



Semantic Video Model for Description, Detection and Retrieval of Visual Events

Ahmed Azough

► To cite this version:

Ahmed Azough. Semantic Video Model for Description, Detection and Retrieval of Visual Events. Artificial Intelligence [cs.AI]. Université Claude Bernard - Lyon I, 2010. English. NNT : 2010LYO10055 . tel-02881033

HAL Id: tel-02881033

<https://theses.hal.science/tel-02881033>

Submitted on 25 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE DE L'UNIVERSITE DE LYON

Délivrée par

**L'UNIVERSITE CLAUDE BARNARD LYON 1
ECOLE DOCTORALE 335**

INFORMATIQUE ET INFORMATION POUR LA SOCIETE

DIPLOME DE DOCTORAT

(arrêté du 7 août 2006)

soutenue publiquement le 08 Mai 2010

par

M AZOUGH Ahmed

TITRE

**MODELE SEMANTIQUE DE LA VIDEO POUR LA DESCRIPTION, LA
DETECTION, ET LA RECHERCHE DES EVENEMENTS VISUELS**

Directeurs de thèse : Pr HACID Mohand-Said

Co-encadrants : Dr. DELTEIL Alexandre, Dr. DE MARCHI Fabien

JURY :

Pr. Hervé MARTIN (Rapporteur)

Pr. Franck NACK (Rapporteur)

Pr. Nicolas SPYRATOS (Examineur)

Pr. Djamel Abdelkader ZIGHED (Examineur)

Pr HACID Mohand-Said (Directeur de thèse)

Dr. Alexandre DELTEIL (Co-encadrant de thèse)

Dr. Fabien DE MARCHI (Co-encadrant de thèse)

PHD THESIS

SEMANTIC VIDEO MODEL FOR
DESCRIPTION, DETECTION AND
RETRIEVAL OF VISUAL EVENTS

Ahmed AZOUGH

Contents

Contents	i
Introduction	3
0.1 Context	3
0.2 Contributions	5
0.3 Content	6
1 An Overview of Video Semantic Indexing Techniques	11
1.1 Introduction	13
1.2 High Level Video Analysis	15
1.3 Semantic Description Languages and Standards	23
1.4 Data-Models and query languages for video Database Management Systems	30
1.5 Discussion and Conclusion	33
2 A Semantic Language for Description and Detection of Visual Events	35
2.1 Introduction	37
2.2 Contribution	37
2.3 Modeling Visual Events	40
2.4 Video Guided monitoring of behavior	42
2.5 MPEG-7 Annotation Validation	48
2.6 Conclusion	55
3 Uncertainty Handling in Semantic Video Retrieval Using Fuzzy Conceptual Graphs	57
3.1 Introduction	59
3.2 Contribution	59
3.3 Fuzzy Conceptual Modeling of Video Data	59
3.4 Graph Matching	76
3.5 Conclusion	83
4 A Database Approach for Expressive Modeling and Efficient Querying of Visual Information	85
4.1 Introduction	87
4.2 Contribution	87
4.3 Basic Definitions	89
4.4 Datalog-like Data Modeling	93
4.5 F-Logic like Data Modeling	100
4.6 Conclusion	112

5	Prototype	113
5.1	Introduction	115
5.2	Object Detectors	115
5.3	Video Annotator Tool	118
5.4	Model Editor	120
5.5	Event Detector	127
5.6	Representation Syntax	127
5.7	Conclusion	130
6	Conclusion and Future Works	131
	Conclusion and Future Works	131
6.1	Summary and Contributions	131
6.2	Future Work	133
6.3	Academic and Industrial Results	134
	Bibliographie	147

List of Figures

1	Example of the description of an image following three abstraction levels.	5
2	Integrated framework for semantic analysis and retrieval of video documents following the three abstraction levels.	7
3	The contents of the thesis organized according to different abstraction levels. . . .	8
1.1	Different semantic indexing approaches of images and videos chronologically classified.	14
1.2	Video scene description in MPEG-7	25
1.3	MPEG-7 description code example	25
1.4	VideoSegment MPEG-7 descriptor used with different semantics.	26
1.5	VideoSegment MPEG-7 descriptor used with different semantics using the DAVP Profile.	27
1.6	Example of a segmentation of a video segment.	31
1.7	Example of a stratification of a video segment.	31
1.8	Example of a temporal cohesion of a video segment.	32
2.1	Positioning chapter contribution within the whole framework.	38
2.2	General Semantic structure of video contents	39
2.3	Example of an event model: Deviated shot on goal from outside the penalty zone.	41
2.4	Hierarchic description of complex objects and events.	42
2.5	Example of hierarchic description of a complex relation	43
2.6	Example of hierarchic representation of a complex visual event	43
2.7	Detecting the event "Goal" in a soccer video	44
2.8	Monitoring protocol construction process.	45
2.9	Video guided Monitoring process.	46
2.10	Matching object instances of a video frame to an event state	47
2.11	Car Theft Monitoring Protocol	51
2.12	Car theft detection in real time video stream	51
2.13	Semantic validation framework architecture	52
2.14	Semantic Validation of a MPEG-7 Description	53
2.15	Example of MPEG-7 video segment description	54
3.1	Different kinds and resources of uncertainty in event retrieval.	60
3.2	Positioning the chapter contribution within the whole framework. It consists of an extension of the event detector with advanced reasoning capabilities.	61
3.3	event graph	65
3.4	Video Graph	67
3.5	Video Sequence	68
3.6	Fuzzy definition of Allen's temporal relations	71
3.7	Allen's temporal relations Trellis	72

3.8	An example of a fuzzy matching of a temporal relation.	73
3.9	Expressing RCC8 relation by using Allen's relations	74
3.10	RCC8 spatial relations Trellis	75
3.11	An example of a fuzzy matching of a spatial relation.	76
3.12	Degree of matching two attributes a and a' regarding the difference of values . . .	77
3.13	Changing from a Concept-Relation view to Arcs view of a conceptual graphs. . . .	79
3.14	The algorithm of heuristic matching performed in many levels.	80
4.1	Positioning the chapter contribution within the whole framework.	88
4.2	An action freekick temporally located according to different frames of reference : Date&Time, soccer time-line and video frames.	90
4.3	Two different references (A soccer field and the camera viewport) and the spatial location of the same object (the ball) in the two references	91
4.4	Spatiotemporal modeling for an event and its composing subevents and subob- jects according to multiple frames of reference.	96
4.5	The proposed data model for representing video content.	103
5.1	Positioning the chapter contribution within the whole framework.	115
5.2	A screen shot of the real-time detection of soccer playefield	117
5.3	Real time multi-players tracking	119
5.4	Evolution of the real-time multi-player tracker	119
5.5	Loading the frames of the video to be annotated to the annotation tool.	120
5.6	Annotating a single frame of the video.	121
5.7	Exporting descriptions of a frame to the next frame in order to facilitate its anno- tation.	122
5.8	Modification of existing descriptions.	123
5.9	Building a finite state machine that represent the temporal composition of an event. .	124
5.10	Constructing the conceptual graph associated to a selected state of the event model. Selecting the type of concepts from a list.	125
5.11	Semi automatic selection of relations between concepts.	126
6.1	Future framework for cross-media semantic analysis and retrieval of video docu- ments.	133

To my family...

Acknowledgment

This work could not be achieved without continuous and strong support of individuals and entities that I would to warmly thank here.

I would like to express my gratitude to my advisor Pr Mohand-Said Hacid who was my tutor during all my high graduated studies in France. His availability, guidance and patience had great impact on my academic and professional career. From him I learned excellence, rigor, creativity, endurance and critical thinking, major values that made of him an imminent scientist in the world. He is and will be my role model in research and innovation world.

I would like to thank my co-advisors Dr Alexandre Delteil and Dr Fabien De-Marchi for their full support during my studies. I am very fortunate to have had the opportunity to work under their supervision. In particular, the benevolence and the guidance of Dr Alexandre Delteil, with which I shared the same office during three years, were very crucial. Without his valuable help, his personal investment, his trust, and his expertise, the work on that multi-disciplinary subject would have literally been impossible.

I would like also to express my gratitude to Pr. Hervé MARTIN, Pr. Franck NACK, Pr. Nicolas SPYRATOS, and Pr. Djamel Abdelkader ZIGHED for serving on my Thesis committee and for their instructive reviews and comments on this work.

On a personal level, and first of all, I would thank my parents Pr. Brahim AZOUGH and Pr. Hayat HAMIDI. I could never reward them enough for my entire being. They were the first to teach me my first words, and it is them who created in me the dream of reaching the highest degrees. I dedicate to them all the success I've had in my life. It is for me a honour, a privilege, but also a responsibility to be their son.

If I were to name a person who deserves the most thanks, it would be my wife Fatima-Zahra KAGHAT. She lived with me this PhD minute by minute and second by second with tireless encouragement. She shared with me the successes and failures, and was my source of happiness and inspiration throughout this period. Behind every great man there's a great woman, and behind this PhD there is Fatima-Zahra. I am indebted to my wife until she fulfils all her dreams.

I can never forget the support of my parents-in-law Nour ed-dine KAGHAT and Najia BERRADA , but also my brothers, my sisters, my big family and my friends. Their unfailing trust and truthful kindness helped me to overcome the hardest difficulties. They have always believed in me, and I could never reward them enough for their continuing prayers and encouragement.

Last but not least, I would like to thank Pr Omar El Beqqali, who was my first mentor in computer sciences and the one who opened to me the way of advanced studies in computer sciences. I would also thank the Outaghzout, Bouhamdan, Ait Bouahia, El-Mousati, El-Arafa, and Ait-lhoussaine families. Without their hospitality and their valuable help, fulfilling my dreams far from my country and my small family could never become a reality.

Introduction

Multimedia and more specially visual information is gaining a lot of importance in daily use. The size and the richness of video collections is in exponential growth. However, storing only the low level features of video resources does not allow users to access the right information in an efficient way. Resources need to be semantically analyzed so that answers to queries can be reliably and quickly computed.

Most videos are published as raw data with poor semantic information. Even for structured data, the information structure is guided by supply and not by demand. While most multimedia resource suppliers omit to efficiently annotate and structure their documents, the unsuitable existing retrieval techniques, often keyword-based, do not enable for an efficient access [88].

0.1 Context

Many approaches have been introduced during the last 50 years in order to efficiently explore and analyze video contents and enable the users to easily and quickly access the right information.

In order access correctly the right information in video documents and perform semantic based exploration of its content, three major tasks should, in general, be accomplished. Those tasks are detection, description and retrieval.

We mean by detection the process of augmenting the binary content by extracting higher conceptual annotations that correspond to semantics appearing or occurring on the video, namely detecting objects and events on video stream. By description we mean all the effort made to correctly associate annotations to a specific document or to a part of it, but also to semantically gather those annotation together. By retrieval we mean the indexing and querying techniques used to explore video databases based on their content.

While humans tend to interpret images and videos using high-level concepts such as objects, events and relations, only low-level features such as color, texture and shape can be automatically extracted in a reliable way [161]. This discrepancy between, on one hand the limited descriptive power of current automatic video analysis generating effectively only low-level features, and on the other side the richness of user semantics and interpretations, is referred to as the “semantic gap” [163]. In general, links between the high-level concepts and the low-level features are very hard to establish [64], and today’s vision systems are still far below human perception system efficiency. One of the promising proposed solutions consists of building many intermediate semantic levels in order to fill the semantic gap and link the raw data with the user semantic concepts.

In [69], Eakins distinguished three levels of image and video analysis and retrieval:

- 1st level: Retrieval by low level features such as color, texture, shape, motion. Typical query is query by example, "find images/videos like this one".
- 2nd level: Retrieval by objects of specific types. Example of this queries: "find images/videos of soccer playfield".
- 3rd level: Retrieval by situations, events and cognitive activities requiring high level spatiotemporal and logical reasoning. Examples of such queries are: "find images/videos showing a penalty" or also "find images/videos of a joyful public".

An example of image description following the three levels is shown figure 1.

In our thesis we focus on retrieval of video content at the 3rd level. For this aim, we introduce three main definitions that shows our comprehension of the content at the highest semantic level.

Situation: a situation can be defined as a configuration of objects satisfying some spatial or logical constraints that remain the same during a time interval or that are valid only at a point in time. The term "state" is sometimes used to refer to a situation. An example of query of a situation is: "find soccer videos showing an offside situation".

Event: an event can be defined as a set of modifications between successive situations that form an evolution from an initial situation into a final situation over a time interval. An event (defined extensionally as a set of event occurrences) can be defined intentionally as a set of situations related by temporal relations representing the possible configurations of state sequence forming an occurrence of the event. An event thus involves a set of objects satisfying some spatiotemporal or logical constraints, whose interactions during time and space fulfill a well defined finality. The term "activity" is sometimes used to refer to an event. Examples of queries of some events are: "find videos showing a plane take-off" or "find videos showing a volcano in eruption".

Cognitive activity: a Cognitive activity can be defined as an interpretation granted by humans to a specific event or situation. Cognitive activities are related to the culture, the context, and the environment where these events have occurred. For instance, some people interpret dancing (event) as happiness (Cognitive activity), while others interpret it as religious prayer (Cognitive activity). Examples of queries of Cognitive activities are: "find videos showing a team celebrating a victory" or "find videos of a joyful public".

One of the principle issues in the research community is to narrow the semantic gap and to develop real-world descriptions and interactions with multimedia documents. The aim is to generate annotations increasing the binary content with semantic descriptions. This requires the conception of algorithms capable of detecting meaningful objects and events happening within these documents.

Using high level description and reasoning formalisms seems to be the best way to combine existing annotations in order to infer new information and detect complex objects and events. Their detection has to be performed as part of a higher-level semantic analysis that includes the identification of their constituent spatial and/or temporal parts but also the use of contextual knowledge already derived about the neighboring segments. In addition, new indexing and query languages should be provided in order to efficiently and reliably access to the right information with respect to the complex data structure of multimedia documents.

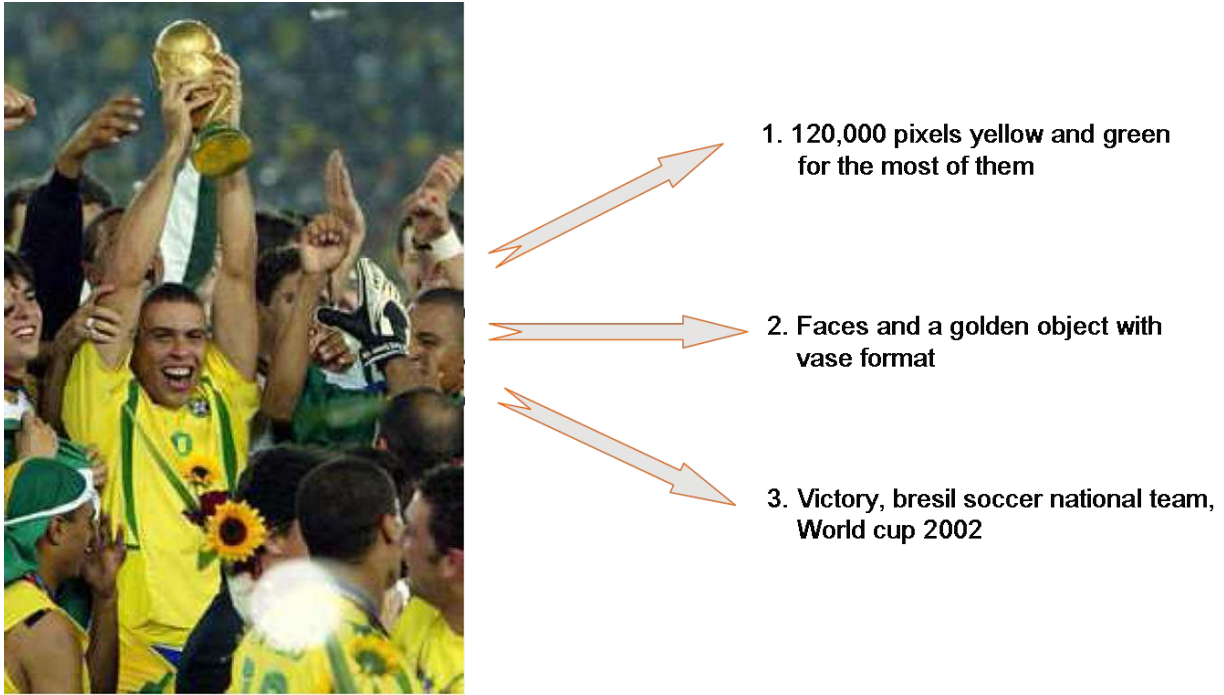


Figure 1: Example of the description of an image following three abstraction levels.

Uncertainty and imprecision is also one of the major challenges related to the semantic gap in multimedia data description and retrieval. It is due not only to errors and imprecisions in content classifications but also to the extended range of user queries and navigation habits. Although this issue is largely studied by the community of Information Retrieval, the integration of techniques resolving the uncertainty problem for event detection and enabling for fuzzy retrieval is still lacking.

0.2 Contributions

The global objective of this thesis is the design of an integrated framework for semantic analysis and retrieval of semantic contents within video documents following the three abstraction levels. The architecture of the framework is shown figure 2.

The first contribution of the work is to design a representation language for spatiotemporal description and detection of visual events. This representation language is built based on two formalisms: Finite State Machines and Conceptual Graphs, to represent respectively temporal and spatial structures of visual events. An event is modeled by a Finite State Machine where each state is associated to a situation occurring within this event. Transitions describe the temporal structure of the event, i.e. the order of occurrence of the situations within the event. Each situation is modeled by a Conceptual Graph that describes its spatial composition. Based on its hierarchical policy for describing high level event, this formalism enables to describe high level semantic concepts using low level concepts and spatiotemporal constraints. These descriptions can therefore be used as models for high-level concepts detection. In addition to be a semantic description language, this representation language is also a tentative to narrow the semantic gap between the object level (2^{nd} abstraction level) and the event level (3^{rd} abstraction level).

Algorithms combining automata acceptance and graph projection are then proposed in order to use the event models produced using this representation language as queries to automatically recognize events in videos. In figure 2, this contribution is modeled by the hexagonal box named "Event Detector", that states between the object base and event base. Using the proposed representation language, we also propose an approach for semantic validation of event descriptions within MPEG-7 documents.

The second contribution is the extension of the previous representation language in order to provide a new variant of fuzzy conceptual graphs more suitable to video content description and retrieval. Emphasis is put on the uncertainty measurement which is inherent in the multimedia content analysis. The approach defines two types of graphs; an event graph that describes a common event where uncertainty is related to human perception in defining relations, and a Video Graph where uncertainty is due to errors and imprecision related to automatic detection algorithms. Moreover, new fuzzy variants of temporal and spatial relations are introduced to reason in a fuzzy manner about relationships between objects and intervals within a video segment. Then, similarity measures are defined to assess the degree of match between the components of video and event graphs. A two-level graph matching is developed to calculate total matching coefficient between video and event graphs. In figure 2, this contribution is also modeled by the hexagonal box named "Event Detector" since it is an extension of the Event Detector proposed in the first contribution.

The third contribution is to propose a hierarchical, hybrid and semistructured data model for representing video data. Based on this model, a declarative, rule-based, constraint query language is presented. The data model allows to associate, in multiple granularity, a segment of space or time to a set of objects, events and relations. In figure 2, this contribution is modeled by the hexagonal box named "Semantic Query Engine". The query language can be used to infer and to retrieve spatial, temporal or semantic relationships from information represented in the model and to intentionally specify relationships among objects and events. Two methods are proposed to model and to query data information: a Datalog-like method and an Object-oriented method. We introduce the concept of temporal and spatial frames of reference that enables for simultaneously locating objects, events, and relations according to different spatio-temporal environments in the real world.

0.3 Content

The content of this thesis is organized as follows (see figure 3). The work contains 6 components:

- The first chapter presents a state of the art of techniques for the extraction, the representation and the indexing and retrieval of semantic information describing video contents. It first reviews how high-level objects and events can be extracted by video analyses techniques. It then presents the description languages that can be used for describing a video with high-level concepts: MPEG-7, logical languages and event models are reviewed. Then it presents the data models and query languages used in the database community for managing video databases. This chapter ends by providing a list of features required for an expressive event model.
- The second chapter presents an expressive language for description and detection of events in video streams. The proposed language combines automata with semantic and spatial constraints and can be used to represent precisely the definition of an event. Given a set of

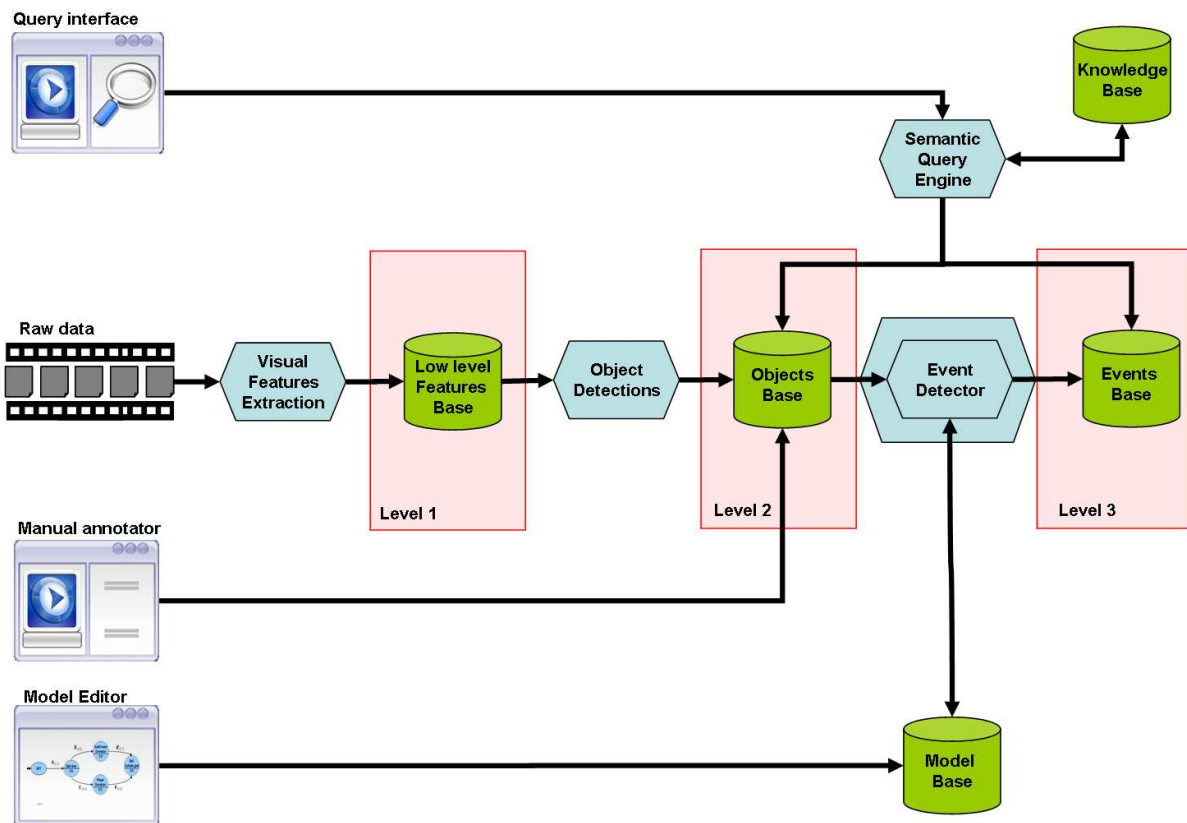


Figure 2: Integrated framework for semantic analysis and retrieval of video documents following the three abstraction levels

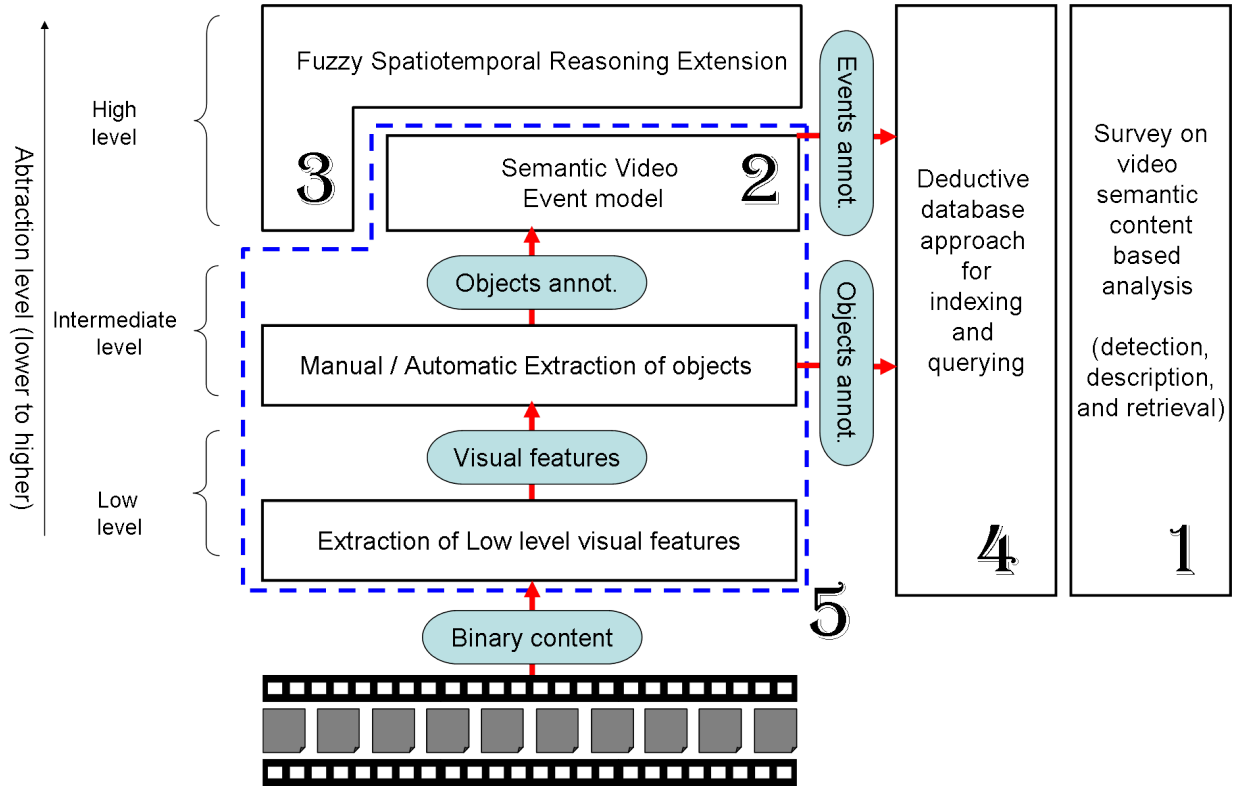


Figure 3: The contents of the thesis organized according to different abstraction levels.

automatic or manual description of objects in a video document, such a definition can also be used to extract high-level event or object annotations and to detect event occurrences. This chapter describes the features and the theoretical properties of the language while the implementation details are provided in Chapter 5.

- The third chapter presents an extension of this representation language in order to add uncertainty for representing and detecting events in videos. Uncertainty is handled by using fuzzy conceptual graphs as states in the automata representing an event.
- The fourth chapter proposes an indexing data model and query language for managing video databases. This data model enables to index events as well as objects that come from automatic or manual description of video documents, but also events that may be detected by the models presented in the chapters 2 and 3. The proposed data model combines spatio-temporal and object relational constraints and the query language is based on Datalog and F-Logic, and enables high-level and spatio-temporal reasoning.
- The fifth chapter describes the implementation details of the event model for description and detection presented in Chapter 2. This model lays on two other processes that were also presented in the same chapter that are:
 - The analysis of video resources in order to extract low level features such as color, texture, shape, etc. These image and video analysis algorithms can be considered as the preliminary step in the process of extracting semantic information.
 - The annotation of objects appearing in videos. Annotations can be provided manually or generated automatically. The software that has been implemented for manual an-

notation of objects appearing in soccer videos is presented. In addition, classification algorithms implemented in order to detect objects, such as players and playfield, in soccer videos are described. Those algorithms were inspired by existing methods for object detection and classification.

The user will benefit from the ability of the technologies to understand more about them to make search more relevant and to make information retrieval more relevant. But it will take time and probably more than people anticipate.

Steve Berkowitz

1

An Overview of Video Semantic Indexing Techniques

▷ *Produced previously by analog devices, video documents gained more importance in daily use due to digitalization. In many domains, such as medicine, news, sport or video surveillance, video is a main resource of information. Nevertheless, multiplication of resources on media collections, and the large diversity of research interests make exploitation of video clips hard and the access to them non trivial.* ◁

Contents of the chapter

1.1	Introduction	13
1.2	High Level Video Analysis	15
1.2.1	Spatio-temporal Segmentation	15
1.2.2	Objects Detection	18
1.2.3	Events Detection and Behavior understanding	20
1.2.4	Discussion	22
1.3	Semantic Description Languages and Standards	23
1.3.1	MPEG-7	24
1.3.2	Discussion	26
1.3.3	Logic-based formalisms	27
1.3.4	Event Models	29
1.4	Data-Models and query languages for video Database Management Systems . .	30
1.4.1	Annotation-based Video Models	30
1.4.2	Object-Relational Data-Models	31
1.5	Discussion and Conclusion	33

1.1 Introduction

The work done in the field of the semantic gap reduction is tremendous and can not be summarized in a single chapter. The proposed approaches can be classified in different ways and following different points of view. One can adopt different classifications by comparing application domains, solutions from distinct research communities, devices used to display documents, off-line or real time processing, etc.

The first efforts for semantic indexing of visual documents resources focused on the Text Based Image Retrieval (TBIR) systems that can be dated back to 1970s [93]. Those systems consist of retrieving images and videos using textual annotations which were manually provided. The work consists then of using a standard Database Management System (DBMS) to perform image indexing and search [119]. Text queries, on the other hand, are more intuitive and natural for users to specify their information needs. However, text based annotation of visual documents faces many challenges. The process of manually annotating resources is tremendous and time and effort consuming. In addition, manual annotations are often inaccurate due to subjectivity of the human perception. In fact, a picture can mean different things to different people. There are many ways to say the same thing, and mistakes are often appearing due to spelling errors. Many researches are still looking for improving TBIR using techniques such as Relevance Feedback [170]. Some works has extended TBIR systems to Web Based Image Retrieval (WBIR) systems that aim to retrieve images and videos within Web pages [58]. In these approaches, images and videos are represented using filename, caption, surrounding text, and text in the HTML document [90]. However, even if the textual information contained in pages can describe the semantic of images these pages contain, texts that are irrelevant to the images can also be found. The problem of identifying relevance between texts and images in the same page is still open [126].

In order to overcome drawbacks of text based retrieval, Content Based Image Retrieval (CBIR) systems were introduced during the 1980s to analyze visual information. CBIR consists of the analysis of visual documents based on their binary contents that are pixels. A pioneering work was published by Chang in 1984 [27], in which the author presented a picture indexing and abstraction approach for pictorial database retrieval. Based on such an approach, some commercial and experimental systems have been developed, such as QBIC [50], Photobook [110], Virage [53], VisualSEEK [129], Netra [97], SIMPLIcity [158], etc. CBIR systems consist of using Low-level features such as color, texture, shape, and motion in case of videos in order to characterize visual documents and provide content based access to them [114, 120].

In this chapter, we tried to address the semantic analysis of video documents starting by the earlier approaches that are text-based and Web-based. Then,

In what follows, we describe how content-based approaches have been proposed for different abstraction levels. Chronological classification of the different approaches we mention in this state of art is shown figure 1.1.

In our work we focus on video analysis and retrieval. However we cite also important works on image retrieval since the two domains are strongly connected as videos are built from still single images. Although motion aspect in videos helps for tracking and then detecting objects, the temporal aspect make this detection harder and more complex as advanced techniques to combine spatial and temporal characteristics are needed. This makes object detection in video documents an error prone process that requires taking into consideration uncertainty in low level feature extraction.

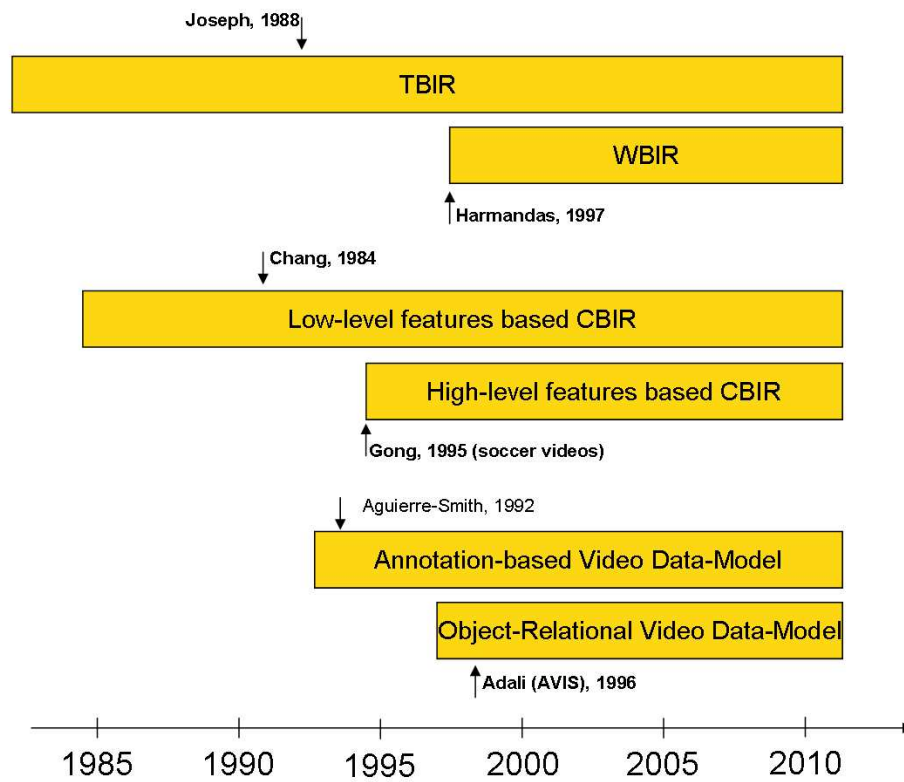


Figure 1.1: Different semantic indexing approaches of images and videos chronologically classified.

This chapter presents a state of the art in three research domains: video analysis, video descriptions standards and languages and video database management systems. The focus of this state of the art is on the semantic level; low level methods and descriptors are mentioned only when they are used as preliminary step for obtaining or using semantic descriptions.

The first section describes the different video analysis methods for generating high level descriptions. Video analysis methods can be distinguished depending on the level of the descriptions they produced. Usually three different abstraction levels are considered: feature/low level, object/intermediate level, and Event/semantic/high level. For the low level, (micro) temporal segmentation and shot boundary detection are presented, for the intermediate level object detection and recognition and for the high level event detection.

The second section is concerned about presenting some important standards and languages used to represent video semantic descriptions. MPEG7, the most popular XML-Based standard for multimedia Description, is first reviewed, then logic-based languages for representing semantic information in video documents are described.

The third section presents an overview of some data-models and query languages used in video database management systems or in video search engines. The segmentation based data-models that indexes videos based on simple schemas and spatio-temporal location are presented first, then those providing advanced representation schemas of video contents using objects, relations, and/or events are described. Finally, we conclude by discussing some requirements not yet fulfilled by the current semantic video models.

1.2 High Level Video Analysis

1.2.1 Spatio-temporal Segmentation

The earlier task for performing a high-level analysis of video is segmentation. Segmentation is the process of partitioning the video content into many logic units in order to simplify its analysis. Image segmentation results in a set of homogeneous regions represented by the contour of their surfaces. Pixels in the same region have the same properties such as color, intensity, or texture. Such a segmentation should be performed before object detection for better understanding of content semantics. Segmentation is based on one or several features (e.g. motion, color, edges, texture) and can be done, regardless of the feature it is based on, by one of the following techniques:

Clustering Methods. One of the techniques consists of choosing randomly k cluster centers, then each pixel is assigned to the closest cluster center, and finally cluster centers are calculated [29]. The operation is repeated until convergence is obtained.

Histogram-Based Methods. In this technique, a histogram is computed from all of the pixels in the image, and the peaks and valleys in the histogram are used to locate the clusters in the image [10].

Region-Growing Methods A region starts with a single pixel. Adjacent pixels are recursively examined and added to the region if they are sufficiently similar to the region. If a pixel is too dissimilar to the current region, it is used to start a new region [98].

Graph Partitioning Methods. In these methods each pixel is a node in the graph, and an edge is formed between every pair of pixels. The weight of an edge is a measure of the similarity between the adjacent pixels. Segmentation is done by removing edges between nodes[140].

Multi-scale Segmentation. Segmentation criteria can be arbitrarily complex and may take into account global as well as local criteria [76].

Semi-automatic Segmentation. The user outlines the region of interest with the mouse and algorithms are applied so that the path that best fits the edge of the image is shown [104].

1.2.1.1 Color and Texture based Segmentation

Segmenting images and video key frames based on color and texture is one of the main preliminary steps for image indexing. It enables for simplifying the processing of images by using more meaningful representation and high level concepts.

Automatically segmenting images is a challenging task. Two features should be taken into consideration when segmenting images [94]. The first feature is color. Many proposed algorithms, essentially on direct clustering methods in color space [32], have been proposed for image segmentation. However, such approaches are efficient only when dealing with homogeneous color regions and color based retrieval systems [61].

The second feature, and the most challenging, is texture [38]. In fact real world images are rich of texture regions with often non homogeneous color. The deal is to correctly segment the regions based on texture homogeneity besides the color one.

The following paragraphs describe important methods used in segmenting images based on both color and texture features.

'JSEG' segmentation. Used in many systems [73, 49], the idea in JSEG¹ segmentation, a region growing method, is to separate the segmentation process into two independent tasks [38]. The first stage, color quantization, aims to quantize colors into classes without considering the spatial distribution. A class-map of the image is created by replacing each pixel with its corresponding class label. The second stage, spatial segmentation, is then performed on the class map, considered as a particular texture composition, using a criterion for "good" segmentation. The segmentation results into a J-image where high and low values are possible region boundaries. A region growing method is then used to define the image segments. Tracking schemes are embedded into region growing when applying this method on video segmentation. JSEG seems the most adapted to provide color-texture homogeneous regions.

Blobworld segmentation. Widely used in different segmentation algorithms [49, 127], Blobworld segmentation, a clustering method, consists in clustering pixels by combining color, texture and position feature spaces [26]. The distribution of color texture and position features is modeled with mixture of Gaussians. The parameters of the model are estimated using an expectation maximization (EM) algorithm [21]. Then, the resulting membership grouping pixels and clusters provides a segmentation of the image. JSEG and Blobworld seems to be the most adapted to provide color-texture homogeneous regions.

