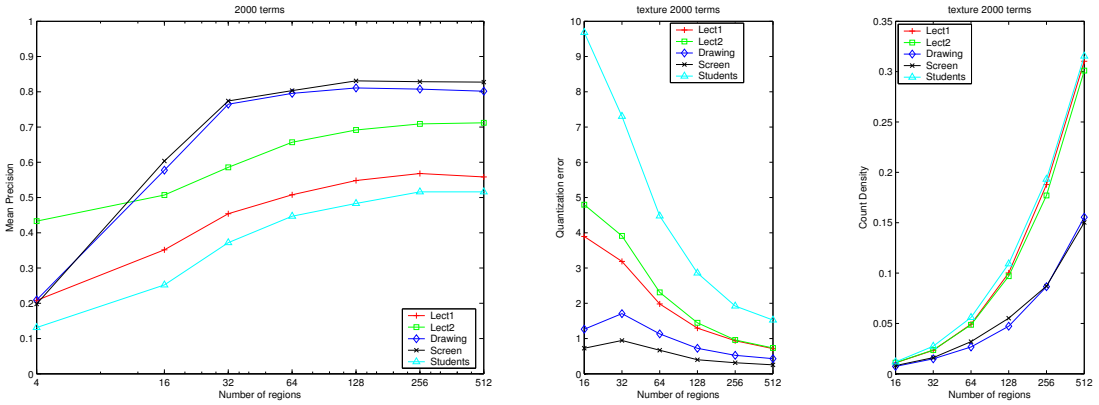
These observations are also verified in fig.4.11. The shark object is clearly described by a few representative terms (fig.4.11(c)), thus obtaining good retrieval results. In the opposite, ball and dog categories have poorer representation in the dictionary with the highest quantization errors (fig.4.11(b)), resulting in weak performance, around 0.3 and 0.4 respectively. For the other categories, the relation between the measures and the performances is not as manifest. What should be taken under consideration is the interaction between categories and the overlap of visual terms. Indeed, some categories can share the same visual environment such as the couples girl-pineapple or shark-turtle. The first one has large background areas in common, which is manifested through low and similar quantization error. It is also the case in the second couple, but the turtle can appear under different views and also other environments. In consequence, a query featuring a turtle may retrieve shots featuring the shark before some other shots where the turtle appears.

These considerations explain how the VSM enables to reduce the video content to a small amount of visual terms. The construction and comparison of these terms depend mostly of the segmentation properties. If the regions are described accurately, compact and discriminative representation can be obtained from a large range of dictionary sizes. Otherwise, the model is penalized when considering variable categories.



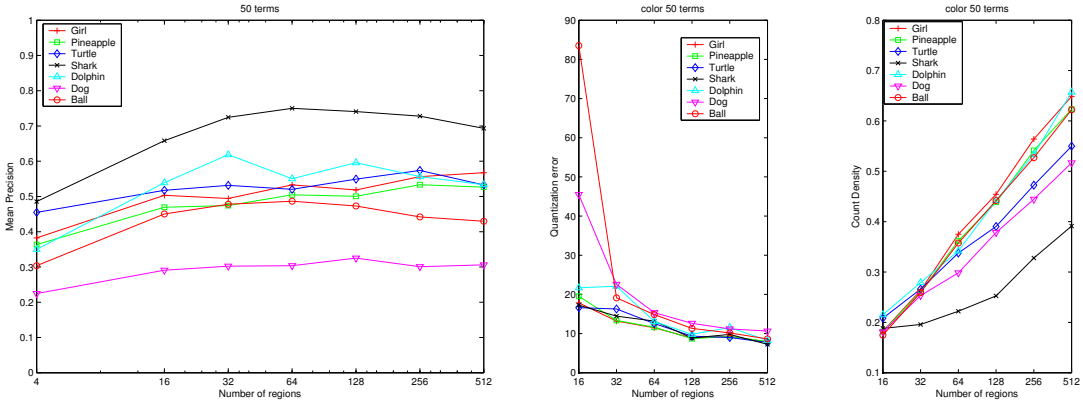
(a) Retrieval Performance (b) Quantization error (c) Count density

Figure 4.9: Analysis of the VSM for different categories with 50 visual terms - Example of the senses video.



(a) Retrieval Performance. (b) Quantization Error. (c) Count density.

Figure 4.10: Analysis of the VSM for different categories with 2000 visual terms - Example of the senses video.



(a) Retrieval Performance (b) Quantization Error (c) Count density

Figure 4.11: Analysis of the VSM for different categories with 50 visual terms - Example of the Docon video.

4.2.4 Comparison between keyframe and spatiotemporal representation

In this section, we evaluate the contribution of the spatiotemporal approach to the VSM. For this purpose, we compare the spatiotemporal representation with keyframe regions obtained from well-known segmentation algorithms.

The first algorithm is watersheding [138], which is based on merging of homogeneous color regions. For its part, the edgeflow algorithm, used in the Netra-V system [32], identifies salient boundaries in the image from the orientation of color and texture flows, followed by merging of regions based on color and texture.

We evaluate the algorithms on different number of frames, except for the edgeflow technique which is too computationally intensive to be used on multiple frames. Table 4.2 recapitulates the list of the algorithms used.

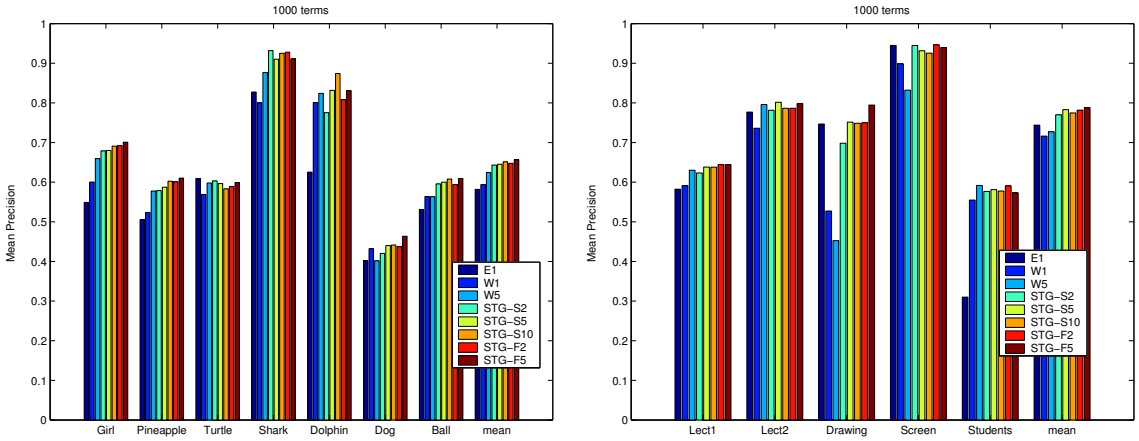
Method	Description
STG-F*	Spatiotemporal segmentation, F-extraction
STG-S*	Spatiotemporal segmentation, S-extraction
W*	Watersheds, regions cumulated over * frames
E*	Edgeflow, regions cumulated over * frames

Table 4.2: Segmentation algorithms.

Results for the two videos are shown fig.4.12(a-b) respectively, with a dictionary size containing 1000 visual terms. The spatiotemporal method performs globally well compared to the edgeflow and the watershed algorithms. Regarding image segmentation methods, good results are obtained with the watershed method for homogeneous color objects such as shark, turtle and static rich colored scenes. However, the results can decrease dramatically for scenes with variable spatial contrast and textured areas (screen, drawing). Not surprisingly, the edgeflow method performs better on this type of scenes. Thus, watershed outperforms the edgeflow method for the Docon video and edgeflow performs better for the senses video.

Globally, we observe gains of 7% (fig.4.12(a)) and 12% (fig.4.12(b)) in the mean precision between the best image segmentation and the best spatiotemporal method (STG-F5). The difference in the retrieval performance can be explained by the fact that, in the senses video, the edgeflow method can give reasonably good results for categories which contain well-defined textured elements, such as the lecturer, drawing and screen. However, it fails to describe accurately scenes with more spatial variations, such as students.

Considering multiple frames (W5) or spatiotemporal volumes contributes to the quality of the shot indexes. In the first case, the density of the shot indexes is increased, so that more common terms can be found. In comparison, in the spatiotemporal representation (STG-S), spatial and temporal variations are attenuated by extracting descriptors on the full volumes. The advantages of these two approaches can be combined by extracting frame descriptors from the spatiotemporal regions (STG-F). In this way, we capture the temporal evolution of the region descriptors. We observe that the extraction process slightly enhances the retrieval performance, up to 6% with respect to STG-S methods with similar number of



(a) Docon video (b) Senses video

Figure 4.12: Retrieval results for different segmentations. The dictionary contains 1000 visual terms.

frames. More precisely, this improvement concerns categories whose visual content undergo important variations between scenes, such as the ball, the dog and girl scenes. Actually, the effect is to accumulate more confidence on the quantization process, as the visual terms are selected using several descriptors from the same volume. This helps to distinguish common terms that remain stable to scene changes from the others.

Another advantage of the spatiotemporal representation is that good results can be achieved considering far less regions than the other methods, as shown in table 4.3. This is noticeable and particularly interesting for processing large databases.

Method	E1	W1	W-5	STG-S2	STG-S5	STG-S10	STG-F2	STG-F5
Nb. reg.	90	135	672	52	88	140	90	254

Table 4.3: Average number of regions for the segmentation algorithms.

Finally, we can notice in fig.4.12 that the performance of spatiotemporal methods does not depend much on the size of the visual dictionary. The results are averaged on all objects for several dictionary sizes. They need between 100 and 1000 terms to reach their best performance, whereas a smaller dictionary can be used for watershed and edgeflow segmentation. This reveals that slightly more visual terms are needed to represent each spatiotemporal region. As mentioned in section 4.2.1 this is not a disadvantage as it adds robustness to the scene variations.

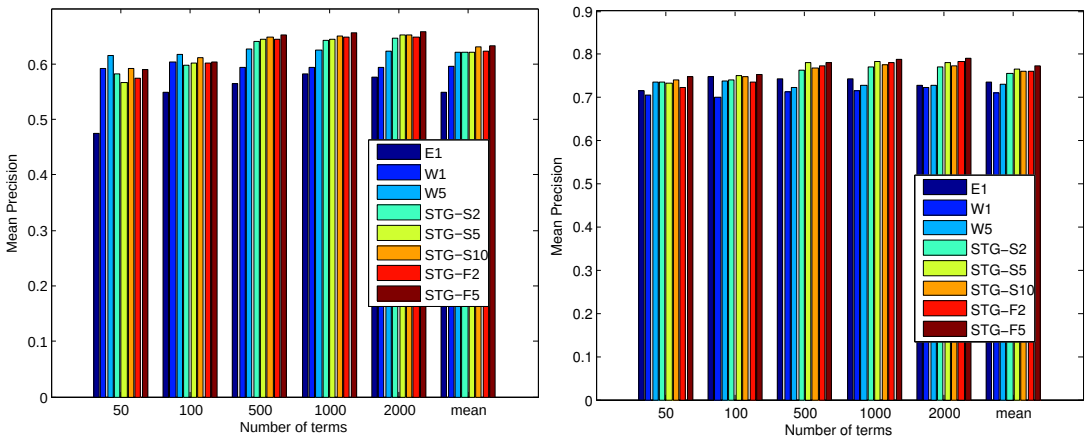


Figure 4.13: Retrieval results for several dictionary sizes. (a) Docon video. (b) Senses video.