¹JSEG: J Segmentation

K-means clustering. K means, a clustering method, are also widely used by some systems to create segmentations that are the most adapted to their context [130, 86]. In [158] an image is first segmented into small blocks. Color and texture features are then extracted from the different blocks. Finally a k-means clustering is applied to assign the feature vectors to several classes. The regions are then created by gathering together the blocks in a same class.

1.2.1.2 Motion-based Segmentation

Motion segmentation refers to the process of detecting regions corresponding to moving objects within an image sequence. Motion segmentation is interesting since it is essential for later processes such as tracking and scene understanding [59]. Several approaches are proposed for motion segmentation. They are mainly based on temporal and spatial information processing: The main approaches can be outlined as the following:

Background subtraction. It is a widely used method for motion segmentation. It is efficient for fixed cameras where background is usually static. Objects are detected by subtracting pixel-by-pixel the current image from the reference background. Even simple to use, it is sensitive to noise and not efficient for changes such as lighting or other environmental conditions. A good background model should allow for reducing the effect of such changes [57, 100, 136].

Temporal differencing. It consists of calculating the difference, pixel-by-pixel, of consecutive two or three frames in order to extract moving regions of the image. Moving sections are generally clustered into motion regions based on connected component analysis [92]. Temporal differencing is very efficient for dynamic backgrounds and moving cameras. However, it can result into errors especially when dealing with pixels inside moving objects, which are generally classified as static [60].

Optical flow. It consists in using flow vectors of moving objects over time to detect moving regions in an image sequence [16]. A typical technique (e. g. [103]) consists in using contour based tracking algorithm to extract articulated objects. Optical-flow-based methods are quite robust for camera motion. However, they are noise sensitive and computationally complex. Their application to real time video streams requires specialized hardware [60].

1.2.1.3 Discussion

Features-based CBIR has enabled for reducing the human effort for annotating multimedia resources, and has improved the efficiency of image and video retrieval especially when dealing with basic image features.

However, as stated in Gestalt theory by the Totality Principle ², each component should be considered as part of a system. In fact, a component in an integrity takes specific properties of its place and function inside the whole object. Therefore, with respect to this theory, an image (or a video) cannot be reduced to the sum of the perceived low level features.

Understanding a behavior occurring in a video or a situation in an image requires not only to analyze visual features but more importantly, to have a synthetic vision and perceive the context

²<http://gestalttheory.net/>

of each feature. This is confirmed by experiential evidence that has shown that the majority of queries in image and video retrieval concerns the function (object) or the behavior (event) of what is shown rather than its appearance. It is then important to develop effective algorithms for detecting objects and event in video documents.

1.2.2 Objects Detection

The majority of methods aiming at object detection within images and videos are based on the use of Machine Learning algorithms [64, 33, 5, 70]. Machine Learning algorithms are based on trained models rather than on explicitly declared information. Statistical analysis and machine learning techniques are used to create models from data and have demonstrated good empirical results. They are quite fast, robust and easily extendable for different applications. Two major classes of machine learning are used in reducing the semantic gap. The first class, called supervised learning, is concerned with classifying an input object based on information about objects in a training set. Whereas the second class, called unsupervised learning, is concerned with describing how the input objects are organized [143] without using any training set. Approaches combining the two learning techniques can be found in the literature of video mining and analysis.

1.2.2.1 Supervised learning

Supervised learning, such as support vector machine (SVM) [54, 128, 152], Bayesian classifier [155], and Neural network, are often used to learn high-level concepts from low-level image features.

SVM (Support Vector Machine). SVM is a method used for classification or regression in image retrieval systems [45, 83]. Originally designed for binary classification, it consists of creating a hyper-plane separating the two categories of data. This should be done while maximizing the margin between the hyper-plane and the nearest data point of each class. The training samples that are closest to the hyper-plane are called 'Support vectors'. A SVM has to be trained for each concept to be retrieved in image. In [128], 23 concepts should be retrieved and a SVM model is trained for each of them. In the testing stage, regions are compared with all models, and then associated to the concept giving the highest positive result.

Bayesian Network. A Bayesian Network (*BN*) is a graphical model for probability relationships among a set of variables (features). The Bayesian network structure *S* is a directed acyclic graph (*DAG*) and the nodes in *S* are in one-to-one correspondence with the features *X*. The arcs represent casual influences among the features while the lack of possible arcs in *S* encodes conditional independencies. Moreover, a feature (*node*) is conditionally independent from its non-descendants given its parents (X_1 is conditionally independent from X_2 given X_3 if $P(X_1|X_2, X_3) = P(X_1|X_3)$ for all possible values of X_1, X_2, X_3) [112].

In [70, 5, 142], Bayesian network are used in capturing high level semantics in indoor/outdoor image classification.

Neural network. Neural Networks are also interesting techniques used for supervised concept learning. A neural network (NN) is an interconnected group of artificial neurons that imitate the properties of biological neurons. It uses a mathematical or computational model for information processing based on a connectionist approach.

In supervised learning context, and given a set of example pairs (x, y) , x in X , y in Y , the aim is to find the mapping $f : X \rightarrow Y$ implied by the data and that matches the examples; The cost function is the mismatch between the mapping f and the data. A commonly used cost is the mean-squared error that tries to minimize the average squared error between the $f(x)$, and y over all the example pairs. NNs are generally applied in tasks of pattern recognition and regression but also to sequential data like speech and gesture recognition. Particularly, in the last years, NNs have been successfully applied to several face analysis tasks.

In [33], first, the author choses 11 categories (concepts): brick, cloud, fur, grass, ice, road, rock, sand, skin, tree, and water. Then a large amount of training data (low-level features of segmented regions) is fed into the neural network classifiers to establish the link between low-level features of an image and its highlevel semantics (category labels). A disadvantage of this algorithm is that it requires large amount of training data and is computationally intensive.

Decision tree. Decision tree techniques are also used to derive semantic features in a supervised learning context. It is used as a predictive model to map observations about an item to conclusions about the item's target value. In these tree structures, leaves represent classifications and branches represent conjunctions of features that lead to those classifications. Decision rules can be obtained by following the paths from the root of the tree to the leaves. Techniques like ID3, C4.5 and CART build up a tree structure by recursively partitioning the input attribute space into a set of non-overlapping spaces [143]. The CART decision tree is used in [64] to obtain decision rules that are used to map color distribution in a given image to textual description. C4.5 decision tree is also used in [122] as model in a RF learning stage to provide relevant image.

1.2.2.2 Unsupervised learning

In contrast to supervised learning, where a training set is needed to guide the learning process, unsupervised learning consists in automatically finding how data is clustered with no need to training samples.

Image clustering is an example of unsupervised learning techniques. It aims at grouping the images in clusters by maximizing the similarity between elements in the same cluster and minimizing it between different clusters.

K-means. A large collection of works on image clustering have used k-means for semantic analysis.

K-means clustering of the color features of a set of training images is applied in [35]. Then, a set of mappings from the low level features and the high level semantics (keywords) is derived. This is done based on statistics measuring the variation with each cluster. Then the new untagged images are indexed based on the extracted mapping rules.

Automatic annotation of an image database is performed in [155]. The system clusters image regions using a variant of k-means clustering (PCK-means) [96]. A set of 59 concepts are defined for the used image database. Then, given a region, the probability of its belonging is calculated using a semi-naive Bayesian method [155]. Thus, new images are annotated by choosing the concepts with highest probabilities.

Spectral clustering method. Spectral clustering methods are introduced in the last few years, to handle cluster data with complex structure [160]. Spectral clustering techniques make use

of the spectrum of the similarity matrix of the data to perform dimensionality reduction for clustering. Techniques like Normalized cut (NCut) have been successfully used for image segmentation, image clustering [71, 13]. NCut has been successfully applied in many fields; however, it cannot produce an explicit mapping function.

Probabilistic classification. Probabilistic classification based on Bayes theory is among the most powerful clustering tools. The common maximum-a-posteriori (or MAP) classifier and its variation maximum-likelihood (or ML) classifier have shown great promise for the CBIR approach [153, 154]. However, traditionally it is difficult to apply the classifiers due to the complexity of the MAP similarity function. In [154], Vasconcelos has shown that the similarity function can be computed efficiently when vector quantizers and Gaussian mixtures are used as models for the probability density functions of the image features.

1.2.3 Events Detection and Behavior understanding

After successfully tracking the moving objects from one frame to another in an image sequence, the problem of understanding object behaviors from image sequences follows naturally.

Shot and scene boundary detection techniques were one of the first proposals for video summarization. In film-making, a shot is defined as the basic block of a film. It represents a series of consecutive frames taken contiguously by a single camera running for an uninterrupted period of time. A scene, in its turn, is a group of consecutive shots that take place in a single location and in continuous time.

The use of a scene/shot based structure to represent video content can be interesting since it enables for an easy navigation of video document and for reduction of key-frame to be interpreted. However, a scene/shot based structure cannot be relevant in many types of video content. For instance, most sports programs or surveillance videos do not have many specific scenes but continuous flow of action. While scenes and shots are very related to the visual features of the frames, events, as defined in the Chapter Introduction, is related to the objects appearing in videos and into semantic interpretation of their behavior. One event may stretch over several scene. While sometimes many events may successively occur in the same scene.

Event-based structure is then necessary for better behavior understanding of content and actions in a video stream. Moreover, event-based access systems have many advantages over shot/scene based video browsing systems. One of these advantages is that event-based access systems return exactly the video segment where the conditions specified in the queries are satisfied. Whereas, in case of shot/scene based systems, the entire scene/shot containing the relevant video segment is returned.

Event detection and behavior understanding can be defined as matching an unknown data sequence with a labeled reference sequence that represents the targeted event or behavior.

A large effort has been devoted to this issue:

Finite-state machine (FSM). Finite state machines were widely used for event detection purpose over video documents. States are used in general to represent the different steps of the event or to decide whether the reference sequence match the test sequence. The transitions are then used to define conditions or constraints to move from one step to another. An important approach for understanding human's every day gestures is presented in [65]. Authors build separate 3D motion models for each part of the body, then compose them across time and space. Detection of these activities is performed by mapping 2D tracks and

3D models, queries are then written using Finite State Machines based language to retrieve videos. In [12], a system that performs automatic annotation of the principal highlights in soccer video is presented. Highlights are modeled using finite state machines. Highlight detection exploits visual cues that are estimated from the video stream, and particularly, ball motion, the currently framed playfield zone, players' positions and colors of players' uniforms. In [4], FSMs was used to analyze the explicit structure of natural gestures. In [25], handcrafted FSM was used to recognize scenarios describing vehicle behaviors in airborne imagery.

Hidden Markov Model (HMM). HMM is a stochastic state machine [164] which allows more sophisticated analysis of data with spatio-temporal variability. Two steps are necessary for using HMMs in video analysis;

1. Training stage: where states of HMM should be specified and the probabilities should be optimized so that the generated symbols can correspond to the observed image features of the examples.
2. Matching stage: where the probability with which a particular HMM generates the test symbol sequence corresponding to the observed image features is computed.

HMMs are widely applied to behavior understanding. In [144], HMMs are used for recognition of sign language. In [68], HMMs was used to detect free kicks, penalties and corner kicks in soccer matches primarily using camera motion. Similarly, HMMs are used in [46] to analyze tennis videos. In [106], authors assert that Coupled Hidden Markov Models (CHMM) outperform HMMs for modeling people behaviors and interactions such as following and meeting. In [3], audio, face and color features are used by a HMM to classify movie sections as either dialog or non-dialog.

Other techniques. Other interesting techniques were used in event based analysis of video documents. In fact, Dynamic time warping (DTW) has been used recently in the matching of human movement patterns [77, 22]. Time-delay neural network (TDNN) has been successfully applied to hand gesture recognition [164] and lip-reading [151]. Syntactic techniques [67] have been recently used for visual behavior recognition [24]. Non-deterministic finite automaton (NFA) are employed in [156] for multi-object behavior recognition. Finally, Self-organizing neural networks are used as an unsupervised learning method in behavior motion recognition, especially when the object motions are unrestricted [74, 141, 109].

1.2.3.1 Use Case : Soccer Highlights Detection

One field where the need for event-based structure can be clearly observed is soccer video analysis. Automatic recognition of events and activities, particularly soccer highlights, has been studied by the image processing community. A pioneering work in soccer video analysis was presented in [52]. In this work, the ball is detected using its chromatic (white regions) and morphological features (circularity). Players are then recognized by detecting peaks in the color histogram of the frame. In [23] and [85] the authors present a semantic video indexing algorithm based on finite state machines and low-level motion indices extracted from the MPEG compressed bit-stream. The proposed algorithm is an example of solution to the problem of finding a semantic relevant event (e.g. scoring of a goal in a soccer game) in case of specific categories of audio-visual programs. To face the semantic indexing problem, an automatic system operates in two steps : first extraction of some low-level indices in order to represent low level information in a

compact way and then a decision-making algorithm to extract a semantic index from the low-level indices.

Another similar approach is the one proposed in [47]. In this work, authors propose an automatic framework for analysis and summarization of soccer videos using cinematic and object-based features. The system can output three types of summaries that are all slow-motion segments in a game, all goals in a game, and slow-motion segments classified according to object-based features. This is performed based on some *low-level* soccer video processing algorithms, such as dominant color region detection, shot boundary detection, shot classification, and on some *higher-level* algorithms like: goal event detection, referee detection, penalty-box detection. Goal events are detected by exploiting in addition to cinematic feature, object-based features that generally follows this events. Those features are emotions of the audience, the close-up views of the actors of the goal celebrating it, but also the slow-motion replays of the goal event.

In [41], a Circle Hough Transform was used to detect soccer ball, however the approach is limited since it requires homogeneity of the ball and does not support occlusions.

Trajectory knowledge is also used in ball detection. Others in [166] use Kalman filter for verifying ball trajectory for off-line detection. While in [7] a Viterbi algorithm is employed to track the ball trajectory.

Other approaches as in [145] start by detecting playfield using color segmentation. Then they apply morphological operations, such as connected component analysis, to detect different other objects like lines, players referees, etc. The main problem in those approaches is the false alarms due to incorrect player and ball detections within the playfield, but also to the occlusions and superposition of players and field lines.

1.2.4 Discussion

Statistical-based techniques for bridging the semantic gap enabled for more human-oriented image and video retrieval. In fact, users become able to query multimedia collections using more meaningful descriptors referring to objects and events.

In most learning-based event detection systems in sport video analysis, only important highlights has gained interests of researchers ([146], [162], [157], [17]). While most of those systems could satisfy a large audience, few approaches have paid attention to other users like training professionals or players, which are, in their side, interested more in specific and personalized event and action detection. Learning-based event detection systems are also as prefabricated black boxes with few possibilities of interaction or modification of the parameters of the recognition process. The proposed algorithms usually summarize extracted events and present them without ability for users to request changes or further information [171].

On the other hand, most learning-based approaches for event detections depend on the training samples used to build the models. However, it is very difficult to provide a large amount of labeled training samples with no errors. Furthermore, changing the application domain requires providing new training samples [54].

Moreover, many approaches of sport videos analysis use smart tricks like emotions or camera motion to characterize and detect important highlights, rather than constituting the complete model of the event by seeking the elementary objects composing it. While the first method enables for relatively rapid responses to event recognition, it is, however, observed that this kind of method may cause excessive false alarms [30].

Video documents comprise extremely rich sources of information and its understanding requires combining multiple contents to infer high semantic information. Developing an learning-based classifier for each event or object is very complicated and time and effort consuming. Using high level reasoning seems to be the best way to combine low-level descriptions in order to infer new information and detect complex objects and events.

One of the major problems that should be solved is the description vocabulary that could provide enough expressivity, interoperability and reuse of video data descriptions.

1.3 Semantic Description Languages and Standards

Recently, several standard description languages for the expression of concepts and relationships in domain ontologies have been defined. These languages enable to produce specific domains and purposes descriptions, yet still remaining interoperable and capable of being processed by standard tools and search systems [20]. The most important were Resource Description Framework Schema (RDFS) [79] and Web Ontology Language (OWL) [99]. However, these semantic web languages were designed for describing all types of resources and do not satisfy specific requirements of multimedia content annotation.

Many vocabularies specifically dealing with multimedia content annotation can be found in the literature [51]. The most widely used ones are the following:

MPEG-7. MPEG-7 is the widely used standard for description of audiovisual features, descriptors, structures, and relationships. A more detailed description of MPEG-7 can be found in the next section.

Dublin Core Element Set. Dublin Core Element Set³ is a commonly used annotation scheme across different domains that can be assigned to any type of resources. It is a small set of relations, identified by domain experts in the field of digital libraries.

VRA. VRA⁴ is a visual representation of physical objects in the cultural heritage domain. The difference between physical object and digital representation has been made explicit. VRA also defines a vocabulary to annotate material in which it makes suggestions to use terms from other vocabularies.

Media Streams. Media Streams[37] (Davis) provides a detailed iconic vocabulary to describe video content, for search and automatic editing with a modular approach to support stratified annotations. The descriptive potential of the used vocabulary is reduced in order to support a large domain, reduce implicit contextual knowledge, and support operational difficulties related to automatic search.

PREMO. PREMO [44] defines a vocabulary which can be used to describe multimedia systems focusing on synchronization and dependencies from a system components perspective. It is a potential candidate to provide specifications on how a presentation specification is rendered to be perceived by a user.

³<http://www.dublincore.org/documents/dces>

⁴<http://www.vraweb.org/vracore3.htm>

COMM. COMM⁵ (Core Ontology for Multimedia) is a well-founded multimedia ontology based on both the MPEG-7 standard and the DOLCE⁶ foundational ontology. The ontology covers the main parts of MPEG-7 and satisfies the main requirements for designing a multimedia ontology such as MPEG-7 compliance, semantic interoperability, syntactic interoperability, Separation of concerns, modularity, and extensibility [11].

1.3.1 MPEG-7

MPEG-7 [1] (formally called Multimedia Content Description Interface) is a multimedia content description standard where low level encoding scheme descriptors to high level content descriptors are merged to describe audio-visual features and their relationships. This description is associated with the multimedia content itself, to allow easy and efficient searching with regards to user's requirements. The standardized description concerns visual and semantic descriptions as well as external information.

To describe a multimedia content, MPEG-7 uses the following tools:

- Descriptors (D): a representation of a feature defined syntactically and semantically.
- Description Schemes (DS) : Specify the structure and semantics of the relations between its components, it can be (D) or (DS)
- Description Definition Language (DDL): XML-based language used to define the structural relations between descriptors, their modification and creation.
- System Tools: tools allowing generation of (D) and (DS), its binarization, synchronization, transport and storage.

The decomposition of multimedia content is described by a set of attributes defining the division type; temporal, spatial, or spatiotemporal. Overlaps can be accepted between segments.

Graph structures are used to describe events occurring in videos segments. A Graph is defined by a set of nodes, each corresponding to a segment, and a set of edges, each corresponding to a relationship between two nodes. Figure 1.2, taken from [1], shows an example of description of a soccer game excerpt. Two *video segments*, one *still region* and three *moving regions* are used to describe the event of dribbling, kicking the ball and then scoring a goal. The first video segment "Dribble and Kick" involves the Ball, the Goalkeeper and the Player. The Ball remains close to the Player who is moving towards the Goalkeeper. The Player appears on the Right of the Goalkeeper. The second video segment "Goal score" involves the same moving regions plus the still region called Goal. In this part of the sequence, the Player is on the Left of the Goalkeeper and the Ball moves towards the Goal.

Figure 1.3 shows the MPEG-7 based syntax description of a hand shake event between a person *A* and a person *B*.

1.3.1.1 MPEG-7 Profiles

In MPEG-7, very different syntactic variations may be used in multimedia descriptions with the same semantics, while remaining valid MPEG-7 descriptions. However, without additional

⁵<http://comm.semanticweb.org/>

⁶<http://www.loa-cnr.it/DOLCE.html>

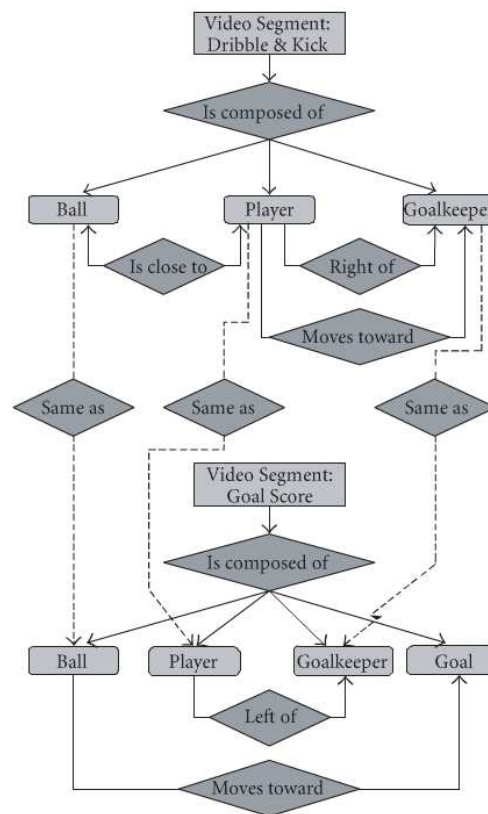


Figure 1.2: Video scene description in MPEG-7

```

<Mpeg7>
  <Description xsi:type="SemanticDescriptionType">
    <Semantics>
      <Label>
        <Name> Shake hands </Name>
      </Label>
      <SemanticBase xsi:type="AgentObjectType" id="A">
        <Label href="urn:example:acs">
          <Name> Person A </Name>
        </Label>
      </SemanticBase>
      <SemanticBase xsi:type="AgentObjectType" id="B">
        <Label href="urn:example:acs">
          <Name> Person B </Name>
        </Label>
      </SemanticBase>
      <SemanticBase xsi:type="EventType">
        <Label><Name> Handshake </Name></Label>
        <Definition>
          <FreeTextAnnotation> Clasping of right hands by two people </FreeTextAnnotation>
        </Definition>
        <Relation type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:agent" target="#A"/>
        <Relation
          type="urn:mpeg:mpeg7:cs:SemanticRelationCS:2001:accompanier" target="#B"/>
        </SemanticBase>
      </Semantics>
    </Description>
  </Mpeg7>

```



Figure 1.3: MPEG-7 description code example

knowledge about how MPEG-7 has been used, semantics about descriptors cannot be correctly inferred from the elements in the description. For instance, the *VideoSegment* descriptor can be used to represent at the same time the whole video content, the shots and the key frames as shown in 1.4.

```
<VideoSegment id="TRECVID2005_1">           <!-- whole video -->
  <MediaLocator>
    <MediaUri>20041116_110000_CCTV4_NEWS3_CHN.mpg</MediaUri>
  </MediaLocator>
  [...]
  <TemporalDecomposition gap="false" overlap="false">
    <VideoSegment id="shot1_1">           <!-- shot -->
      <MediaTime>
        <MediaTimePoint>T00:00:00:0F30000</MediaTimePoint>
        <MediaDuration>PT00H00M03S26116N30000F</MediaDuration>
      </MediaTime>
      <TemporalDecomposition>
        <VideoSegment id="shot1_1_RKF">   <!-- key frame -->
          <MediaTime>
            <MediaTimePoint>T00:00:01:27057F30000</MediaTimePoint>
          </MediaTime>
        </VideoSegment>
      </TemporalDecomposition>
    </VideoSegment>
    [...]
  </TemporalDecomposition>
</VideoSegment>
```

Figure 1.4: VideoSegment MPEG7 descriptor used with different semantics.

In order to avoid semantic ambiguities induced by this flexibility, *Profiles*, such as Detailed Audiovisual Profile (DAVP) proposed by Troncy et al. [148], enable to specify a number of semantic constraints. These *Profiles* have three parts:

1. Description tool selection: subsets of definition tools included in the profile.
2. Description tool constraints: restrictions on cardinality or on attributes,
3. Semantic constraints: describe the use of tools in the context of the profile

Additional elements include the *StructuralUnit* element on the segments that specify its semantic type (shot or key frame) and the *criteria* attribute of the decompositions. With these additional elements and attributes, it is then possible to distinguish between the different types of elements for which *VideoSegment* has been used.

1.3.2 Discussion

Some annotation tools have appeared in order to assist users in the task of annotating the video contents. As an example to this, one can cite VideoAnnEx annotation tool [131] that enables users to add new description to video segments using MPEG-7. Annotations consisting of objects, events or other lexicon sets can be attached to each video shot. The tool enables for saving MPEG-7 description files, but also to open them to display annotations associated to video sequences.

Despite the large capabilities offered by multimedia annotation vocabularies and tools for representing low-level feature (**what the video looks like?**) and syntactic information (**how is**

```

<VideoSegment id="TRECVID2005_1">
  <StructuralUnit
    href="urn:x-mpeg-7-davp:cs:StructuralUnitCS:2005:vis.programme"/>
  [...]
  <TemporalDecomposition gap="false" overlap="false"
    criteria="visual shots">
    <VideoSegment xsi:type="ShotType" id="shot1_1">
      <StructuralUnit
        href="urn:x-MPEG-7-davp:cs:StructuralUnitCS:2005:vis.shot"/>
      <MediaTime>
        <MediaTimePoint>T00:00:00:0F30000</MediaTimePoint>
        <MediaDuration>PT00H00M03S26116N30000F</MediaDuration>
      </MediaTime>
      <TemporalDecomposition criteria="key frames">
        <VideoSegment id="shot1_1_RKF">
          <StructuralUnit
            href="urn:x-MPEG-7-davp:cs:StructuralUnitCS:2005:vis.keyframe"/>
          <MediaTime>
            <MediaTimePoint>T00:00:01:27057F30000</MediaTimePoint>
          </MediaTime>
        </VideoSegment>
      </TemporalDecomposition>
    </VideoSegment>
  [...]
</TemporalDecomposition>
</VideoSegment>

```

Figure 1.5: VideoSegment MPEG-7 descriptor used with different semantics using the DAVP Profile.

the video structured?), these vocabularies are not precise enough to describe semantic information (what is happening in the video?). In addition, these vocabularies are not associated with query mechanisms that enable users to explore and retrieve semantic information through annotated video documents. There is a real need of video models that could efficiently manage the three kinds of information but also that provide the necessary powerful query facilities.

1.3.3 Logic-based formalisms

Many approaches have investigated the use of logical formalisms for expressing semantics.

1.3.3.1 Ontology based Multimedia Indexing

A formal ontology is a controlled vocabulary where relations are explicitly expressed in an ontology representation language [134]. A corresponding language has a grammar for using vocabulary terms to express something meaningful within a specified domain of interest. The grammar contains formal constraints (e.g., specifies what it means to be a well-formed statement, assertion, query, etc.) on the way terms in the ontology controlled vocabulary can be used together. An ontology consists of concepts, concept properties, and relationships between concepts represented by terms.

In video semantic indexing, ontologies are generally used to enable for semantic annotation of video documents either manually by associating the terms of the chosen ontology with video segments, or automatically by associating the terms of the ontology with appropriate classifiers and knowledge models that define the combination of low and mid level visual features representing the terms [18]. In [55] domain specific ontology was used for automatically creating a semantic description of soccer video documents using a reasoning engine. The ontology was provided with the ability of using multilingual terms and cross document merging.

In [121] an ontology integrating the scene knowledge and the system knowledge, with the purpose of detecting the objects and events in a video scene for surveillance, is presented. Scene knowledge is formulated using objects, relations showing how scene objects and simple or complex events can be described, while system knowledge allows determining the best configuration of the processing schemes for detecting the objects/events of the scene.

An approach to semantic video object detection is presented in [36]. Semantic concepts for a given domain are defined in an RDF(S) ontology together with qualitative attributes (e.g. color homogeneity), low-level features (e.g. model components distribution) and object spatial relations and multimedia processing methods (e.g. color clustering).

1.3.3.2 Multimedia Enriched Ontologies

Multimedia Enriched Ontologies are specific ontologies where concepts and categories are not only linguistic terms but also visual or auditory data that would be more appropriate in describing a particular category of video content. They were created due to the experimental observation that linguistic terms are not rich enough when they must describe specific patterns of objects, events or video entities.

Multimedia Enriched Ontologies were first introduced in [72] where text information available in videos and visual features are extracted and manually assigned to concepts, properties, or relationships in the ontology.

In [19], linguistic ontologies were extended by visual concepts that enable for enriched video annotations. Video documents of highlights are first linked to linguistic concepts, then videos in the same concept class are clustered into subclasses according to their perceptual similarity. Visual concepts are then defined as the centers of each cluster such that each visual concept represents a specific pattern. Based on these pictorially enriched ontologies, MOM (Multimedia Ontology Manager) system was presented in [18]. This system enables for creating and updating of multimedia ontologies, automatically annotating and commenting video sequences, and also querying video databases based on the ontology itself.

1.3.3.3 Conceptual Graphs based Multimedia Indexing

Conceptual graphs are very useful and powerful formalisms for representing structured knowledge. In [101] Mechkour presents an extended model for image representation and retrieval called *EMIR*², this model combines different interpretations of the image to build a complete description of it, each interpretation being represented by a particular view. Based on Conceptual Graphs, the model defines four types of relations between concepts each corresponding to a specific view within still image content: the structural view involving relations between parts and subparts, the spatial view, involving 2D (within the image) or 3D (in the real world represented in the image) relations between image parts, the symbolic view involving relations that define symbolic properties of image parts, and finally the perceptive view involving relations that define perceptive properties (color, texture,) of image parts. The work proposes a correspondence function that estimates the similarity between two images. In [28], authors introduce a conceptual model for video content description presented as an extension of *EMIR*². In addition to *EMIR*² view, The proposed model adds further views like; an extended structural view involving relations between temporal segments, a temporal view involving temporal relations between video segments, an extended perceptive view, involving relations that define additional percep-

tive properties (mainly about motion) of video parts, and finally an event-based view, involving relations that define what happens to or between video parts.

Another interesting approach in this sense is the work presented in [66]. Authors present an experimentation concerning the description of audio-visual documents used in medicine and based on relational indexing schemas. This description rests on the concept of patterns of indexing based on existing usage scenarios, and exploits technologies resulting from the semantic Web. The authors show that the combination of several ontologies and rules of inference enables for more complete structured description.

1.3.3.4 Description Logic based Multimedia Indexing

In [138] the simple description logic *ALC* was extended with fuzzy logic in order to support reasoning about imprecise concepts. A concept *C* of the fuzzy DL is interpreted as a fuzzy set and the assertions associating an individual to a concept or a couple of individuals to a role are given a truth value in $[0,1]$ representing a degree of membership. *SHOIN(D)* description logic [95] is a powerful language allowing to reason with concrete data types such as strings or integers using so-called concrete domains. In [139], an extension of the *SHOIN(D)* with fuzzy logics is presented. It provides further capabilities especially by using fuzzy sets of concrete domains and fuzzy modifiers, and by allowing values from the interval $[0, 1]$ for subsumption relationship. In [91] an application of *ALC* description logic in the multimedia context is presented. This application aims to improve semantic search of multimedia resources in the e-learning tool CHEST (Computer History Expert System). It takes as input a question about computer history expressed in natural language, translates it into a formal DL expression, and returns as output the list of multimedia clips whose description is subsumed by the formal query.

1.3.4 Event Models

Events are essential information for humans, and they are important concepts for multimedia. Several models and ontologies have been proposed during the last few years in order to model events, their spatiotemporal properties, their causality and their composition. Among the proposed ontologies, one can cite:

- CIDOC CRM [39] and ABC [84] that are two ontologies enabling to describe historical events,
- EO[115] intends to describe musical events,
- EventsML-G2⁷ designed for exchanging events information among news providers,
- DUL⁸ that is a simplification of DOLCE foundational ontology⁹ for representing the social aspects.
- LODÉ [125] that is presented as a linked data event model enabling for representing events by their "factual" aspects that are agentivity, space, time, participation and causality.

⁷<http://www.iptc.org/EventsML/>

⁸<http://www.loa-cnr.it/ontologies/DUL.owl>

⁹<http://www.loa-cnr.it/DOLCE.html>

However, while spatiotemporal, causal, and purpose aspects of events were well studied, those concerning composition, participation and object involvement still need more research effort. A brief survey on the properties provided by the above models for linking objects to events can result to the following; ABC defines **hasPresence** properties to assert that an agent (object) was present at an event. It provides also **hasParticipate** to distinguish those agents who had major role in the occurrence of the event. CIDOC in his turn respectively uses **P11.had_Participant** and **P11.carried_out_by** for the previous purposes. DUL uses **hasParticipant** for linking an event to an agent while EO provides the **agent** property for the same purpose. LODE defines two properties **involved** and **involvedAgent** to link an event respectively to a thing or an agent. In order to link an event to another as a part of it, CIDOC proposes **P9.consists_of** property, ABC uses **isSubEventOf** property, while EO defines **sub_event** property.

Although these previous ontologies enable for linking events to their composing objects/agents or events, they are not expressive enough to describe the complete structural aspect of the event, the order of participation of objects during the events or the spatial relation established between them. They also do not enable for an advanced description of the different steps (situations) that an event go through during its occurrence. Furthermore, those ontologies and data-models enable to represent only the composition of event occurrences not the composition of event types. Providing such a representation would enable for detecting event occurrence by detecting its composing objects and events.

1.4 Data-Models and query languages for video Database Management Systems

The size and the richness of multimedia collections is in constant growth. Resources need to be semantically indexed so that answers to queries can be quickly computed.

Studies [118] have shown that most user needs correspond to queries expressed using high level (i.e. semantic) concepts. But structured and accurate manual annotation is still lacking for the vast majority of multimedia documents.

1.4.1 Annotation-based Video Models

The segmentation approach [31] was one of the first video indexing scheme for video documents. The video is split into independent time segments annotated individually (Figure 1.6). However the strict and crisp partitioning of video documents hinder to describe intervals other than segments which leads to imprecise and insufficient description. The early models introduced to remedy these limitations are referred to as annotation-based models, also called stratification approaches [7].

Stratification enables for annotating video content individually by associating to each annotation a temporal interval called *Stratums*. This is reduced to the creation of several layers of descriptions on the top of video stream. The user is able to access only interesting fact regardless of other descriptions. Annotations can be either keywords, free text, or structured data (Figure 1.7). Despite their facility and flexibility, annotation-based models have limited expressive power and query facilities. Relations can not be specified between annotations and only keyword-based queries are allowed.

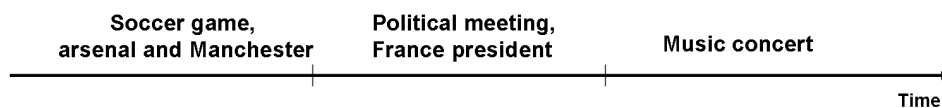


Figure 1.6: Example of a segmentation of a video segment.

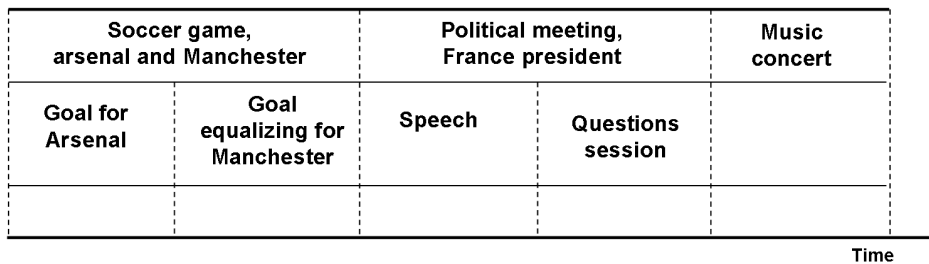


Figure 1.7: Example of a stratification of a video segment.

Some interesting Annotation-based video query systems can be found in the literature. QBIC [50] was one of the first systems designed to explore large image data bases. It enables querying based on keywords, sketches, color, shape, texture, but it is limited to low level features.

CCM [89] makes a compromise between the property-inheritance mechanism of strong type models and flexible facilities for dynamic schema update as in weak type models. Later, Smart VideoText [80] uses conceptual graphs to capture the semantic associations among the concepts described in text annotations of video data and achieves more effective querying of the semantic content of video data.

M-OntoMat-Annotizer [111] is an integrated framework developed under aceMedia¹⁰; a project aiming to discover and exploit knowledge inherent in multimedia content in order to automate annotation at all levels and so to make content more relevant to the user. This framework enables for linking RDF(S) domain ontologies with low-level MPEG-7 visual descriptors. M-OntoMat-Annotizer presents a graphical interface for loading images and videos, enables the user to select regions of interest from images and then apply low-level visual features extraction procedure to associate regions with appropriate semantic description. The tool also supports automatic segmentation of the images.

1.4.2 Object-Relational Data-Models

User requirements for video data exploration need more powerful models to be fulfilled. Object-Relational data-models then have been introduced in order to provide users with more sophisticated capabilities. They enable to represent the real world objects appearing in a video, events, and relationships between objects. They also provide richer query facilities such as query by attributes, relationships, temporal ordering, and browsing [167].

¹⁰<http://www.acemedia.org/aceMedia>

The video database system OVID [108], is one of the earliest object-relational data models. A video frame sequence (an interval of video frames) is modeled as an object called video objects. Video objects can have attributes to describe their contents. Inheritance notion is defined based on the interval inclusion relationship. TA SQL-based query language VideoSQL for retrieving video-objects is presented. Semantic objects are represented as values of attributes of video-objects. However there is no possibility to define spatiotemporal relationships between semantic objects.

AVIS [6] is one of the earliest object-relational video data-models that enables indexing and querying objects and relationships between them. It divides video into fixed duration intervals. Then, it locates objects and events in the time line by using an association map that corresponds to the line segment of the x axis of the Cartesian plane. It adopts a frame segmented tree where each node represents a frame sequence and the objects and events occurring in it.

Another example of object-relational Video models is Videx [150] that uses UML to represent the structure and semantics of video data in an object-oriented manner. ExIFO2 [14] is an extension of the data model *ExIFO₂*, a conceptual data model able to handle complex objects along with their uncertain and imprecise properties. Another semantic video model is the one proposed in [47]. In this work, authors propose a video model for analysis and summarization of soccer videos using cinematic and object-based features. The model rests on Entity-relationship (ER) model enriched with object-oriented concepts.

Hacid et al. [56] have extended the stratification concept by defining temporal cohesion. While in the stratification approach, a time segment is associated with a description, temporal cohesions allow a set of time segments to be associated with the same description (Figure 1.8). However, this work was limited to temporal modeling and did not take into account spatial modeling.

Spatial modeling is also inherent to video data and the spatial dimension of entities appearance and event occurrence should be considered in order to build a satisfactory video data management system and query engine.

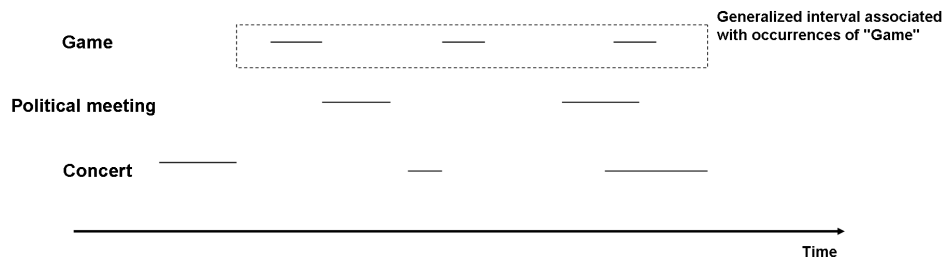


Figure 1.8: Example of a temporal cohesion of a video segment.

Many investigations regarding the spatial dimension of the entities and the events in video document exist. In [81], an extension of AVIS system to spatial dimension [6] is presented. It extends the association map used in AVIS so it can store spatial properties of objects as well as temporal properties. However, objects have no other attributes and few semantics can be inferred and stored in the data model.

BilVideo system was presented in [40] as an original query system allowing to represent spatial, temporal and semantic information of objects in video documents. Based on this system, a natural language-based interface for querying video contents was developed in [82] using En-

glish language. Nevertheless, the system deals only with the spatial properties corresponding to the coordinates of objects on the screen. However, an object or an event can be spatially positioned following different references (e.g. soccer players should be located according to their position in the playfield, to their position on the screen, but also to the city where the match is played). Therefore, spatiotemporal queries can be limited without considering additional spatiotemporal frames of reference. In [48], two time dimensions were described, the *story time* that correspond to the time dimension during which the story of video “takes place”, and the *video time* during which the story is “shown”. In order to differentiate the two dimensions temporal relations are postfixed by *_V* for video time and *_S* for story time. Nevertheless, only events can be connected by temporal relations and not durations of appearance of objects. In real world, many story times can exist. For example, in order to locate soccer actions, in addition to the hour and date the game takes place, the minute of play is also interesting. Other temporal referential and spatial referential can exist depending on the application, users should be able to define and use them in their video database systems.

1.5 Discussion and Conclusion

Some works have focused on the evaluation of semantic video models and setting criteria to characterize their expressivity and power. Geurts et al. proposed in [51] a set of requirements that a language of multimedia content description should fulfill for an efficient annotation. These requirements include: lightweight and extensibility, reuse of existing vocabularies, relating concepts to media assets, and annotation structuring. Authors in [159] focused on the recommendations for event centric video models. They mentioned for example the ability to express spatial-temporal constraints, taking into consideration uncertainty, offering different levels of granularity in description, being independent from storage format and satisfying interoperability between description frameworks. One of the most complete evaluation of semantic video models is the one presented by Yu Wang et al. in [167] where semantic video models are compared according to 21 criteria that concern three major requirements that are expressive power, acquisition of semantic information and query supporting capability.

In addition to the above requirements and according to which many evaluations were done, additional requirements are, in our point of view, very important to produce integrated Object-Relational video models :

Objects/Events modeling. the model should enable the users to define and use objects and events to describe contents within video documents. Hierarchical connections should also be allowed between objects of different types. Users should be able to connect objects with events. Most video models mentioned above enable this criteria. However, the definition of events in those approaches is restricted. Objects are connected to events only in terms of role attributes (Hacid [56], AVIS [6]). Few possibilities are given in order to add more specification to the order of appearance of objects within the described events or even their spatiotemporal positions during the occurrence of the event.

Objects/Events detection. A video model should be able to integrate components for detecting object and events within video documents and then to feed the semantic information database. Some of the video models above (such as *MOntoMatAnnotizer*) enable objects detection by performing extraction of low level features and then linking particular classes to domain concepts used to define objects. However, few video models incorporate inde-

pendent component for detecting events with respect to their structure and the types of objects involved in without the need of user queries.

Manual annotations. Although humans do not have much time to describe video content in an efficient way, some manual annotations can be relevant for detecting or inferring new information. The video model should enable users to populate the database with manual annotations using appropriate tools. Some video models (such as Bilvideo), enable the annotation of video segment and region with fine granularity, however, only MBR (Minimum Boundary Rectangle) is used to associate a region of image to a specific object. This form is not convenient for many objects such as lines, balls, etc.

Support for uncertainty. Uncertainty is one of the major reasons of semantic gap that has not been sufficiently studied. In fact, uncertainty can appear in many forms and can concern many issues. Uncertainty should be taken into consideration when detecting objects and events. Attributing a region to a specific type of object is not obvious and may be a source of uncertainty. Users also should be enabled to flexibly specify their queries and especially the relations between objects. They use, in general, vague concepts (such as bigger, near,...) that can not be represented with exact measure but only with approximations. Few video models support uncertainty. Extended ExIFO2 supports uncertainty only at attribute-level, class/object level, and class/subclass level.

Spatiotemporal positioning. Users locate objects and events according to multiple spatial and temporal environment following their needs. While the position of the object in the screen or in the image is important, other spatial frames of reference are important such as the city where the event happens, the geometrical position with regards to a specific zone (position in a playfield), etc. The same fact can be established for temporal localization where measures other than the time of occurrence in the video are important according to the domain of application. Users should be able to freely define and use new spatial and temporal frames of reference in order to locate their objects and events. Few efforts have been done in this direction except the work reported in [48] where two time dimensions were defined, that is "*story time*" and "*video time*".

The rest of this manuscript is devoted to our contribution towards trying to provide solutions to some of the mentioned problems:

- intuitive and human centric description of the event structure using elementary composing objects.
- expressive description model directly connected to logical reasoning
- enabling fuzzy spatiotemporal reasoning for extraction of complex events.
- indexing object and events according to multiple spatiotemporal frames of reference.

There are no little events in life, those we think of no consequence may be full of fate, and it is at our own risk if we neglect the acquaintances and opportunities that seem to be casually offered, and of small importance.

Amelia E. Barr

2

A Semantic Language for Description and Detection of Visual Events

▷ Our objective in this chapter is to design a semantic representation language enabling for complete specification and modeling of visual events. A model specified by the language will be used to automatically retrieve visual events from a video database or in real time video broadcast. This modeling language is also used for semantic validation of MPEG-7 based description of video sequences. This chapter is organized as follows. First we explain our contributions and the requirements that are fulfilled by our multimedia resource description language. Then, the language devoted to describe visual events is presented in (Section 2.2) and its applications to automatic event detection (Section 2.4.2), to video guided monitoring (Section 2.4) and to semantic validation of MPEG-7 based visual descriptions (Section 2.5) are presented. ◁

Contents of the chapter

2.1	Introduction	37
2.2	Contribution	37
2.2.1	Video Semantic Structure	37
2.3	Modeling Visual Events	40
2.3.1	Conceptual Graphs for Video Content Description	40
2.3.2	Finite State Machines and Video indexing	40
2.3.3	Formal Model	40
2.3.4	Hierarchical description	41
2.4	Video Guided monitoring of behavior	42
2.4.1	Monitoring protocol construction	44
2.4.2	Event detection using monitoring protocols	45
2.4.3	Use Case: Car Theft	48
2.5	MPEG-7 Annotation Validation	48
2.6	Conclusion	55

2.1 Introduction

For an efficient event detection in video document, every system should implement layers corresponding to objects classification, tracking, personal identification and then behavior understanding. But while works on the first layers have produced quite reliable results, the research on the last layer still remains in the preliminary stages.

Many researchers have focused on the semantic interpretation of video contents [59]. Semantic video models have been introduced since the early 90's (section 1.4) in order to remedy some of the drawbacks of appearance-based semantic video analysis, especially those concerning expressivity and power of query and modeling languages. Despite their importance, many of these approaches remains domain-restricted, and, on the other hand, do not involve high level reasoning necessary for large knowledge processing.

Most users of media search engines specify their queries using common concepts expressed in natural or weak formal language. For this reason, we have designed a semantic representation language allowing for an intuitive and complete description of visual events. Users can thus build high-level descriptors combining intermediate concepts from lower abstraction levels, and then use them as queries for event detection within the video.

2.2 Contribution

2.2.1 Video Semantic Structure

The proposed formalism is a combination of Timed Finite State Machines (*tFSM*) [9] and Conceptual Graphs (CG) [132]. It enables to describe complex events by expressing spatial and temporal constraints on involved objects. In addition to their flexibility and power to represent qualitative information, CG were chosen to express spatial constraints due to their adaptability to build graph structures that are much more adapted to express spatial positioning of video contents. The semantic video model we present in this chapter is especially designed for real time video analysis, which justified the use of *tFSM*. The part of the integrated framework concerned by this chapter is highlighted in 2.1.

Descriptions of semantic contents are needed for different levels of abstraction. To this aim, some notions for structuring content semantics within videos were proposed:

Basic Object. Elementary item representing low and mid level concepts that can be expressed either using image processing detectors or by manual annotations (car, person, Zidane,...).

Relation. A spatial or logical constraint gathering together two separate objects (in, contains, bigger than, ...).

Complex Object. A semantic entity composed of connected objects using spatial or logical relations (occupied car, empty zone,...)

Situation. A set of objects related by spatial or logical relations describing a configuration that remains true during one or several consecutive frames (ball in penalty zone,...)

Event. A set of situations related by temporal relations representing the possible configurations of the different state sequence forming an occurrence of the event (penalty, goal, car theft,...)

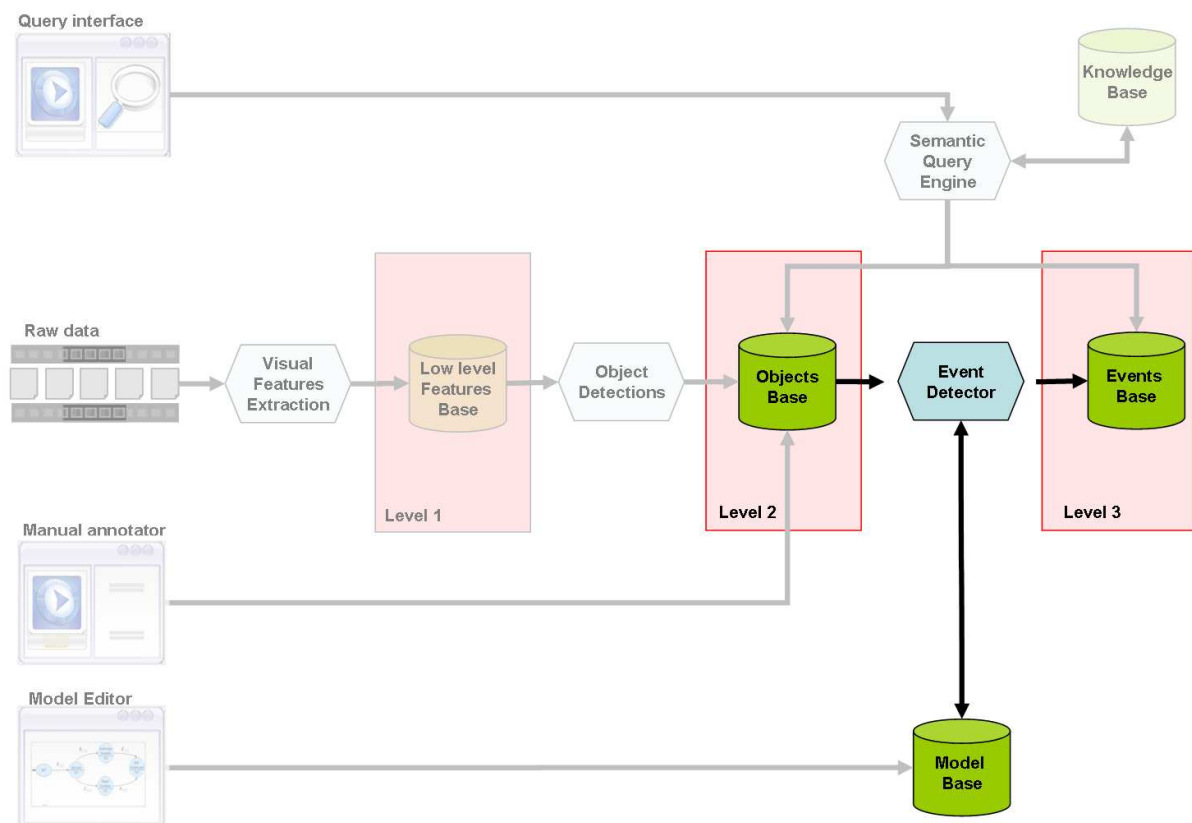


Figure 2.1: Positioning chapter contribution within the whole framework.

Figure 2.2 depicts our semantic structure to describe video.

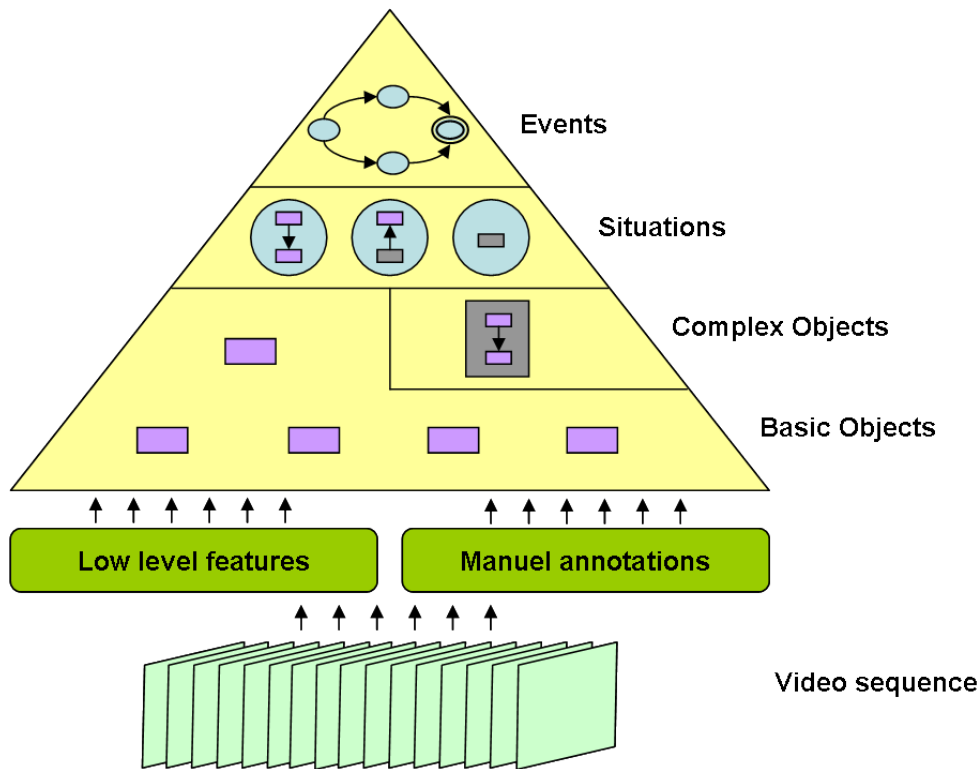


Figure 2.2: General Semantic structure of video contents

The present video model fulfills the following set of requirements. Those requirements are cited by researchers as mandatory to be satisfied by event-based models for multimedia applications [159].

- Combine manual and automatic annotations.
- Design of generic detectors for complex events.
- Provide expressive language.
- Enable for human centric and comprehensive event descriptions.
- Use intermediate level to narrow the semantic gap.
- Produce Domain-independent description models.
- Express Spatial and temporal constraints.
- Enable for hierarchical description of objects and events.
- Include objects, events, and attributes to describe video content.
- Distinguish activities and events.
- Offer different levels of granularity in description.

- Be Independent from storage format.
- Satisfy interoperability between description frameworks.

Other requirements such as uncertainty support and expressive querying will be handled by the event model extension proposed in the next chapters 3 and 4

2.3 Modeling Visual Events

2.3.1 Conceptual Graphs for Video Content Description

Conceptual graphs (CG) are known to be flexible and expressive. They are suitable for representing semantic knowledge about video content since they are directly connected to first order logic. In fact, in [133] an operator Φ which permits to associate to each graph u a formula $\Phi(u)$ expressed in first order logic and then translate all the semantics is defined. A document D expressed by a graph G_D is relevant to a query Q expressed by a graph G_Q if G_D is a specialization of G_Q [48]. A query expressed as a CG can be answered either by a direct matching with a CG of the KB or an indirect matching using inference rules. All of the algorithms defined on CGs are domain-independent and every semantic domain can be described through a purely declarative set of CGs .

2.3.2 Finite State Machines and Video indexing

Timed Finite State Machines (tFSM) have expressive power and are frequently used for pattern detection and recognition [113]. Moreover, while learning based approaches suffer from erroneous classifications due to incomplete training sets, FSM-based models enable for explicitly declaring rules about event structures and objects and relations composing them. FSM models are also relevant for monitoring and understanding behavior for real time video analysis by using states to reflect meaningful changes of phases and transition to define conditions to move from one state to another one. Results are very promising provided that elementary objects are appropriately selected and classified and that the structure of event is known and precise.

2.3.3 Formal Model

Let C be a set of object types and R a set of topological and spatial binary relations. A Basic object, (the lowest semantic components) is represented by a referent f and an object type o from C . Complex objects and situations are represented by a referent f , a set O of basic objects composing the complex object, and a graph $G=\{(o_1, o_2, r) | (o_1, o_2) \in O^2, r \in R\}$ describing the spatial and logical links between the composing objects.

An Event Model M is defined as $M=(P, S, \delta)$ where :

- P is a *tFSM* representing the temporal segmentation of the event.
- S is a set of CG corresponding to different 2D compositions of objects involved in the event.
- $\delta : S_P \rightarrow S$ is the function associating to each state of P a CG of S .

A spatial situation is described using a *CG* (*CG*). $G(C,R,A)$ is a directed graph formed using three types of components: concepts, relations, and attributes, grouped respectively in the sets C , R , and A . They are defined by:

- **concept** (e, t) where e is a referent, and t is a concept type.
- **relation** (r) where r is a relation type.
- **attribute** $[a, v]$ where a is an attribute type, and v is a value.

The used *tFSM* is defined as $P = (S, s_0, F, M, R)$ where S is the set of spatial states (each one will be described by a conceptual graph), s_0 is the initial state, F is the set of final states, M is the set of labels in the form $X_{after(MinSec-MaxSec)}$ where *MinSec* and *MaxSec* are respectively the minimal and the maximal duration in secondes separating the two states; $R = \{(s_i, s_j, m) | s_i \in S, s_j \in S, m \in M\}$ is the set of all possible transitions between states.

An example of an event described using the presented formalism is shown in figure 2.3.

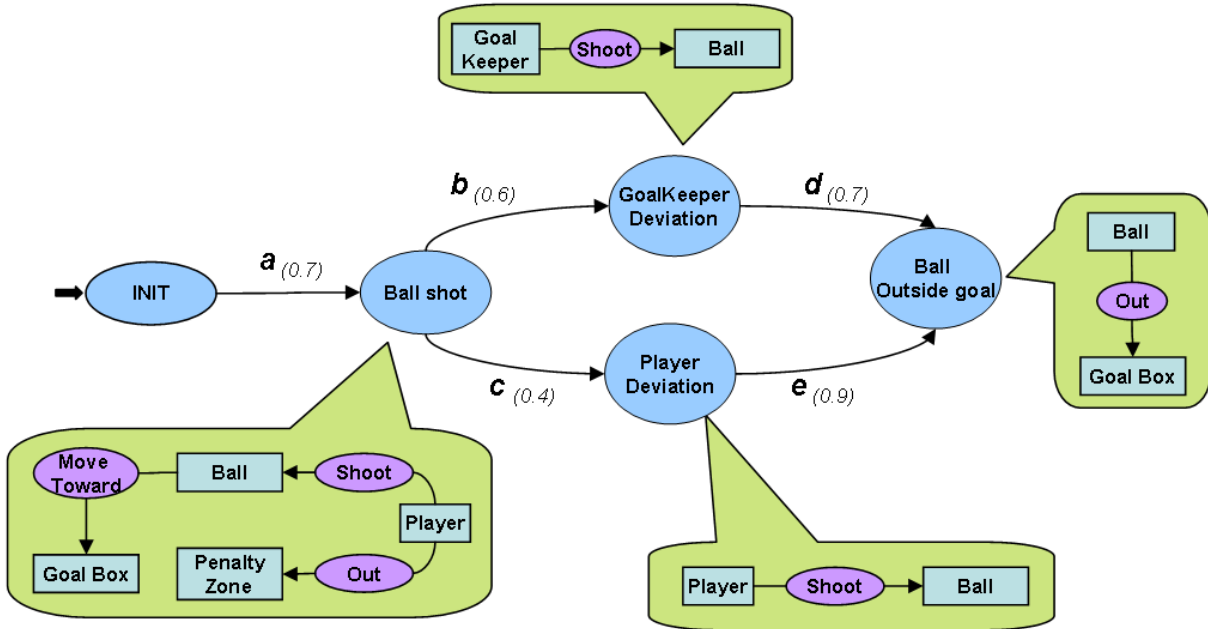


Figure 2.3: Example of an event model: Deviated shot on goal from outside the penalty zone.

2.3.4 Hierarchical description

This representation formalism can be seen as a bridge between different levels of semantic abstraction. The lower level can contain basic features that can be detected using image processing techniques and the most used tags in manual annotation. Concepts of this layer can be used to describe more complex ones in an upper level that can be used at their turn to form richer and more complex events in a higher abstraction level. This leads to a hierarchical description of events combining concepts from different abstraction levels (Figure 2.4).

This concept is very useful during the design of high level detectors, rather than expressing the event using the low level features (color, shape,...), midlevel concepts (composing objects)

can be used to describe concepts. This also helps to satisfy interoperability between description systems and to keep the definition of the event correct in all conditions. Logical and spatial re-

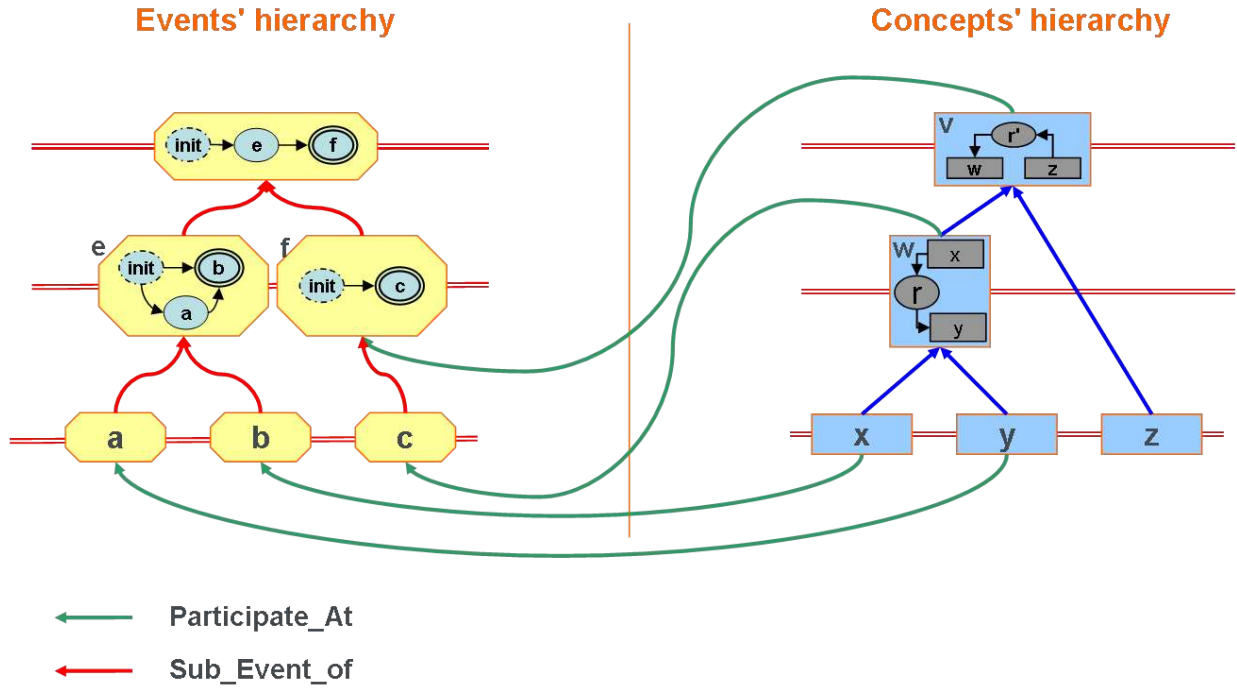


Figure 2.4: Hierarchic description of complex objects and events.

lations relying objects to form a situation can be defined in a hierarchical way too. While the relations in the lower level should be related to algorithms dealing with visual features and enabling for their verification within a frame, complex relations should be defined using logical operators. Figure 2.5 depicts the definition of the relation *IN* based on the basic relations *EQ* (Equal), *TPP* (Tangential Proper Part) and *NTPP* (Non Tangential Proper Part) in a 2D vision context. These binary relations are parts of the well known system of spatial relations RCC8 [117]. We can write :

$$IN(x, y) = EQ(x, y) \cap TPP(x, y) \cap NTPP(x, y)$$

where x and y are the objects concerned with the relation *IN*.

Figure 2.6 depicts a hierarchic definition of the event of "*Stranger enters a car*" which is pretty difficult to automatically detect. Using two automatic detectors for the objects *Car* and *Person* and a face recognition algorithm for identifying the "*Authorized*" people to get in the car, the detection of this complex event becomes easier. For this aim, we define an intermediate level containing the two situations *Empty Car* and *Occupied Car*. Using this two situations and the concepts detections, we define the *car theft* event model. The model expresses the fact that the event *Stranger enters a car* occurs when the supervised car becomes occupied by a person who is not authorized to access the car.

2.4 Video Guided monitoring of behavior

The language is designed in order to bridge the gap between the high and low level visual concepts and to facilitate recognition of complex events in video clips based on low level objects

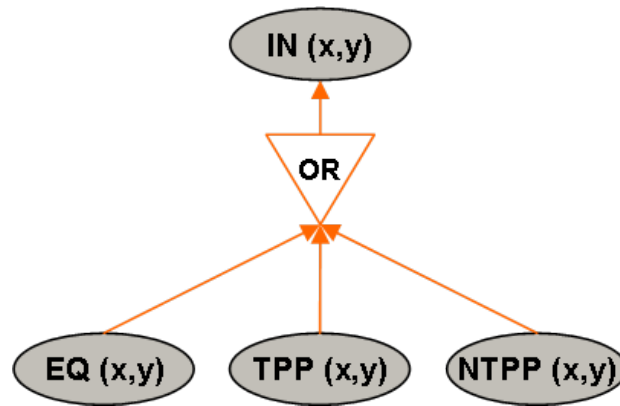


Figure 2.5: Example of hierarchic description of a complex relation

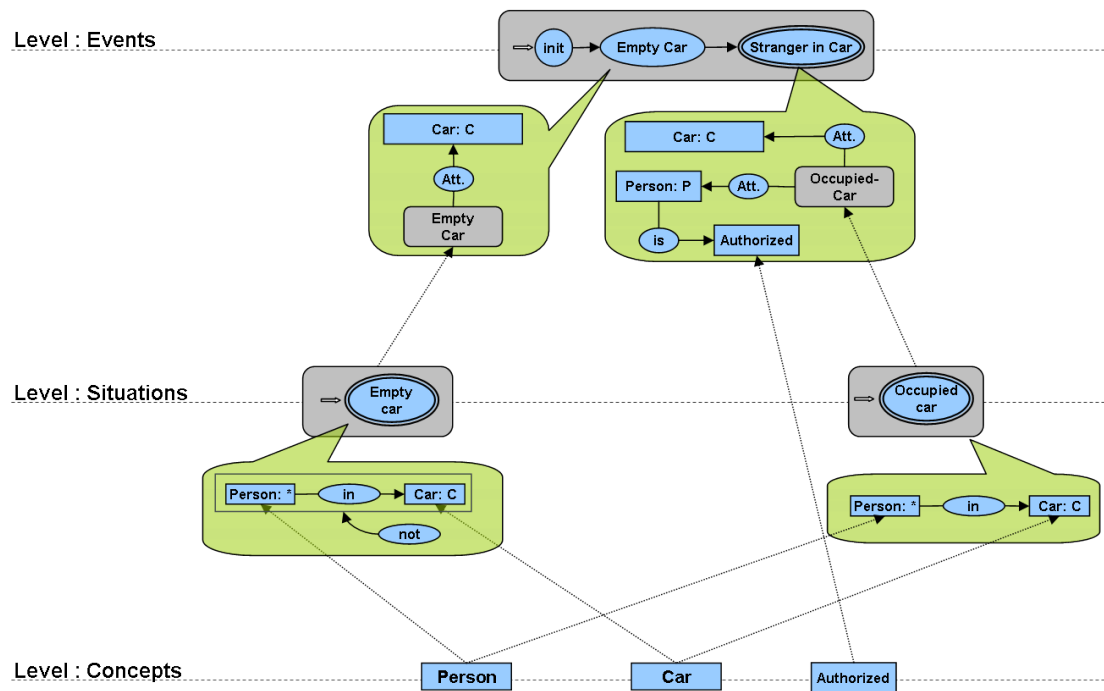


Figure 2.6: Example of hierarchic representation of a complex visual event

and relations. Figure 2.7 shows the correspondence between the event model of *Goal* and a real video sequence. The used event models can either be specified by domain experts or by non expert users aiming to describe and detect their typical events on video clips. This is done via appropriate interactive interfaces.

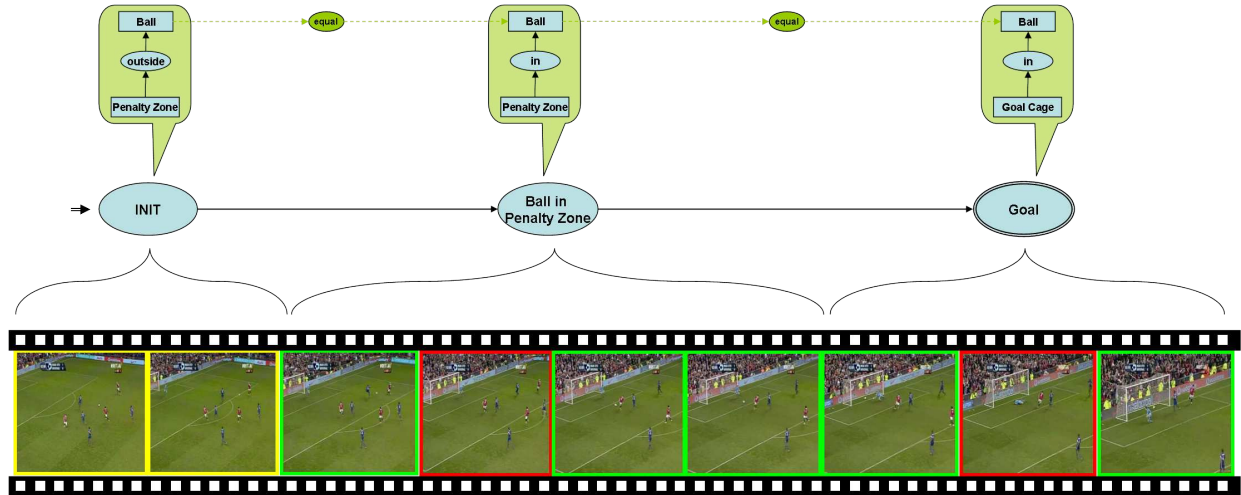


Figure 2.7: Detecting the event "Goal" in a soccer video

Based on our modeling language, we build a framework for real time behavior monitoring within video broadcast. This application requires the description of the monitored behavior using the appropriate model of the event to detect. It also requires the use of fast low level detection algorithms. All the spatial and temporal constraints should be expressed using the formalism taking into consideration all possible behavior cases. Automata are extended by modules for the definition of alerts, strategies and decisions that have to be launched depending on the current state.

2.4.1 Monitoring protocol construction

Monitoring behaviors via real time broadcast allows for video surveillance of a limited zone where the used cameras moves rarely. The behavior is then described by constituting the model describing it in a semiautomated way (figure 2.8). This is done according to the following next steps:

- Automatic background extraction: the framework automatically extracts the background of the monitored area.
- Manual segmentation and description : the user segments the background into regions and annotates them using predefined domain ontology. The actors entering in the execution of the behavior are also added to the description model. Low level features (shape, texture, color ...) related to regions and to the added actors are also extracted.
- Defining interesting objects: the user is invited to introduce pictures of objects to be recognized and identified during the occurrence of the event. In the case of figure 2.8, we are interested in detecting the event of "car theft". The interesting objects are the photos of the

faces of people "authorized" to get in the car and those of the monitored car taken from different sides.

- Interactive definition of event model: the framework performs an interactive process for the specification of the automata representing the event to monitor. We call that event model the monitoring protocol. The user is assisted in specifying the important states that compose the protocol, in identifying the corresponding conceptual graphs for each state and in associating the adequate decisions to be taken and the alerts to be launched in each state.

At the end of this operation, the monitoring protocol is complete and ready to be used.

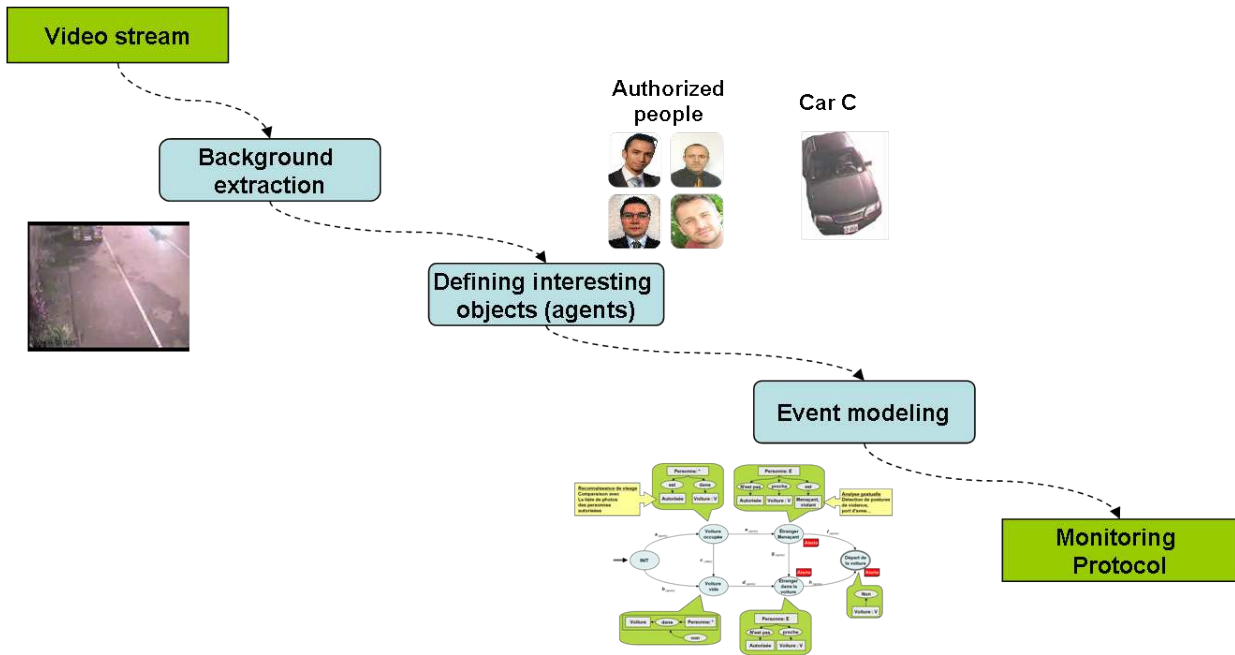


Figure 2.8: Monitoring protocol construction process.

2.4.2 Event detection using monitoring protocols

2.4.2.1 General process

After producing the *Monitoring Protocol*, the real time video guided behavior monitoring is performed by the *Event Detector* module of the framework (figure 2.9). The user can fix a number N so that the algorithm will analyze one frame each N frames of the video stream. We mention this frame the "chosen frame"

1. At the beginning, the monitoring index, a simple pointer used to specify the current state of event, is set to the initial state of the model of the event to be monitored.
2. Then, the algorithm gets the next frame to analyze (the N^{th} frame), the detector extracts its content graph making use of low-level objects detectors and recognition algorithms to identify the interesting objects.

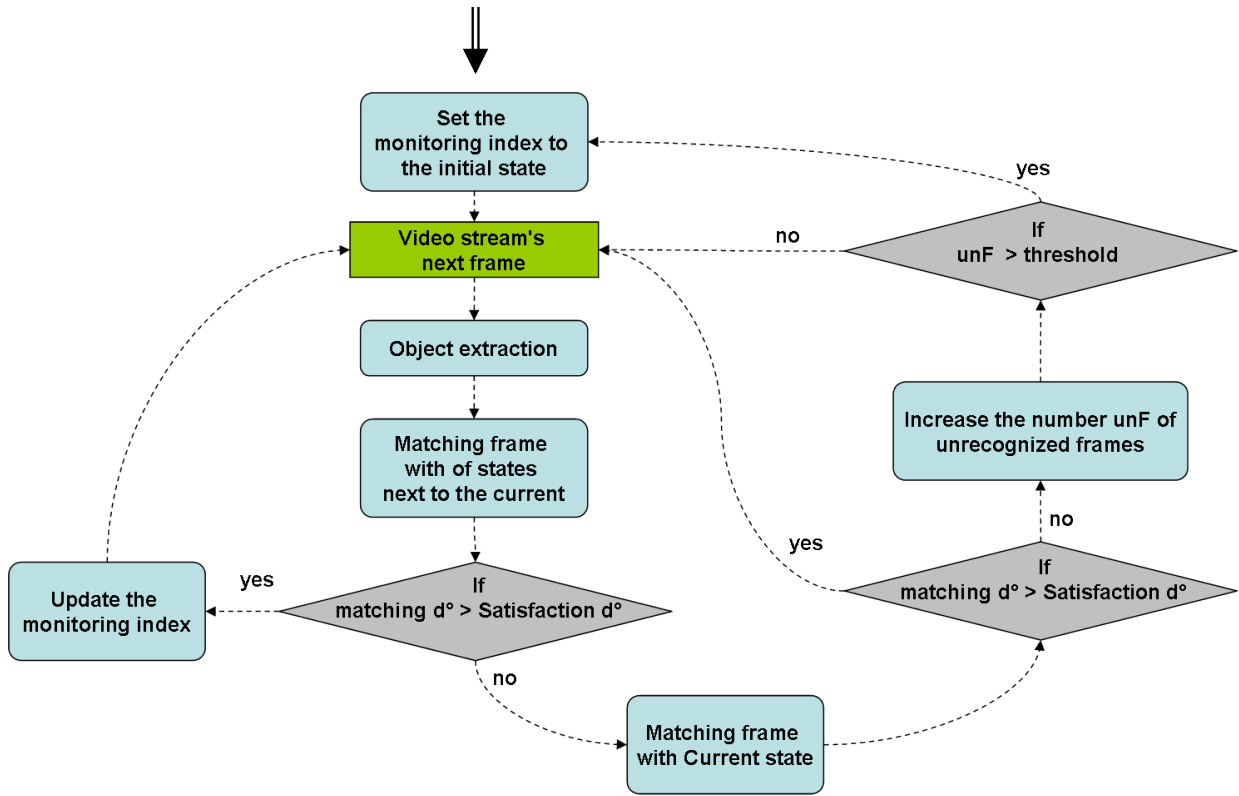


Figure 2.9: Video guided Monitoring process.