4.3 Video shot matching

The VSM indexing model enables to take into account statistical structure of the spatiotemporal representation by reducing the variability of the region description with visual keywords. However, this approach requires quantification from a visual dictionary, and particular types of content can be insufficiently represented with the visual keywords. Furthermore, spatiotemporal organization of the volumes is not considered in the signatures.

Comparing directly the ARG representations enables to take into account both visual and structural similarities. In this section, we propose to construct a similarity between graphs. A matching technique is developed to recognize visually close object structures while tolerating a certain variability in the description.

4.3.1 Visual and structural similarities

When comparing object structures in different shots, an important point is that a volume in one shot can appear as split into several volumes in another shot due to scene changes, occlusion and several other factors. In order to address this problem and limit the complexity of the matching, we develop a many-to-one technique. Each volume in the two graphs is constrained to have at most one match, but it is tolerated that two volumes have the same match. An example is given in fig.4.14(a). Let $G_1(V_1, E_1, \nu_1, \xi_1)$ and $G_2(V_2, E_2, \nu_2, \xi_2)$ two ARGs. The matching between V_1 and V_2 is represented as a directed bipartite graph $L(V, E, W)$. The arcs (v_1, v_2) and (v_2, v_1) in E represents respectively a match from V_1 to V_2 and V_2 to V_1. We introduce also the weight of a match from V_1 to V_2, w_{v_1,v_2}. For two vertex sets $Q_1 \subset V_1$ and $Q_2 \subset V_2$, we extend the notation w_{Q_1,Q_2} as the sum of the weights from Q_1 to Q_2. Figure 4.14(b) gives an example of the matching graph.

The constraints defined for the matching impose the following restrictions on the node

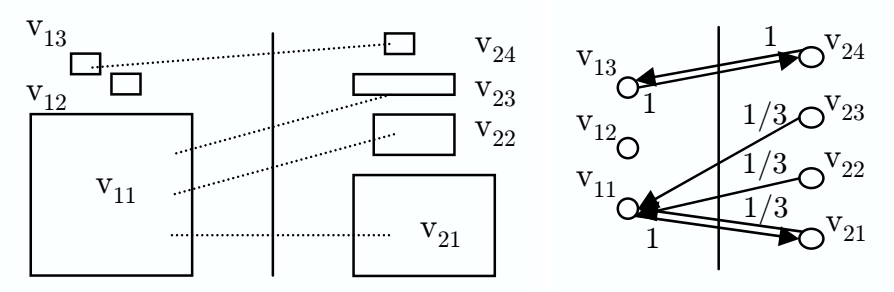


Figure 4.14: (a) Example of matching between video volumes. (b) The corresponding bipartite matching graph.

degrees. For the indegree of a node v_i, we have $deg^+(v_i) \in [0, |L/V_i|]$ and for its outdegree $deg^-(v_i) \in [0, 1]$. We set the weights of the matches incoming to a node in order that they are distributed uniformly. This is defined as follows:

$$w_{v_1,v_2} = \begin{cases} 0 & \text{if } deg^+(v_2) = 0 \\ \frac{1}{deg^+(v_2)} & \text{else} \end{cases} \tag{4.8}$$

The weight w_{v_2,v_1} is defined symmetrically. The idea is that when one node has been matched, we do not consider furthermore its properties, but only its relationships in these graphs. To establish the matches we will rely on the visual attributes of the ARG, but the weights themselves do not depend of the visual similarity.

The basic similarity between two nodes takes into account both the structural and the visual properties. For $(v_1, v_2) \in V_1 \times V_2$, $\xi_N(v_1, v_2)$ is defined as:

$$\xi_N(v_1, v_2) = \alpha\xi_D(v_1, v_2) + \beta\xi_S(v_1, v_2) \tag{4.9}$$

ξ_D is the overall similarity between the volume visual descriptors, as defined in eq.3.22. The structural similarity ξ_S between two nodes is defined from the matching graph L. For $(v_1, v_2) \in G_1 \times G_2$, the similarity is based on their neighborhoods $N_1(v_1)$ and $N_2(v_2)$ in G_1 and G_2. The principle is inspired by the normalized cuts [113] between two subgraphs. The measure computes the ratio of total edge connection between two subgraphs to their total connection to the whole graph. In the matching graph L, we compare the flow incoming to $N_2(v_2)$ from $N_1(v_1)$ to the total flow incoming to $N_2(v_2)$. This gives the strength of the match from $N_1(v_1)$ to $N_2(v_2)$. Moreover, when v_2 and one of its neighbors $n_2 \in N_2(v_2)$ matches both v_1, v_2 is excluded from $N_2(v_2)$. Indeed in this case v_2 and n_2 are likely to correspond to subparts of v_1, i.e. they should be merged in a single node. The reasoning is the same for the matches from $N_2(v_2)$ to $N_1(v_1)$. We note $M_2(v_1) = \{n_2 \in N_2(v_2) | w_{v_2,v_1} \neq 0\}$ and $M_1(v_2) = \{n_1 \in N_1(v_1) | w_{v_1,v_2} \neq 0\}$ these excluded vertex sets in the neighborhood of v_2 and v_1, respectively. Thus, we consider restricted neighborhood of v_2 to $N_2^*(v_2) =$

$N_2(v_2)/M_2(v_1)$ and $N_1^*(v_1) = N_1(v_1)/M_1(v_2)$. Formally, the similarity is defined as :

$$\xi_S(v_1, v_2) = \frac{1}{2}\left(\frac{w_{N_1(v_1),N_2^*(v_2)}}{w_{V_1,N_2^*(v_2)}} + \frac{w_{N_2(v_2),N_1^*(v_1)}}{w_{V_2,N_1^*(v_1)}}\right) \tag{4.10}$$

If there are no matched nodes in $\mathcal{N}_1(v_1)$ or $\mathcal{N}_2(v_2)$, the similarity is set to zero, as there are no common matches between the neighborhoods.

Finally, the total similarity between complete ARGs is obtained from the set of matched pairs $E_S \subset V_1 \times V_2$:

$$\xi(G_1, G_2) = \frac{1}{|E_S|} \sum_{(v_1,v_2)\in E_S} \xi_N(v_1, v_2) \tag{4.11}$$

4.3.2 Matching algorithm

The selection of the matched pairs is based on the visual similarity. First we compute a similarity matrix between V_1 and V_2 and find the best matches E_S^1 from V_1 to V_2 and E_S^2 from V_2 to V_1. When the best match for a volume v_1 is not reliable, we further compare $N_1(v_1)$ to the neighborhood of the possible candidates. For each node in $N_1(v_1)$, we find the best match in the candidate neighborhood. Then, we compute the average distance on the k-best matches, where k is the minimum cardinality of the neighborhoods. In this way more visual information is considered to select the match.

When all matches have been established, the next step consists in pruning the matches which are the most visually different. First, they are not likely to represent the same element, and secondly computing the structural similarity will be not relevant. One method is to consider the distribution of similarities and choose the number of matches $|E_S|$ from the x-percentile of the distribution. Given a fixed percentile, $|E_S|$ is low when a few volume matches clearly distinguish from the other, and high if all the distribution of matches is uniform. Finally, the algorithm to compute the similarity measure is summarized below.

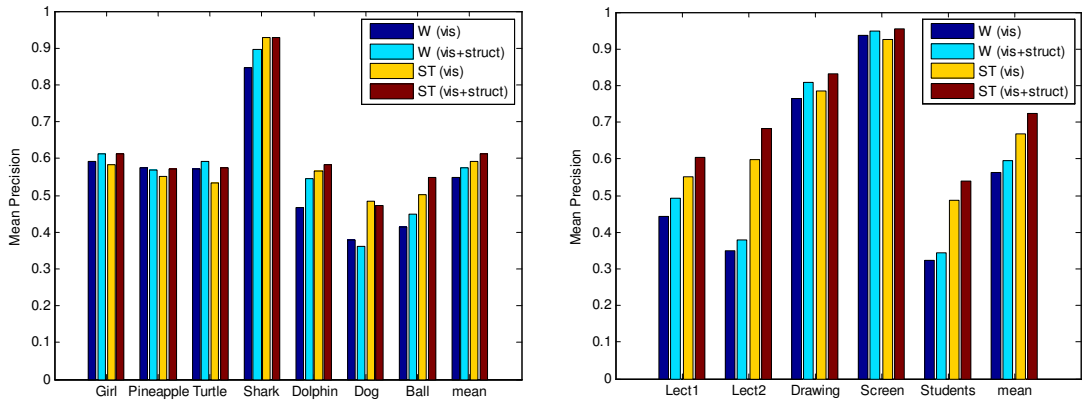
1. Compute all the visual similarities between V_1 and V_2.
2. Find the matches E_S^1 from V_1 to V_2 and E_S^2 from V_2 to V_1.
3. Build the selection E_S from E_S^1 and E_S^2.
4. Build the matching graph L from E_S.
5. Compute the structural similarity for the matches in E_S.
6. Compute the total distance from visual and structural similarities in E_S.

Table 4.4: Building of the similarity measure.

4.3.3 Experiments

To highlight the advantages of the proposed framework, we conducted experiments for the task of video retrieval in the data set presented in section 4.2.2.

We compare different approaches for shot representation and matching. The first approach is based on keyframe segmentation based on the watersheding technique (W) and



(a) (b)

Figure 4.15: Retrieval performance. (a) Docon video. (b) Lecture video.

matching using the visual descriptors only (*vis*). For the second approach, we consider the full similarity measure, including visual and structural parts (*vis+struct*). Third and fourth approaches use the spatiotemporal representation (ST) instead of keyframe regions.

Mean average precision results are reported fig.4.15(a) and fig.4.15(b) using Color Structure and Edge Histogram as visual descriptors. Globally, the performance observed for each category is function of the variability of the layout and of the extracted descriptors. The best categories retrieved are depicted globally with discriminative visual descriptors (screens, shark), whereas the other ones with more variable descriptors and less common elements are more difficult to retrieve (girl, ball, students).