3. Using the algorithm "Matching" (Algorithm 3), the frame graph is mapped to each graph of successor states of the current state, but also to the graph of the current state. For each state-to-frame matching, a matching degree is saved.
4. The algorithm selects the state returning the higher matching degree, updates the monitoring index to point to that state, and lunches the appropriate alert associated to the new state. If no matching degree exceeds a fixed threshold (a satisfaction degree chosen by the user), the frame is indicated as unrecognized and an indicator unF of the number of unrecognized frames is increased.
5. If unF exceeds a threshold uTh fixed by the user, the algorithm goes to step n°1, otherwise, it goes to step n°2.

2.4.2.2 Detection Algorithms

Let $V = \{f_0, f_1, f_2, \dots, f_n\}$ be the analyzed video represented by its frames, and $M=(P,G)$ the model of the tracked event within this video. The detection of the event in the video is done using three major algorithms.

ModelOccurrence The algorithm *ModelOccurrence* (Algorithm 1) is recursive. It takes, at each step, as input the current frame f_i , the current state s , the number of unrecognized frames during event occurrence unF , and the frame where the event starts at each occurrence *start-Frame*. It returns the list of all occurrences of the event M in V . The algorithm starts with

($i=0$). If the frame f_i matches the state s with a degree higher than the satisfaction match threshold sTh , the algorithm initializes **startFrame** if s is initial and checks the next frame unless s is final. If f_i does not match s , the algorithm tries to match the f_i with all the successors of s in P . If no successor of s matches f_i the algorithm increases **unF** if the event has began. The algorithm then checks the next frame f_{i+1} but only if **unF** is lower than the threshold uTh . The algorithm continues frame by frame until the end of the video and returns the list of correct occurrences of the event M in the video.

objectInstances This algorithm is recursive (Algorithm 2). It aims at extracting all instances of a set of objects G in a specified video frame f . It begins by extracting all the stored manual annotations of the frame f and verifies the occurrence of objects of G in these annotations. Then, it extracts annotations of each object o in G . If o is a *basic object* it applies the detection algorithms on the frame f , otherwise (o is a *complex object*), it extracts the set of objects composing o and calls **objectInstances** on this set.

Matching Algorithm 3 computes the degree of satisfaction of the visual graph associated to a state in an event model s by the object instances of a video frame f (figure 2.10). It takes as input a set of instances I and a state s of an event model. The algorithm computes the best combination of instances that returns the higher certainty coefficient of satisfaction of the relations in the graph of s based on the responses provided by algorithms dealing with low level features algorithms. Combinations are calculated as follows:

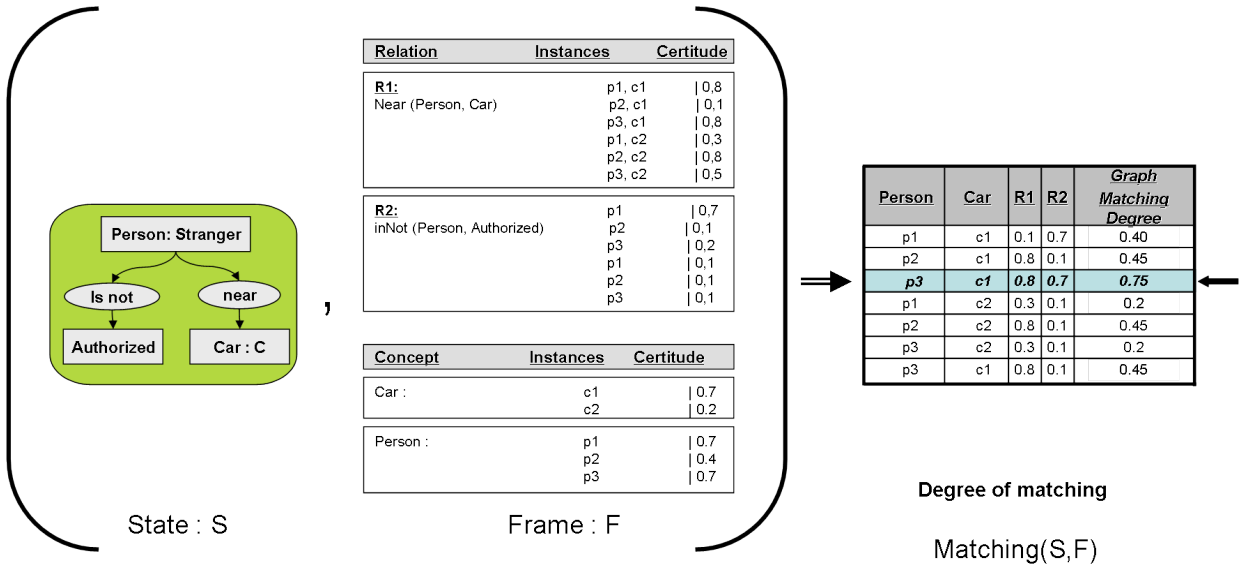


Figure 2.10: Matching object instances of a video frame to an event state

Let a state S the associated to a conceptual graph $G_S = (C_S, R_S, A_S)$ where C_S is the set of concepts involved in the situation, R_S is the set of relations, while A_S is the set of attributes. Consider a frame where a set of concepts C_I were detected. The objective is to verify whether the set of concepts C_I matches the conceptual graph G_S . First, among C_I concepts, we constitute all sets of concepts that are candidate to match the concepts C_S . Let $L_S = [c_1, \dots, c_n]$ be a n -ary tuple containing all concepts of C_S arbitrary ordered. We define $\Gamma(L_S, C_I)$ as the set of all n -ary tuples of concepts of $C_I = c'_1, \dots, c'_m$ that respect the order of concept types of L_S . $\Gamma(L_S, C_I)$ is defined as follows:

$$\begin{aligned}
L_S &= [c_1, \dots, c_n], \\
C_I &= \{c'_1, \dots, c'_m\} \\
\Gamma(L_S, C_I) &= \{[c'_{p+1}, \dots, c'_{p+n}] \in (C_I)^n \mid \\
&\quad \text{type}(c'_{p+i}) = \text{type}(c_i) \forall i \in \{1, \dots, n\}\},
\end{aligned}$$

Let U be a tuple in Γ , and $U = [c'_{p+1}, \dots, c'_{p+n}]$. We note $\text{corresp}(c_i) = c'_{p+i}$.

For instance, let us consider :

$$\begin{aligned}
L_S &= [(c_1, \text{vehicle}), (c_2, \text{box}), (c_3, \text{person})], \\
C_I &= \{(c'_1, \text{car}), (c'_2, \text{box}), (c'_3, \text{bus}), (c'_4, \text{building}), (c'_5, \text{tree}), (c'_6, \text{man}), (c'_7, \text{woman})\}
\end{aligned}$$

We have :

$$\begin{aligned}
C_{I/\text{type}(c_1)} &= \{c'_1, c'_3\}, \\
C_{I/\text{type}(c_2)} &= \{c'_2\}, \\
C_{I/\text{type}(c_3)} &= \{c'_6, c'_7\},
\end{aligned}$$

and then $\Gamma(L_S, C_I)$ is defined as :

$$\begin{aligned}
\Gamma(L_S, C_I) &= \{U_1 = [c'_1, c'_2, c'_6], \\
&\quad U_2 = [c'_1, c'_2, c'_7], \\
&\quad U_1 = [c'_3, c'_2, c'_6], \\
&\quad U_2 = [c'_3, c'_2, c'_7]\}
\end{aligned}$$

The matching between a state graph G_S and image concepts C_I is performed by using algorithm 3.

Automatic referring and video surveillance are relevant fields to apply this process.

2.4.3 Use Case: Car Theft

In order to illustrate our approach, a monitoring protocol related to the use case of *car theft* is designed and then used to detect the event of theft of a car in real time video stream. The monitoring protocol of figure 2.11 describes the main possible situations where car theft can occur. Using simple objects like Car, Person, and Gun, the protocol describes three scenarios. The path P1={INIT,a,Occupied car,f,Stranger enters car,h,car departure} that describes a forced car theft, the path P2={INIT,a,Occupied car,e,Stranger with gun,g,Stranger enters car,h,car departure} that describes a forced car where the thief uses especially a gun, and then the path P3={INIT,b,Empty-Car,d,Stranger-With-Gun,g,Stranger-Enters-Car,h,car departure} that describes a stranger who steals the car when it is empty. Figure 2.12 shows the result of a performed *car theft* detection using the monitoring protocol and the detection framework.

2.5 MPEG-7 Annotation Validation

MPEG-7 is a description standard used to create complex and comprehensive metadata descriptions of multimedia contents [1]. However, XML Schema, used by MPEG-7, is a language for constraining syntactic structures and not for describing semantics. High level visual concepts


```

end if
return occList

```

Algorithm 2 *objectInstances(G,f)*

Require: $M(f) \leftarrow \text{ManualAnnotations}(f)$ I : Object instances List**Ensure:** *objectInstances(G,f)* $I \leftarrow \text{empty}$ **for each** $c \in G$ **do** $I \leftarrow \text{instances of } c \text{ in } M(f)$ **if** c is *basic object* **then** $I \leftarrow I \cup \text{BasicDetection}(c,f)$ **else** $T \leftarrow \text{graph of objects of } c$ $I \leftarrow I \cup \text{objectInstances}(T,f)$ **end if****end for****return** I

Algorithm 3 *Matching(S,I)*

Require: $G_S = (C_S, R_S) \leftarrow \text{conceptual graph associated to } S$ $C_I \leftarrow \text{set of objects composing the frame } I$ $\Gamma \leftarrow \Gamma(G_S, C_I)$ set of tuples of I candidate to match G_S $R \leftarrow \text{set of relations in the state graph}$ **Ensure:** *Matching(S,I)* $\text{maxCoef} \leftarrow 0$ **for all** $U \in \Gamma$ **do** $\text{matchCoef} \leftarrow 0$ **for all** $r \in R$ **do****if** $r(\text{corresp}(r.\text{source}, U), \text{corresp}(r.\text{target}, U))$ is true **then** $\text{matchCoef} \leftarrow \text{matchCoef} + 1$ **end if****end for** $\text{matchCoef} \leftarrow \text{matchCoef} / \text{size}(C_S)$ **if** $\text{maxCoef} < \text{matchCoef}$ **then** $\text{maxCoef} \leftarrow \text{matchCoef}$ **end if****end for****return** maxCoef

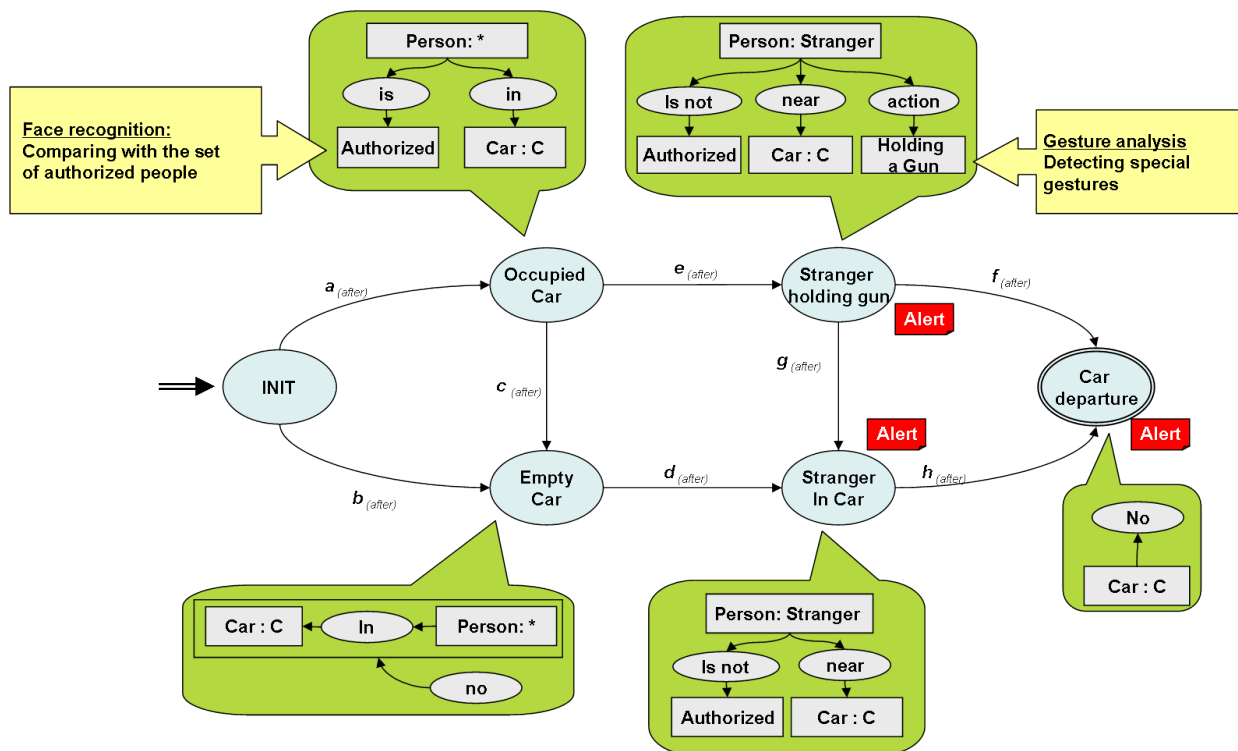


Figure 2.11: Car Theft Monitoring Protocol

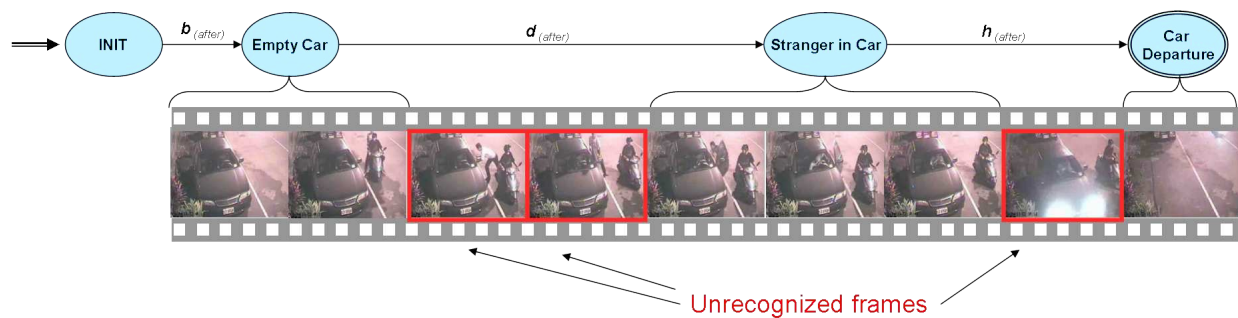


Figure 2.12: Car theft detection in real time video stream

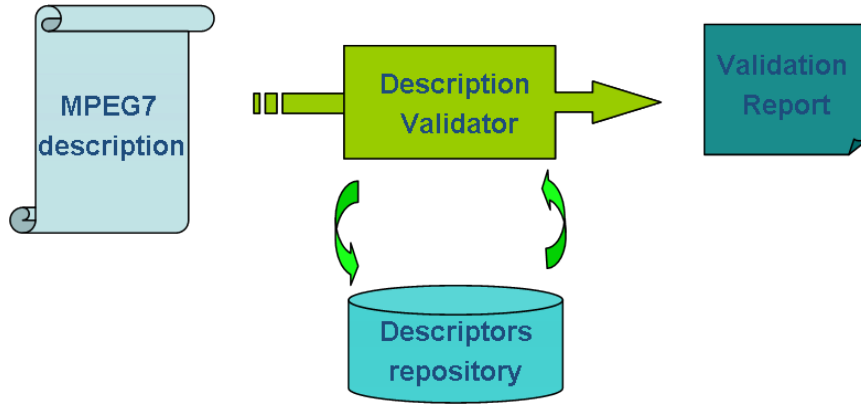


Figure 2.13: Semantic validation framework architecture

and events can be described in multiple ways, which affect the interoperability and the automatic use of MPEG-7 based descriptions. A lot of works have been devoted to the structural and syntactical validation of MPEG-7 based video descriptions [63]. Detailed Audiovisual Profile (DAVP) has to been proposed to specify the use of the descriptors in a particular context and then to avoid semantic ambiguities. Nevertheless, the semantic constraints of the DAVP cannot be formalized using XML Schema and thus cannot be checked for full semantic conformance. Descriptions would still conform to the schema, although they would violate the non-formally represented semantic constraints of DAVP. In [147], authors have been interested in checking the consistency of temporal and spatial descriptions (such as invalid time specification or negative segment duration) while using DAVP. They have proposed the use of semantic Web languages to express DAVP constraints, and proposed inference tools to check the semantic consistency of the descriptions.

However, few works have been devoted to the validation of the semantic composition of event occurrences and then their semantic validation. In fact, a goal event in soccer game should begin by shooting the ball and then the entering of the same ball in the goal box. The verification of the order of situation in occurrence of events inside a MPEG-7 description of videos is not easy. In our system, we enable the validation of event MPEG-7 based descriptions and the satisfaction of the correct semantic temporal and spatial structure of events. Considering an event modeled using the description language defined previously (section 2.2), each MPEG7 file describing the occurrence of such an event is mapped to the model of this event to validate its spatio-temporal decomposition. The process of MPEG7 description validation is defined as follows:

- Extraction of the execution paths from the event model: each execution path represent a possible way an event can occur within a video. It represents a correct chronological decomposition of this event. Since each event is associated with an automaton, where the states are the spatial situations happening during the occurrence of the event, then the execution path corresponds to a sequence of states that start from the initial state of the automaton and ends with its final state. The set of such paths is defined as $E_M = \{e = s_0.s_1 \dots s_n | s_n \in F(M), s_0 = s_0 = \text{initState}(M)\}$, where M is the event model. In figure 2.14: $E_M = \{A.B.D, A.C.D, A, B, C, D\}$ where $S_M = \{A, B, C, D, E\}$.
- Extraction of spatial structure corresponding to each state in the automata: Each state is associated to a conceptual graph that describes the spatial objects occurring in the situation and the spatial relations gathering them together. The set of such structure is de-

defined as $Ss_M = \{(C, R, T) | \exists s \in S(M), (C, R, T) = \text{graph}(s)\}$ (section 2.2). In the same figure, $Ss_M = \{\text{graph}(A) = (a, r1(a, a), \emptyset), \text{graph}(B) = (a, b, r2(a, b), \emptyset), \text{graph}(C) = (a, c, r3(a, c), \emptyset), \text{graph}(D) = (a, b, c, r2(a, b), r3(a, c), r4(b, c), \emptyset)\}$

- Extraction of spatial and temporal structures of the MPEG7 visual description. This information is contained in the *Location* and *Basic Elements* descriptors which represent the region locator, the spatiotemporal locator and the spatial 2D coordinates [149] of the described resource. From the description file we extract the temporal segments of the event, and then for each segment we extract the still regions composing it. Then we compute the relations between the regions based on their 2D coordinates. In the same example, the description is decomposed into the temporal sequences X, Y, Z , where the state X is decomposed into regions x, y and z . Figure 2.15 depicts the MPEG-7 based description from where the information where extracted.
- Finally, the matching algorithm (Algorithm 3) is then used to verify whether the spatio-temporal decomposition $\{X, Y, Z\}$ of the event in the MPEG-7 file can be mapped to an allowed possible execution path in the event model.

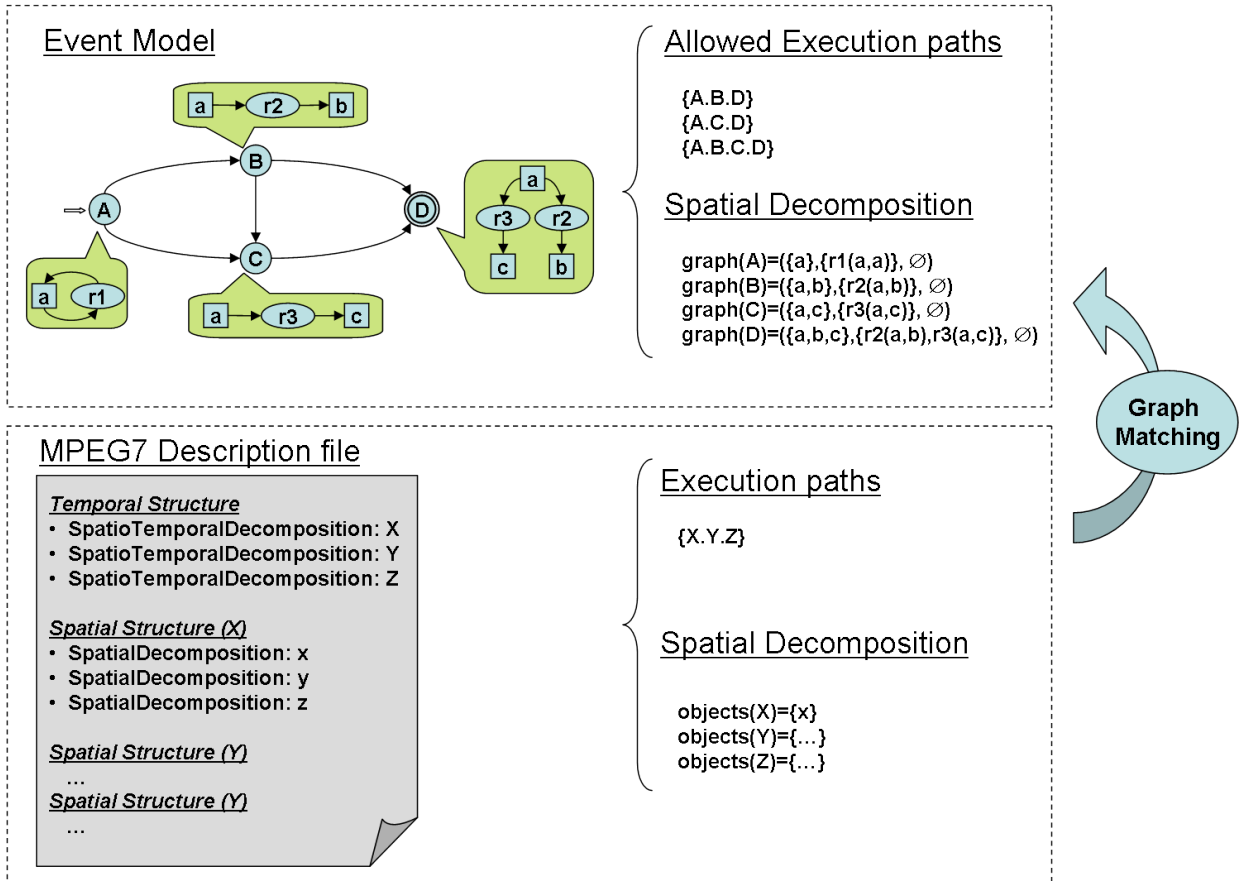


Figure 2.14: Validation method of a MPEG-7 Description

```

<MPEG7 type="complete" xmlns="http://www.mpeg7.org/2001/MPEG-7_Schema"
  xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
  xsi:schemaLocation="http://www.mpeg7.org/2001/MPEG-7_Schema">
  <ContentDescription xsi:type="ContentEntityType">
    <MultimediaContent xsi:type="VideoType">
      <Video>
        <TemporalDecomposition gap="false" overlap="false">
          <VideoSegment>
            <MediaTime>
              <MediaTimePoint> T00:00:00 </MediaTimePoint>
              <MediaDuration> PT0M15S </MediaDuration>
            </MediaTime>
            <TextAnnotation type="scene description" relevance="1" confidence="1">
              <FreeTextAnnotation> Event M </FreeTextAnnotation>
            </TextAnnotation>
            <SpatioTemporalDecomposition>
              <TextAnnotation type="scene description" relevance="1" confidence="1">
                <FreeTextAnnotation> X </FreeTextAnnotation>
              </TextAnnotation>
              <StillRegion>
                <MediaIncrTimePoint timeUnit="PT1N30F"> 421 </MediaIncrTimePoint>
                <SpatialDecomposition>
                  <StillRegion>
                    <TextAnnotation>
                      <FreeTextAnnotation> x </FreeTextAnnotation>
                    </TextAnnotation>
                    <SpatialLocator>
                      <Poly>
                        <CoordsI> 41 135 290 135 290 230 41 230 </CoordsI>
                      </Poly>
                    </SpatialLocator>
                  </StillRegion>
                </SpatialDecomposition>
                ...
                <SpatialDecomposition>
                  ...
                  <FreeTextAnnotation> y </FreeTextAnnotation>
                </SpatialDecomposition>
                ...
                <SpatialDecomposition>
                  ...
                  <FreeTextAnnotation> z </FreeTextAnnotation>
                </SpatialDecomposition>
                ...
              </SpatioTemporalDecomposition>
            </SpatioTemporalDecomposition>
            ...
            <FreeTextAnnotation> Y </FreeTextAnnotation>
          </SpatioTemporalDecomposition>
        </SpatioTemporalDecomposition>
        ...
        <FreeTextAnnotation> Z </FreeTextAnnotation>
      </SpatioTemporalDecomposition>
    </VideoSegment>
  </TemporalDecomposition>
</Video>
</MultimediaContent>
</ContentDescription>
</MPEG7>

```

Figure 2.15: Example of MPEG-7 video segment description

2.6 Conclusion

In this chapter, we introduced a new generic language of semantic representation of video content. This language is built based on two formalisms, finite state machines and conceptual graphs, to represent respectively temporal and spatial structure of visual events and to recognize them using a detection framework. The resulting formalism enables the description of high level semantic descriptions and bridges the gap between the different abstraction levels. In addition to automatically extract events from a video database, the frameworks allow to monitor behavior in a video-surveillance setting and to validate MPEG7-based descriptions.

The proposed semantic video model is especially designed for real time video monitoring. In this case one can suppose that experts have a clear idea about the structure of events and can translate the human knowledge needed for representing events into well-defined rules. However, cautions should be taken while building the monitoring protocols, the defined models should be neither too general nor too detailed. The first case will cause false detections, while the latter will result in many false rejections. An example of a too general description is the one defining a playfield by turf surface. Indeed turf can be also found in parks and gardens. An example of a too detailed description is the one asserting that the width of the soccer playfield is about 90 meters. In fact, dimensions of soccer palyfields vary from a stadium to th other. It is of major importance that all major objects, relations, attributes, and situations identifying the event, and only them, be included in the description of the event model. For this aim, heuristics, automatic learning or manual inquiry on videos containing occurrences of the event to be detected can be performed. Results can help the user to identify the key object, relations, attributes, and situations composing an events.

We suppose also, for this kind of applications, that concepts and relations are crisp since in general the question is to decide whether or not the video stream shows that a phase of execution is achieved. No ranking of results is expected after detecting the event in such applications. Implementation details about this chapter are provided in Chapter 5.

Another problem with this model is that it imposes a crisp definition of concepts and relations between them. However, for other applications such as off-line video analysis, additional capabilities should be provided while reasoning about spatio-temporal features of video contents. The image processing algorithms and the textual annotation of videos are also error-prone processes which make it necessary to take into consideration uncertainty in description and detection of events. This issue is addressed in the next chapter.

There is nothing worse than a sharp image of a fuzzy concept.

Ansel Adams (1902 - 1984)

3

Uncertainty Handling in Semantic Video Retrieval Using Fuzzy Conceptual Graphs

▷ *In this chapter, we introduce a new method for classifying video segments and detecting complex events based on graph matching by taking into account uncertainty. The work makes a distinction between the Event Model Graph that represents the query, and the Video Segment Graph that represents the video contents. It proposes a new variant of fuzzy temporal and spatial relations issued from temporal Allen's algebra and spatial RCC8 relations. It also introduces new similarity measures between components of the two graphs and then proposes an algorithm for matching the two graphs and verifying the occurrence of specific events within the video documents.* ◁

Contents of the chapter

3.1	Introduction	59
3.2	Contribution	59
3.3	Fuzzy Conceptual Modeling of Video Data	59
3.3.1	Fuzzy Conceptual Graphs	62
3.3.2	Fuzzy Spatial and Temporal relationships	66
3.4	Graph Matching	76
3.4.1	Match degrees	76
3.4.2	Matching algorithms	78
3.5	Conclusion	83

3.1 Introduction

Due to the inherent nature of video information, uncertainty is often taken into account when representing semantic content. Two major sources of uncertainty in semantic media content descriptions can be distinguished.

- The first source is the human perception of media content. In fact, while describing the events they wish to retrieve, users are not always sure about relations gathering objects involved in the occurrence of the event. This encourage them to use vague or subjective concepts commonly used in real world (near, far, crowded, small,...) [118]. As an example, users may look for an event "a man near a car". They explicitly specify the objects 'man' and 'car' but they rarely specify how much the two objects are near to each other (4m, 1m,...).
- The second source of the uncertainty is imprecision and errors issued from automatic classification of video objects. Indeed, there can be a lot of variability in the appearance of an object in a multimedia resource. After the segmentation process, associating a segment to a concept type is generally an imprecise process that is associated with probability rate due to the variability. This variability makes recognition a difficult and error-prone process. Variability comes from the object itself (different spatial orientations, different colors, ...), or from the scene in which the object is immersed (occlusions with others objects, different lightning and shadows ...). Additionally, manual annotations generates errors due to limitations of annotation tools that do not provide descriptors sufficiently adapted to the format of video objects, but also to the limited time and effort devoted by the user to annotate the video object. This difference in sources of uncertainty is shown figure 3.1

3.2 Contribution

In this chapter, a new variant of fuzzy conceptual graphs, suitable for handling uncertainty in visual event description and retrieval, is presented. We deal with two types of graphs according to the sources of uncertainty. New variant of fuzzy spatial and temporal relationships are defined to capture imprecision in video content spatiotemporal features. Moreover, similarity measures and matching algorithms are defined to assess the degree of match between the components of video and the event model and then to localize events within video segments. This work can be seen as an extension of the model proposed in the previous chapter by including handling of uncertainty and providing more expressive formalism. The part of the integrated framework concerned by this chapter is highlighted in the 3.2.

3.3 Fuzzy Conceptual Modeling of Video Data

Among Knowledge Representation (KR) formalisms, *Conceptual Graphs* [135] constitute an interesting formalism for representing content knowledge within video documents. However, they are not suitable to deal with the inherent issue of uncertain multimedia content description.

Developed by Lotfi Zadeh, *Fuzzy Logic* [168] is a superset of *Boolean logic* that has been extended to handle the concept of partial truth. The main concept of this theory is to represent truth

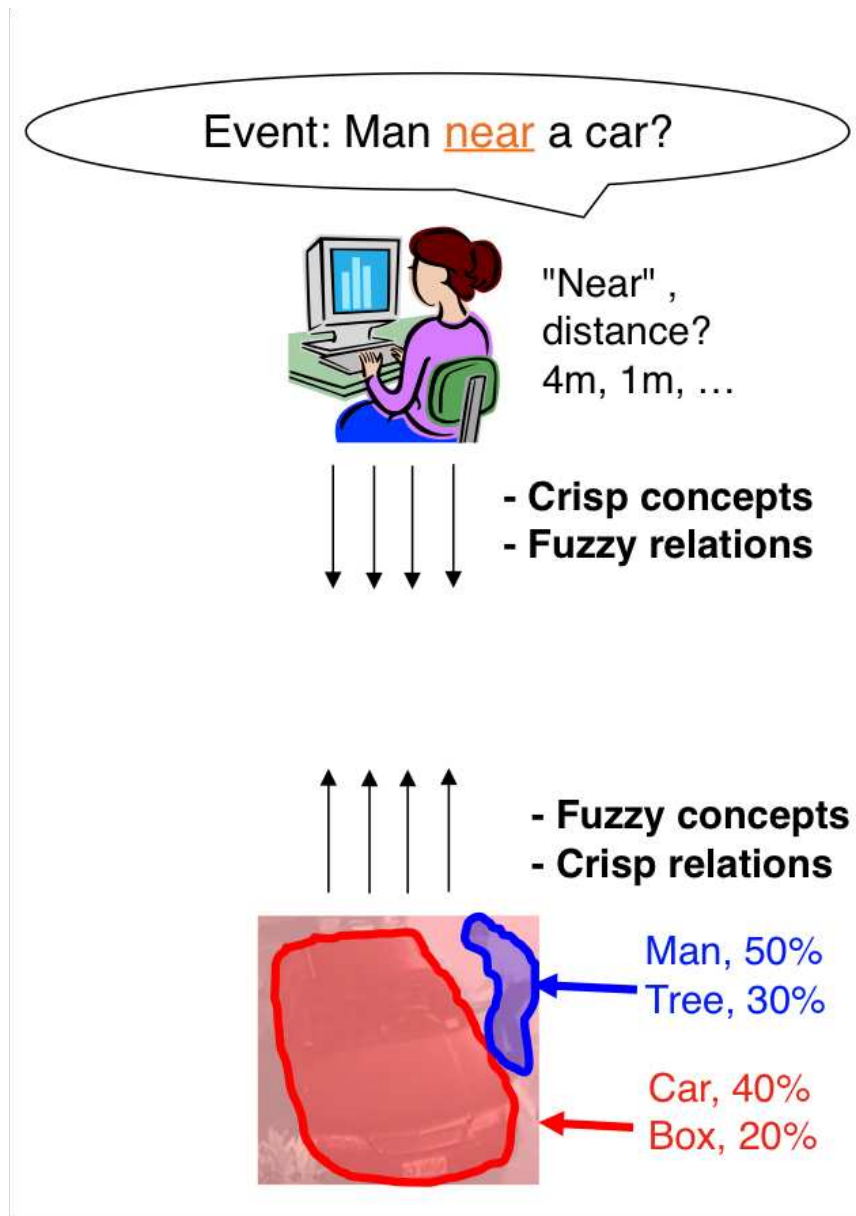


Figure 3.1: Different kinds and resources of uncertainty in event retrieval.

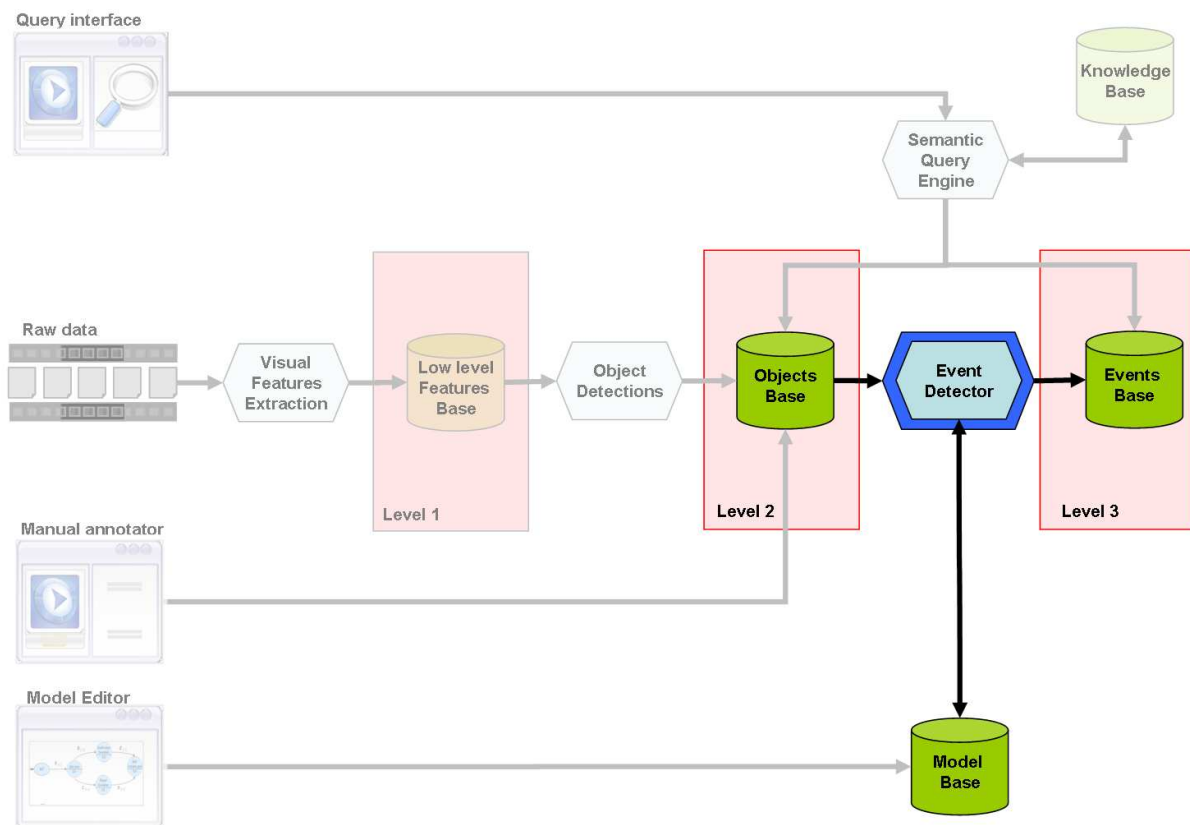


Figure 3.2: Positioning the chapter contribution within the whole framework. It consists of an extension of the event detector with advanced reasoning capabilities.

values or membership values by a value on the range $[0.0, 1.0]$, with 0.0 for absolute Falseness and 1.0 for absolute Truth.

Based on this logic, fuzzy Conceptual Graphs were introduced as a new KR formalism more adapted to uncertain information. In [105], concepts, relations and relation attributes are used to introduce a variation of fuzzy conceptual graphs for image classification and analysis. This approach defines two types of fuzzy conceptual graphs: the *Model Graph* that describes a known scene and an *Image Graph* that describes an input image. Then, it proposes a similarity measure between these two types of graphs in terms of graph projection. This approach considers only errors and imprecision in assigning image segments to concept types. However, it is clear that spatial and temporal positioning of objects occurrence is also a major source of errors and imprecision, which affects spatial and temporal relations in the Image Graphs. Also, this work was restricted to uncertainty related to spatial properties of image objects. Applying such an approach to video documents requires the study of uncertainty related to temporal properties of objects occurrences.

Other interesting works for incorporating uncertainty representation into semantic models can be found in the literature. One of them is the work presented by Stoilos et al. in [137]. This work concerns OWL DL, one of the key languages in Semantic Web, yet not supporting in specification of vague and imprecise information. Extension based on Fuzzy logic was proposed to enable OWL to deal with fuzzy concepts. However, no additional extensions, specific to multimedia context, were described.

In [138] the simple description logic ALC was extended with fuzzy logic in order to support reasoning about imprecise concepts. A concept C of the fuzzy DL is interpreted as a fuzzy set and the assertions associating an individual to a concept or a couple of individuals to a role are given a truth value in $[0,1]$ representing a degree of membership. SHOIN(D) description logic [95] is a powerful language allowing to reason with concrete data types such as strings or integers using so-called concrete domains. In [139], an extension of the SHOIN(D) with fuzzy logics is presented. It provides further capabilities especially by using fuzzy sets of concrete domains and fuzzy modifiers, and by allowing values from the interval $[0, 1]$ for subsumption relationship.

3.3.1 Fuzzy Conceptual Graphs

The complex nature of visual events requires new models able to support spatial, temporal and logical relationships gathering together basic component of video content in order to semantically express high level contents within video documents. Our model is built by combining conceptual graphs, fuzzy logic, region connection calculus (RCC8), *Allen's* interval algebra and logical reasoning within a unique and powerful core language more suitable for video content description. We define a fuzzy conceptual graph fCG as $G(C, R, A)$ where C is a set of fuzzy concepts, R is a set of fuzzy relations, and A is a set of fuzzy attributes. These descriptors are defined as follows:

- A **fuzzy concept** c is a couple (e, S) where e is a referent and $S = \{(t_1, f_1), \dots, (t_n, f_n)\}$ where $(t_i, i \in \{1, \dots, n\})$ is a concept type and f_i ($i \in \{1, \dots, n\}$) is the degree of certitude that the object whose referent is e is of type t_i ($0 \leq f_i \leq 1, \forall i \in \{1, \dots, n\}$). t_i can be a generic concept type but also an individual object, a proper noun, etc. A crisp concept is a particular fuzzy concept where S is a singleton $\{(t, 1)\}$. In our model, concepts correspond to objects involved in event composition.

- A **fuzzy relation** r is a couple (t, f) where t is a relation type and $f \in [0, 1]$ is the probability of occurrence of the relation t . A crisp relation is a particular fuzzy relation where $f = 1$. When linking spatial objects, t is an RCC8 spatial relation, while when linking temporal intervals, t is an Allen's temporal relation. t can be a complex relation formed by a conjunction or a disjunction of primitive relations.
- A **fuzzy attribute** a is a triple (t, v, f) where t is an attribute type, v is a value and $f \in [0, 1]$ indicates the probability that the value of the attribute t is v . A crisp attribute is a particular fuzzy attribute where $f = 1$. A *fuzzy attribute* can be associated either with a *fuzzy concept* or with a *fuzzy relation*.

Uncertainty in video content descriptions stems from two major sources, the human perception of the event which causes different types of queries, and the errors and imprecisions in description and extraction of video indexes. Usually, in Information Retrieval processes, it is common to present three distinct parts: the query model, the document model, and the matching function to project the document into the query. In order to correctly deal with uncertainty representation and narrowing the semantic gap while extracting complex visual events, we introduce four types of fuzzy conceptual graphs, two document graphs, and two query graphs :

- Content Graphs (Document):
 - Image Graph: It is a conceptual graph that describes a still image by citing the appearing content, the spatial relations between the objects, and attributes related to those relations and to objects. The same Image graph can correspond to a sequence of consecutive images as long as spatial relations and attributes remain unchanged.
 - Video Graph: It is a conceptual graph that describes the content in a video segment. It is composed of many image graphs that can be connected together using temporal relations. Image graphs within a video graph can also be associated to attributes that describe their temporal interval of occurrence.
- Model Graphs (Query):
 - Situation Graph: It is a conceptual graph that describes the semantic structure of a spatial situation that involves objects connected by spatial relations.
 - Event Graph: It is a conceptual graph that describes semantic structure of a spatio-temporal event involving objects that change their positions and attributes over time. It is composed of many situation graphs that are connected through temporal relations.

The *Event Graph* represents the perception a human can have regarding the composition of a specified event (query), and the *Video Graph* describes the contents of a specific video segment produced by automatic or manual annotations. The goal is to correctly match the two types of graphs to answer the user query.

3.3.1.1 Event Graph.

It represents the semantic structure of an event the user is targeting to retrieval within a video database. The definition of event structure is very subjective. Users generally agree about the inherent objects involved in the occurrence of an event. However, they express differently the

way these objects are connected to each other and the attributes that can characterize these objects. The issue is then to enable a flexible definition of relations and attributes while it should be crisp for object concepts. An event is generally composed of many distinct situations describing the positions of the objects and the relations between them. These situations are also connected to each other by temporal relations describing their order of occurrence. A *Penalty* event in soccer games, for example, begins by placing the ball on the penalty point, waiting for the referee whistling, and then shooting the ball by a player while all the other players are outside the penalty zone. The *eventgraph* is then described as a fuzzy conceptual graph composed of elementary fuzzy conceptual subgraphs. Each subgraph describes a specific situation composed of concepts gathered together by spatial relations. Subgraphs are themselves connected to each other by temporal relations. All concepts are crisp inside this model while relations and attributes are fuzzy. This means that a system verifying the model should contain all the concepts appearing in that model, while the relations and attributes of the model should be verified by the system depending on the certainty degree associated to each relations or attribute (see below). Figure 3.3 shows an event graph describing a person leaving a bag near a crowd of people. This graph decomposes the event into four situations:

- The first situation (S_1) describes a person holding a bag: the relation holding a bag is captured using the spatial fuzzy relations *externally connected* (*ec*) and *partially overlapping* (*po*) that will be defined later. The fuzzy relations are associated with a low degree of confidence 0.4 since it is difficult to verify that a man holds a bag. The appearance of the two objects *bag* and *person* is respectively associated with the degrees of confidence 0.6 and 0.8. According to the graph, this situation occurs within the event with a degree of confidence set to 0.7.
- The second situation (S_2) describes the same person leaving the bag away. The fuzzy relation *disconnected* between the two objects *bag* and *person* is associated with an attribute *range*. The user wishes that the range gets a value equal to *very far* with a confidence set to 0.8. The present situation occurs within the event, according to the user, with confidence degree equal to 0.8.
- The third situation (S_3) describes the person disappearing from the scene. This is expressed by verifying that no person appearing in the scene is the person who leaves the bag at the former situation. This situation is connected to the event concept by a fuzzy relation *contain* with a confidence degree set to 0.3. In fact, the user wishes to capture the event even if the person leaving the bag does not leave the scene. She expresses this wish by associating a low degree of confidence to the fuzzy relation *contain* between the event and the current situation.
- The fourth situation (S_4) describes a crowd of people composed of more than 4 people. The situation S_4 lasts a long period of time within which all the other situations occur successively. This is expressed using the fuzzy relations *before* and *during* between the situations.

3.3.1.2 Video Graph.

It represents the description of contents appearing in the video document or in a part of it. This description is the result of three steps: spatial segmentation of the frames composing the video, classification of spatial segments into elementary objects with different degrees of confidence,

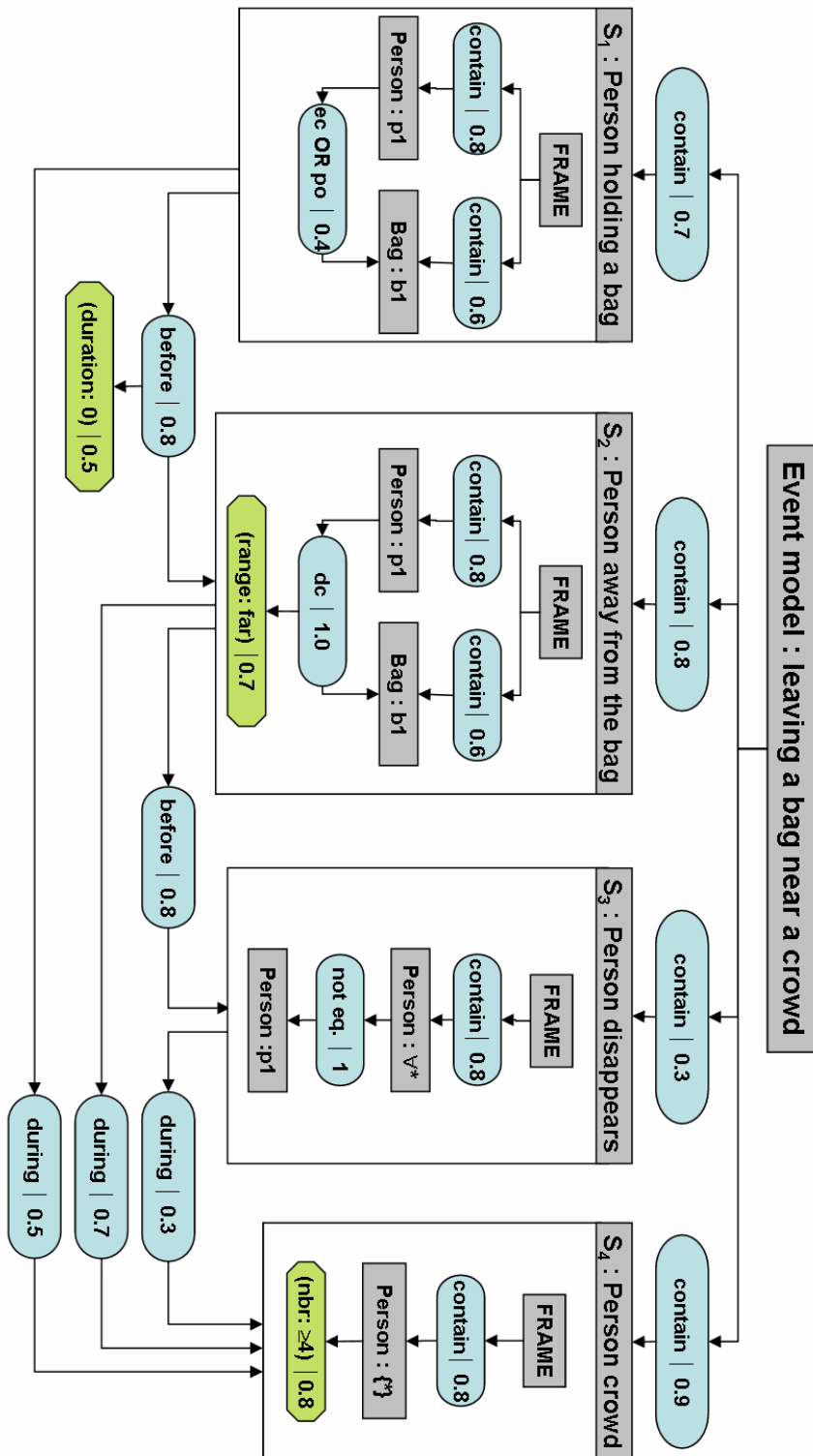


Figure 3.3: event graph

and then clustering the different frames into temporal segments regarding the objects in frames and the spatial relations gathering them. Each temporal segment is related to a set of objects where the spatial relations gathering them remain true for all the interval of time.

The *Video Graph* is a fuzzy conceptual graph composed itself of elementary fuzzy conceptual subgraphs. Each subgraph is related to a temporal segment and has an attribute *duration* showing the time interval of occurrence within the video segment. Each object inside the subgraph is also related to an attribute *MBR* (for Minimum Boundary Rectangle) that describes its position in the frame. In contrast with the event graphs, uncertainty is mainly coming from the imprecision in recognition and classification of the elementary objects. In fact, after image segmentation, each region can be associated with different concept types with different degrees of confidence (e.g. a blue region in an image can be recognized as *sky* with 60 percent confidence, as *sea* with 50 percent confidence, and as *playfield* with 10 percent confidence regarding the low level visual features of the region). However, spatial relations gathering the image regions, temporal relations between video intervals and the attributes are crisp since they can be directly computed from the video frames. As sample *Video Graphs*, figure 3.4 depicts the conceptual graph of a video segment that has as id #3. The video segment is composed of five consecutive temporal segments. Each segment describes a particular position. All the objects in the graph are fuzzy objects associated each one of them to a degree of confidence. For example, the object *o2* is assigned to two concept types: *Rock* with confidence degree 0.5 and *bag* with confidence degree 0.6. Figure 3.5 can be described by a Video Graph such as the one depicted in figure 3.4. In that figure, situation *S3* of the event graph shown figure 3.3 is not verified, but this video graph still corresponds to the event graph figure 3.3 since the confidence degree corresponding to the occurrence of the situation *S3* is very low. This is described by the relation (*contain*|0.3) linking the *event graph* : *leaving a bag near a crowd* and the situation *S3*.

3.3.2 Fuzzy Spatial and Temporal relationships

In order to define and constitute fuzzy graphs and compute partial or complete similarity between them, spatial and temporal relations gathering together video contents should be defined in a fuzzy manner. That should enable for taking into account uncertainty due, on the one hand, to errors and imprecision of video content descriptions, and on the other hand, to handle queries formulated by users using vague concepts like "very close", "far from", "in 70%", etc. This section defines new fuzzy variant for the most used relation systems for representing temporal and spatial relations between objects.

3.3.2.1 Fuzzy Temporal Relationships

Allen has proposed, in [8], an interval-based temporal logic to represent relations between time intervals based on 13 basic relations:

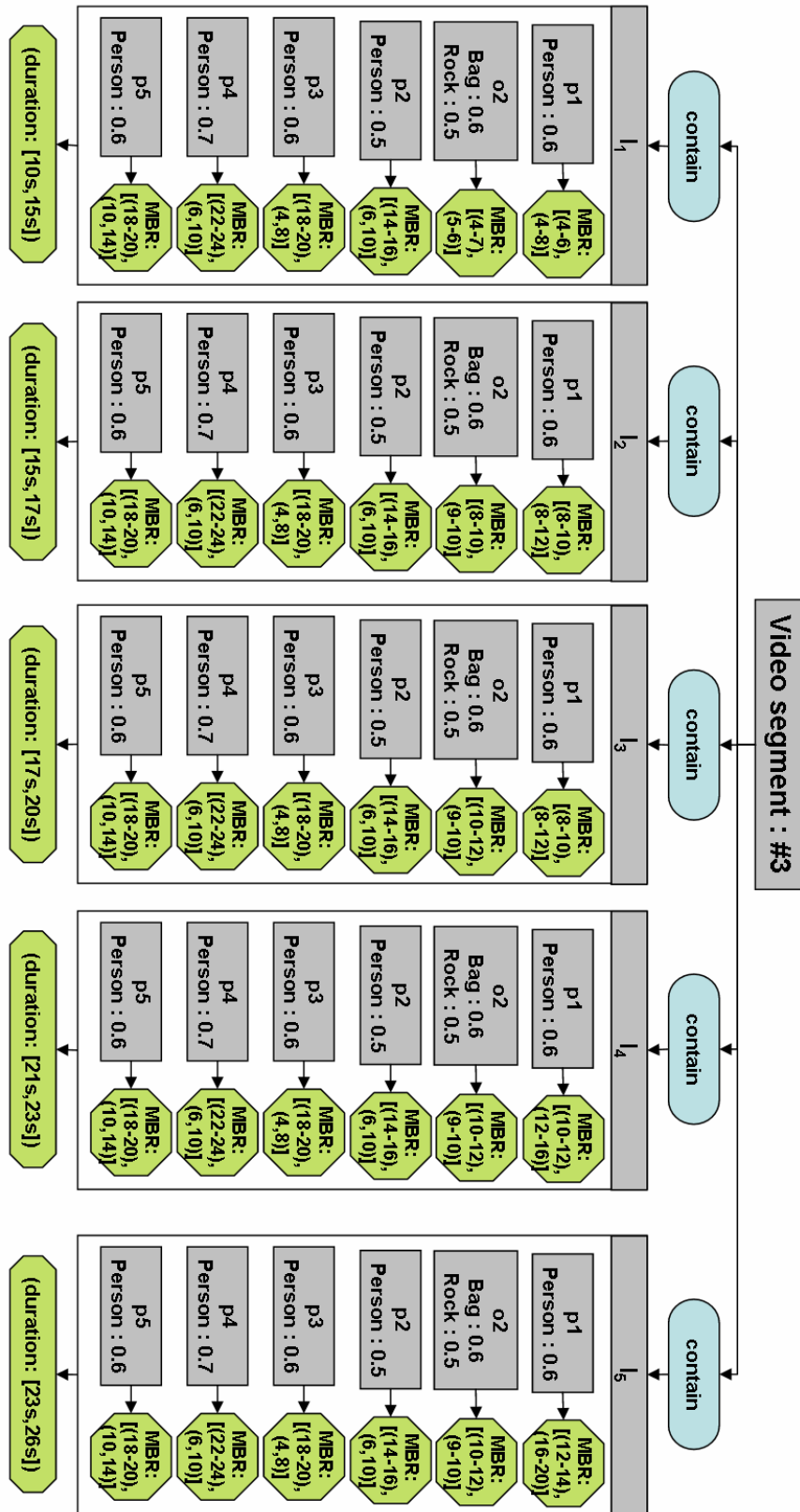


Figure 3.4: Video Graph

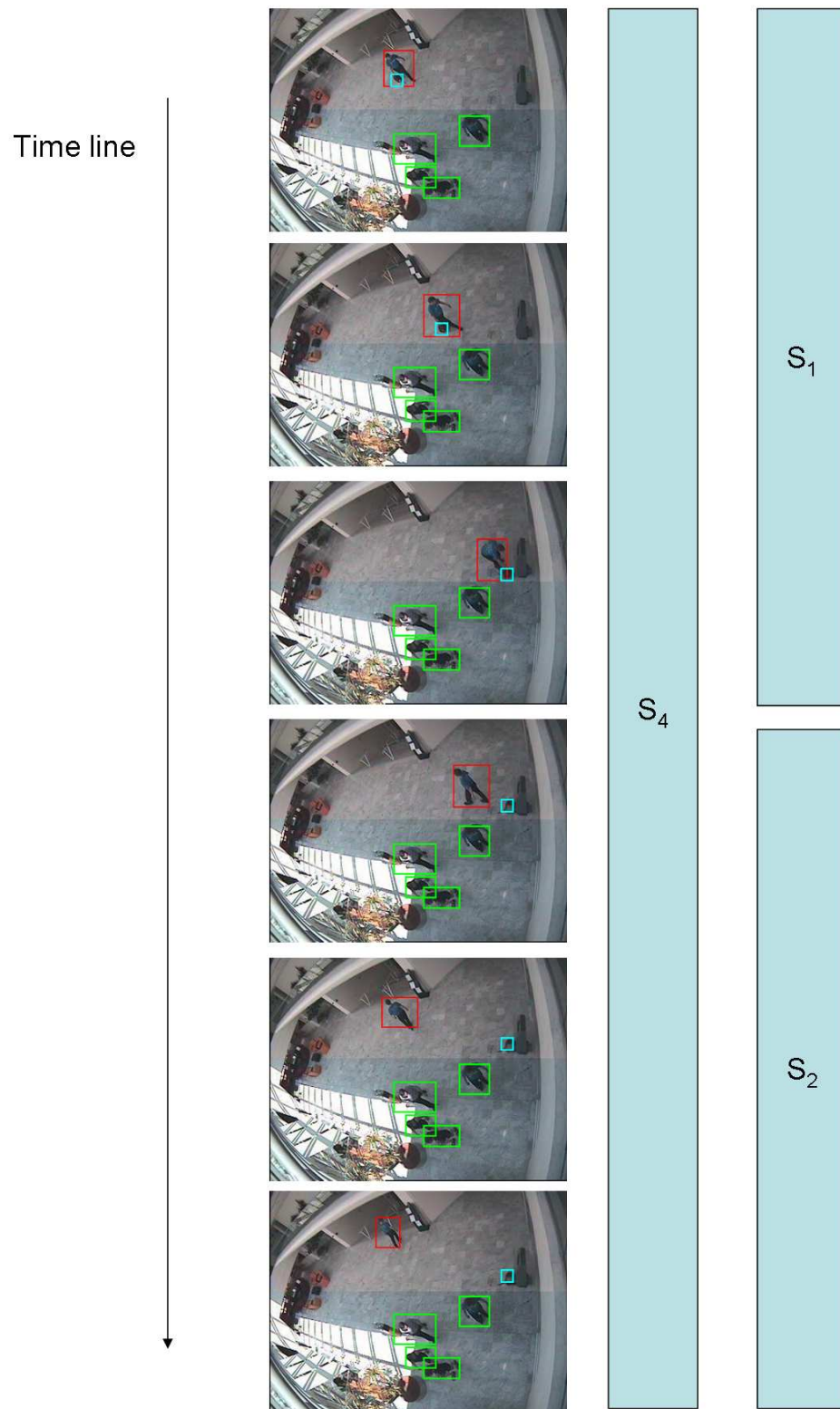


Figure 3.5: Video Sequence

rel_1	$=$	<i>equal</i>
rel_2	$=$	<i>before</i>
rel_3	$=$	<i>after</i>
rel_4	$=$	<i>during</i>
rel_5	$=$	<i>during</i> ^{<i>i</i>}
rel_6	$=$	<i>overlaps</i>
rel_7	$=$	<i>overlaps</i> ^{<i>i</i>}
rel_8	$=$	<i>meets</i>
rel_9	$=$	<i>meets</i> ^{<i>i</i>}
rel_{10}	$=$	<i>starts</i>
rel_{11}	$=$	<i>starts</i> ^{<i>i</i>}
rel_{12}	$=$	<i>finishes</i>
rel_{13}	$=$	<i>finishes</i> ^{<i>i</i>}

A constraint between two intervals is a set of atomic relations which can hold between them. The imprecision and vagueness are inherent in temporal knowledge representation domain. However, based on this algebra, it is impossible to express more refined knowledge regarding the priority of the constraints or about the uncertainty affecting them.

Badaloni et al. have presented, in [15], a fuzzy Interval Algebra extending the classical Interval Algebra (IA) based on the possibility theory [42, 169] and offering a rich and powerful setting for the representation and treatment of the time information pervaded with imprecision and uncertainty.

Allen's Interval Algebra can be viewed as a special case of Constraint Satisfaction Problem (CSP), since an interval can be interpreted as an element of \mathbb{R}^2 and a relation between a pair of intervals as a subset of $\mathbb{R}^2 \times \mathbb{R}^2$. Badaloni et al. present an extended Interval Algebra IA^{fuz} , as a special case of Fuzzy Constraint Satisfaction Problem (FCSP) [43]. The definition of constraints is relaxed by making the subset $\mathbb{R}^2 \times \mathbb{R}^2$ fuzzy, assigning a preference degree α_i to every atomic relation rel_i . Relations between intervals I_1 and I_2 are then expressed in the form:

$$I_1(rel_1[\alpha_1], rel_2[\alpha_2], \dots)I_2$$

where α_i is the preference degree of rel_i ($i = 1, \dots, 13$), belonging to the interval $[0, 1]$. If α_i belongs to $\{0, 1\}$ we obtain the classical framework.

The aim of this section is to propose a method for automatically computing the degree of confidence in satisfaction of a specific temporal relation between two temporal intervals. This degree will be considered as a preference degree and used later for reasoning about the temporal information of video contents. To calculate this degree, we define for each relation rel_i ($i = 1, \dots, 13$) two values:

- a *margin_i* that corresponds to an inherent distance that changes regarding the position of the two intervals.
- a *threshold_i* that represents a maximum value for *margin_i* beyond which the relation rel_i will be considered as totally non satisfied.

For instance, to verify the satisfaction of the relation *before* between two intervals A and B , we consider that if less than half of the length of A lasts before the beginning of B , the relation *before* is not satisfied. However, if more than half of the length of A lasts before B , then the

relation *before* is satisfied with a degree of confidence that varies according to the length of intersection of *A* and *B*. The *margin* for the relation *before* is set to the intersection of *A* and *B* and the *threshold* is set to half of the length of *A*.

If the *margin* $\in [0, \text{threshold}]$ the degree of confidence of the relation *A.before.B* is equal to $1 - \text{margin}/\text{threshold}$. Let *A* and *B* be two intervals, we note A^+ and A^- respectively the superior and inferior bounds of the interval *A*. Figure 3.6 summarizes the margin, threshold values and degrees of confidence corresponding to the satisfaction of each of the Allen's relations. In our framework, users have the ability to modify the *margin* and *threshold* values to fit with the requirements of specific applications.

In the following, we present the expressions of degrees of confidence of Allen's relations between two intervals *A* and *B*.

The degree of confidence for the satisfaction of the relation *A.before.B* is expressed by:

$$f(\text{before}, A, B) = \begin{cases} 1 & \text{if } M < 0 \\ 1 - M/T & \text{if } 0 \leq M < T \\ 0 & \text{if } T \leq M \end{cases}$$

with $M = A^+ - B^-$, and $T = (A^+ - A^-)/2$

Similarly, the degree of confidence of the satisfaction of the relation *A.meets.B* is :

$$f(\text{meets}, A, B) = \begin{cases} 1 - M/T & \text{if } 0 \leq M \leq T \\ 0 & \text{otherwise} \end{cases}$$

with $M = \|B^- - A^+\|$, and $T = (A^+ - A^-)/2$

The degree of confidence of the satisfaction of the relation *A.overlaps.B* is defined as:

$$f(\text{overlaps}, A, B) = \begin{cases} 1 & \text{if } M < 0 \\ 1 - M/T & \text{if } 0 \leq M < T \\ 0 & \text{if } T \leq M \end{cases}$$

with $M = B^- - A^+$, and $T = (A^+ - A^-)/2$

The degree of confidence of the satisfaction of the relation *A.during.B* is :

$$f(\text{during}, A, B) = \begin{cases} 1 & \text{if } M_1 < 0 \text{ AND } M_2 < 0 \\ 0 & \text{if } T \leq M_1 \text{ OR } T \leq M_2 \text{ OR } T \leq M_1 + M_2 \\ 1 - M/T & \text{otherwise} \end{cases}$$

with $M_1 = A^+ - B^+$, $M_2 = A^- - B^-$, $M = M_1 + M_2$, and $T = (A^+ - A^-)/2$

The degree of confidence of the satisfaction of the relation *A.finishes.B* is expressed by:

$$f(\text{finishes}, A, B) = \begin{cases} 1 - M/T & \text{if } 0 \leq M < T \\ 0 & \text{if } T \leq M \end{cases}$$

with $M = \|A^+ - B^+\|$, and $T = (A^+ - A^-)/2$

Finally, the degree of confidence of the satisfaction of the relation *A.starts.B* is expressed with:

<u>Relation</u>	<u>Schema</u>	<u>Margin</u>	<u>Threshold</u>	<u>Degree of confidence</u>
A.before.B		$M = A^{(+)} - B^{(-)}$	$T = (A^{(+)} - A^{(-)}) / 2$	$f = 1$; if $M < 0$ $f = 1 - M / T$; if $0 \leq M < T$ $f = 0$; if $T \leq M$
A.meets.B		$M = B^{(-)} - A^{(+)} $	$T = (A^{(+)} - A^{(-)}) / 2$	$f = 1 - M / T$; if $0 \leq M < T$ $f = 0$; if $T \leq M$
A.overlaps.B		$M = B^{(-)} - A^{(+)}$	$T = (A^{(+)} - A^{(-)}) / 2$	$f = 1$; if $M < 0$ $f = 1 - M / T$; if $0 \leq M < T$ $f = 0$; if $T \leq M$
A.during.B		$M_1 = A^{(+)} - B^{(+)}$ $M_2 = A^{(-)} - B^{(-)}$ $M = M_1 + M_2$	$T = (A^{(+)} - A^{(-)}) / 2$	$f = 1$; if $M_1 < 0$ AND $M_2 < 0$ $f = 0$; if $T \leq M_1$ OR $T \leq M_2$ OR $T \leq M_1 + M_2$ $f = 1 - M / T$; otherwise
A.finishes.B		$M = A^{(+)} - B^{(+)} $	$T = (A^{(+)} - A^{(-)}) / 2$	$f = 1 - M / T$; if $0 \leq M < T$ $f = 0$; if $T \leq M$
A.starts.B		$M = A^{(-)} - B^{(-)} $	$T = (A^{(+)} - A^{(-)}) / 2$	$f = 1 - M / T$; if $0 \leq M < T$ $f = 0$; if $T \leq M$

Figure 3.6: Fuzzy definition of Allen's temporal relations

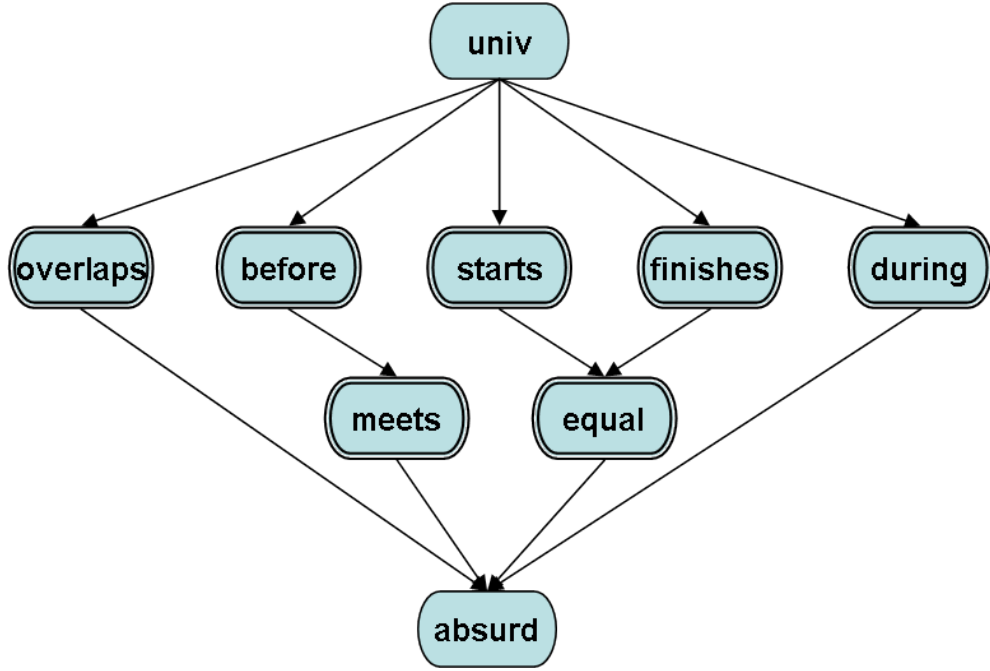


Figure 3.7: Allen's temporal relations Trellis

$$f(starts, A, B) = \begin{cases} 1 - M/T & \text{if } 0 \leq M < T \\ 0 & \text{if } T \leq M \end{cases}$$

with $M = \|A^- - B^-\|$, and $T = (A^+ - A^-)/2$

Figure 3.8 shows an example of matching of temporal relation 'meets' with different image graphs. The formulas for calculating degrees of satisfaction of the temporal relations presented above will be used in matching a crisp temporal relation to a fuzzy temporal relation as described below in the section 3.4.1.2.

3.3.2.2 Fuzzy Spatial Relations

Many systems have been introduced to represent and qualitatively or quantitatively reason about spatial properties and relations between objects. One of the most used approaches is the Region Connection Calculus (*RCC8*) proposed by Randell et al in [116]. *RCC8* introduces a set of 8 jointly exhaustive and pairwise disjoint relations: disconnected (*dc*), externally connected (*ec*), equal (*eq*), partially overlapping (*po*), tangential proper part (*tpp*), tangential proper part inverse (*tpp*⁻¹), non-tangential proper part (*ntpp*), and non-tangential proper part inverse (*ntpp*⁻¹).

The definition of spatial relations differs according to the model used to represent the spatial property of video objects. In order to simplify the representation, we formulate these relationships reasoning based on *MBR* (Minimum Boundary Rectangle) of video objects. The *MBR* of an object in 2-Dimension(x,y) coordinate system is specified by [(min(x) - max(x)), (min(y) - max(y))].

Let *A* be a 2-D video object represented by its *MBR*. We note *A_x* the projection of *A* on the *X* axis. Similarly, *A_y* denotes the projection of *A* on the *Y* axis. *A_x* and *A_y* are then one dimensional intervals and can be considered as temporal intervals. Therefore, we use the previous definitions

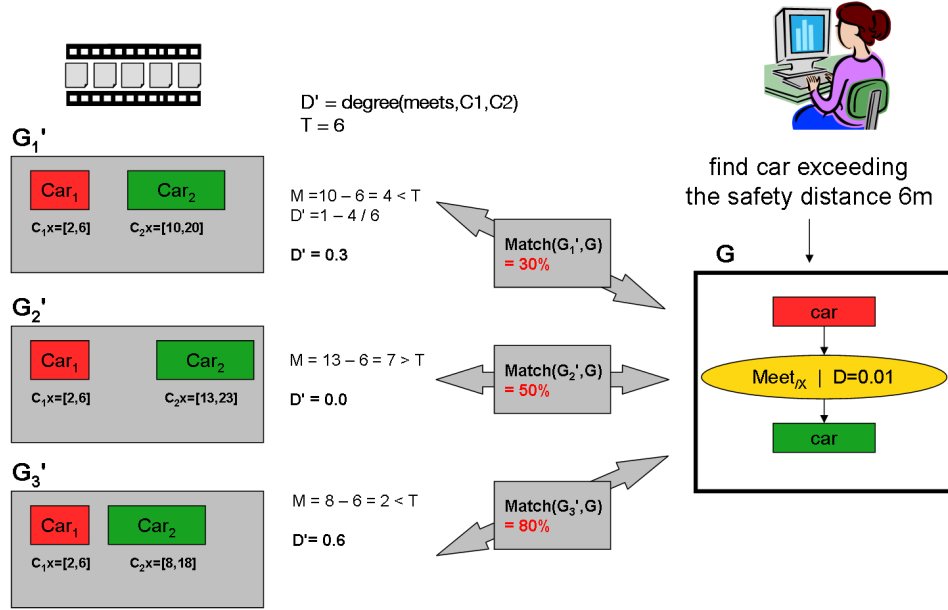


Figure 3.8: An example of a fuzzy matching of a temporal relation

of temporal relationships in order to define the qualitative spatial RCC8 relationships. Figure 3.9 depicts the definition of the *RCC8* spatial relations based on Allen's temporal relations.

Therefore, the calculation of degrees of confidence associated to the *RCC8* relations is done based on the degree of confidence of the Allen's relations.

The degree of confidence corresponding to the relation $A.dc.B$ is defined as follows :

$$f(dc, A, B) = \max\{f(\text{before}, A_x, B_x), f(\text{before}, A_y, B_y), f(\text{before}, B_x, A_x), f(\text{before}, B_y, A_y)\}$$

The relation $A.ec.B$ is defined as follows :

$$f(ec, A, B) = \max\{ \begin{aligned} &(\max\{f(\text{meets}, A_x, B_x), f(\text{meets}, B_x, A_x)\} \\ &- \max\{f(\text{before}, A_y, B_y), f(\text{before}, B_y, A_y)\}), \\ &(\max\{f(\text{meets}, A_y, B_y), f(\text{meets}, B_y, A_y)\} \\ &- \max\{f(\text{before}, A_x, B_x), f(\text{before}, B_x, A_x)\}) \end{aligned} \}$$

The degree of confidence corresponding to the relation $A.tpp.B$ can be defined by:

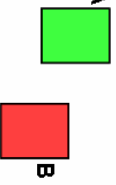
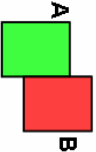


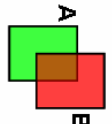
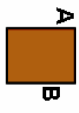
<u>Relation</u>	<u>Schema</u>	<u>Formula</u>
A.dc.B		$A_x.before.B_y \text{ OR } A_y.before.B_x$ OR $B_x.before.A_x \text{ OR } B_y.before.A_y$
A.ec.B		$(A_x.meets.B_x \text{ OR } B_x.meets.A_x) \text{ AND NOT}(A_y.before.B_y \text{ OR } B_y.before.A_y)$ OR $(A_y.meets.B_y \text{ OR } B_y.meets.A_y) \text{ AND NOT}(A_x.before.B_x \text{ OR } B_x.before.A_x)$
A.tpp.B		$(A_x.during.B_x) \text{ AND } (A_y.starts.B_y \text{ OR } A_y.finishes.B_y)$ OR $(A_y.during.B_y) \text{ AND } (A_x.starts.B_x \text{ OR } A_x.finishes.B_x)$
A.n TPP.B		$(A_x.during.B_x) \text{ AND NOT}(A_y.starts.B_y \text{ OR } A_y.finishes.B_y)$ OR $(A_y.during.B_y) \text{ AND NOT}(A_x.starts.B_x \text{ OR } A_x.finishes.B_x)$
A.po.B		$(A_x.overlaps.B_x \text{ OR } B_x.overlaps.A_x) \text{ AND NOT}(A_y.before.B_y \text{ OR } B_y.before.A_y)$ OR $(A_y.overlaps.B_y \text{ OR } B_y.overlaps.A_y) \text{ AND NOT}(A_x.before.B_x \text{ OR } B_x.before.A_x)$
A.eq.B		$A_x.starts.B_x \text{ AND } A_y.finishes.B_y$ AND $A_y.starts.B_y \text{ AND } A_x.finishes.B_x$

Figure 3.9: Expressing RCC8 relation by using Allen's relations

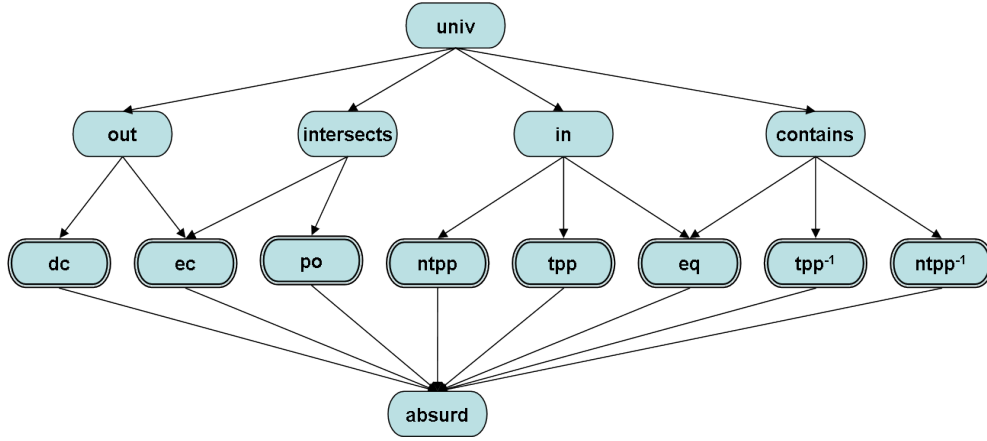


Figure 3.10: RCC8 spatial relations Trellis

$$f(tpp, A, B) = \max\{ \begin{aligned} & (f(during, A_x, B_x) \\ & + \max\{f(starts, A_y, B_y), f(finishes, A_y, B_y)\}), \\ & (f(during, A_y, B_y) \\ & + \max\{f(starts, A_x, B_x), f(finishes, A_x, B_x)\}), \\ & \end{aligned} \}$$

The degree of confidence corresponding to the relation $A.ntpp.B$ can be defined by :

$$f(ntpp, A, B) = \max\{ \begin{aligned} & (f(during, A_x, B_x) \\ & - \max\{f(starts, A_y, B_y), f(finishes, A_y, B_y)\}), \\ & (f(during, A_y, B_y) \\ & - \max\{f(starts, A_x, B_x), f(finishes, A_x, B_x)\}), \\ & \end{aligned} \}$$

The degree of confidence corresponding to the relation $A.po.B$ can be defined as follows :

$$f(po, A, B) = \max\{ \begin{aligned} & (\max\{f(overlaps, A_x, B_x), f(overlaps, B_x, A_x)\} \\ & - \max\{f(before, A_y, B_y), f(before, B_y, A_y)\}), \\ & (\max\{f(overlaps, A_y, B_y), f(overlaps, B_y, A_y)\} \\ & - \max\{f(before, A_x, B_x), f(before, B_x, A_x)\}) \\ & \end{aligned} \} / 2$$

The degree of confidence corresponding to the relation $A.eq.B$ can be defined with :

$$f(eq, A, B) = \frac{(f(starts, A_x, B_x) + f(finishes, A_x, B_x) + f(starts, A_y, B_y) + f(finishes, A_y, B_y))}{4}$$

Figure 3.11 shows an example of the matching of spatial relation 'ec' with different image graphs.