With respect to shot representation, our spatiotemporal approach outperforms the keyframe approach, in particular in the lecture video where keyframe regions can be inaccurate and do not reflect well visual elements in the shot. Using the graph structure leverages the results for the categories with more variable descriptors, but when a common structure still remains between shots. The improvement is noticeable for several categories such as lecturer, girl and ball. This effect is also more remarkable on the spatiotemporal representation, as the neighborhood is enlarged and more reliable matches can be found between shots.

4.4 Conclusion

This chapter was dedicated to the use of spatiotemporal representation for video indexing and retrieval. We have proposed to adapt the Vector Space Model for the description of shots from video regions. The model enables to exploit spatial and temporal redundancies provided by the spatiotemporal volumes for the construction of compact and representative shot signatures.

In the second part of the chapter, we have evaluated the indexing and retrieval system on various types of visual scenes. Experimental results reveal that the performance of the

VSM tightly depends of the region extraction process, the use of accurate region descriptors enhancing the specificity of the shot signatures. The comparison of the retrieval results with different image segmentation methods used for region-based CBIR systems shows that the new representation leverages retrieval performance. The use of spatiotemporal volumes enforces the robustness of the indexing process, obtaining more consistent results between different visual scenes.

Finally, we have introduced a second approach based for the comparison of shots directly based on the spatiotemporal graph representation. Volumes are accurately described by a set of visual descriptors, and structural relations are described through the adjacency graph. With an adapted graph matching technique, the proposed description enables to compute shot similarities in an efficient way that is interesting for indexing applications.

Chapter 5

Conclusions and perspectives

In this thesis we addressed different problems linked to the extraction and exploitation of objects in videos sequences. We adopted a logical plan for the understanding and solving each of these issues, first proposing a global framework for the spatiotemporal modeling of the video shot structure and associated algorithms to build this representation, and then exploit it for applications such as semantic labeling and content-based video indexing and retrieval.

In this chapter, we will provide a short summary of the work achieved in the thesis, and explore possible directions for future research.

5.1 Summary

Manipulation of video objects plays an important role nowadays for organizing and building video databases. Efficient representation among with techniques for object extraction and indexing are to be developed in this field. Among the possible approaches, the expressiveness of graph representation makes it a good candidate to describe the organization and the internal structure of objects in the scene. Its use in the spatiotemporal domain is motivated by perceptual grouping principles, which suggest that the interpretation of a scene is achieved considering spatial and temporal cues simultaneously. Therefore a graph representation of the scene can be obtained from successive spatiotemporal groupings representing different content entities.

The first chapter introduces the graph-based framework developed in the thesis. We examined the description of a video sequence by exploiting the MPEG-7 description tools for visual and semantic analysis. From this study, we deduced a spatiotemporal representation with graphs to describe the content structure into objects, volumes, pixels along with the properties of which they are composed.

In the second chapter, we interested in the issue of spatiotemporal segmentation. After a reminder of the possible approaches, we proposed a new approach based on 2D+T scheme. First we extended an efficient region growing technique based on minimum spanning tree to the spatiotemporal domain. To this aim different grouping algorithms were proposed in function of the level of representation. The benefits of the proposed method are i) to reduce significantly the complexity comparing with the approaches that segment the sequence as

a single block and ii) to achieve good compromise between the span and the consistency of the regions. Then we employed the spatiotemporal representation for the extraction of objects of interest. To compensate lack of salient motion information that may occur in some parts of a sequence, we proposed a scheme based on detection and propagation of moving objects. To that end, motion models were built and used as a cue for grouping volumes with significant motion. Then, a matching procedure enabled to propagate objects in areas where motion information is not consistent. The advantage of this approach is that the detection of objects does not depend on the initial frame of the sequence. Nevertheless the cost of motion estimation for the volumes still remains important.

The third chapter emphasized the work on the spatiotemporal segmentation for the semantic description of video objects. We introduced the exploited knowledge base and the semantic labeling of objects in an image. Thanks to domain knowledge, semantic labeling enables to group elements that belong to the same concept, whereas spatiotemporal segmentation can propagate semantic information within the shot. These two aspects were exploited to propose an efficient strategy for extending the labeling from images to video shots. In this way, the cost of semantic labeling is considerably reduced while providing a spatiotemporal segmentation for the whole sequence. This framework offers an interesting perspective to bridge the gap between the low-level visual description and high-level description of semantic objects, leading to a global interpretation of the shot.

The last chapter examined the application of the spatiotemporal representation for video indexing and retrieval of video shots. We analyzed the benefits of the spatiotemporal approach into a region-based system where each shot is indexed with a compact signature. The adopted content model (VSM) is based on the statistical structure of the description, assigning a set of visual keywords to each volume. Evaluation of results from different types of segmentation underlines that spatiotemporal segmentation can provide more more robust and discriminant signatures which are less dependent of the type of content. Then, we proposed to employed directly the graph representation for indexing and retrieval task. This approach does not require the use of the dictionary which construction can be problematic for large database size and enables to describe the spatiotemporal organization of the shot, a feature which was not used in the previous model. We proposed a similarity measure between graphs that tolerates to a certain extent variability in the spatiotemporal representation. Experiment shows that with these framework, spatiotemporal representation improves the shot description, while using the graph structure helps for the retrieval of certain categories.

5.2 Perspectives

The scope of issues addressed in this thesis is rather large and several points still deserved to be worked on. Concerning the problem of spatiotemporal segmentation, a multiresolution approach that builds nested partitions would be advantageous. First, it would have the ability to lift ambiguities while grouping spatiotemporal regions by looking through their sub-regions. Secondly, the spatiotemporal representation will not be dependent on the fixed granularity of the segmentation. Thus, efficient techniques to handle the complexity of this

type of approach are still needed.

In the adopted representation, the low-level description of the video content is based on region descriptors, typically color or texture from the MPEG-7 standard. The descriptors are accurate but remains to some extent sensitive to changes in illumination, image noise, scaling, and changes in viewpoint. Completing the spatiotemporal description with local interest point descriptors (such as SIFT) would help matching of objects when its color or texture are not well-defined. The visual patches can be assigned to keywords for efficient indexing [116].

We have introduced graph similarity measures for comparing visually shots and objects. Matching a query graph with every graph in the database makes the search operation costly. When the similarity is known to be a metric, efficient techniques based on indexing trees can be employed [16]. However, most common similarity measures between graph are non-metric. For instance, the number of matched nodes between two graphs is not a metric. For unsupervised organization of object graph, the solution would then reside in clustering. Recently pairwise clustering algorithms such as affine propagation [35] has been proposed for the identification of good prototypes within a large set of a examples.

Appendix A

Mpeg-7 Description Example

A.1 Classification scheme (CS)

```
<?xml version="1.0" encoding="utf-8"?>
<Mpeg7>
  <Description xsi:type="ClassificationSchemeDescriptionType">
    <ClassificationScheme uri="urn:ex:cs:SemanticCS">
      <Term termID="RoadDriving">
        <Name xml:lang="en">RoadDriving</Name>
        <Definition xml:lang="en">Depicts a vehicule driving along the road</Definition>
      </Term>
      <Term termID="Road">
        <Name xml:lang="en">Road</Name>
        <Definition xml:lang="en">Depicts a road</Definition>
      </Term>
      <Term termID="Vehicule">
        <Name xml:lang="en">Vehicule</Name>
        <Definition xml:lang="en">Any medium of transport</Definition>
        <Term termID="Car">
          <Name xml:lang="en">Car</Name>
        </Term>
        <Term termID="Trunk">
          <Name xml:lang="en">Trunk</Name>
        </Term>
        <Term termID="Motorcycle">
          <Name xml:lang="en">Motorcycle</Name>
        </Term>
      </Term>
      <Term termID="War">
        <Name xml:lang="en">War</Name>
        <Definition xml:lang="en">Depicts military actions in period of war</Definition>
      </Term>
      <Term termID="Roadblock">
        <Name xml:lang="en">Roadblock</Name>
        <Definition xml:lang="en">barricades, blocking on a road</Definition>
        <Term termID="MilitaryRoadblock">
          <Name xml:lang="en">MilitaryRoadblock</Name>
          <Definition xml:lang="en">Roadblock controlled by militaries</Definition>
        </Term>
      </Term>
      <Term termID="Walk">
        <Name xml:lang="en">Walk</Name>
        <Definition xml:lang="en">Depicts a person walking</Definition>
      </Term>
      <Term termID="Stop">
        <Name xml:lang="en">Stop</Name>
        <Definition xml:lang="en">Depicts an objects wich stops moving.</Definition>
      </Term>
      <Term termID="Verify">
        <Name xml:lang="en">Verify</Name>
        <Definition xml:lang="en">Depicts someone searching an object or person
        </Definition>
      </Term>
    </ClassificationScheme>
  </Description>
</Mpeg7>
```

A possible classification scheme for the ”RoadDriving” and ”MilitaryRoadblock” narrative worlds.

A.2 Semantic Description

```
<Mpeg7>
 <Description xsi:type="SemanticDescriptionType">
  <Semantics id="RoadDriving-FA">
   <Label>
    <Name>Formal Abstraction of Road Driving</Name>
   </Label>
   <!-- Semantic entities -->
   <SemanticBase xsi:type="SemanticType" id="RoadDriving">
    <Label href="urn:ex:cs:SemanticCS:RoadDriving">
     <Name>Road Driving</Name>
    </Label>
    <SemanticBase xsi:type="EventType" id="Drive">
     <Label href="urn:ex:cs:SemanticCS:Drive">
      <Name>Drive</Name>
     </Label>
     <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agentOf"
      target="#Vehicule"/>
     <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:pathOf"
      target="#Road"/>
    </SemanticBase>
    <SemanticBase xsi:type="AgentObjectType" id="Vehicule">
     <Label href=" urn:ex:cs:SemanticCS:Vehicule">
      <Name>Vehicule</Name>
     </Label>
     <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:partOf"
      target="#RoadDriving"/>
    </SemanticBase>
    <SemanticBase xsi:type="SemanticPlaceType" id="Road">
     <Label href=" urn:ex:cs:SemanticCS:Road">
      <Name>Road</Name>
     </Label>
     <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:partOf"
      target="#RoadDriving"/>
    </SemanticBase>
   </SemanticBase>
  </Semantics>
 </Description>
</Mpeg7>
```