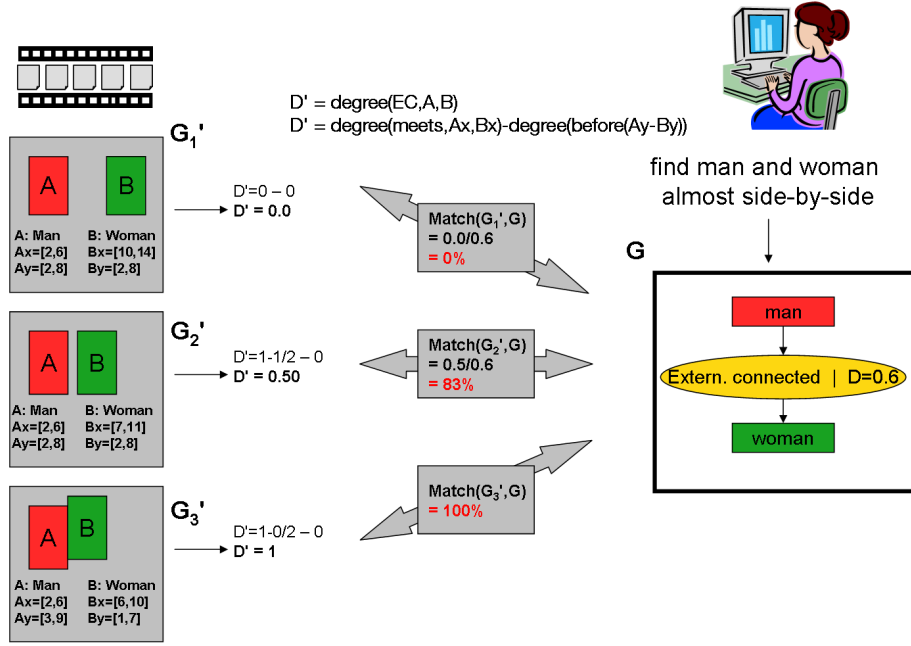


Figure 3.11: An example of a fuzzy matching of a spatial relation

3.4 Graph Matching

Retrieving complex events within video segments is possible by calculating the match degree between the event graph and the different graphs of video segments. The formulas for calculating degrees of satisfaction of the spatial relations presented above will be used in matching a crisp spatial relation to a fuzzy spatial relation as described below in the section 3.4.1.2.

3.4.1 Match degrees

3.4.1.1 Matching of concepts.

The degree of match $M(c, c')$ between a crisp concept in an event graph $c = t$ and a fuzzy concept in a Video Graph $c' = (e, (t_1, f_1), \dots, (t_n, f_n))$ is defined as follows:

$$M_C(c, c') = \begin{cases} \max(f_i) & \text{if } (\exists i \in \{1, \dots, n\}) \text{ where } t_i \sqsubseteq t \\ 0 & \text{otherwise} \end{cases}$$

3.4.1.2 Matching of relations.

As mentioned above in section 3.3.2, a relation r' between two video objects is calculated with a degree of satisfaction f related to errors in object positions extraction. On the other hand, a relation r' in the model graph is associated with a confidence degree f' that depends on the conceptual representation of the event. Therefore, the degree of match $M(r, r')$ between a crisp

relation $r = (t, 1)$ calculated with a satisfaction degree f in a video graph and a fuzzy relation $r' = (t', f')$ in an event graph is defined as follows:

$$M_R(r, r') = \begin{cases} 1 & \text{if } (f > f') \text{ and } (t \sqsubseteq t') \\ f/f' & \text{if } (f' > f) \text{ and } (t \sqsubseteq t') \\ 0 & \text{otherwise} \end{cases}$$

3.4.1.3 Matching of attributes.

Matching of attributes from two graphs means estimating how the values of the two attributes are close or different. However, this mainly depends on the type of the attribute but also the perception the user can have regarding the values of an attribute. We address this issue by proposing a solution that is mainly applicable for attributes that take values from ordered concrete domains. Let $a' = (t', v', f')$ be a fuzzy attribute from the event graph and $a = (t, v)$ the crisp attribute from the Video Graph. We define a threshold w for the difference of values $d = \text{abs}(v - v')$. If d exceeds w , then the two attributes are not considered similar. Whereas, if this d is less than w , the degree of similarity f of the two values of the attributes is calculated regarding the value of d such as $f = 1 - d/w$. Therefore, the degree of match between the two attributes $a = (t, v)$ and $a' = (t', v', f')$ is defined as follows:

$$M_A(a, a') = \begin{cases} 1 & \text{if } (f > f') \text{ and } (t \sqsubseteq t') \\ f/f' & \text{if } (f < f') \text{ and } (t \sqsubseteq t') \\ 0 & \text{otherwise} \end{cases}$$

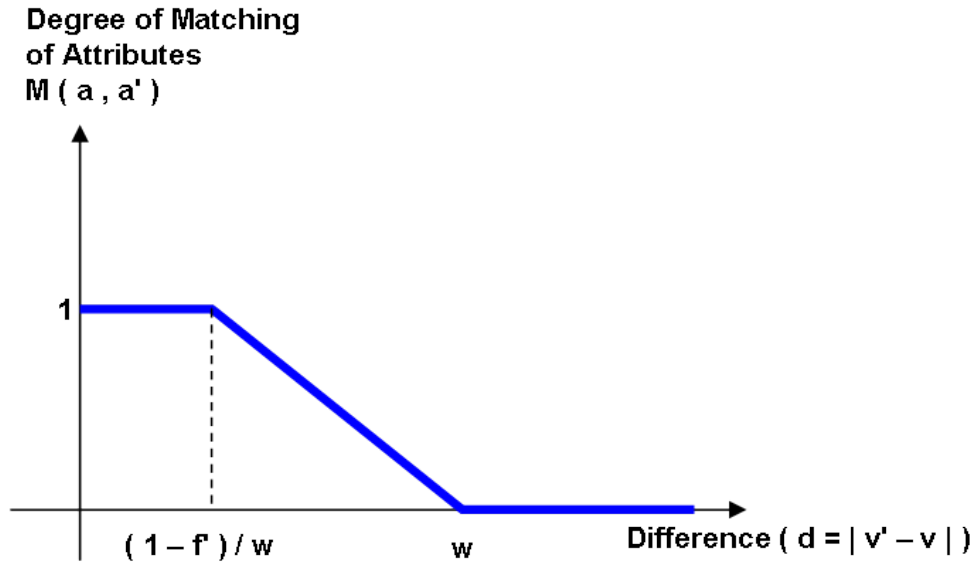


Figure 3.12: Degree of matching two attributes a and a' regarding the difference of values

Let $G(C, R, A)$ and $G'(C', R', A')$ be two fuzzy graphs. A projection π from G to G' is a mapping such that :

- $\forall c \in C, \pi(c) \in C'$ and $M(c, \pi(c)) > 0$,

- $\forall r \in R, \pi(r) \in R'$ and $M(r, \pi(r)) > 0$,
- $\forall a \in A, \pi(a) \in A'$ and $M(a, \pi(a)) > 0$,

Given two graphs $G(C, R, A)$ and $G'(C', R', A')$, and a projection π from G to G' , the matching degree M_G between G and $\pi(G)$ is defined as :

$$M_G(G, G') = \frac{1}{|G|} [\alpha_c \sum_{c_i \in C} M_C(c_i, \pi(c_i)) + \alpha_r \sum_{r_i \in R} M_R(r_i, \pi(r_i)) + \alpha_a \sum_{a_i \in A} M_A(a_i, \pi(a_i))]$$

Where $|G|$ is the sum of the number of concepts, the number of relations and the number of attributes in the graph G . α_c , α_r , and α_a are weights associated respectively with concepts, relations and attribute matching calculus.

3.4.2 Matching algorithms

Matching between an event graph and a Video Graph is done in two steps. The first step is an *intra-image-sequence* matching to link each situation in the event model with its or their corresponding image sequences. The second one is *inter-image-sequence* intended to verify whether the video situations satisfy the temporal relations fixed between their corresponding situations in the event model.

3.4.2.1 Intra-image-sequence matching.

In this step, we calculate for each image sequence graph, the spatial relations gathering its fuzzy concepts. Then, each situation graph from the event model is compared to all image sequence graphs in the Video Graph. The matching problem becomes the one of finding the best matching subgraph g that maximizes $M_G(g, \pi(g))$.

We have developed two types of algorithms that can be used to perform intra-image-sequence matching.

- Heuristic Matching
- Exhaustive Matching

Heuristic Matching

The used matching algorithm is inspired by the *Breadth – firstsearch* for tree parsing and is based on Messmer and Bunke's error-tolerant subgraph isomorphism algorithm [102] and the matching algorithm in [105]. The goal of this algorithm is to find the best matching between the two subgraphs. The best image subgraph that matches the current situation graph

Given two graphs G_S and G_I , the graphs are decomposed into *arche* sets which are respectively H_S and H_I . Each arch consists of a source concept s , a target concept t , a relation r and one or more attributes. Figure 3.13 shows how a graph can be seen as a set of connected arcs.

The idea is to start with a single arch a_S in the situation graph that matches another arch a_I in the image sequence graph. This pair of arches (a_S, a_I) forms the initial matching model and image subgraphs. Subsequently, other model arches that have matching image arches are added to the matching subgraphs. This is done by considering the pair (a_S, a_I) as a tree root and

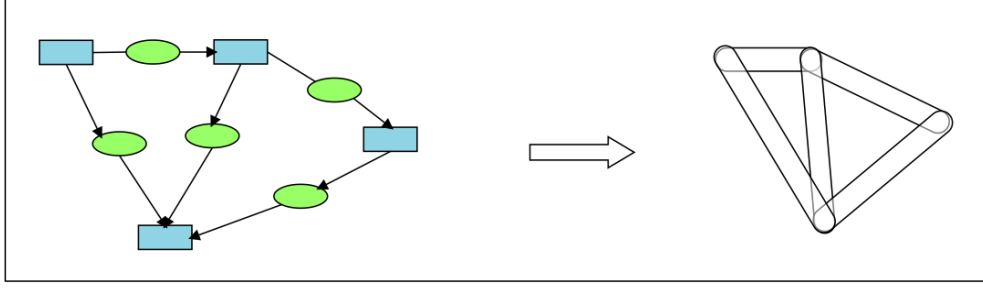


Figure 3.13: Changing from a Concept-Relation view to Arcs view of a conceptual graphs.

then finding the children subgraphs $((a_S, b_S), (a_I, b_I))$ where b_S matches b_I and are respectively connected with g_S and g_I . We name this operation extending the matching pair (a_S, a_I) . At the first level, we begin by keeping a first level priority queue Q_1 that contains all the possible matching arch pairs. Pairs in the queue are ranked decreasingly based on the degree of match of each pair. From this queue, only the first N pairs are kept in a set S_1 . Then, each element in S_1 is extended by calculating all its children. The resulting elements of all pair of extensions are added to a priority queue Q_2 . Then, the first N elements are grouped in S_2 . We continue building the tree until arriving to a set S_l where no element is extended. Figure 3.14 gives an idea about the execution of the proposed algorithm.

The main differences with the algorithm in [102] are:

- the algorithm in [102] conserves at each extension level only one matching pair that has the highest matching degree, except the first level of single arch pairs. This policy helps for minimizing the processing time however it can cause algorithm to be caught in a local optimum and missing a better (deeper) optimum. In our algorithm we choose to conserve at each level the N best matching pairs.
- the algorithm in [102] returns only one best solution, our algorithm returns the N solutions.
- our algorithm is recursive while the algorithm in [102] is iterative.

The previous procedure is done through a recursive algorithms (algorithm 4).

Exhaustive Matching

In this variant of matching algorithm, the goal is to find total matching between all the components of the two graphs. Each situation can be seen as a subgraph and defined as $G_S = (C_S, R_S, A_S)$ where C_S is the set of concepts included in the situation, R_S is the set of relations, while A_S is the set of attributes. Similarly, each image sequence can be defined as a subgraph $G_I = (C_I, R_I, A_I)$ composed of the concept set C_I , the relation set R_I and the attribute set A_I .

Let L_S be a tuple composed of the elements in C_S ordered in a chosen way. Before calculating the matching degree between the two graphs, the set $\Gamma(L_S, C_I)$ of all tuples of concepts in C_I that corresponds to concepts type of L_S is computed. $\Gamma(L_S, C_I)$ is defined as follows:

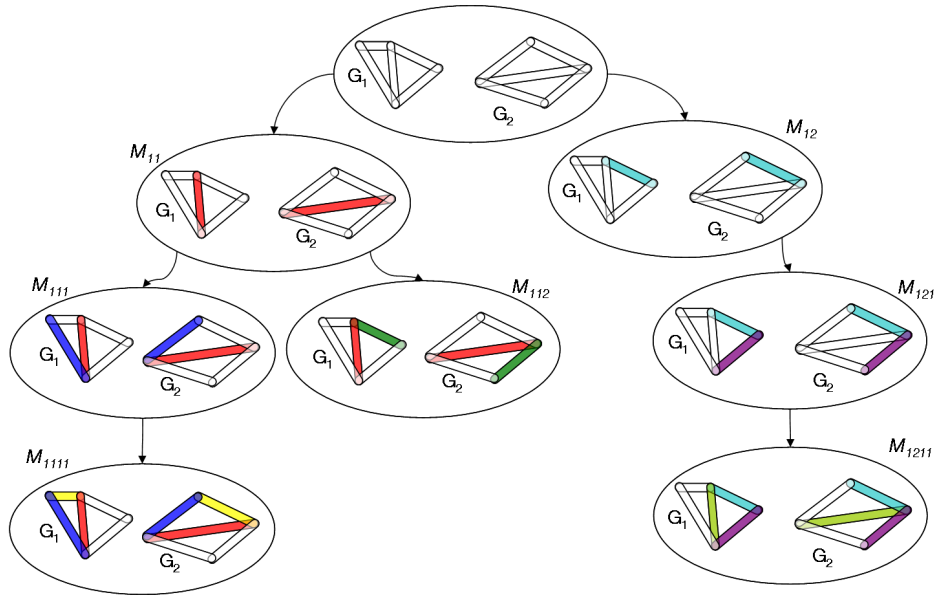


Figure 3.14: The algorithm of heuristic matching performed in many levels.

Algorithm 4 *HeuristicGraphMatching*(S_i)**Require:** $S_i \leftarrow$ set of couples of matching subgraphs at level i (S_0 is empty). $H_I = toArchs(G_I), H_S = toArchs(G_S)$ **Ensure:** *HeuristicGraphMatching*(S_i) $Q_{tmp} = newQueue()$ **for all** $(g_I, g_S) \in S_i$ **do** **while** $\exists (h_I, h_S) \in H_I \times H_S, M_G(h_I, h_S) > 0, connect(h_I, g_I), connect(h_S, g_S)$ **do** $(g_I, g_S) = (g_I \sqcup h_I, g_S \sqcup h_S)$ **if** $f = M_G(g_I, g_S) > thresh$ **then** $enqueue_{wrt(f)}(Q_{tmp}, (g_I, g_S))$ **end if** **end while****end for****if** $isEmpty(Q_{tmp})$ **then** **return** S_i **else** $S_{i+1} = Q_{tmp}[1 : N]$ $HeuristicGraphMatching(S_{i+1})$ **end if**

$$\begin{aligned}
L_S &= [c_1, \dots, c_n], \\
C_I &= (c'_1, \dots, c'_m) \\
\Gamma(L_S, C_I) &= \{[c'_{p+1}, \dots, c'_{p+n}] \in (C_I)^n \mid \\
&\quad c'_{p+i} \in C_{I/c_i.type} \forall i \in \{1, \dots, n\}, \\
&\quad c'_{p+i} \neq c'_{p+j} \forall i, j \in \{1, \dots, n\}\}
\end{aligned}$$

Let L be a tuple in Γ , and $L = [c'_{p+1}, \dots, c'_{p+n}]$. We note $corresp(c_i) = c'_{p+i}$.

For instance, let :

$$\begin{aligned}
L_S &= [c_1(e_1, \{(vehicle, 1)\}), \\
&\quad c_2(e_2, \{(car, 1)\}), \\
&\quad c_3(e_3, \{(person, 1)\})], \\
C_I &= \{c'_1(e'_1, \{(car, 0.8), (box, 0.3)\}), \\
&\quad c'_2(e'_2, \{(bus, 0.8), (building, 0.3)\}), \\
&\quad c'_3(e'_3, \{(Tree, 0.7), (man, 0.6)\}), \\
&\quad c'_4(e'_4, \{(woman, 0.5)\})\}
\end{aligned}$$

We have :

$$\begin{aligned}
C_{I/c_1.type} &= \{c'_1, c'_2\}, \\
C_{I/c_2.type} &= \{c'_1\}, \\
C_{I/c_3.type} &= \{c'_3, c'_4\},
\end{aligned}$$

and then $\Gamma(L_S, C_I)$ is defined as :

$$\begin{aligned}
\Gamma(L_S, C_I) &= \{L'_1 = [c'_2, c'_1, c'_3], \\
&\quad L'_2 = [c'_2, c'_1, c'_4]\}
\end{aligned}$$

We note $c_i.Att$ (resp. $r_i.Att$) the subset of attributes included in A and whose elements are linked to the concept c_i (resp. relation r_i).

The matching between a situation graph G_S and an image sequence graph G_I is performed according to the algorithm 5.

3.4.2.2 Inter-image-sequence matching.

After getting matching between situations in event graph and the image sequence graphs in the Video Graph, the task is then to verify whether the identified image sequences verify the temporal relations linking their corresponding situations. Before verifying each temporal relation in the event graph, we begin by gathering together consecutive image sequences corresponding to the same situation in a unique image sequence. The temporal interval of this image sequence is the union of all the sequence intervals. The temporal relations regrouping the different image sequences are calculated and then the first level matching algorithm is recalled to map the fuzzy situation graph and the event graph.

A summary of the matching process between an event graph G_E and a Video Graph G_V is described by the algorithm 6.

Algorithm 5 *ExhaustiveGraphMatching*(G_S, G_I)**Ensure:** ExhaustiveGraphMatching(G_S, G_I)

```

 $maxCoe f = 0$ 
for all  $L \in \Gamma$  do
   $matchCoe f = 0$ 
   $index = 0$ 
  for all  $c \in C_S$  do
     $corresp(c) = c'_i$ 
     $matchCoe f = matchCoe f + Matching(c_i, c'_i)$ 
     $matchAtt = 0$ 
     $nbrAtt = 0$ 
    for all  $a' \in c'.Att$  do
      if  $\exists a \in c.Att$  where  $a.type = a'.type$  then
         $matchAtt = matchAtt + Matching(a, a')$ 
         $nbrAtt = nbrAtt + 1$ ;
      end if
    end for
     $matchCoe f = matchCoe f + matchAtt/nbrAtt$ 
     $index = index + 1$ 
  end for
  for all  $r \in R_S$  do
    if  $\exists r' \in R_I, corresp(r.start) = r'.start, corresp(r.end) = r'.end,$ 
     $r'.type = r.type$  then
       $matchCoe f = matchCoe f + Matching(r_i, r'_i)$ 
       $matchAtt = 0$ 
       $nbrAtt = 0$ 
      for all  $a' \in r'.Att$  do
        if  $\exists a \in r.Att$  where  $a.type = a'.type$  then
           $matchAtt = matchAtt + Matching(a, a')$ 
           $nbrAtt = nbrAtt + 1$ ;
        end if
      end for
       $matchCoe f = matchCoe f + matchAtt/nbrAtt$ 
       $index = index + 1$ 
    end if
  end for
   $matchCoe f = matchCoe f / index$ 
  if  $maxCoe f < matchCoe f$  then
     $maxCoe f = matchCoe f$ ;
  end if
end for
return  $maxCoe f$ 

```

Algorithm 6 *EventMatching*(G_E, G_V)**Require:** $S_1 \leftarrow$ set of subgraphs matching results $S_2 \leftarrow$ set of final matching results**Ensure:** *EventMatching*(G_E, G_V) $S_g = \text{newQueue}()$ **for all** (G_S, G_I) $\in \text{sequences}(G_E) \times \text{situations}(G_V)$ **do** Calculate spatial relations in G_I $S_1 = S_1 \cup \text{FirstLevelMatching}(G_S, G_I)$ **end for**

Group adjacent graphs refereing to the same situations

Calculate temporal relations R_1 between S_1 elementsConstitute the video graph $G_V = (S_1, R_1)$ $S_2 = \text{FirstLevelMatching}(G_E, G_V)$ **return** the element of S_2 with the highest matching degree

3.5 Conclusion

This chapter presents a new variant of fuzzy conceptual graphs more suitable to video content description and retrieval than the former approaches for representing video contents. Emphasis is put on the uncertainty measurement which is inherent to the multimedia content access. The approach defines two types of graphs; an event graph that describes a common event where uncertainty is related to human perception in defining relations, and a Video Graph where uncertainty is due to errors and imprecision in calculating object positions. Moreover, new fuzzy variants of Allen's temporal algebra and RCC8 spatial relations are introduced to reason in a fuzzy manner about relationships between objects and intervals within a video segment. Then, similarity measures are defined to assess the degree of match between the components of video and event graphs. A two level graph matching is developed to calculate total matching coefficient between video and event models.

The proposed semantic video model is especially designed for handling uncertainty related to 2D spatial relations and does not integrate 3D spatial relations. An easy extension of the definition of fuzzy 3D spatial relations is affordable. This can be done by adopting the same strategy method used in this chapter to define (n+1)D fuzzy relations using definitions of (n)D fuzzy relations. Furthermore, the proposed model supposes that it is the user who fixes the confidence degrees related to the event graph (query). This can be difficult when the graph contains many confidence degrees to fix or when the user has no idea about the correct confidence degrees to use. Therefore, this work can be extended by adding a learning based algorithm that would calculate and recommend the confidence degrees that would enable for efficiently detecting visual events.

The ultimate search engine, ... would understand exactly what you mean and give back exactly what you want.

Larry Page

4

A Database Approach for Expressive Modeling and Efficient Querying of Visual Information

▷ *In this chapter, a novel declarative rule based language for modeling and querying semantic contents in video documents is presented. It allows to reason with objects, events and spatiotemporal constraints. Queries can refer to both objects and events semantics and audio visual layers. They can be specified in fine granularity with the possibility to retrieve only the segment of the video where the conditions given in the query are satisfied. Spatial, temporal and semantic conditions are specified as predicates which make it easier and more intuitive to formulate complex query conditions. We introduce the concept of temporal and spatial frame of reference which allow to simultaneously locate video contents according to multiple spatiotemporal environments in real world. Our model and query language are extensible, application independent, expressive and quite suitable for multimedia information retrieval.* ◁

Contents of the chapter

4.1	Introduction	87
4.2	Contribution	87
4.3	Basic Definitions	89
4.3.1	Temporal FoRs	89
4.3.2	Spatial FoRs	89
4.4	Datalog-like Data Modeling	93
4.4.1	Example	96
4.5	F-Logic like Data Modeling	100
4.5.1	Data Definition Language	100
4.5.2	Example	102
4.5.3	Rule-based Query Language	105
4.5.4	Inferring new relations	107
4.6	Conclusion	112

4.1 Introduction

By the quick development of multimedia technologies, the number of available video resources is always increasing and the need for efficient video modeling, indexing and retrieval techniques is growing.

Major databases rest on traditional technologies such as hierarchical models, network models or relational models. Such models are very effective and work well in many domains that require simple data structuring. However, these models are not competitive when dealing with advanced applications using complex data structures.

Conventional database systems, based on relational data model for the most of them, are often lacking facilities that provide effective management of video contents [165]. Spatial relations as well as temporal relations are essential to represent the semantic structure of video content and to link objects and events occurring within videos. However, relational models do not provide convenient indexing techniques to manage this kind of information. Moreover, a video database management system requires knowledge and inference techniques for casting raw data into high level contents. Such facilities are not provided with conventional DB systems. Finally, users should be able to describe video content hierarchically. In fact, there can be composed objects containing elementary objects. Events can themselves be composed of smaller events. Therefore, the database system should provide facilities in order to retrieve contents through hierarchical structures.

4.2 Contribution

In this work we propose to use deductive databases based system which would, in our view, help to overcome the above drawbacks. Streaming from the combination of Logic Programming and Relational Databases, deductive databases enable to store semantic information concerning objects and events using facts and then to use a declarative language to specify rules and infer new information. Another fundamental motivation for investigating deductive databases is that of enhancing the expressive power of the relational algebra [87], enabling us to express more complex queries for visual information retrieval. For instance, deductive databases would enable as to express recursive queries and rules that can not be expressed within basic relational databases.

In contrast to *Annotation – based* approaches that only attach to each segment of frame a set of objects or events and do not enable for describing relations between those objects (see section 1.4.2), we adopt an *Object – Relational* based approach to attach to each object or event in the database its corresponding positions and durations where it occurs, and to describe relations between objects and events (see section 1.4.2).

This allows for adding a lot more expressivity and for considerably reducing the time of answering queries and thus augmenting the efficiency of the search engine. Indeed most multimedia content queries correspond to common objects or events rather than specific video segments [2]. This is also interesting for multicamera behavior video monitoring and surveillance. In addition, instead of computing spatiotemporal relations online during the query processing which is a costly operation, our rule based approach enables for considerably cutting down the query response time by calculating offline all the relationships between objects in the video based on defined predicates within the query system. The part of the integrated framework concerned by this chapter is highlighted in the 4.1.

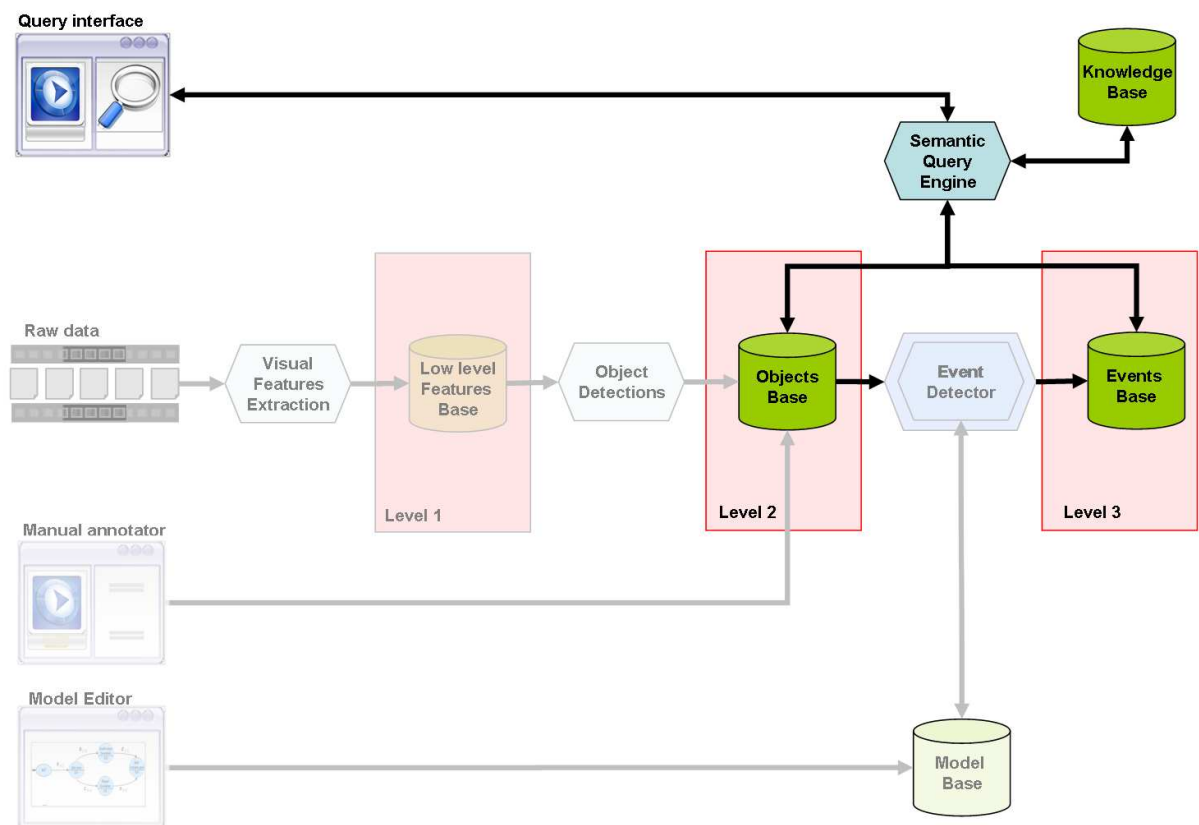


Figure 4.1: Positioning the chapter contribution within the whole framework.

4.3 Basic Definitions

Our video data model and the underlying query language provide wider and more expressive spatiotemporal description. Some basic concepts relevant to our formal model are defined in this section. We mainly introduce the notion of spatial and temporal Frame of Reference which allow to locate video contents simultaneously according to multiple environments in real world.

Definition 1 (Spatio-temporal Frames of Reference). *A frame of reference (FoR) is a coordinate system used to measure the position of objects in it. It consists of an origin, a set of axis and a variable on each axis.*

In our work, we are interested in two kinds of FoRs: *Spatial FoRs* that spatially locate contents and *Temporal FoRs* that temporally locate them.

4.3.1 Temporal FoRs

Temporal References that will be used in this work are :

- **Date FoR:** This FoR corresponds to the date following the Gregorian Calendar. Time is measured by days, months, and years. The origin is in 0 BC.
- **Time FoR:** That FoR corresponds to the time of the day. Time is measured inter alia by hours, minutes, and seconds. Predefined statements such as morning, afternoon, evening,.. can be used. Although they are subjective, those statement can be mapped to the day time as the following: morning(06:00-12:00), afternoon(12:00-17:00), evening (17:00-21:00).
- **Date&Time FoR:** That FoR corresponds to the combination of the two precedent FoRs, Time FoR and Date FoR. Time is measured by years, months, days, hours, minutes, and seconds.
- **Video Frames FoR:** This FoR is associated with the video sequence itself. It takes its origin at the first frame of the video sequence. The time measure assigned to a specific frame is equal to the number of frames separating that frame from the first frame. This FoR would enable to affirm, for example, that an object O appears from the frame f_a of the video to the frame f_b of the same video.
- **Soccer Time-line FoR:** This FoR typically enables to measure the video in minutes and seconds following the timing of the soccer match. Time origin is the first second of play. It is important to consider such FoR since there are many stops in the game, which make this FoR different from the Time FoR. This is an example of a domain-specific FoR that the user can freely define to localize objects and events within.

Figure 4.2 shows an event "free kick" located according to multiple temporal frames of reference.

4.3.2 Spatial FoRs

Spatial References that used in this work are :

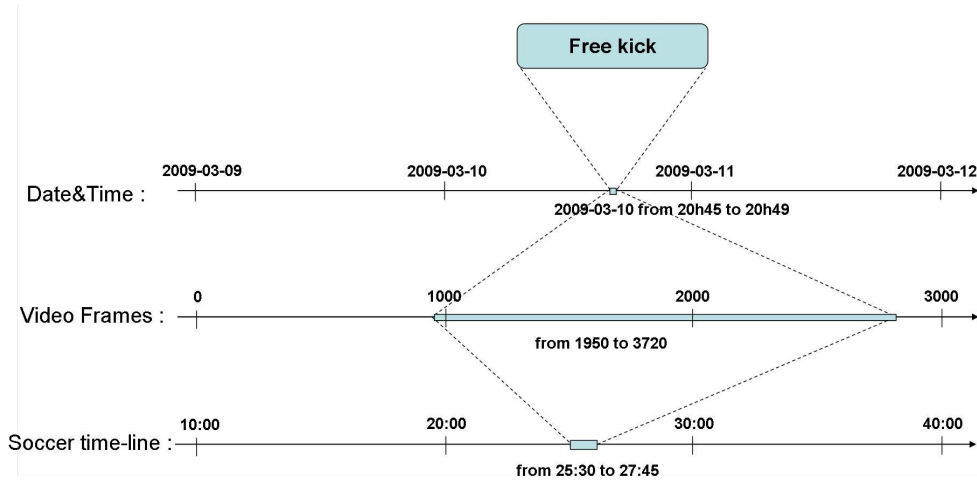


Figure 4.2: An action freekick temporally located according to different frames of reference : Date&Time, soccer time-line and video frames.

- **Geographic FoR:** This FoR corresponds to the system commonly used to locate a position on earth. Names of cities and countries can be used, but also longitude and latitude of the position.
- **Screen FoR:** It corresponds to a region of the 2D euclidean space \mathbb{N}^2 with origin at the top left corner of the image as displayed on a screen. the x-axis is horizontal and oriented from left to right, and the y-axis is vertical and oriented top down. The measure unit is the pixel.
- **2D soccer field FoR:** The 2D soccer field can be seen as a region of the 2D euclidean space \mathbb{R}^2 , in which it is possible to define the position of players and ball in a soccer playfield using, for instance, the meter as a measuring unit.
- **3D soccer field FoR:** The 3D soccer field can be seen as a region of the 3D euclidean space \mathbb{R}^3 used for positioning objects in real world space.

Similarly, the two later FoRs are domain-specific spatial frames of reference that can be defined by the user for domain restricted applications. Definition of new FoRs is possible using predicate symbols as explained below (Definition 5). Figure 4.3 shows the object "ball" that is located according to multiple spatial frames of reference.

Definition 2 (Temporal Constraint). *An atomic temporal constraint is a formula of the form $t \Theta t'$ or $t \Theta c$ where t and t' are variables, c is a constant and Θ is one of $=, \leq, <, \neq, \geq, >$. A complex temporal constraint is a boolean combination built from (atomic or complex) constraints by using logical connectives.*

For instance the constraint $interval_{[a,b]}(t)$ defined by $t \geq a \wedge t < b$ is a complex constraint that refers to the time interval delimited by the two instants a and b . We use "[a" and "]a" to indicate whether the endpoint "a" is to be included or excluded from the set.

Two reasoning tasks will be used on time constraints: entailment and satisfiability. The entailment of two constraints $c_1(t)$ and $c_2(t)$ is denoted by $c_1(t) \Rightarrow c_2(t)$ and is true iff $c_1(t) \wedge \neg c_2(t)$ is unsatisfiable (e.g. $interval_{[2,3]}(t) \Rightarrow interval_{[1,3]}(t)$ is true). Techniques for checking satisfiability and entailment can be found in [34].

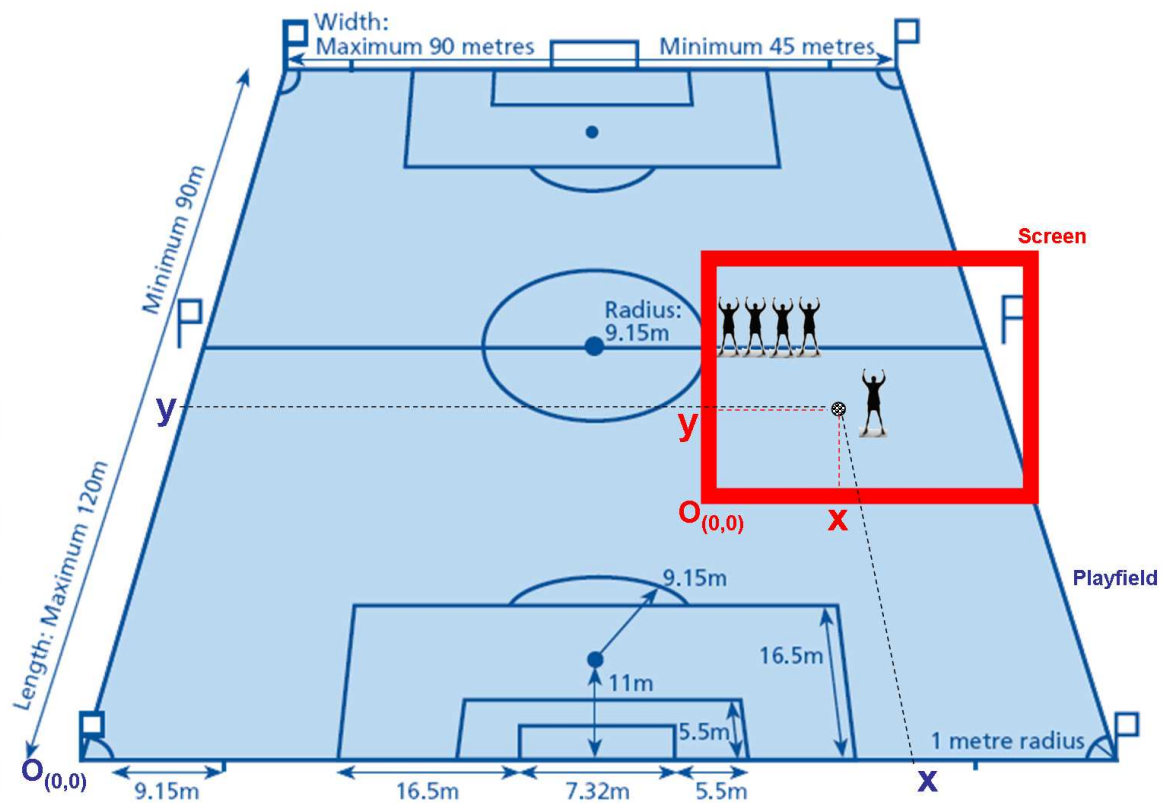


Figure 4.3: Two different references (A soccer field and the camera viewport) and the spatial location of the same object (the ball) in the two references

Let $c(t)$ be a temporal constraint. We define $\inf(c(t)) = \operatorname{argmin}_t\{c(t)\}$ and $\sup(c(t)) = \operatorname{argmax}_t\{c(t)\}$ respectively as the lower and upper bound of $c(t)$.