Formal abstraction of the ”RoadDriving” narrative world using the Semantic DS.

```
<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics id=" MilitaryRoadblock-FA" >
      <Label>
        <Name>Formal Abstraction of Military Roadblock</Name>
      </Label>
      <!-- Semantic entities -->
      <SemanticBase xsi:type="SemanticType" id="MilitaryRoadblock">
        <Label href="urn:ex:cs:SemanticCS:MilitaryRoadblock">
          <Name>MilitaryRoadblock</Name>
        </Label>
        <!--Events-->
        <SemanticBase xsi:type="EventType" id="Stop">
          <Label href="urn:ex:cs:SemanticCS:Stop">
            <Name>Stop</Name>
          </Label>
        </SemanticBase>
        <SemanticBase xsi:type="EventType" id="Walk">
          <Label href="urn:ex:cs:SemanticCS:Walk">
            <Name>Walk</Name>
          </Label>
        </SemanticBase>
        <SemanticBase xsi:type="EventType" id="Verify">
          <Label href="urn:ex:cs:SemanticCS:Verify">
            <Name>Verify</Name>
          </Label>
        </SemanticBase>
        <SemanticBase xsi:type="AgentObjectType" id="Vehicule">
          <Label href=" urn:ex:cs:SemanticCS:Vehicule">
            <Name>Vehicule</Name>
          </Label>
        </SemanticBase>
        <SemanticBase xsi:type="AgentObjectType" id="Military">
          <Label href=" urn:ex:cs:SemanticCS:Military">
            <Name>Military</Name>
          </Label>
        </SemanticBase>
        <SemanticBase xsi:type="SemanticPlaceType" id="Roadblock">
          <Label href="urn:ex:cs:SemanticCS:Roadblock">
            <Name>Roadblock</Name>
          </Label>
        </SemanticBase>
        <SemanticBase xsi:type="ContextType" id="War">
          <Label href="urn:ex:cs:SemanticCS:War">
            <Name>War</Name>
          </Label>
        </SemanticBase>
      </SemanticBase>
      <!-- graph of semantic relations -->
      <Graph>
        source="#Vehicule" target="#Wait"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agentOf"
         source="#Vehicule" target="#Stop"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:locationOf"
          source="#RoadBlock" target="#Stop"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agentOf"
         source="#Military" target="#Walk"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:destinationOf"
         source="#Vehicule" target="#Walk"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agentOf"
         source="#Military" target="#Verify"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:patientOf"
         source="#Vehicule" target="#Verify"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:partOf"
          source="#Military" target="#MilitaryRoadblock"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:partOf"
          source="#Roadblock" target="#Roadblock"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:contextFor"
          source="#War" target="#MilitaryRoadblock"/>
      </Graph>
    </Semantics>
  </Description>
</Mpeg7>
```

Formal abstraction of the ”MilitaryRoadblock” narrative world using the Semantic DS.

```
<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <!--narrative worlds -->
    <Semantics id="worlds">
      <Label>
        <Name>Description of narrative worlds</Name>
      </Label>
      <SemanticBase xsi:type="SemanticType" id="W1">
        <Label href="urn:ex:cs:SemanticCS:RoadDriving">
          <Name>RoadDriving</Name>
        </Label>
        <MediaOccurrence>
          <MediaInformationRef idref="video-1"/>
          <Mask xsi:type="TemporalMaskType">
            <SubInterval>
              <MediaTimePoint>T00:00:00:0F25</MediaTimePoint>
              <MediaDuration>PT00H00M02S0N25F</MediaDuration>
            </SubInterval>
          </Mask>
        </MediaOccurrence>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:examplifies"
         target="#RoadDriving-FA"/>
        <!--Semantic relations -->
        <Graph>
          <node id="W1-n1" idref="E1"/>
          <node id="W1-n2" idref="O1"/>
        </Graph>
      </SemanticBase>
      <SemanticBase xsi:type="SemanticType" id="W2">
        <Label href="urn:ex:cs:SemanticCS:MilitaryRoadblock">
          <Name>MilitaryRoadblock</Name>
        </Label>
        <MediaOccurrence>
          <MediaInformationRef idref="video-1"/>
          <Mask xsi:type="TemporalMaskType">
            <SubInterval>
              <MediaTimePoint>T00:00:02:0F25</MediaTimePoint>
              <MediaDuration>PT00H00M09S0N25F</MediaDuration>
            </SubInterval>
          </Mask>
        </MediaOccurrence>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:examplifies"
         target="#MilitaryRoadBlock-FA"/>
        <Graph>
          <node id="W2-n1" idref="E2"/>
          <node id="W2-n2" idref="E3"/>
          <node id="W2-n3" idref="E4"/>
          <node id="W2-n4" idref="O2"/>
          <node id="W2-n5" idref="O3"/>
        </Graph>
      </SemanticBase>
    </Semantics>
    <!--Events-->
    <Semantics id="Events">
      <Label>
        <Name>Description of events</Name>
      </Label>
      <SemanticBase xsi:type="EventType" id="E1">
        <Label>
          <Name>Truck drives on the road</Name>
        </Label>
        <MediaOccurrence>
          <MediaInformationRef idref="video-1"/>
          <Mask xsi:type="TemporalMaskType">
            <SubInterval>
              <MediaTimePoint>T00:00:00:0F25</MediaTimePoint>
              <MediaDuration>PT00H00M02S0N25F</MediaDuration>
            </SubInterval>
          </Mask>
        </MediaOccurrence>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:examplifies"
          target="#Drive"/>
        <!--Object links -->
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"
          target="#O1"/>
      </SemanticBase>
```

```
      <SemanticBase xsi:type="EventType" id="E2">
        <Label>
          <Name>Military walks towards the truck</Name>
        </Label>
        <MediaOccurrence>
          <MediaInformationRef idref="video-1"/>
          <Mask xsi:type="TemporalMaskType">
            <SubInterval>
              <MediaTimePoint>T00:00:03:0F25</MediaTimePoint>
              <MediaDuration>PT00H00M03S0N25F</MediaDuration>
            </SubInterval>
          </Mask>
        </MediaOccurrence>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:examplifies"
          target="#Walk"/>
        <!--Object links -->
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"
          target="#O3"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:patient"
         target="#O2"/>
      </SemanticBase>
      <SemanticBase xsi:type="EventType" id="E3">
        <Label>
          <Name>Truck stops</Name>
        </Label>
        <MediaOccurrence>
          <MediaInformationRef idref="video-1"/>
          <Mask xsi:type="TemporalMaskType">
            <SubInterval>
              <MediaTimePoint>T00:00:03:12F25</MediaTimePoint>
              <MediaDuration>PT00H00M01S00N25F</MediaDuration>
            </SubInterval>
          </Mask>
        </MediaOccurrence>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:examplifies"
          target="#Stop"/>
        <!--Object links -->
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"
         target="#O2"/>
      </SemanticBase>
      <SemanticBase xsi:type="EventType" id="E4">
        <Label>
          <Name>Military verifies the truck</Name>
        </Label>
        <MediaOccurrence>
          <MediaInformationRef idref="video-1"/>
          <Mask xsi:type="TemporalMaskType">
            <SubInterval>
              <MediaTimePoint>T00:00:06:00F25</MediaTimePoint>
              <MediaDuration>PT00H00M05SN25F</MediaDuration>
            </SubInterval>
          </Mask>
        </MediaOccurrence>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:examplifies"
          target="#Wait"/>
        <!--Object links -->
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent"
          target="#O3"/>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:patient"
         target="#O2"/>
      </SemanticBase>
    </Semantics>
    <!--objects-->
    <Semantics id="Objects">
      <Label>
        <Name>Description of objects</Name>
      </Label>
      <SemanticBase xsi:type="ObjectType" id="O1">
        <Label>
          <Name>Truck O1</Name>
        </Label>
        <MediaOccurrence>
          <MediaInformationRef idref="video-1"/>
          <Mask xsi:type="TemporalMaskType">
            <SubInterval>
              <MediaTimePoint>T00:00:00:00F25</MediaTimePoint>
```

```
            <MediaDuration>PT00H00M02SN25F</MediaDuration>
          </SubInterval>
        </Mask>
      </MediaOccurrence>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:examplifies"
        target="#Vehicule"/>
      <!--Object links -->
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agentOf"
        target="#E1"/>
    </SemanticBase>
    <Label>
      <Name>Description of objects</Name>
    </Label>
    <SemanticBase xsi:type="ObjectType" id="O1">
      <Label>
        <Name>Truck O2</Name>
      </Label>
      <MediaOccurrence>
        <MediaInformationRef idref="video-1"/>
        <Mask xsi:type="TemporalMaskType">
          <SubInterval>
            <MediaTimePoint>T00:00:03:12F25</MediaTimePoint>
            <MediaDuration>PT00H00M07S13N25F</MediaDuration>
          </SubInterval>
        </Mask>
      </MediaOccurrence>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:examplifies"
        target="#Vehicule"/>
      <!--Object links -->
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:identifies"
        target="#O1"/>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agentOf"
        target="#E3"/>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:patientOf"
        target="#E4"/>
    </SemanticBase>
    <Label>
      <Name>Description of objects</Name>
    </Label>
    <SemanticBase xsi:type="ObjectType" id="O3">
      <Label>
        <Name>Military O3</Name>
      </Label>
      <MediaOccurrence>
        <MediaInformationRef idref="video-1"/>
        <Mask xsi:type="TemporalMaskType">
          <SubInterval>
            <MediaTimePoint>T00:00:02:0F25</MediaTimePoint>
            <MediaDuration>PT00H00M09SN25F</MediaDuration>
          </SubInterval>
        </Mask>
      </MediaOccurrence>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:examplifies"
        target="#Military"/>
      <!--Object links -->
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agentOf"
        target="#E2"/>
      <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agentOf"
        target="#E4"/>
    </SemanticBase>
  </Semantics>
 </Description>
</Mpeg7>
```

MPEG-7 Semantic Description for the audiovisual content of fig.1.16.

The description describes hierarchically the narrative worlds (W_1, W_2), the events occurring in the video (E_1 to E_4), and the different objects in the sequence. Objects and events intervening in the description of a narrative world are referenced by a Graph DS. The Relation DS is used to link events to their associated objects and vice-versa. The occurrences of each semantic entity in the video are finally reported with the MediaOccurrence DS.