Definition 3 (spatial Constraint). *An atomic spatial constraint on a n -tuple of variables $(x_1, \dots, x_n) \in \mathbb{R}^n$ is a formula of the form $f(x_1, \dots, x_n) \Theta 0$, where n is called the dimension of the constraint and Θ is one of $=, \leq, <, \neq, \geq, >$. A complex spatial constraint is a boolean combination built from (atomic or complex) constraints by using logical connectives.*

For instance, the spatial constraint $\operatorname{point}_{(a,b)}(x, y)$ representing the point $(a - x = 0) \wedge (b - y = 0)$.

For instance, the spatial constraint $\operatorname{disk}_{(a,b,r)}(x, y)$ representing the area inside a circle of center (a, b) and radius r is represented by: $(x - a)^2 + (y - b)^2 - r^2 < 0$. We use $\overline{\operatorname{disk}}_{(a,b,r)}(x, y)$ to refer to a closed disk.

The complex spatial constraint $\operatorname{rect}_{(x_1,x_2,y_1,y_2)}(x, y)$ refers to the area inside the rectangle represented by: $(x_1 - x < 0) \wedge (x - x_2 < 0) \wedge (y_1 - y < 0) \wedge (y - y_2 < 0)$. We use $\operatorname{rect}_{(\overline{x_1},x_2,y_1,y_2)}(x, y)$ to include a rectangle side into the constraint (here $x = x_1$).

Similarly, the complex spatial constraint $\operatorname{box}_{(x_1,x_2,y_1,y_2,z_1,z_2)}(x, y, z)$ refers to the area inside the box represented by: $(x_1 - x < 0) \wedge (x - x_2 < 0) \wedge (y_1 - y < 0) \wedge (y - y_2 < 0) \wedge (z_1 - z < 0) \wedge (z - z_2 < 0)$.

Let $c(x, y)$ be a spatial constraint. Borders of $c(x, y)$ are defined using the following formulas:

- $\inf_x(c(x, y)) = \operatorname{argmin}_x\{c(x, y)\},$
- $\sup_x(c(x, y)) = \operatorname{argmax}_x\{c(x, y)\},$
- $\inf_y(c(x, y)) = \operatorname{argmin}_y\{c(x, y)\},$
- $\sup_y(c(x, y)) = \operatorname{argmax}_y\{c(x, y)\},$

For instance, $\inf_x(\operatorname{disk}_{(a,b,r)}(x, y)) = a - r,$

Let $c(x, y)$ be a spatial constraint. We note by $c(x, y).X$ (resp. $c(x, y).Y$) the projection of the spatial constraint $c(x, y)$ on the X axis (resp. Y axis). For instance:

$$\begin{aligned} \operatorname{rect}_{(x_1,x_2,y_1,y_2)}(x, y).X &= \operatorname{interval}_{[x_1,x_2]}(x) \\ \operatorname{rect}_{(x_1,x_2,y_1,y_2)}(x, y).Y &= \operatorname{interval}_{[y_1,y_2]}(y) \end{aligned}$$

Spatial constraints can be referred to by predefined nouns like names of cities, countries, streets, geo-positions (latitude and longitude)etc. The following instances are examples of nominated spatial constraints: $\operatorname{city}(\text{London})$, $\operatorname{address}(\text{1 place Vendôme, Paris})$, $\operatorname{country}(\text{Morocco})$, $\operatorname{geoPosition}(\text{21.422510, 39.826169})$. Such nominated spatial constraints are necessary to locate objects using common geographical places instead of using particular coordinates.

Definition 4 (Set-order Constraint). *Let D be a domain. A set-order constraint is one of the following types: $c \in X$, $s \subseteq X$, where c is a constant of type D , s is a set of constants of type D and X is a set variables that range over finite sets of elements of type D .*

Most of the time, people refer to video content essentially using the following descriptors :

- **Object:** Entity of interest appearing in a video sequence such as: person, car, building, etc.

- **Event:** Sequential facts that occur during an interval of time and involving some objects: weeding, soccer game, etc.
- **Attributes:** It is necessary to identify objects and events by specific parameters such as spatial or temporal measures, semantic information or any other characteristics.
- **Relations:** Relations are essential for an expressive description of video content. They include semantic, logic, spatial or temporal links connecting objects or events.

To build the data model of the video database, we assume the existence of the following countably infinite and disjoint sets:

- video identifiers : $\mathcal{ID}_{video} = \{vid_1, vid_2, \dots\}$
- object identifiers : $\mathcal{ID}_{obj} = \{oid_1, oid_2, \dots\}$
- event identifiers : $\mathcal{ID}_{evt} = \{eid_1, eid_2, \dots\}$
- object types : $\mathcal{C}_{obj} = \{C_1, C_2, \dots\}$
- event types : $\mathcal{E}_{evt} = \{E_1, E_2, \dots\}$
- spatial Frames of Reference : $\mathcal{SFOR} = \{sf_1, sf_2, \dots\}$,
- temporal Frames of Reference : $\mathcal{TFOR} = \{tf_1, tf_2, \dots\}$,
- relations : $\mathcal{R} = \{R_1, R_2, \dots\}$,
- attributes : $\mathcal{A} = \{A_1, A_2, \dots\}$,
- (atomic) constants : $\mathcal{D} = \{d_1, d_2, \dots\}$.

4.4 Datalog-like Data Modeling

In this first modeling syntax, we describe resources inside the deductive databases using a Datalog-like syntax based on predicates. In this modeling method, all data types are put in the same level. For this, we define two classes of predicates: the first class *EDB* (for Extensional Database) that contains the predicates corresponding to the stored tables ("the database") and *IDB* (for Intensional Database) that represents predicates appearing in the heads of rules only and enabling for inferring new information not materialized in the database.

4.4.0.1 Syntax

In order to express queries, elements from two sets are used;

- A set \mathcal{C} of constant symbols. These constant symbols refer to values, objects, or events.
- A set \mathcal{V} of variable symbols. These variable symbols refer to value variables, object variables, or event variables.

Definition 5 (Predicate Symbol). *In order to define the predicate symbol set, we assume the existence of the following:*

- Each n-ary relation P in R is associated with a predicate symbol P of arity n .
- A 7-ary predicate symbol $Event$ represents the occurrence of an event within a video following the syntax:

$$Event(eid, name, type, vid, sConst, tConst, certainty_Degree)$$

where :

- $eid \in \mathcal{ID}_{evt}$ is the event identifier,
 - $name$ is a given name to the event eid that can be *null*,
 - $type$ is the type of the event eid . The management of the type of the event is left to the user who can either select the type from a thesaurus to assure interoperability or to write free text.
 - $vid \in \mathcal{ID}_{video}$ is the video identifier where the event is shown.
 - $sConst$ is the spatial constraint referring to the screen zone where the event is shown. It can be for example a rectangle of the screen expressed by $rect_{(a_1, a_2, b_1, b_2)}(x, y)$. One can use the keyword *ANY* to ignore the screen zone, the event then would be considered as shown at any zone of the screen.
 - $tConst$ is the temporal constraint referring to the interval of time of the video sequence when the event is shown. It can be an interval of time $interval_{[t_1, t_2]}(t)$. Similarly, the keyword *ANY* can be used to ignore the time interval.
 - $certainty_Degree \in [0, 1]$ is a the degree of certainty that is, an event of type $type$ that is shown at the mentioned spatiotemporal sequence of video.
- By analogy, a 7-ary predicate symbol $Object$ represents the appearance of an object within a video following the syntax:

$$Object(oid, name, type, vid, sConst, tConst, certainty_Degree)$$

where :

- $oid \in \mathcal{ID}_{obj}$ is the object identifier,
 - $name$ is a given name to the object oid that can be *null*,
 - $type$ is the type of the object oid . Similarly, the management of object types is left to users.
 - $vid \in \mathcal{ID}_{video}$ is the video identifier where the object appears.
 - $sConst$ is the spatial constraint referring to the screen zone where the object appears. Here also, the keyword *ANY*, if used, expresses ignorance of the screen zone.
 - $tConst$ is the temporal constraint referring to the interval of time of the video sequence when the object appears. We use *ANY* to ignore the time interval.
 - $certainty_Degree \in [0, 1]$ is a degree of certainty that is, an object of type $type$ that is shown at the mentioned spatiotemporal sequence of video.
- A predicate $AtDuration$ expresses the occurrence of a video (or an object) within a period of time according to a temporal frame of reference. The following expression:

$$AtDuration(id, tConst_f, tFor, vid, tConst_v)$$

says that an event (or object) id was in $tConst_f$ according to $tFor$, and that this event was shown in the interval of time $tConst_v$ of the video vid . For example, $AtDuration(eid_1, t = "2009 - 03 - 10", \text{Gregorian_Calendar}, vid_4, interval_{[10:00,20:00]}(t))$ expresses that the video vid_4 shows between the 10th and the 20th minute an event eid_1 that occurs at October 3rd, 2009.

- A predicate $AtPosition$ expresses the occurrence of an event (or an object) within a position according to a spatial frame of reference. The following expression:

$$AtPosition(id, sConst_f, sFor, vid, tConst_v)$$

says that an event (or object) id was in $sConst_f$ according to $sFor$, and that this was shown in the interval of time $tConst_v$ of the video vid . For example, $AtPosition(oid_1, city(\text{Fez}), \text{Geographic}, vid_5, interval_{[40:00,50:00]}(t))$ expresses that the video vid_5 shows, between the 40th and the 50th minute, the object oid_1 in the city of Fez.

- A 2-ary predicate $SubObject$:

$$SubObject(oid, eid)$$

that expresses that the object oid is involved in the occurrence of the event eid .

- A 2-ary predicate $SubEvent$:

$$SubEvent(eid_1, eid_2)$$

that expresses that the event eid_1 is a sub event of the event eid_2 .

- A unary predicate $SpatialFoR(sf)$ that enables the users to define a new spatial frame of reference sf . A binary predicate $LocationAtSFoR(v, SFoR)$ that enables the users to add a new spatial location v to the spatial frame of reference $sFoR$. A user can, for instance, define a spatial FoR $myHome$ and add to it new spatial locations like "bedroom", "kitchen", "hall", "guest_room" as the following:

$SpatialFoR(myHome),$
 $LocationAtSFoR(\text{bedroom}, myHome),$
 $LocationAtSFoR(\text{kitchen}, myHome),$
 $LocationAtSFoR(\text{hall}, myHome),$
 $LocationAtSFoR(\text{guest_room}, myHome),$

The user can then use this spatial FoR in order to locate the objects existing in her/his environment and for the understanding of events.

- Similarly, the unary predicate $TemporalFoR(tf)$ enables the users to define a new temporal frames of reference tf . Then, the binary predicate $LocationAtTFoR(v, tFoR)$ enables the users to add a new temporal location v to the temporal FoR $tFoR$.
- Users are able to extend the syntax with additional predicate symbols they need in order to represent their data. They can also define rules combining data model and user defined predicate symbols. The query engine will exploit the defined rules to answer the query.

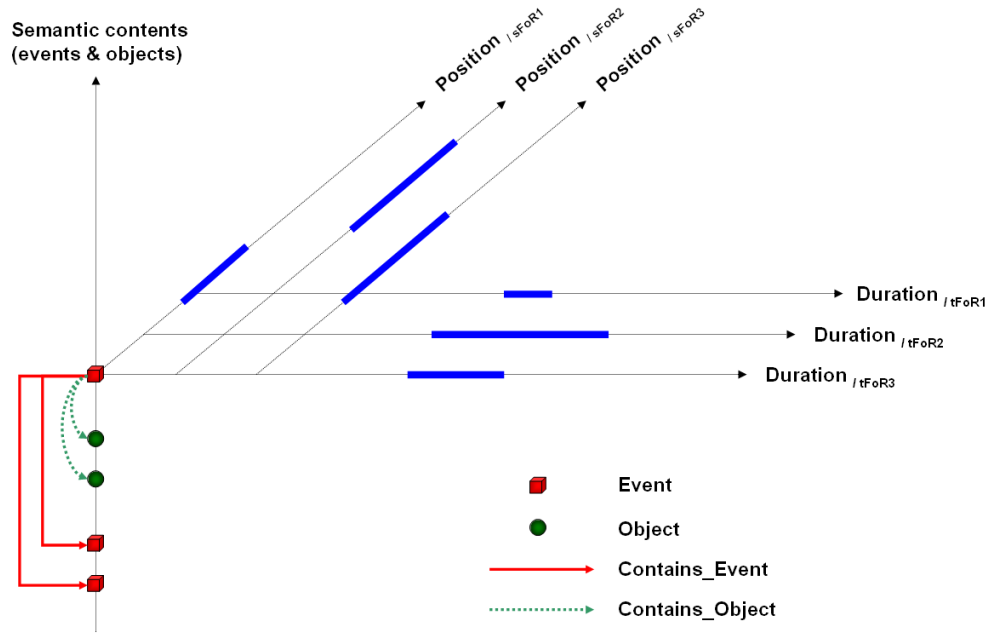


Figure 4.4: Spatiotemporal modeling for an event and its composing subevents and subobjects according to multiple frames of reference.

Figure 4.4 summarizes the structure of data model used to index contents within video databases.

Definition 6 (Rule). We define a rule r by the formula: $r : T \leftarrow L_1, \dots, L_n, c_1, \dots, c_m$ where L_i is a positive literal for all i in $\{1, \dots, n\}$, and where c_i is a constraint for all i in $\{1, \dots, m\}$. T is called the head of the rule while $L_1, \dots, L_n, c_1, \dots, c_m$ is called the body of the rule. A rule is called location-restricted if all the variables of the rule occurs in the body of the rule.

Definition 7 (Program). A program is a collection of location-restricted rules.

Definition 8 (Query). A query is defined as a formula of the form: $q(s)$ where q is a query predicate and s is a tuple of variables and constants.

4.4.1 Example

Let us consider the following event: Monday, March 10th 2009, in Liverpool, the Liverpool soccer player "Gerrard" scores a penalty against the Real-Madrid goalkeeper "Casillas" in the 27th minute of the game. Here, penalty is composed of three elementary events, preparing penalty, shooting ball and scoring goal.

A simple EDB (Extensional Database) representing the event can be described based on our data model as follows:

N.B.:

It is primary to note that this chapter is concerned about the representation aspect of video content within the data model. The way how to get this information, manually or automatically, is not the subject of this chapter.

Event(e_1 , null, "Penalty", v_5 , ANY, *interval*_[02:00,03:00](t), 0.6)
AtPosition(e_1 , *city*(Liverpool), "geographic", v_5 , *interval*_[02:00,02:59](t))
AtPosition(e_1 , "penaltyzone", "2Dsoccer – field", v_5 , *interval*_[02:00,02:59](t))
AtDuration(e_1 , $t = "2009 - 03 - 10"$, "calendar", v_5 , *interval*_[02:00,02:59](t))
AtDuration(e_1 , *interval*_[27:00,27:59](t), "soccertime", v_5 , *interval*_[02:00,02:59](t))
SubEvent(e_{11} , e_1)
SubEvent(e_{12} , e_1)
SubEvent(e_{13} , e_1)
SubObject(o_1 , e_1)
SubObject(o_2 , e_1)
SubObject(o_3 , e_1)

Event(e_{11} , null, "PreparePenalty", v_5 , ANY, *interval*_[02:00,02:25](t))
AtPosition(e_{11} , "penaltyzone", "2Dsoccer – field", v_5 , *interval*_[02:00,02:25](t))
AtDuration(e_{11} , *interval*_[27:00,27:25](t), "soccertime", v_5 , *interval*_[02:00,02:25](t))
SubObject(o_1 , e_{11})
SubObject(o_2 , e_{11})
SubObject(o_3 , e_{11})

Event(e_{12} , null, "ShootBall", v_5 , ANY, *interval*_[02:26,02:30](t))
AtPosition(e_{12} , "penaltyzone", "2Dsoccer – field", v_5 , *interval*_[02:26,02:30](t))
AtDuration(e_{12} , *interval*_[27:26,27:30](t), "soccertime", v_5 , *interval*_[02:26,02:30](t))
SubObject(o_1 , e_{12})
SubObject(o_2 , e_{12})

Event(e_{13} , null, "ScoreGoal", v_5 , ANY, *interval*_[02:31,03:00](t))
AtPosition(e_{13} , "penaltyzone", "2Dsoccer – field", v_5 , *interval*_[02:31,02:59](t))
AtDuration(e_{13} , *interval*_[27:31,27:59](t), "soccertime", v_5 , *interval*_[02:31,02:59](t))
SubObject(o_2 , e_{13})
SubObject(o_3 , e_{13})

Object(o_1 , null, "SoccerBall", v_5 , *point*_(500,400)(x, y), *interval*_[02:00,03:00](t))
AtPosition(o_1 , "PenaltyPoint", "2Dsoccer – field", v_5 , *interval*_[02:00,02:25](t))
AtPosition(o_1 , "GoalBox", "2Dsoccer – field", v_5 , *interval*_[02:31,02:59](t))
AtDuration(o_1 , *interval*_[27:00,27:59](t), "soccertime", v_5 , *interval*_[02:00,02:59](t))

Object(o_2 , "Gerrard", "Player", v_5 , *rect*_(575,625,450,550)(x, y), *interval*_[02:00,02:25](t))
AtPosition(o_2 , *point*_(45,16)(x, y), "2Dsoccer – field", v_5 , *interval*_[02:00,02:25](t))
AtPosition(o_2 , *point*_(45,13)(x, y), "2Dsoccer – field", v_5 , *interval*_[02:31,02:40](t))
AtDuration(o_2 , *interval*_[27:00,27:59](t), "soccertime", v_5 , *interval*_[02:00,02:59](t))

Object(o_3 , "Casillas", "GoalKeeper", v_5 , *rect*_(375,425,250,350)(x, y), *interval*_[02:00,02:25](t))
Object(o_3 , "Casillas", "GoalKeeper", v_5 , *rect*_(325,375,300,400)(x, y), *interval*_[02:26,02:59](t))
AtPosition(o_3 , *point*_(45,0)(x, y), "2Dsoccer – field", v_5 , *interval*_[02:00,02:25](t))
AtPosition(o_3 , *point*_(40,1)(x, y), "2Dsoccer – field", v_5 , *interval*_[02:31,02:40](t))
AtDuration(o_3 , *interval*_[27:00,27:59](t), "soccertime", v_5 , *interval*_[02:00,02:59](t))

foul(o_3 , o_2 , (v_5 , [02 : 00, 02 : 25], "x ∈ [375, 425], y ∈ [250, 350]"))

This EDB can be extended by an IDB (Intensional Database) that specifies rules that can be used to infer new relations. Among the relations, the spatiotemporal relations between video objects and events. In the following, the rule *idb*₀ defines the relation *before* between two intervals (temporal constraint), the rule *idb*₁ tells whether the event e_1 is shown before the event e_2 in the

video sequence v_5 . The rule idb_2 says whether the event e_1 occurs before the event e_2 according to the a specified temporal frame of reference $tFor$. The rule idb_3 says whether the relation $before(e_1, e_2)$ is satisfied at least with a confidence degree α . This is calculated based on the formulas cited in the section 3.3.2.1. The rule idb_4 states whether the relation $before(e_1, e_2, tFor)$ is satisfied at least with a confidence degree α . Similarly, rules can be used to infer information about satisfaction of the other allen's relations between events in videos.

$$idb_0 : before(I_1, I_2) : - \\ Sup(I_1) < Inf(I_2)$$

$$idb_1 : before(e_1, e_2) : - \\ Event(e_1, x_1, y_1, vid, sZone_1, I_1), Event(e_2, x_2, y_2, vid, sZone_2, I_2), \\ Sup(I_1) < Inf(I_2)$$

$$idb_2 : before(e_1, e_2, tFor, vid) : - \\ AtDuration(e_1, Ir_1, tFor, vid, I_1), AtDuration(e_2, Ir_2, tFor, vid, I_2), \\ Sup(Ir_1) < Inf(Ir_2)$$

$$idb_3 : before(e_1, e_2, \alpha) : - \\ Event(e_1, x_1, y_1, vid, sZone_1, I_1), Event(e_2, x_2, y_2, vid, sZone_2, I_2), \\ M = Sup(I_1) - Inf(I_2), T = Sup(I_1) - Inf(I_1), \beta \geq \alpha, \\ ((\beta = 1 \wedge M < 0) \vee (\beta = 1 - M/T \wedge 0 < M \wedge M < T) \vee (\beta = 0 \wedge T < M))$$

$$idb_4 : before(e_1, e_2, tFor, vid, \alpha) : - \\ AtDuration(e_1, Ir_1, tFor, vid, I_1), AtDuration(e_2, Ir_2, tFor, vid, I_2), \\ M = Sup(Ir_1) - Inf(Ir_2), T = Sup(Ir_1) - Inf(Ir_1), \beta \geq \alpha, \\ ((\beta = 1 \wedge M < 0) \vee (\beta = 1 - M/T \wedge 0 < M \wedge M < T) \vee (\beta = 0 \wedge T < M))$$

RCC8 relations defined in the previous chapter, can also be defined through IDB rules as follows. The IDB rules idb_5 and idb_6 say whether the object o_1 and the object o_2 are externally connected.

$$idb_5 : EC(o_1, o_2) : - \\ (meet(o_1.X, o_2.X) \vee meet(o_2.X, o_1.X)), Not(before(o_1.Y, o_2.Y) \vee before(o_2.Y, o_1.Y)) \\ (meet(o_1.X, o_2.X) \vee meet(o_2.X, o_1.X)), Not(before(o_1.Y, o_2.Y) \vee before(o_2.Y, o_1.Y))$$

$$idb_6 : EC(o_1, o_2) : - \\ (meet(o_1.Y, o_2.Y) \vee meet(o_2.Y, o_1.Y)), Not(before(o_1.X, o_2.X) \vee before(o_2.X, o_1.X))$$

New types of events and new semantic relations can be inferred using the IDB predicates. At the following, we describe the *Penalty shootout* event that is a set of penalty kicks shoot at the end of the soccer game (120 min) and used to decide which team is the owner. In order to define the event *Penalty shootout*, idb_7 describes *Penalty shootout start* by catching the first penalty E_{first} played after the 120th min and verifying that no penalty E_{ant} was played after the 120th min and before E_{first} . Then, idb_8 describes *Penalty shootout end* by catching the last penalty played after the 120th min E_{last} and verifying that no penalty E_{post} was played after the 120th min and after E_{last} . Finally, idb_9 gather the first and the last penalties together and returns the time Interval I_P that ranges from the event *Penalty shootout start* and the event

Penalty shootout end. Each of the two events is calculated with a degree of confidence, the degree of confidence of the *Penalty shootout* event is the mean of the degrees of confidence of the two events.

$idb_7 : \text{Event}(Eid_{start}, ANY, "Penalty\ shootout\ start", vid, ANY, I_{start}, Deg_{start}) : -$
 $\text{Event}(E_{first}, ANY, "Penalty", vid, ANY, I_{first}, Deg_{first}),$
 $\text{AtDuration}(E_{first}, Time_{first}, "soccertimeline", vid, ANY), \text{Sup}(Time_{first}) > 120,$
 $\text{Not}(\text{Event}(E_{ant}, ANY, "Penalty", vid, ANY, ANY, ANY),$
 $\text{AtDuration}(E_{ant}, Time_{ant}, "soccertimeline", vid, ANY), \text{Sup}(Time_{ant}) > 120,$
 $\text{before}(E_{ant}, E_{first}, "soccertimeline", vid)),$
 $Deg_{start} = Deg_{first}, I_{start} = I_{first}.$

$idb_8 : \text{Event}(Eid_{end}, ANY, "Penalty\ shootout\ end", vid, ANY, I_{end}, Deg_{end}) : -$
 $\text{Event}(E_{last}, ANY, "Penalty", vid, ANY, I_{last}, Deg_{last}),$
 $\text{AtDuration}(E_{last}, Time_{last}, "soccertimeline", vid, ANY), \text{Sup}(Time_{last}) > 120,$
 $\text{Not}(\text{Event}(E_{post}, ANY, "Penalty", vid, ANY, ANY, ANY),$
 $\text{AtDuration}(E_{post}, Time_{post}, "soccertimeline", vid, ANY), \text{Sup}(Time_{post}) > 120,$
 $\text{before}(E_{last}, E_{post}, "soccertimeline", vid)),$
 $Deg_{end} = Deg_{last}, I_{end} = I_{last}$

$idb_9 : \text{Event}(Eid, ANY, "Penalty\ shootout\ end", vid, ANY, I, Deg) : -$
 $\text{Event}(Eid_{start}, ANY, "Penalty\ shootout\ start", vid, ANY, I_{start}, Deg_{start})$
 $\text{Event}(Eid_{end}, ANY, "Penalty\ shootout\ end", vid, ANY, I_{end}, Deg_{end})$
 $I = \text{interval}_{[Inf(I_{start}), Sup(I_{end})]}(t)$
 $Deg = (Deg_{start} + Deg_{end})/2$

4.4.1.1 Query Types

By integrating spatial modeling into our data model, more complex and specialized queries can be formulated. In this section we give examples of such queries and explain how it could be formulated and processed in our system. In what follows, we refer to variables by uppercase letters and to constants by lowercase letters. Queries are used to retrieve salient objects from different videos, different time intervals, different space locations, following different references and satisfying some spatiotemporal constraints.

The query "List all the segments of films where Tom Cruise appears and the roles he was playing" can be expressed by the following rule. We assume the existence of a predicate role that associates a role with an object within a specified video interval.

$$q(V, I, R) \leftarrow \text{Object}(O, "Tom\ Cruise", "Actor", V, ANY, I),$$

$$\text{Role}(O, R, V, I).$$

The query "List all the athletes that have participated in the 100m sprint running during the Olympic games of Beijing" can be written as follows:

$$q(N) \leftarrow \text{Event}(E_1, "Beijing\ Olympic\ Games", "Olympic\ Games", V_1, S_1, T_1),$$

$$\text{Event}(E_2, "100m\ Spring\ Running", "Spring\ Running", V_2, S_2, T_2),$$

$$\text{SubEvent}(E_2, E_1),$$

$$\text{SubObject}(O, E_1), \text{Object}(O, N, "Athlet", V_3, S_3, T_3)$$

The query "List the names of events that have occurred at January 1st, 1945 and the videos where they are recorded" is expressed as:

$$q(N, V) \leftarrow \text{Event}(E, N, P, V, S, T), \\ \text{AtDuration}(E, t = "1945/01/01", "Date", V, T)$$

The query "List the video sequences shot at April the 1st 2009 in Buckingham Palace where the president Barack Obama appears in the left of the Queen Elizabeth" is expressed by:

$$q(V, I) \leftarrow \text{Object}(O_1, "Barak Obama", "Statesman", V, Sc_1, I), \\ \text{Object}(O_2, "Elizabeth II", "Stateswoman", V, Sc_2, I), \\ \text{AtDuration}(O_1, t = "2009/04/01", "Date", V, I), \\ \text{AtDuration}(O_2, t = "2009/04/01", "Date", V, I), \\ \text{before}(Sc_1.X, Sc_2.X),$$

Let's consider a multi-camera surveillance system installed in a metro station, with a camera fixed on the check point and sending a video stream V_c , and a second camera fixed on the hall of the station and sending a video stream V_h . We can express the query "list the people entering the station hall without crossing the check point" by:

$$q(O) \leftarrow \text{Object}(O, ANY, "Person", V_h, S_h, I_h), \\ \neg(\text{Object}(O, ANY, "Person", V_c, S_c, I_c), \text{before}(I_c, I_h)),$$

4.5 F-Logic like Data Modeling

In the previous section, we made use of a Datalog-like syntax for modeling semantic video data. Although this conventional syntax use logic as a computational formalism and as a data specification language, however it relies on a flat data model and does not support data abstraction. On the other hand, object oriented languages became the most popular programming approach since they enable a number of concepts and capabilities such as complex objects, object identity, methods, encapsulation, typing, and inheritance. F-Logic has been introduced by Kifer [78] to combine the two paradigms and then overcoming the problem of impedance mismatch between programming languages for writing applications and languages for data retrieval. In this section we use F-Logic to represent and retrieve video semantic contents in an object oriented syntax that will enable representing inheritance, abstraction and complex objects, major characteristics of video object and events.

4.5.1 Data Definition Language

Definition 9 (Video Location). A video location l is defined as a triple $(vid, c_t(t), c_s(x, y))$ where vid is a video identifier, $c_t(t)$ is a temporal constraint (cf. Def 2) representing a temporal segment of video vid , and $c_s(x, y)$ is a spatial constraint (cf. Def 3) representing the position of the frames in the temporal segment $c_t(t)$.

For a given video location l , we denote $l.vid$ the video identifier, $l.tmp_const$ the temporal constraint and $l.spc_const$ the spatial constraint.

Definition 10 (Value). The set of values is the smallest set containing the union of the sets **OID**, **OT**, **EID**, **ET**, **SREF**, **TREF** and **D**, and such that, if v_1, \dots, v_n ($n \geq 1$) are values, then so is $\{v_1, \dots, v_n\}$.

The value of attributes associated to objects and events can change during the video, hence value is true within a video location.

Definition 11 (Attribute Value set). Let A be an attribute in \mathcal{A} . We define an attribute value set associated to A as a set of couples (val_i, l_i) where val_i is a value and l_i is the video location where the attribute A_i gets val_i as value.

Example :

- Associating an attribute *color* with the value set $\{(RED, (vid_5, [3s, 5s], ANY)), (BLUE, (vid_5, [6s, 9s], ANY))\}$ means that the color is red between the 3th and the 5th second and blue between the 6th and the 9th second in vid_5 .

Definition 12 (Position). A position is a pair $(c(x_1, \dots, x_n), sF)$ where $c(x_1, \dots, x_n)$ is a spatial constraint and sF is a spatial frame of reference.

For a given position p , we denote $p.spc_const$ its spatial constraint and $p.for$ its spatial frame of reference.

Definition 13 (Duration). A Duration is a pair $(c(t), tF)$ where $c(t)$ is a temporal constraint (t is a variable), and tF is a temporal frame of reference.

For a given duration d , we denote $d.tmp_const$ its temporal constraint and $p.for$ its temporal frame of reference.

Definition 14 (Spatial and Temporal Locations). A spatial location is a couple (S, l) where S is a Position and l the video location. A temporal location is a couple (T, l) where T is a Duration and l the video location.

Example :

- Given a person *Alex* and her attribute *spatiallocation* with a value set $\{((London, Geographic), (vid_5, [5min, 15min], ANY)), ((Paris, Geographic), (vid_4, [4min, 10min], ANY))\}$. That means that from the 5th minute to the 15th of video vid_5 , *Alex* is filmed in *London* whereas from the 4th minute to the 10th minute of video vid_4 , *Alex* is filmed in *Paris*.
- Given an event *goal* and its attribute *temporallocation* associated with a value set $\{((t \in [5 : 27, 6 : 27], soccer_time), (vid_1, [6min, 7min], ANY)), ((t = "1997 - 12 - 01", calender), (vid_1, [6min, 7min], ANY))\}$. That means that the fragment 6th to 7th minute of video vid_1 shows an event *goal* that happens from 5 : 27 to 6 : 27 minutes of soccer match in December 1st, 1997.

Definition 15 (Object). An object is defined by a couple (oid, val) where:

- $oid \in \mathbf{OID}$ is the object identifier.
- val is a n-tuple $[(A_1, v_1), \dots, (A_n, v_n)]$ where $A_i \in \mathcal{A}$, v_i is a value, and $\forall i, j \in [1, n], A_i \neq A_j$ and :
 - $\exists j \in [1, n]$ where $A_j = \text{"type"}$ and $v_j = \{(c_1, d_1), (c_2, d_2), \dots\}$, where $c_i \in \mathcal{C}_{obj}$ and $d_i \in [0, 1]$ is the degree of confidence that oid is of type c_i .
 - $\exists j \in [1, n]$ where $A_j = \text{"video locations"}$ and v_j is the set of video locations where oid appears.

- $\exists j \in [1, n]$ where $A_j = \text{"spatial locations"}$ and v_j is an attribute value set (cf. definition 11) composed of couples (p, l) where p is a *Position* (cf. definition 12) and l is the video location where *oid* was filmed in p .
- $\exists j \in [1, n]$ where $A_j = \text{"temporal locations"}$ and v_j is an attribute value set (cf. definition 11) composed of couples (d, l) where d is a *Duration* (cf. definition 13) and l is the video location where *oid* was filmed at d .
- $\exists j \in [1, n]$ where $A_j = \text{"sub_objects"}$, v_j is the set of elementary objects composing *oid*.

Let $e = (o, val)$ be an object. We note $attr(e) = \{A_1, \dots, A_n\}$ and $value(e) = \{v_1, \dots, v_n\}$.

Definition 16 (Event). An event is defined by a couple (eid, val) where:

- $eid \in EID$ is the object identifier.
- val is a n -tuple $[(A_1, v_1), \dots, (A_n, v_n)]$ where $A_i \in \mathcal{A}$, v_i is a value, and $\forall i, j \in [1, n], A_i \neq A_j$ and :
 - $\exists j \in [1, n]$ where $A_j = \text{"type"}$ and $v_j = \{(c_1, d_1), (c_2, d_2), \dots\}$, where $c_i \in \mathcal{E}_{obj}$ and $d_i \in [0, 1]$ is the degree of confidence that eid is of type c_i .
 - $\exists j \in [1, n]$ where $A_j = \text{"video locations"}$ and v_j is the set of video locations where eid occurs.
 - $\exists j \in [1, n]$ where $A_j = \text{"spatial locations"}$ and v_j is an attribute value set (cf. definition 11) composed of couples (p, l) where p is a *Position* (cf. definition 12) and l is the video location where eid was filmed in p .
 - $\exists j \in [1, n]$ where $A_j = \text{"temporal locations"}$ and v_j is an attribute value set (cf. definition 11) composed of couples (d, l) where d is a *Duration* (cf. definition 13) and l is the video location where eid was filmed at d .
 - $\exists j \in [1, n]$ where $A_j = \text{"sub_events"}$, v_j is the set of elementary events composing eid .
 - $\exists j \in [1, n]$ where $A_j = \text{"objects"}$, v_j is the set of objects appearing within eid .

We refer to the value v_j by $oid.A_j$. Let X be a content (object or event) and $X.spc_loc$ the set of values corresponding to the spatial locations where X is located. We denote by $X.spc_loc_{(sFoR)}$ the subset of $X.spc_loc$ containing only spatial locations calculated according to the spatial frame of reference $sFoR$. Similarly, $X.tmp_loc_{(tFoR)}$ refers to the subset of $X.tmp_loc$ whose elements are temporal locations calculated according to the temporal reference $tFoR$.

4.5.2 Example

The example presented in the previous section (Monday, March 10th 2009, in Liverpool, the Liverpool soccer player "Gerrard" scores a penalty against the Real-Madrid goalkeeper "Casillas" in the 27th minute of the game) can be described using our object oriented DDL as follows: A penalty is composed of three elementary events, preparing penalty, shooting ball and scoring goal. A simple database representing the event can be described based on our data model as follows:

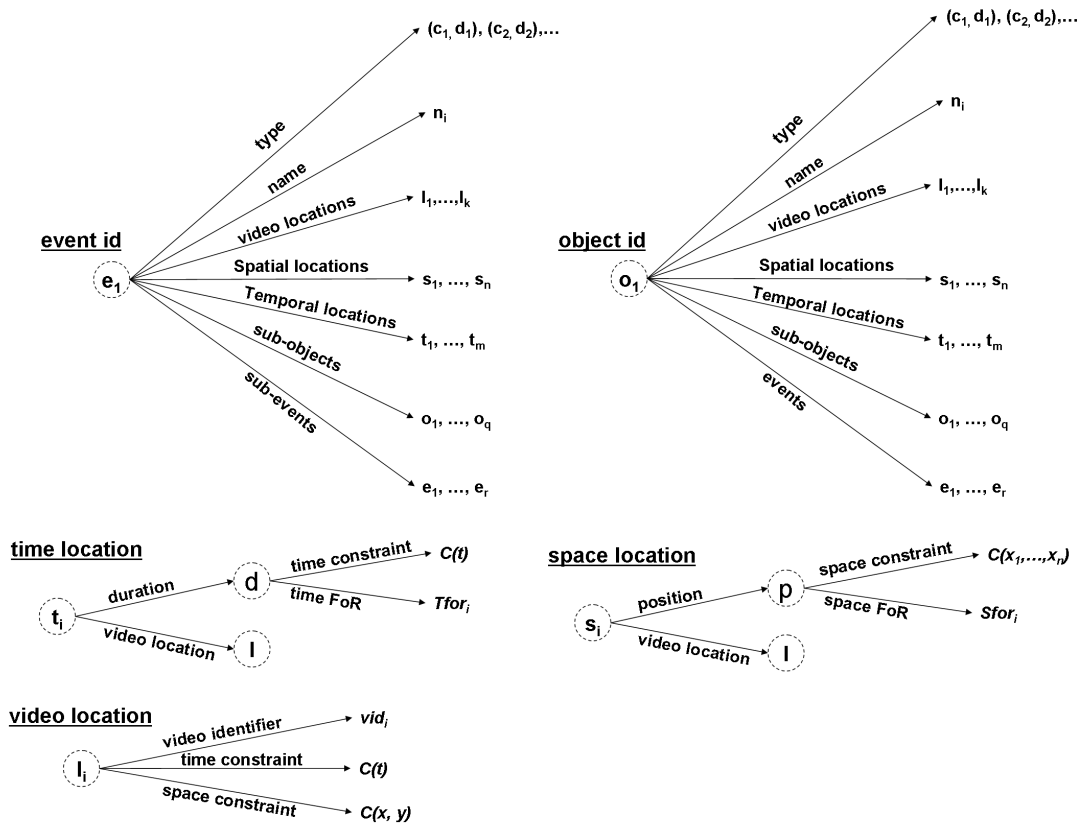


Figure 4.5: The proposed data model for representing video content.