A.3 Visual and Structural Description

```
<Mpeg7>
  <Description xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video id="video-1">
        <MediaLocator>
          <MediaUri>video-1.mpg</MediaUri>
        </MediaLocator>
        <MediaTime>
          <MediaTimePoint>T00:00:00:0F25</MediaTimePoint>
          <MediaDuration>PT00H06M39S24N25F</MediaDuration>
        </MediaTime>
        <TemporalDecomposition>
          <!-- Shot S1  -->
          <VideoSegment xsi:type="ShotType" id="S1" >
            <MediaTime>
              <MediaTimePoint>T00:00:00:0F25</MediaTimePoint>
              <MediaDuration>PT00H00M02S0N25F</MediaDuration>
            </MediaTime>
            <!-- Semantic description of the shot-->
            <SemanticRef idref="#W1"/>
            <!-- Decomposition  into moving regions-->
            <SpatioTemporalDecomposition>
              <MovingRegion id="mr1">
                <MediaTime>
                  <MediaTimePoint>T00:00:00:0F25</MediaTimePoint>
                  <MediaDuration>PT00H00M02S0N25F</MediaDuration>
                </MediaTime>
                <SemanticRef idref="#O1"/>
                <!-- Localisation -->
                <SpatioTemporalLocator>
                  <ParameterTrajectory motionModel="still">
                    <InitialRegion>
                      <Box mpeg7:dim="3 3">20 10 0 279 282 50</Box>
                    </InitialRegion>
                  </ParameterTrajectory>
                  <ParameterTrajectory motionModel="affine">
                    <InitialRegion>
                      <Box mpeg7:dim="2 2">20 10 277 275</Box>
                    </InitialRegion>
                    <Params keyPointNum="2">
                      <WholeInterval>
                        <MediaDuration>PT00H00M02S0N25F</MediaDuration>
                      </WholeInterval>
                      <InterpolationFunctions>
                        <KeyValue type="startPoint">-2.5612</KeyValue>
                        <KeyValue type="startPoint">-2.3643</KeyValue>
                      </InterpolationFunctions>
                      <InterpolationFunctions>
                        <KeyValue type="startPoint">-2.4233</KeyValue>
                        <KeyValue type="startPoint">-2.2153</KeyValue>
                      </InterpolationFunctions>
                      <InterpolationFunctions>
                        <KeyValue type="startPoint">-0.0234</KeyValue>
                        <KeyValue type="startPoint">-0.0231</KeyValue>
                      </InterpolationFunctions>
                      <InterpolationFunctions>
                        <KeyValue type="startPoint">0.0021</KeyValue>
                        <KeyValue type="startPoint">0.0019</KeyValue>
                      </InterpolationFunctions>
                      <InterpolationFunctions>
                        <KeyValue type="startPoint">-0.0262</KeyValue>
                        <KeyValue type="startPoint">-0.0251</KeyValue>
                      </InterpolationFunctions>
                      <InterpolationFunctions>
                        <KeyValue type="startPoint">0.0019</KeyValue>
                        <KeyValue type="startPoint">0.0012</KeyValue>
                      </InterpolationFunctions>
                    </Params>
                  </ParameterTrajectory>
                </SpatioTemporalLocator>
```

```
            <!-- visual descriptors -->
            <VisualDescriptor xsi:type="ParametricMotion" motionModel="affine">
              <Duration>
                <MediaIncrDuration  mediaTimeUnit="PT1N25F">50</MediaIncrDuration>
              </Duration>
              <Parameters>-2.5612 -2.4233 -0.0234 0.0021 -0.0262 0.0012</Parameters>
            </VisualDescriptor>
            <VisualDescriptor xsi:type="ColorStructureType" colorQuant="1">
              <Values>
                0 0 0 0 0 1 0 0 4 3 0 0 2 50 14 0 19 84 17 0 5 3 0 0 2 3 3 2 0 0 0 0
              </Values>
            </VisualDescriptor>
            <VisualDescriptor xsi:type="EdgeHistogramType" >
              <BinCounts>
                5 3 2 2 0 3 3 2 0 0 0 2 3 7 2 5 0 7 1 1 4 2 3 1 0 5 0 2 0 0 4 1 7 1 2
                6 0 0 0 0 5 0 0 0 0 5 2 2 1 1 4 5 3 2 0 6 1 2 1 0 2 1 0 1 0 3 2 0 5 1
                3 2 4 0 1 4 1 1 0 0
              </BinCounts>
            </VisualDescriptor>
          </MovingRegion>
        </SpatioTemporalDecomposition>
        <TemporalDecomposition overlap="true" gap="true">
          <VideoSegment id="S1-1">
            <MediaTime>
              <MediaTimePoint>T00:00:00:0F25</MediaTimePoint>
              <MediaDuration>PT00H00M02S0N25F</MediaDuration>
            </MediaTime>
            <semanticRef idref="E1"/>
            <SpatioTemporalDecomposition>
              <MovingRegionRef idref="mr1"/>
            </SpatioTemporalDecomposition>
          </VideoSegment>
        </TemporalDecomposition>
      </VideoSegment>
      <!-- Shot S2  -->
      <VideoSegment xsi:type="ShotType" id="S2" >
        <MediaTime>
          <MediaTimePoint>T00:00:02:0F25</MediaTimePoint>
          <MediaDuration>PT00H00M09S0N25F</MediaDuration>
        </MediaTime>
        <SemanticRef idref="#W2"/>
        <SpatioTemporalDecomposition>
          <MovingRegion id="mr2">
            <MediaTime>
              <MediaTimePoint>T00:00:03:12F25</MediaTimePoint>
              <MediaDuration>PT00H00M01S0N25F</MediaDuration>
            </MediaTime>
            <!-- Semantic description-->
            <SemanticRef idref="#O2"/>
            <!-- Localisation -->
            <SpatioTemporalLocator>
              <!--bounding box only-->
              <ParameterTrajectory motionModel="still">
                <InitialRegion>
                  <Box mpeg7:dim="3 3">0 50 0 255 183 25</Box>
                </InitialRegion>
              </ParameterTrajectory>
            </SpatioTemporalLocator>
            <!-- visual descriptors -->
            <VisualDescriptor xsi:type="ColorStructureType" colorQuant="1">...
            </VisualDescriptor>
            <VisualDescriptor xsi:type="EdgeHistogramType" >...
            </VisualDescriptor>
            <!-- Object Decomposition -->
            <SpatioTemporalDecomposition>
              <MovingRegion id="mr2-1">
                <MediaTime>
                  <MediaTimePoint>T00:00:03:12F25</MediaTimePoint>
                  <MediaDuration>PT00H00M01S0N25F</MediaDuration>
                </MediaTime>
                <!-- Semantic description-->
                <SemanticRef idref="#O2"/>
                <!-- Localisation -->
                <SpatioTemporalLocator>
                  <ParameterTrajectory motionModel="affine">
                    <InitialRegion>
```

```
                    <Box mpeg7:dim="2 2">0 50 120 159</Box>
                  </InitialRegion>
                  <Params keyPointNum="2">
                    <WholeInterval>
                      <MediaDuration>PT00H00M01S0N25F</MediaDuration>
                    </WholeInterval>
                    <InterpolationFunctions>...</InterpolationFunctions>
                    <InterpolationFunctions>...</InterpolationFunctions>
                    <InterpolationFunctions>...</InterpolationFunctions>
                    <InterpolationFunctions>...</InterpolationFunctions>
                    <InterpolationFunctions>...</InterpolationFunctions>
                    <InterpolationFunctions>...</InterpolationFunctions>
                  </Params>
                </ParameterTrajectory>
              </SpatioTemporalLocator>
              <!-- Visual descriptors -->
              <!--motion model changes (decelerate)-->
              <VisualDescriptor xsi:type="ParametricMotion" motionModel="affine">
                <Duration>
                  <MediaIncrDuration  mediaTimeUnit="PT1N25F">12</MediaIncrDuration>
                </Duration>
                <Parameters>3.253 -0.005 -0.0002 0.0005 -0.0012 0.0009
                </Parameters>
              </VisualDescriptor>
              <VisualDescriptor xsi:type="ParametricMotion" motionModel="affine">
                <Duration>
                  <MediaIncrDuration  mediaTimeUnit="PT1N25F">13</MediaIncrDuration>
                </Duration>
                <Parameters>0.85 -0.002  -0.0003 0.0007  -0.0008 0.0011
                </Parameters>
              </VisualDescriptor>
              <VisualDescriptor xsi:type="ColorStructureType" colorQuant="1">...
              </VisualDescriptor>
              <VisualDescriptor xsi:type="EdgeHistogramType" >...
              </VisualDescriptor>
            </MovingRegion>
            <MovingRegion id="mr2-2">
              <MediaTime>
                <MediaTimePoint>T00:00:04:12F25</MediaTimePoint>
                <MediaDuration>PT00H00M06S13N25F</MediaDuration>
              </MediaTime>
              <!-- Semantic description-->
              <SemanticRef idref="#O2"/>
              <!-- Localisation -->
              <SpatioTemporalLocator>
                <ParameterTrajectory motionModel="affine">
                  <InitialRegion>
                    <Box mpeg7:dim="2 2">0 50 120 159</Box>
                  </InitialRegion>
                  <Params keyPointNum="2">
                    <WholeInterval>
                      <MediaDuration>PT00H00M01S0N25F</MediaDuration>
                    </WholeInterval>
                    <InterpolationFunctions>...</InterpolationFunctions>
                    <InterpolationFunctions>...</InterpolationFunctions>
                    <InterpolationFunctions>...</InterpolationFunctions>
                    <InterpolationFunctions>...</InterpolationFunctions>
                    <InterpolationFunctions>...</InterpolationFunctions>
                    <InterpolationFunctions>...</InterpolationFunctions>
                  </Params>
                </ParameterTrajectory>
              </SpatioTemporalLocator>
              <!-- Visual descriptors -->
              <VisualDescriptor xsi:type="ParametricMotion" motionModel="affine">
                <Duration>
                  <MediaIncrDuration  mediaTimeUnit="PT1N25F">12</MediaIncrDuration>
                </Duration>
                <Parameters>3.253 -0.005 -0.0002 0.0005 -0.0012 0.0009
                </Parameters>
              </VisualDescriptor>
              <VisualDescriptor xsi:type="ParametricMotion" motionModel="affine">
                <Duration>
                  <MediaIncrDuration  mediaTimeUnit="PT1N25F">13</MediaIncrDuration>
                </Duration>
                <Parameters>0.85 -0.0002 -0.0003 0.0007 -0.0008 0.0011
                </Parameters>
```

```
                </VisualDescriptor>
                  <VisualDescriptor xsi:type="ColorStructureType" colorQuant="1">...
                  </VisualDescriptor>
                  <VisualDescriptor xsi:type="EdgeHistogramType" >...
                  </VisualDescriptor>
                  <TemporalDecomposition gap="true">
                    <MovingRegion id="mr2-2-E4">
                      <MediaTime>
                        <MediaTimePoint>T00:00:06:0F25</MediaTimePoint>
                        <MediaDuration>PT00H00M05S0N25F</MediaDuration>
                      </MediaTime>
                    </MovingRegion>
                  </TemporalDecomposition>
                </MovingRegion>
              </SpatioTemporalDecomposition>
            </MovingRegion>
            <MovingRegion id="mr3">
              <MediaTime>
                <MediaTimePoint>T00:00:02:00F25</MediaTimePoint>
                <MediaDuration>PT00H00M09S0N25F</MediaDuration>
              </MediaTime>
              <!-- Semantic description-->
              <SemanticRef idref="#03"/>
              <!-- Localisation -->
              <SpatioTemporalLocator>
                <!--bounding box only-->
                <ParameterTrajectory motionModel="still">
                  <InitialRegion>
                    <Box mpeg7:dim="3 3">175 58 0 106 195 225</Box>
                  </InitialRegion>
                </ParameterTrajectory>
              </SpatioTemporalLocator>
              <!-- visual descriptors -->
              <VisualDescriptor xsi:type="ColorStructureType" colorQuant="1">...
              </VisualDescriptor>
              <VisualDescriptor xsi:type="EdgeHistogramType" >...
              </VisualDescriptor>
              <!-- Object Decomposition -->
              <SpatioTemporalDecomposition>
                <MovingRegion id="mr3-1">
                  <MediaTime>
                    <MediaTimePoint>T00:00:02:00F25</MediaTimePoint>
                    <MediaDuration>PT00H00M01S0N25F</MediaDuration>
                  </MediaTime>
                  <!-- Semantic description-->
                  <SemanticRef idref="#03"/>
                  <!-- Localisation -->
                  <SpatioTemporalLocator>
                    <ParameterTrajectory motionModel="still">
                      <InitialRegion>
                        <Box mpeg7:dim="2 2">202 55  65 173</Box>
                      </InitialRegion>
                    </ParameterTrajectory>
                  </SpatioTemporalLocator>
                  <!-- Visual descriptors -->
                  <VisualDescriptor xsi:type="ColorStructureType" colorQuant="1">...
                  </VisualDescriptor>
                  <VisualDescriptor xsi:type="EdgeHistogramType" >...
                  </VisualDescriptor>
                </MovingRegion>
                <MovingRegion id="mr3-2">
                  <MediaTime>
                    <MediaTimePoint>T00:00:03:00F25</MediaTimePoint>
                    <MediaDuration>PT00H00M03S0N25F</MediaDuration>
                  </MediaTime>
                  <!-- Semantic description-->
                  <SemanticRef idref="#03"/>
                  <!-- Localisation -->
                  <SpatioTemporalLocator>
                    <ParameterTrajectory motionModel="affine">
                      <InitialRegion>
                        <Box mpeg7:dim="2 2">198 58  64 171</Box>
                      </InitialRegion>
                      <Params keyPointNum="2">
                        <WholeInterval>
                          <MediaDuration>PT00H00M01S0N25F</MediaDuration>
```

```
                        </WholeInterval>
                        <InterpolationFunctions>...</InterpolationFunctions>
                        <InterpolationFunctions>...</InterpolationFunctions>
                        <InterpolationFunctions>...</InterpolationFunctions>
                        <InterpolationFunctions>...</InterpolationFunctions>
                        <InterpolationFunctions>...</InterpolationFunctions>
                        <InterpolationFunctions>...</InterpolationFunctions>
                      </Params>
                    </ParameterTrajectory>
                  </SpatioTemporalLocator>
                  <!-- Visual descriptors -->
                  <!--motion model north-west, then ouest -->
                  <VisualDescriptor xsi:type="ParametricMotion" motionModel="affine">
                    <Duration>
                      <MediaIncrDuration  mediaTimeUnit="PT1N25F">45</MediaIncrDuration>
                    </Duration>
                    <Parameters> -1.2534 -1.1745 -0.0042 0.0023 -0.0054 0.0048
                     </Parameters>
                  </VisualDescriptor>
                  <VisualDescriptor xsi:type="ParametricMotion" motionModel="affine">
                    <Duration>
                      <MediaIncrDuration  mediaTimeUnit="PT1N25F">30</MediaIncrDuration>
                    </Duration>
                    <Parameters>-0.9835 0.0123 0.0014 0.0021 -0.0007 0.0017
                    </Parameters>
                  </VisualDescriptor>
                  <VisualDescriptor xsi:type="ColorStructureType" colorQuant="1">...
                  </VisualDescriptor>
                  <VisualDescriptor xsi:type="EdgeHistogramType" >...
                  </VisualDescriptor>
                </MovingRegion>
                <MovingRegion id="mr3-3">
                  <MediaTime>
                    <MediaTimePoint>T00:00:06:00F25</MediaTimePoint>
                    <MediaDuration>PT00H00M05S0N25F</MediaDuration>
                  </MediaTime>
                  <!-- Semantic description-->
                  <SemanticRef idref="#03"/>
                  <!-- Localisation -->
                  <SpatioTemporalLocator>
                    <ParameterTrajectory motionModel="affine">
                      <InitialRegion>
                        <Box mpeg7:dim="2 2">183 55  26 156</Box>
                      </InitialRegion>
                      <Params keyPointNum="5">
                        <WholeInterval>
                          <MediaDuration>PT00H00M01S0N25F</MediaDuration>
                        </WholeInterval>
                        <InterpolationFunctions>...</InterpolationFunctions>
                        <InterpolationFunctions>...</InterpolationFunctions>
                        <InterpolationFunctions>...</InterpolationFunctions>
                        <InterpolationFunctions>...</InterpolationFunctions>
                        <InterpolationFunctions>...</InterpolationFunctions>
                        <InterpolationFunctions>...</InterpolationFunctions>
                      </Params>
                    </ParameterTrajectory>
                  </SpatioTemporalLocator>
                  <!-- Visual descriptors -->
                  <!--complex motion-->
                  <VisualDescriptor xsi:type="ParametricMotion" motionModel="affine">...
                  </VisualDescriptor>
                  <VisualDescriptor xsi:type="ParametricMotion" motionModel="affine">...
                  </VisualDescriptor>
                  <VisualDescriptor xsi:type="ParametricMotion" motionModel="affine">...
                  </VisualDescriptor>
                  <VisualDescriptor xsi:type="ParametricMotion" motionModel="affine">...
                  </VisualDescriptor>
                  <VisualDescriptor xsi:type="ParametricMotion" motionModel="affine">...
                  </VisualDescriptor>
                  <VisualDescriptor xsi:type="ColorStructureType" colorQuant="1">...
                  </VisualDescriptor>
                  <VisualDescriptor xsi:type="EdgeHistogramType" >...
                  </VisualDescriptor>
                </MovingRegion>
              </SpatioTemporalDecomposition>
            </MovingRegion>
```

```
            </SpatioTemporalDecomposition>
            <!--temporal decomposition  into the events -->
            <TemporalDecomposition overlap="true" gap="true">
              <!--military walk-->
              <VideoSegment id="S2-1">
                <MediaTime>
                  <MediaTimePoint>T00:00:03:0F25</MediaTimePoint>
                  <MediaDuration>PT00H00M02S0N25F</MediaDuration>
                </MediaTime>
                <semanticRef idref="E2"/>
                <SpatioTemporalDecomposition>
                  <MovingRegionRef idref="mr2-1"/>
                  <MovingRegionRef idref="mr3-2"/>
                  <Graph>
                    <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:rightOf"
          source="mr3-2" target="mr2-1"/>
                  </Graph>
                </SpatioTemporalDecomposition>
              </VideoSegment>
              <!-- truck stop-->
              <VideoSegment id="S2-2">
                <MediaTime>
                  <MediaTimePoint>T00:00:03:12F25</MediaTimePoint>
                  <MediaDuration>PT00H00M01S0N25F</MediaDuration>
                </MediaTime>
                <semanticRef idref="E3"/>
                <SpatioTemporalDecomposition>
                  <MovingRegionRef idref="mr2-1"/>
                </SpatioTemporalDecomposition>
              </VideoSegment>
              <VideoSegment id="S2-3">
                <MediaTime>
                  <MediaTimePoint>T00:00:06:0F25</MediaTimePoint>
                  <MediaDuration>PT00H00M05S0N25F</MediaDuration>
                </MediaTime>
                <semanticRef idref="E4"/>
                <SpatioTemporalDecomposition>
                  <MovingRegionRef idref="mr2-2-E4"/>
                  <MovingRegionRef idref="mr3-3"/>
                  <Graph>
                    <Relation type="urn:mpeg:mpeg7:cs:SpatialRelationCS:2001:overlaps"
          source="#mr3-3" target="#mr2-2-E4"/>
                  </Graph>
                </SpatioTemporalDecomposition>
              </VideoSegment>
            </TemporalDecomposition>
          </VideoSegment>
        </TemporalDecomposition>
      </Video>
    </MultimediaContent>
  </Description>
</Mpeg7>
```

MPEG-7 Content description for the example fig.1.16.

The video is decomposed hierarchically in shots and objects with the TemporalDecomposition and SpatioTemporalDecomposition DSs, respectively. The shot is decomposed into objects represented with the MovingRegion DS, and can be further decomposed spatiotemporally into sub-objects. In the example each sub-object is related to corresponding event (drive, stop, walk, verify). The shot can be also structured temporally with events. The objects intervening in the event are referenced within a SpatioTemporalDecomposition DS. Object segments that overlaps the event are aligned temporally with the event using a TemporalDecomposition DS, as a parent segment should contain all its children segments.

Bibliography

[1] G. Adiv. Inherent ambiguities in recovering 3d motion and structure from a noisy field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):477–489, 1989.

[2] H. Agius and M.C. Angelides. Enriching Mpeg-7 User Model with Content Metadata. In *2nd International Workshop on Semantic Media Adaptation and Personalization SMAP'06)*, pages 151–156, Washington, DC, USA, 2006. IEEE Computer Society.

[3] G. Ahanger and T.D.C. Little. A survey of technologies for parsing and indexing digital video. *Journal of Visual Communication and Image Representation*, 7(1):28–43, March 1996.

[4] G. Akrivas, G. B. Stamou, and S. Kollias. Semantic association of multimedia document descriptions through fuzzy relational algebra and fuzzy reasoning. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(2), March 2004.

[5] J.F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, Nov. 1983.

[6] Arnon Amir, Shih-Fu Chang, Martin Franz, Giridharan Iyengar, John R. Kender, Ching-Yung Lin, Milind R. Naphade, Apostol Natsev, John R. Smith, and Jelena Tesic. Ibm research trecvid-2004 video retrieval system. In *NIST TRECVID-2004*, March 2005.

[7] M.C. Angelides and A.Agius. Semantic integration and retrieval of multimedia metadata. In *Fifth International Workshop on Knowledge Markup and Semantic Annotation at the Fourth International Semantic Web Conference*, Ireland, 2005.

[8] M.C. Angelides and A.Agius. A mpeg-7 scheme for semantic content modelling and filtering of digital video. In *ACM Multimedia Systems)*, pages 320–339, 2006.

[9] E. Ardizzone, M. La Cascia, and D. Molinelli. Motion and color based video indexing and retrieval. In *ICPR*, pages III: 135–139, 1996.

[10] Th. Athanasiadis, Ph. Mylonas, Y. Avrithis, and S. Kollias. Semantic image segmentation and object labeling. *IEEE Transactions On Circuits And Systems For Video Technology*, 17(3), March 2007.

[11] Th. Athanasiadis, V. Tzouvaras, V. Petridis, F. Precioso, Y. Avrithis, and Y. Kompatsiaris. Using a multimedia ontology infrastructure for semantic annotation of multimedia content. In *5th International Workshop on Knowledge Markup and Semantic Annotation*, pages 236–247, Galway, Ireland, November 2005 2005.

[12] W. Bailer and P. Schallauer. The detailed audiovisual profile: Enabling interoperability between mpeg-7 based systems. In *IEEE Multimedia Modelling Conference*, pages 217–224, Jan. 2000.

[13] H.G. Barrow and R.J. Popplestone. Relational descriptions in picture procressing. machine intelligence. *Machine Intelligence*, 6:377–396, 1971.

[14] A. Basharat, Y. Zhai, and M. Shah. Content based video matching using spatiotemporal volumes. *Journal of Computer Vision and Image Understanding (CVIU)*, 2008.

[15] R.E. Bellman. *Adaptive Control Processes*. Princeton University Press, 1961.

[16] S. Berretti, A. del Bimbo, and E. Vicario. Efficient matching and indexing of graph models in content-based retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10):1089–1105, October 2001.

[17] L. Bonnaud and C. Labit. Multiple objects tracking using a non-redundant boundary-based representation for image sequence interpolation after decoding. In *ICIP'97*, volume 2, pages 426–429, 1997.

[18] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[19] E. Bruno and D. Pellerin. Series expansion for video indexing and retrieval. In *Advances in Visual Information System,Visual*, pages 327–337, Lyon, 2000.

[20] E. Bruno and D. Pellerin. Video structuring, indexing and retrieval based on global motion wavelet coefficients. In *International Conference on Pattern Recognition*, pages III: 287–290, 2002.

[21] H. Bunke, C. Irniger, and M. Neuhaus. Graph matching - challenges and potential solutions. In *International Conference on Image Analysis and Processing (ICIAP 05)*, September 2005.

[22] C. Carson, M. Thomas, and S. Belongie. Blobworld: a system for region-based indexing and retrieval. In *VISUAL*, pages 509–516, 1999.

[23] M.M. Chang, A.M. Tekalp, and M.I. Sezan. Motion-field segmentation using an adaptive map criterion. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 33–36 vol.5, 27-30 Apr 1993.

[24] S.F. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong. Videoq: An automatic content-based video search system using visual cues. In *ACM Multimedia*, 1997.

[25] S.F Chang, T. Sikora, and A. Puri. Overview of the mpeg-7 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):688–695, June 2001.

[26] Shih-Fu Chang, W. Chen, H.J. Meng, H. Sundaram, and Di Zhong. A fully automated content-based video search engine supporting spatiotemporal queries. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8(5):602–615, Sept 1998.

[27] S.K. Chang, Q.Y. Shi, and C.W. Yan. Iconic indexing by 2-d strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3):413–428, May 1987.

[28] F. Chevalier, M. Delest, and J.P. Domenger. A heuristic for the retrieval of objects in video in the framework of the rough indexing paradigm. *Signal Processing : Image Communication*, 22(7-8):622–634, August 2007.

[29] S. Dagtas, W Al-Khatib, A. Ghafoor, and R.L. Kashyap. Models for motion-based video indexing and retrieval. *IEEE Transactions on Image Processing*, 9(1):88–101, 2000.

[30] A. del Bimbo, P. Pala, and L. Tanganelli. Video retrieval based on dynamics of color flows. In *ICPR*, pages 1: 851–854, 2000.

[31] D. DeMenthon and D. Doermann. Video retrieval using spatio-temporal descriptors. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 508–517, Berkeley, CA, USA, November 2003.

[32] Y. Deng and B.S Manjunath. Netra-v toward an object-based video representation. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):616–627, 1998.

[33] Y. Deng and B.S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8):800–810, August 2001.

[34] Nevenka Dimitrova and Forouzan Golshani. Motion recovery for video content classification. *ACM Transactions on Information Systems*, 13(4):408–439, 1995.

[35] D. Dueck and B.J. Frey. Non-metric affinity propagation for unsupervised image categorization. In *International Conference on Computer Vision*, pages 1–8, 2007.

[36] F. Dufaux, F. Moscheni, and A. Lippman. Spatio-temporal segmentation based on motion and static segmentation. *ICIP*, 1:306, 1995.

[37] H. Eidenberger. How good are the visual mpeg-7 features? *Visual Communications and Image Processing*, 5150:476–488, 2003.

[38] Horst Eidenberger and Christian Breiteneder. Vizir - a framework for visual information retrieval. *J. Vis. Lang. Comput.*, 14(5):443–469, 2003.

[39] M.A. El Saban and B.S. Manjunath. Video region segmentation by spatio-temporal watersheds. In *ICIP'03*, volume 1, pages 349–352, Barcelona, Spain, September 2003.

[40] R. Fablet, P. Bouthemy, and M. Gelgon. Moving object detection in color image sequences using region-level graph labeling. In *ICIP'99*, volume 2, pages 939–943, Kobe, Japan, October 1999.

[41] R. Fablet, P. Bouthemy, and P. Perez. Nonparametric motion characterization using causal probabilistic models for video indexing and retrieval. *IEEE Transactions on Image Processing*, 11(4):393–407, April 2002.

[42] Ronan Fablet and Patrick Bouthemy. Spatio-temporal segmentation and general motion characterization for video indexing and retrieval. In *10th DELOS Workshop on Audio-Visual Digital Libraries*, Santorini, Greece, June 1999.

[43] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, September 2004.

[44] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, David Steele, and Peter Yanker. Query by image and video content: The qbic system. *Computer*, 28(9):23–32, 1995.

[45] G. Foret, P. Bertolino, and D. Cibaud. Partition projection in videos by global and local block-matching. In *ICIP'02*, volume 3, pages 409–412, 2002.

[46] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–224, February 2004.

[47] Frameline. http://frameline.tv/.

[48] M. Galler, E. Sharon, R. Basri, and A. Brandt. An improved equivalence algorithm. In *Communications of the ACM*, volume 7, pages 301–303, May 1964.

[49] E. Galmar and B. Huet. Graph-based spatio-temporal region extraction. In *ICIAR*, pages 236–247, 2006.

[50] M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *ICCV*, 2003.

[51] M. Gelgon and P. Bouthemy. A region-level motion-based graph representation and labeling for tracking a spatial image partition. *Pattern Recognition*, 33(4):725–740, April 2000.

[52] S. Gephstein and M. Kubocy. The emergence of visual-objects in space-time. In *Proceedings of the National Acamedy of Sciences*, volume 97, pages 8186–8191, 2000.

[53] C. Gomila and F. Meyer. Graph-based object tracking. In *ICIP'03*, volume 2, pages 41–44, 2003.

[54] R.C. Gonzales and R.E Woods. *Digital Image Processing*. Addison Wesley Longman, 1992.

[55] H. Greenspan, J. Goldberger, and A. Mayer. Probabilistic space-time video modeling via piecewise gmm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):384–396, March 2004.

[56] C. Gu and M.-C. Lee. Semi-automaticsegmentation and tracking of video objects. *IEEE Transactions on Circuits and Systems for Video Technology*, 8:572–584, 1998.

[57] R.M Haralick, K. ShanMugam, and I. Dinstein. Texture features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6), 1973.

[58] K. Haris, S.N. Efstratiadis, N. Maglaveras, and A.K. Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *Image Processing, IEEE Transactions on*, 7(12):1684–1699, Dec 1998.

[59] Berthold K. P. Horn and Brian G. Schunck. *Determining optical flow*, pages 389–407. Jones and Bartlett Publishers, Inc., USA, 1992.

[60] M. Irani, B. Rousso, and S. Peleg. Computing occluding and transparent motions. *IJCV*, 12:5–16, 1994.

[61] S. Jeannin and R. Jasinschi. Mpeg-7 visual motion descriptors. *Signal Processing: Image Communication Journal*, 16(1-2):59–85, Sept 2000.

[62] Qifa Ke and Takeo Kanade. A subspace approach to layer extraction. In *CVPR*, volume 1, pages 255–262, Los Alamitos, CA, USA, 2001.

[63] K. I. Kim, K. Jung, S. H. Park, and H. J. Kim. Support vector machines for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1542–1550, 2002.

[64] Vikrant Kobla, David S. Doermann, King-Ip Lin, and Christos Faloutsos. Compressed-domain video indexing techniques using DCT and motion vector information in MPEG video. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 200–211, 1997.

[65] Wolfgang Kohler. *Gestalt Psychology*. Liveright, 1938.

[66] Vladimir Kolmogorov and Ramin Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[67] I. Kompatsiaris, G. Manzaras, and M. G. Strintzis. Spatiotemporal segmentation and tracking of objects in color image sequences. In *ISCAS*, pages 709–713, May 2000.

[68] J. Konrad. *Image and Video Processing Handbook*, chapter Motion Detection And Estimation. Academic Press, 2000.

[69] I. Koprinska and S. Carrato. Temporal video segmentation : A survey. *Signal Processing : Image Communication.*, 16:451–460, 2001.

[70] M. Koprulu, N.K. Cicekli, and A. Yazici. Spatio-temporal querying in video databases. *Information Sciences*, 160(1-4):131–152, March 2004.

[71] Kishore Korimilli and Sudeep Sarkar. Motion segmentation based on perceptual organization of spatio-temporal volumes. *ICPR*, 03:3852, 2000.

[72] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, Feb 1956.

[73] M.P. Kumar, P.H.S Torr, and A. Zisserman. Objcut. In *CVPR'05*, 2005.

[74] J. Lee, J. Oh, and S. Hwang. Strg-indexing, spatio-temporal region graph indexing for large video databases. In *ACM SIGMOD*, pages 718–729, 2005.

[75] Steven Lehar. *The World in Your Head*. Lawrence Erlbaum Associates, 2003.

[76] Y. Li, J. Sun, and H-Y. Shum. Video object cut and paste. In *SIGGRAPH*, 2005.

[77] Joo-Hwee Lim. Learning visual keywords for content-based retrieval. *Multimedia Computing and Systems, 1999. IEEE International Conference on*, 2:169–173, Jul 1999.

[78] Jobst Löffler, Konstantin Biatov, Christian Eckes, and Joachim Köhler. Ifinder: an mpeg-7-based retrieval system for distributed multimedia content. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 431–435, New York, NY, USA, 2002. ACM.

[79] Wei-Ying Ma and B. S. Manjunath. Netra: A toolbox for navigating large image databases. *Multimedia Systems*, 7(3):184–198, 1999.

[80] M.K. Mandal, F. Idris, and S. Panchanathan. A critical evaluation of image and video indexing techniques in the compressed domain. *Image and Vision Computing Journal*, 17:513–529, May 1999.

[81] B.S. Manjunath, J.-R. Ohm, V.V. Vasudevan, and A. Yamada. Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):703–715, June 2001.

[82] D. Marr. *Vision*. W. H. Freeman and Co, 1982.

[83] A. Massmann, S. Posch, and G. Sagered. Relational descriptions in picture procressing. machine intelligence. In *International Conference on Image Processing (ICIP)*, volume 2, pages 207–210, 1997.

[84] J. Matas, R. Marik, and J.V. Kittler. On representation and matching of multicoloured objects. In *International Conference on Computer Vision*, pages 726–732, 1995.

[85] R. Megret and D. DeMenthon. A Survey of Spatio-Temporal Grouping Techniques. Technical Report LAMP-TR-094,CS-TR-4403,UMIACS-TR-2002-83,CAR-TR-979, University of Maryland, College Park, 2002.

[86] T. Meier and K.N. Ngan. Automatic segmentation of moving objects for automatic video object plane generation. *IEEE Transactions on Circuits and Systems for Video Technology*, 8:525–538, Sept 1998.

[87] O.J. Morris, M.J. Lee, and A.G. Constantinides. Graph theory for image analysis: An approach based on the shortest spanning tree. In *Inst. Elect. Eng., vol. 133*, pages 146–152, Galway, Ireland, April 1986.

[88] Fabrice Moscheni, Sushil Bhattacharjee, and Murat Kunt. Spatiotemporal segmentation based on region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):897–915, 1998.

[89] Information technology - multimedia content description interface-part 3: Visual, MPEG ISO/IEC 15938-3, ISO/IEC/JTC1/SC29/WG11/N4358, July 2001.

[90] Information technology - multimedia content description interface-part 4: Audio, MPEG ISO/IEC 15938-4, ISO/IEC/JTC1/SC29/WG11/N4224, July 2001.

[91] Information technology - multimedia content description interface - part 5 : Multimedia description, MPEG ISO/IEC 15938-5, ISO/IEC/JTC1/SC29, May 2003.

[92] Mpeg-7 visual part of experimentation model version 9.0, MPEG ISO/IEC 15938-5, ISO/IEC JTC1/SC29/WG11 N3914 2001.

[93] Ph. Mylonas, Th. Athanasiadis, and Y. Avrithis. Improving image analysis using a contextual approach. In *7th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2006)*, Seoul, Korea, April 2006.

[94] Frank Nack and Lynda Hardman. Towards a Syntax for Multimedia Semantics. Technical Report INS-R0204, CWI, April 2002.

[95] Frank Nack and Adam T. Lindsay. Everything you wanted to know about mpeg-7: Part 1. *IEEE MultiMedia*, 06(3):65–77, 1999.

[96] Michel Neuhaus, Kaspar Riesen, and Horst Bunke. Fast suboptimal algorithms for the computation of graph edit distance. In *Syntactical and Structural Pattern Recognition/Statistical Pattern Recognition (SSPR/SPR)*, pages 163–172, 2006.

[97] N.E O'Connor, E. Cooke, H. Le Borgne, M. Blighe, and T. Adamek. The acetoolbox: Low-level audiovisual feature extraction for retrieval and classification. In *Advances in Large Margin Classifiers*, 2006.

[98] J.M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, December 1995.

[99] Timo Ojala, Markus Aittola, and Esa Matinmikko. Empirical evaluation of mpeg-7 xm color descriptors in content-based retrieval of semantic image categories. *icpr*, 02:21021, 2002.

[100] N. Paragios and G. Tziritas. Adaptive detection and localization of moving objects in image sequences. *Signal Processing:Image Communications*, 14:277–296, Feb 1999.

[101] Dong Kwon Park, Yoon Seok Jeon, and Chee Sun Won. Efficient use of local edge histogram descriptor. In *MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia*, pages 51–54, New York, NY, USA, 2000. ACM.

[102] I. Patras, E. A. Hendricks, and B. S. Manjunath. Video segmentation by map labeling of watershed segments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):326–332, March 2001.

[103] K. Petridis, S. Bloehdorn, C. Saathoff, N. Simou, S. Dasiopoulou, V. Tzouvaras, S. Handschuh, Y. Avrithis, Y. Kompatsiaris, and S. Staab. Knowledge representation and semantic annotation of multimedia content. *Visual Image Signal Processing*, 153(3):255–262, June 2006.

[104] J.C Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages pp.61–74, 1999.

[105] Herwig Rehatschek, Werner Bailer, Helmut Neuschmied, Sandra Ober, and Horst Bischof. A tool supporting annotation and analysis of videos. In *Reconfigurations. Interdisciplinary Perspectives on Religion in a Post-Secular Society.*, Oct. 2007.

[106] Ricoh movie tool. http://www.ricoh.co.jp/src/multimedia/movietool/.

[107] Y. Rubner and C. Tomasi. Texture-based image retrieval without segmentation. In *International Conference on Computer Vision*, pages 1018–1024, 1999.

[108] Y. Rui, T. Huanga, and S. Chang. Image retrieval: current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation*, 4(10):39–62, April 1999.

[109] G. Salton, A. Wong, and C.S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[110] H.S. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):814–830, 1996.

[111] Dennis Shasha and Kaizhong Zhang. *Approximate Tree Pattern Matching*, pages 341–371. Oxford University Press, 1997.

[112] J. Shi and J. Malik. Motion segmentation and tracking using normalized cuts. In *Proceedings of the Sixth International Conference on Computer Vision (ICCV98)*, pages 1154–1160, Bombay, India, January 1998.

[113] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(2):888–905, August 2000.

[114] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, WA, USA, 1994.

[115] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[116] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, pages 1470–1477, 2003.

[117] John R. Smith and Ana B. Benitez. Conceptual modeling of audio-visual content. In *IEEE International Conference on Multimedia and Expo (II)*, pages 915–919, 2000.

[118] J.R. Smith and S.F. Chang. Visualseek, a fully automated content-based system. In *ACM Multimedia*, pages 87–98, 1996.

[119] S.M. Smith. Asset-2: real-time motion segmentation and shape tracking. *iccv*, 00, 1995.

[120] A. Smolic and J.R. Ohm. Robust global motion estimation using a simplified m-estimator approach. In *International Conference on Image Processing (ICIP)*, pages Vol I: 868–871, 2000.

[121] Fabrice Souvannavong, Bernard Mérialdo, and Benoit Huet. Region-based video content indexing and retrieval. In *CBMI*, 2005.

[122] C. Stiller and J. Konrad. Estimating motion in image sequences. *Signal Processing Magazine, IEEE*, July 1999.

[123] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, November 1991.

[124] T.F. Syeda-Mahmood, S. Srinivasan, A. Amir, D. Ponceleon, B. Blanchard, and D. Petkovic. Cuevideo: a system for cross-modal search and browse of video databases. In *International Conference on Computer Vision and Pattern Recognition*, pages II: 786–787, 2000.

[125] H. Tamura, T. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8:460–473, June 1978.

[126] Hai Tao, Harpreet S. Sawhney, and Rakesh Kumar. Object tracking with bayesian estimation of dynamic layer representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):75–89, 2002.

[127] A.M. Tekalp. *Digital Video Processing*. Prentice Hall, 1995.

[128] Raphaël Troncy. Integrating structure and semantics into audio-visual documents. In *2nd International Semantic Web Conference (ISWC'03)*, volume LNCS 2870, pages 566–581, Sanibel Island, Florida, USA, October 2003.

[129] Y. Tsaig and A. Averbuch. Automatic segmentation of moving objects in video sequences: A region labelling approach. *IEEE Trans. on Circuits and Systems for Video Technology*, 12(7):597–612, July 2002.

[130] C. Tsinaraki, P. Polydoros, and S. Christodoulakis. Integration of owl ontologies in mpeg-7 and tvanytime compliant semantic indexing. In A. Persson and J. Stirna, editors, *Advanced Information Systems Engineering, 16th International Conference*, volume 3084 of *Lecture Notes in Computer Science*, pages 398–413. Springer, 2004.

[131] Chrisa Tsinaraki, Panagiotis Polydoros, and Stavros Christodoulakis. Interoperability support for ontology-based video retrieval applications. In *CIVR*, volume 3115 of *Lecture Notes in Computer Science*, pages 582–591. Springer, 2004.

[132] O. Utsumi, K. Miura, I. Ide, S. Sakai, and H. Tanaka. An object detection method for describing soccer games from video. In *ICME '02, Proceedings International Conference on Multimedia and Expo*, volume 1, Aug 2005.

[133] W.R. Uttal. *Computational Modeling Of Vision*. CRC Press, 1999.

[134] A. Vailaya and A.K. Jain. Image retrieval using color and shape. In *Asian Conference on Computer Vision*, pages II: 529–533, 1995.

[135] Nuno Vasconcelos and Andrew Lippman. Empirical bayesian motion segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):217–221, 2001.

[136] Shankar Vembu, Malte Kiesel, Michael Sintek, and Stephan Baumann. Towards bridging the semantic gap in multimedia annotation and retrieval. In *Proceedings of the First International Workshop on Semantic Web Annotations for Multimedia*, 2006.

[137] C Venters and M. Cooper. A Review of Content-Based Image Retrieval Systems. Technical report, 2000.

[138] L. Vincent and P. Soille. Watersheds in digital space: an efficient algorithm based on immersion simulations. *IEEE Trans. PAMI*, 13(6):583–598, 1991.

[139] V.V. Vinod. Activity based video shot retrieval and ranking. In *International Conference on Pattern Recognition*, pages Vol I: 682–684, 1998.

[140] D. Wang. Unsupervised video segmentation based on watersheds amd temporal tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(6):539–546, Sept. 1996.

[141] J.Y.A. Wang and E.H. Adelson. Representing moving images with layers. *Image Processing, IEEE Transactions on*, 3(5):625–638, Sep 1994.

[142] J.Z. Wang, J. Li, and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, September 2001.

[143] M. Wertheimer. Principles of perceptual organization. *Readings in Perception*, 6:115–135, 1958.

[144] H. Xu, A. Younis, and M.R. Kabuka. Automatic moving object extraction for content-based applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(6):796–812, June 2004.

[145] S.X. Yu and J. Shi. Multiclass spectral clustering. In *ICCV03*, pages 313–319, 2003.

[146] C. Yuan, Y. F. Ma, and H. J. Zhang. A graph theoritic approach to video object segmentation in 2d+t space. Technical report, MSR, 2003.

[147] Z. Zaharia and F. Preteux. Parametric motion models for video content description within the mpeg-7 framework. *SPIE*, may 2001.

[148] W. Zeng, W. Gao, and D. Zhao. Description of cost 211 analysis model. In *Cost 211*, july 1998.

[149] N. Zlatoff, B. Tellez, and A. Baskurt. Vision gestalt et connaissances : une approche générique à l'interprétation d'images. In *Compression et Représentation des Signaux Audiovisuels*, 2004.