- $e_1 = (eid_1, [type=\{"penalty",1\}], vid_loc=\{(v5, [02:00,03:00],ANY)\}$
 $spc_loc=\{[(Liverpool, geographic),(v5,[02:00,03:00],ANY)],$
 $[(penalty\ zone, soccer\ playfield),(v5,[02:00,03:00],ANY)]\},$
 $tmp_loc=\{[(10/03/2009,calendar),(v5,[02:00,03:00],ANY)],$
 $[(t>27:00,t<28:00, soccer\ time),(v5,[02:00,03:00],ANY)]\},$
 $sub_events=\{e_{11}, e_{12}, e_{13}\}\ objects=\{o_1, o_2, o_3\}\}$.
- $e_{11} = (eid_{11}, [type=\{"preparing\ penalty",1\}], vid_loc=\{(v5, [02:00,02:25],ANY)\}$
 $spc_loc=\{[(penalty\ zone, soccer\ playfield),(v5,[02:00,02:25],ANY)]\},$
 $tmp_loc=\{[(t>27:00,t<27:25, soccer\ time),(v5,[02:00,02:25],ANY)]\},$
 $objects=\{o_1, o_2, o_3\}\}$.
- $e_{12} = (eid_{12}, [type=\{"shooting\ ball",1\}], vid_loc=\{(v5, [02:26,02:30],ANY)\}$
 $spc_loc=\{[(penalty\ zone, soccer\ playfield),(v5,[02:26,02:30],ANY)]\},$
 $t_loc=\{[(t>27:25,t<27:30, soccer\ time),(v5,[02:26,02:30],ANY)]\},$
 $objects=\{o_1, o_2\}\}$.
- $e_{13} = (eid_{13}, [type=\{"scoring\ goal",1\}], vid_loc=\{(v5, [02:31,03:00],ANY)\}$
 $spc_loc=\{[(penalty\ zone, soccer\ playfield),(v5,[02:31,02:59],ANY)]\},$
 $tmp_loc=\{[(t>27:31,t<27:59, soccer\ time),(v5,[02:31,02:59],ANY)]\},$
 $objects=\{o_1, o_2, o_3\}\}$.
- $o_1 = (oid_1, [type=\{"soccer\ ball",1\}],$
 $vid_loc=\{(v5,[02:00,02:29],"x=500,y=400"),(v5,[02:30,03:00],"x=450,y=250")\},$
 $spc_loc=\{[(\{"Penalty\ Point",soccer\ playfield\}), (v5,[02:00,02:25],ANY)],$
 $[(\{"goal\ box",soccer\ playfield\}), (v5,[02:31,03:00],ANY)]\},$
 $tmp_loc=\{[(\{"t>27:00,t<27:59", soccer\ time\}), (v5,[02:00,03:00],ANY)]\},$
- $o_2 = (oid_2, [type=\{"soccer\ player",1\}],name="Gerrard"$
 $vid_loc=\{(v5,[02:00,02:25],"x\in[575,625],y\in[450,550]"),$
 $(v5,[02:26,03:00],"x\in[475,525],y\in[350,450]"),$
 $spc_loc=\{[(\{"x=45,y=16", soccer\ playfield\}), (v5,[02:00,02:25],ANY)],$
 $[(\{"x=45,y=13",soccer\ playfield\}), (v5,[02:31,03:40],ANY)]\},$
 $tmp_loc=\{[(\{"t>27:00,t<27:59", soccer\ time\}), (v5,[02:00,03:00],ANY)]\},$
- $o_3 = (oid_3, [type=\{"soccer\ player",1\}],name="Casillas"$
 $vid_loc=\{(v5,[02:00,02:25],"x\in[375,425],y\in[250,350]"),$
 $(v5,[02:26,03:00],"x\in[325,375],y\in[300,400]"),$
 $spc_loc=\{[(\{"x=45,y=0", soccer\ playfield\}), (v5,[02:00,02:25],ANY)],$
 $[(x=40,y=1,soccer\ playfield),(v5,[02:31,02:40],ANY)]\},$
 $tmp_loc=\{[(\{"t>27:00,t<27:59", soccer\ time\}), (v5,[02:00,03:00],ANY)]\},$
- $foul(o_3, o_2, (v5, [02 : 00, 02 : 25], "x \in [375, 425], y \in [250, 350]"))$

4.5.3 Rule-based Query Language

In this section, a novel declarative rule based language is presented. It allows to reason with objects, events and spatiotemporal constraints in the previous video data model. Queries can refer to both semantic and audio visual layers. They can be specified in fine granularity with the possibility to retrieve only the part of the video where the conditions given in the query are satisfied. Spatial, temporal and semantic conditions are specified as predicates which make it easier and more intuitive to formulate complex query conditions.

4.5.3.1 Syntax

In order to express queries, elements from two sets are used; a constant symbol set \mathbb{C} whose elements refer to values, objects, or events, and a variable symbol set \mathbb{V} whose elements refer to value, object, and event variables.

Definition 17 (Atom). An atom is an expression $P(t_1, \dots, t_n)$ where P is a predicate symbol and t_i is a term for all i in $\{1, \dots, n\}$.

Definition 18 (Predicate Symbol). In order to define the predicate symbols set, we assume that :

- Each relation P in \mathbf{R} of arity n is associated to a predicate symbol P of arity n ,
- Two unary predicate symbols Event and Object represent respectively the events and objects classes,
- Unary predicate symbols spatial_location and Temporal_location refer respectively to spatial_location and Temporal_location attributes. We assume the existence of a predicate Period that returns the period of time that a given video object or event appears within the video or regarding a specific temporal FoR.

$$Period(x, v) \leftarrow \bigvee (l.tmp_const | \forall l \in x.vid_loc \wedge l.vid = v)$$

$$Period(x, v, tf) \leftarrow \bigvee (d.tmp_const | \forall (d, l) \in x.spc_loc_{(tf)}, l.vid = v)$$

The result of predicate Period is a temporal constraint. We use an object oriented syntax to denote $Period(x, v)$ by $x.prd_{(v)}$ and $Period(x, v, tf)$ by $x.prd_{(v, tf)}$.

Definition 19 (Atom). An atom is an expression $P(t_1, \dots, t_n)$ where P is a predicate symbol and t_i is a term for all i in $\{1, \dots, n\}$.

Definition 20 (Rule). We define a rule r by the formula: $r : T : -L_1, \dots, L_n, c_1, \dots, c_m$ where L_i is a positive literal for all $i \in \{1, \dots, n\}$, and where c_i is a constraint for all $i \in \{1, \dots, m\}$. T is called the head of the rule while $L_1, \dots, L_n, c_1, \dots, c_m$ is called the body of the rule. A rule is called location-restricted if all the variables of the rule occurs in the body of the rule.

Definition 21 (Program). A program is a collection of location-restricted rules.

Definition 22 (Query). A query is defined as a formula of the form: $q(s)$ where q is a query predicate and s is a tuple of variables and constants.

4.5.3.2 Query Types

the queries given in the previous chapter can be expressed using an object-oriented syntax as follows:

The query "List all the segments of films where Tom Cruise appears, and the roles he was playing" can be expressed by the following rule:

$$q(V, T_1, T_2, R) \leftarrow \begin{aligned} & \text{Object}(O), ("actor", 1) \in O.type, O.name = "Tom Cruise", \\ & L \in O.vid_loc, L.video = V, L.tmp_const \Rightarrow t \in [T_1, T_2], \\ & (R, L) \in O.role. \end{aligned}$$

The query "List all the athletes that have participated in the 100m sprint running during the olympic games of Beijing" can be written as follows:

$$q(N) \leftarrow \begin{aligned} & \text{Event}(E_1), \text{Event}(E_2), ("100m spring running", 1) \in E_1.type, \\ & ("Beijing Olympic Games", 1) \in E_2.name, E_1 \in E_2.sub_events, \\ & \text{Object}(O), O \in E_1.objects, O.type = \{"athlete"\}, N \in O.name,. \end{aligned}$$

The query "List the events that have occurred at January 1st, 1945 and the videos where they are filmed" is expressed as:

$$q(E, V) \leftarrow \begin{aligned} & \text{Event}(E), \text{Duration}(D), \text{Vid_Loc}(L), (D, L) \in E.tmp_loc(Calendar), \\ & D.tmp_const \Rightarrow \{t = "1945-01-01"\}, V = L.vid \end{aligned}$$

The query "List the video sequences shot at April the 1st 2009 in Buckingham Palace where the president Barack Obama appears in the left of the Queen Elizabeth" is expressed by:

$$q(E) \leftarrow \begin{aligned} & \text{Event}(E), \text{Object}(O_1), \text{Object}(O_2), \{O_1, O_2\} \subset E.objects, \\ & O_1.name = \{Barak Obama\}, O_2.name = \{Queen Elizabeth\}, \\ & (P_1, L_1) \in O_1.spc_loc("Residences"), (P_2, L_2) \in O_1.spc_loc("Residences") \\ & P_1.spc_const \Rightarrow "Buckingham", P_2.spc_const \Rightarrow "Buckingham", \\ & L_1.tmp_const.sup_{(x)} > L_2.tmp_const.sup_{(x)}, \\ & (D_1, L_1) \in O_1.tmp_loc("Calendar"), (D_2, L_2) \in O_1.spc_loc("Calendar") \\ & D_1.tmp_const \Rightarrow \{t = "2009-04-01"\}, D_2.spc_const \Rightarrow \{t = "2009-04-01"\} \end{aligned}$$

Using this definition of the relation *shaking_hands*, we can formulate the query "List the video sequences where Secretary-General of the Arab League Amr Moussa is shaking hands with the UN Secretary-General Ban Ki-moon".

$$q(V, T_1, T_2) \leftarrow \begin{aligned} & \text{Object}(O_1), \text{Object}(O_2), \{O_1, O_2\} \subset E.objects, \\ & O_1.name = \{Amr Moussa\}, O_2.name = \{Ban Ki - moon\}, \\ & shaking_hands(O_1, O_2, L), \text{Vid_loc}(L), L.tmp_const = t \in [T_1, T_2]. \end{aligned}$$

Let's consider a multi-camera surveillance system installed in a metro station, with a camera fixed on the check point and sending a video stream V_c , and a second camera fixed on the hall of the station and sending a video stream V_h . We can express the query "list the people entering the station hall without crossing the check point" by:

$$q(O) \leftarrow \begin{aligned} & \text{Object}(O), \text{forall}(L \in O.vid_loc), L.vid \neq V_c, \\ & \text{exist}(L_h \in O.vid_loc), L_h.vid \neq V_h, \} \end{aligned}$$

The query "List the video sequences of free kicks scored by the Arsenal soccer team, where the ball was deviated by the opposite team, list the ball shooter and the player who deviated the ball".

$$\begin{aligned}
 q(O_1, O_2, E) \quad \leftarrow \\
 & \text{Event}(E), E.type = \text{"freekick"}, \\
 & \text{Event}(E_1), (\text{"shooting ball"}, 1) \in E_1.type, \\
 & \text{Event}(E_2), (\text{"touching ball"}, 1) \in E_2.type, \\
 & \text{Event}(E_3), (\text{"scoring goal"}, 1) \in E_3.type, \\
 & T_1 \in E_1.Time, E_1 \in G.subEvents, \\
 & T_2 \in E_2.Time, E_2 \in G.subEvents, \\
 & T_3 \in E_3.Time, E_3 \in G.subEvents, \\
 & T_1.value.ref = T_2.value.ref = T_3.value.ref, \\
 & T_1.dur_{(soccer-time)}.sup < T_2.dur_{(soccer-time)}.inf, \\
 & T_2.dur_{(soccer-time)}.sup < T_3.dur_{(soccer-time)}.inf, \\
 & \text{Object}(O_1), O_1 \in E1.Objects, \\
 & (\text{"soccer player"}, 1) \in O_1.type, \\
 & O_1.team = \{\text{"Arsenal Football Club"}\} \\
 & \text{Object}(O_2), O_2 \in E2.Objects, \\
 & (\text{"soccer player"}, 1) \in O_2.type, \\
 & \{\text{"Arsenal Football Club"}\} \not\subseteq O_2.team
 \end{aligned}$$

4.5.4 Inferring new relations

4.5.4.1 Advanced Temporal Relations

Rules can be used to specify new relationships between objects and events within video sequences. In this section we use the object-oriented modeling in order to define new spatial and temporal relations between objects and events.

Given the above temporal representation of events, temporal Allen's relations [8] can be expressed by rule-based queries. Such relations will be defined with arity 3 if the spatial or the temporal reference is specified. If they are defined or used with arity 2, this will be for reasoning based on the absolute spatial and temporal references.

Let x_1 and x_2 be two video contents (objects or events). The temporal relations of Allen between durations of the two contents following any temporal reference can be expressed as follows:

$$\begin{aligned}
 before(x_1, x_2, tRef) \quad : - \quad & x_1.dur_{(tRef)}.sup < x_2.dur_{(tRef)}.inf. \\
 meets(x_1, x_2, tRef) \quad : - \quad & x_1.dur_{(tRef)}.sup = x_2.dur_{(tRef)}.inf. \\
 overlaps(x_1, x_2, tRef) \quad : - \quad & x_1.dur_{(tRef)}.sup > x_2.dur_{(tRef)}.inf \\
 & \wedge \\
 & x_1.dur_{(tRef)}.inf < x_2.dur_{(tRef)}.inf.
 \end{aligned}$$

$$\begin{aligned}
 \text{during}(x_1, x_2, tRef) & : - & x_1.dur_{(tRef)}.inf > x_2.dur_{(tRef)}.inf \\
 & & \wedge \\
 & & x_1.dur_{(tRef)}.sup < x_2.dur_{(tRef)}.sup. \\
 \\
 \text{finishes}(x_1, x_2, tRef) & : - & x_1.dur_{(tRef)}.sup = x_2.dur_{(tRef)}.sup. \\
 \\
 \text{starts}(x_1, x_2, tRef) & : - & x_1.dur_{(tRef)}.inf = x_2.dur_{(tRef)}.inf. \\
 \\
 \text{equal}(x_1, x_2, tRef) & : - & x_1.dur_{(tRef)}.inf = x_2.dur_{(tRef)}.inf \\
 & & \wedge \\
 & & x_1.dur_{(tRef)}.sup = x_2.dur_{(tRef)}.sup.
 \end{aligned}$$

The previous relations can be defined in a fuzzy way for two reasons. In one hand to take into account uncertainty due to errors and imprecision of video content descriptions, and on the other hand to handle queries formulated by users using vague concepts like "very close", "far from"...

An event A occurs before an event B if more than the half of the duration of event A lasts before the beginning of event B. If only the half of the duration of A occurs before B, the relation is satisfied with certainty degree 0. If all A's duration is before B the relation is satisfied with degree 1 (100%).

Let α be the minimum accepted degree of certainty for the satisfiability of the relation before.

The relation before will be formulated as follows :

$$\begin{aligned}
 \text{before}(x_1, x_2, \alpha, tRef) & : - & D = x_1.dur_{(tRef)}.length/2, \\
 & & H = x_1.dur_{(tRef)}.sup - x_2.dur_{(tRef)}.inf, \\
 & & (1 - H/D) \geq \alpha,
 \end{aligned}$$

Similarly, the relation *meets* can be defined fuzzily as follows.

$$\begin{aligned}
 \text{meets}(x_1, x_2, \alpha, tRef) & : - & D = x_1.dur_{(tRef)}.length/4, \\
 & & H = abs(x_1.dur_{(tRef)}.sup - x_2.dur_{(tRef)}.inf), \\
 & & (1 - H/D) \geq \alpha,
 \end{aligned}$$

the relation *overlaps* can be defined as follows:

$$\begin{aligned}
 \text{overlaps}(x_1, x_2, \alpha, tRef) & : - & D = x_1.dur_{(tRef)}.length/2, \\
 & & H = x_1.dur_{(tRef)}.sup - x_2.dur_{(tRef)}.inf - D/2, \\
 & & H/D \geq \alpha,
 \end{aligned}$$

the relation *during* can be defined as follows:

$$\begin{aligned}
 \text{overlaps}(x_1, x_2, \alpha, tRef) & : - & D = x_1.dur_{(tRef)}.length/2, \\
 & & C = x_1.dur_{(tRef)} \wedge x_2.dur_{(tRef)}, \\
 & & H = C.length - x_1.dur_{(tRef)}.length, \\
 & & (1 - H/D) \geq \alpha,
 \end{aligned}$$

the relation *finishes* can be defined fuzzily as follows.

$$\begin{aligned}
 \text{finishes}(x_1, x_2, \alpha, tRef) & : - & D = x_1.dur_{(tRef)}.length/4, \\
 & & H = abs(x_1.dur_{(tRef)}.sup - x_2.dur_{(tRef)}.sup), \\
 & & (1 - H/D) \geq \alpha,
 \end{aligned}$$

the relation *starts* can be defined fuzzily as follows.

$$\begin{aligned}
D &= x_1.dur_{(tRef)}.length/4, \\
starts(x_1, x_2, \alpha, tRef) &: - \quad H = abs(x_1.dur_{(tRef)}.inf - x_2.dur_{(tRef)}.inf), \\
&\quad (1 - H/D) \geq \alpha,
\end{aligned}$$

the relation *equals* can be defined fuzzily as follows.

$$equals(x_1, x_2, \alpha, tRef) : - \quad starts(x_1, x_2, \alpha, tRef) \wedge finishes(x_1, x_2, \alpha, tRef)$$

4.5.4.2 Advanced Spatial Relations

The definition of spatial relations differs following the model used to represent the spatial property of video objects. In order to simplify the representation, we formulate these relationships reasoning on MBR (Minimum Boundary Rectangle) of the video objects.

Spatial constraints of multiple dimensions can be decomposed into elementary spatial constraint having a unary dimension. These unary dimensional spatial constraints can be considered as temporal constraints. Therefore, we use the previous definitions of temporal relationships in order to define the advanced qualitative spatial RCC8 relationships.

The spatial relation disconnected (DC) can be defined as follows :

$$DC(x_1, x_2, sRef) : - \quad x_1.dur_{(sRef)} \wedge x_2.dur_{(sRef)} = \emptyset$$

The spatial relation Externally Connected (EC) can be defined as follows :

$$\begin{aligned}
EC(x_1, x_2, sRef) &: - \\
&c_{1X} = x_1.dur_{(sRef)}.dim_X, c_{2X} = x_2.dur_{(sRef)}.dim_X, \\
&c_{1Y} = x_1.dur_{(sRef)}.dim_Y, c_{2Y} = x_2.dur_{(sRef)}.dim_Y, \\
&((meets(c_{1X}, c_{2X}, sRef) \vee meets(c_{2X}, c_{1X}, sRef)) \\
&\wedge c_{1Y} \wedge c_{2Y} \neq \emptyset) \\
&\vee \\
&((meets(c_{1Y}, c_{2Y}, sRef) \vee meets(c_{2Y}, c_{1Y}, sRef)) \\
&\wedge c_{1X} \wedge c_{2X} \neq \emptyset)
\end{aligned}$$

The spatial relation Tangential Proper Part (TPP) can be defined as follows :

$$\begin{aligned}
TPP(x_1, x_2, sRef) &: - \\
&c_{1X} = x_1.dur_{(sRef)}.dim_X, c_{2X} = x_2.dur_{(sRef)}.dim_X, \\
&c_{1Y} = x_1.dur_{(sRef)}.dim_Y, c_{2Y} = x_2.dur_{(sRef)}.dim_Y, \\
&x_1.dur_{(sRef)} \Rightarrow x_2.dur_{(sRef)} \wedge \\
&(starts(c_{1X}, c_{2X}, sRef) \\
&\vee starts(c_{1Y}, c_{2Y}, sRef) \\
&\vee starts(c_{2X}, c_{1X}, sRef) \\
&\vee starts(c_{2Y}, c_{1Y}, sRef) \\
&\vee finishes(c_{1X}, c_{2X}, sRef) \\
&\vee finishes(c_{1Y}, c_{2Y}, sRef) \\
&\vee finishes(c_{2X}, c_{1X}, sRef) \\
&\vee finishes(c_{2Y}, c_{1Y}, sRef))
\end{aligned}$$

The spatial relation Non-Tangential Proper Part (NTPP) can be defined as follows:

$$\begin{aligned}
NTPP(x_1, x_2, sRef) &: - \\
&x_1.dur_{(sRef)} \Rightarrow x_2.dur_{(sRef)} \wedge \\
&\neg TPP(x_1, x_2, sRef)
\end{aligned}$$

The spatial relation Partially Overlapping (P0) can be defined as follows :

$$\begin{aligned}
 PO(x_1, x_2, sRef) \quad : - \\
 & c_{1X} = x_1.dur_{(sRef)}.dim_X, c_{2X} = x_2.dur_{(sRef)}.dim_X, \\
 & c_{1Y} = x_1.dur_{(sRef)}.dim_Y, c_{2Y} = x_2.dur_{(sRef)}.dim_Y, \\
 & (x_1.dur_{(sRef)} \wedge x_2.dur_{(sRef)} \neq \emptyset) \wedge \\
 & (\neg meets(c_{1X}, c_{2X}, sRef) \\
 & \wedge \neg meets(c_{1Y}, c_{2Y}, sRef) \\
 & \wedge \neg meets(c_{2X}, c_{1X}, sRef) \\
 & \wedge \neg meets(c_{2Y}, c_{1Y}, sRef))
 \end{aligned}$$

The spatial relation Equal (EQ) can be defined as follows :

$$\begin{aligned}
 EQ(x_1, x_2, sRef) \quad : - \\
 & x_1.dur_{(sRef)} \Leftrightarrow x_2.dur_{(sRef)}
 \end{aligned}$$

The spatial relation Tangential Proper Part inverse (TPP^{-1}) can be defined as follows:

$$\begin{aligned}
 TPP^{-1}(x_1, x_2, sRef) \quad : - \\
 & TPP(x_2, x_1, sRef),
 \end{aligned}$$

The spatial relation Non-Tangential Proper Part inverse ($NTPP^{-1}$) can be defined as follows:

$$\begin{aligned}
 NTPP^{-1}(x_1, x_2, sRef) \quad : - \\
 & NTPP(x_2, x_1, sRef),
 \end{aligned}$$

With $abs(x)$ refers to the absolute value of x . overlaps also can be defined fuzzily using a similar method. Using this definition of the relation externally connected (EC), we can formulate the query "List the video sequences where the president "Obama" appears shaking hands with the UN general-secretary "Ban Ki-moon".

$$\begin{aligned}
 q(V, T_1, T_2) \quad \leftarrow \\
 & Object(O_1), Object(O_2), \\
 & O_1.role = \{ "USA president" \}, \\
 & O_1.name = \{ Obama \}, \\
 & O_2.role = \{ "UN general secretary" \}, \\
 & O_2.name = \{ Ban Ki - moon \}, \\
 & L_1 \in O_1.locations, L_2 \in O_2.locations, \\
 & L_1.v = L_2.v = V, \\
 & L_1.I = L_2.I = [T_1, T_2], \\
 & meets(L_1.P, L_2.P) \vee meets(L_2.P, L_1.P),
 \end{aligned}$$

4.5.4.3 Trajectory Queries

Objects and events are associated, in the data model, with their spatiotemporal locations following specified references. Trajectory can then be easily inferred by ordering spatial locations of the tracked object through a specific time interval. While most previous approaches propose only queries about similarity of object trajectories (e.g. [40]), our architecture can support other kinds of trajectory queries and with more extended definition of uncertainty degree. In addition, while the majority of previous approaches were able to identify the trajectory of the object with regards to its positions in the frame, the present work enables to calculate this trajectory according to any spatial reference and temporal reference. Hence, we can track a player according to her/his po-

sition in the playfield, a car in the city avenues, a man in house rooms, etc. All this according to video timing, day timing, or event timing (e.g. soccer).

Three types of trajectory queries can be formulated: indemnifying objects' trajectories, comparing trajectories of different objects and retrieving objects following specified trajectories:

4.5.4.4 Trajectory identification

This is the first category where the aim is to get the route of the tracked object on the video in consecutive frames.

Example: The query : "Get the trajectory followed in a Car chase event". In this case, the spatial frame of reference would be a user defined spatial FoR "city map" where spatial constraints are the streets and the avenues of the city. This can be defined using the predicates *SpatialFoR* and *LocationAtSFoR* presented by the definition 5. The temporal reference will be Time.

$$\begin{aligned}
 Q(V, c, P, I) \quad \leftarrow \\
 & \text{Object}(c), c.type = "Car", \\
 & (S, L_1) \in c.space_{(city_map)}, S.const = P, \\
 & (T, L_2) \in c.time_{(Time)}, \\
 & L_1.inter = L_2.inter, L_1.vid = L_2.vid = V, \\
 & T.const = I
 \end{aligned}$$

4.5.4.5 Trajectory similarity

Given two video objects evolving in a specific environment, users can ask about the similarity of the trajectory followed by the two objects. Using our formalism, the user can ask for an approximate similarity rather than the exact similarity proposed by the majority of cited approaches. Let o_1 and o_2 be two video objects. The similarity of the trajectory of o_1 and the trajectory of o_2 depends on two parameters:

- α : Incertitude degree corresponding to how many positions visited by o_2 are also visited by o_1 .
- β : Incertitude degree corresponding in how the neighbor positions of the two objects are exactly matching.

$$sameTrajectory(o_1, o_2, sRef, tRef, \alpha, \beta) \quad :-$$

$$\begin{aligned}
 & P_1 = \{T_1 | (T_1, L_1) \in o_1.time_{tRef}, \\
 & (T_2, L_2) \in o_2.time_{tRef}, \\
 & (S_1, L_1) = o_2.space_{sRef}, \\
 & (S_2, L_2) = o_2.space_{sRef}, \\
 & L_1.vid = L_2.vid, L_1.inter = L_2.inter, \\
 & during(S_1.const, S_2.const, sRef, \beta)\}, \\
 & P_2 = \{T_2 | (T_2, L) \in o_2.time_{tRef}\}, \\
 & length(P_1) \geq \alpha \times length(P_2),
 \end{aligned}$$

4.5.4.6 Trajectory-based retrieval

In this category of queries, the aim is to retrieve all the objects that have been located in specified positions. The temporal locations can be fixed or not fixed in case they were not important.

Let $Tr = \{(["eiffel\ tour", "monuments"], ["10h00-11h00", "day\ time"]), (["Arc\ de\ Triomphe", "monuments"], ["11h00-12h00", "day\ time"]), (["Louvre\ Museum", "monuments"], ["12h00-18h00", "day\ time"])\}$ be trajectory for visiting Paris. The query "List all the video related to visiting Paris monuments where tourists have followed the same trajectory in (Tr)"

Assume that $sRef = "monuments"$ and $tRef = "daytime"$. $Q(V, sRef, tRef, \alpha, \beta) : -$

$Object(O), O.type = "tourist",$
 $P_1 = \{T_1 | (S_1, L_1) \in O.space(sRef),$
 $(T_1, L_2) \in O.time(tRef),$
 $(S_2, T_2) \in Tr,$
 $L_1.vid = L_2.vid, L_1.inter = L_2.inter,$
 $EQ(S_1.const, S_2.const, sRef, \beta)\},$
 $during(T_1.const, T_2.const, tRef, \beta)\},$
 $length(P_1) \geq \alpha \times length(Tr),$

4.6 Conclusion

In this chapter, a novel framework, for modeling, indexing and querying semantic objects and events from video documents has been presented. The framework is based on a novel data model enabling spatial, temporal, and semantic modeling of events and objects occurring in video documents. The proposed model is hierarchical since it allows to describe video segments with regards to multiple levels and variable granularity. Its architecture makes it easily extendable and tailored for specific requirements. To the best of our knowledge, our model is one of the first proposals allowing to represent the temporal and spatial information following multiple references, and combining objects, events and relations for modeling video data. In order to efficiently explore and access the right information a declarative, rule based, constraint query language is proposed. Spatiotemporal relations are represented as predicates and new spatial, temporal, or semantic relationships can be easily inferred. Our model combines objects, events and relations for specifying semantics of video data specified by multiple spatial and temporal frames of reference. Many interesting directions can be pursued:

1. This work can be extended to the feature and content layer by attaching to predicate symbols appropriate programs extracting the features from the visual content.
2. The problem of sequence presentation is not studied in this chapter. A declarative and graphical language connected to our query language can provide adequate flexibilities to this issue.

When in doubt, use brute force.

Butler Lampson, 1984 - "Hints for Computer System Design"

5

Prototype

▷ *In this chapter, we present a brief overview of the first release of the prototype we have developed. It is composed of four main parts: object detectors, event model editor, manual annotator, and the event detector.*

◁

Contents of the chapter

5.1	Introduction	115
5.2	Object Detectors	115
5.2.1	Playfield Detector	116
5.2.2	Players Detection	116
5.2.3	Line borders Detection	118
5.3	Video Annotator Tool	118
5.4	Model Editor	120
5.5	Event Detector	127
5.6	Representation Syntax	127
5.6.1	Concept syntax	127
5.6.2	Occurrence syntax	128
5.6.3	Event model syntax	129
5.7	Conclusion	130

5.1 Introduction

While most detection frameworks of visual events usually summarize the extracted events and present them with no ability for users to request additional information, our detection framework is designed in order to be more adapted to user needs. This framework would enable users to precisely express the structure of the events they want to retrieve from a video collection. The objective is to combine annotations provided by manual annotator tool and those emanated from automatic object classifications in order to detect complex events. The framework is composed of many important packages integrated together in order to fulfill the task of event detection. The implemented packages of the framework are highlighted in the figure 5.1.

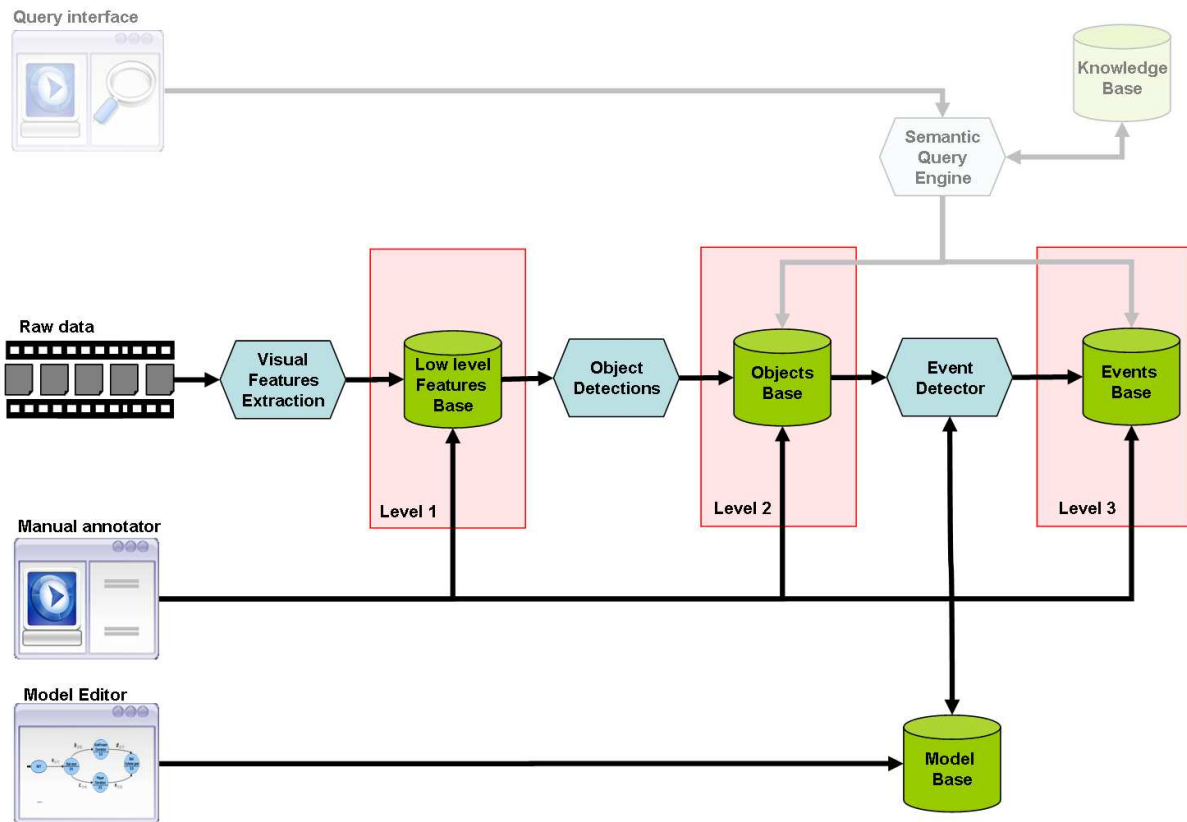


Figure 5.1: Positioning the chapter contribution within the whole framework.

5.2 Object Detectors

The first important package is the one containing all the algorithms for detecting and classifying elementary objects. In our case, we have focused on soccer videos in order to detect important highlights. For this, we have implemented and improved some existing algorithms for detecting important objects such as playfield, players, lines...

5.2.1 Playfield Detector

Detecting playfield is the a major step for extracting further semantics and understanding soccer videos. A Soccer playfield often consists of grass. The idea then is to detect pixels corresponding to color and texture of grass. For this aim we propose to use a simple but powerful method widely used in soccer video analysis [62] as well as for skin detection [75].

This method is based on color histogram learning techniques. A training set is composed of soccer video frames, then color models for playfield pixels and non-playfield pixels are learned. Color models are composed of RGB color histogram with N bins per channel in the RGB color space. Each playfield pixel is placed into the appropriate bin of the playfield histogram. We proceed similarly to learn color models for non-playfield pixels. Afterwards, we use the statistics of the two histograms to calculate, for each bin (r, g, b) , the following discrete probability distribution:

$$P((r, g, b)|playfield) = \frac{N_{((r,g,b)|playfield)}}{N_{(playfield)}}$$

$$P((r, g, b)|nonPlayfield) = \frac{N_{((r,g,b)|nonPlayfield)}}{N_{(nonPlayfield)}}$$

where $N_{((r,g,b),playfield)}$ is the number of pixels in the bin (r,g,b) labeled as playfield pixels, $N_{((r,g,b),nonPlayfield)}$ is the number of pixels in the bin (r,g,b) labeled as non-playfield pixels, $N_{(playfield)}$ is the number of pixels in the playfield histogram, and $N_{(nonPlayfield)}$ is the number of pixels in the non-playfield histogram. Therefore, a pixel classifier is derived using the likelihood ratio as follows:

$$R((r, g, b)) = \frac{P((r,g,b)|playfield)}{P((r,g,b)|nonPlayfield)}$$

$$\begin{cases} (r, g, b) \text{ is playfield pixel} & \text{if } 0 \leq R \geq \theta \\ (r, g, b) \text{ is nonplayfield pixel} & \text{if } 0 \leq R \geq \theta \end{cases}$$

where $\theta > 0$ is a threshold which can be adjusted to separate between correct detections and false positives [62]. Figure 5.2 shows the result of the detection of soccer playfield using the method described above.

5.2.2 Players Detection

After performing playfield pixel detection on each video frame, a binary mask where foreground consists of non-playfield pixels and background consists of playfield pixels. Morphological filtering [123](Erosion+Dilatation) is performed on each frame to eliminate the noise and to smooth the borders of playfield. Then, Connected Components Analysis (CCA) scans the pixels of the frame and gathers the ones that have the same color into connected regions using an 8-connectivity neighborhood [124]. The biggest background region is assigned to playfield while the biggest foreground region is assigned to the public. The remaining components are ideally foreground regions inside the playfield that correspond to players, referees and the ball. Automatic detection of regions corresponding to TV logos score area is not performed in this work. The position of such a component in general remains unchanged during a specified soccer game video. In our case the positions of these components are manually specified for each video.



Figure 5.2: A screen shot of the real-time detection of soccer playfield

Generally, the sizes of components corresponding to humans inside the playfield are comparable. Only the ball is clearly smaller than all the other components. The mean size of components inside the extracted playfield area is calculated. The components whose size exceeded on-third of the calculated mean size are then labeled as players and referee. The components whose size is below on-third of the mean size are considered as candidate components for corresponding to the ball. Figures 5.3 and 5.4 display the result of real-time multi-player tracking performed by our player detection algorithm.

5.2.3 Line borders Detection

In order to detect lines on playfield, the zone of detection is first delimited. This one corresponds to the zone delimited by borders of the playfield component detected in the previous stage. The pixels corresponding to players inside the playfield are also subtracted. Then, a standard Sobel edge detection is performed on resulting zone. Then, a morphological dilatation is performed on the resulting image. A Hough Transform is then used to detect the line parameters as described in [107]. For this, we use the sinusoidal equation of lines: $\rho = x \times \cos(\theta) + y \times \sin(\theta)$. After Hough Transform is done, the resulting lines are filtered based on their angles with the x-axis. We exploit the fact that there is no line intersection in soccer playfield except for perpendicular lines. We begin by dividing the resulting lines into two sets. The lines that have positive angle with $x - axis$ are put in one set and those that have negative angle with $x - axis$ are put in the other set. The two sets correspond to the horizontal and vertical lines of the playfield. Then, there should be no intersection between two lines from the same set. If an intersection is found between two lines in the same set, only the line that has the higher number in hough accumulator is maintained.

Once detected, object occurrences are stored in an XML file following the syntax shown in section 5.6.2.

5.3 Video Annotator Tool

In addition to automatic classification algorithms for detecting objects of interests, manual annotation can be used to populate the object base of the framework.

The Video-Annotator module enables the user to easily annotate segments of the frames in a video using a set of descriptors referring to some objects appearing in the video. This enables for populating the system's object database with this data to be used for semantic video queries and event detections. The user can annotate the video using domain-specific concepts that are stored in XML files following the syntax shown in section 5.6.1. One concept file chosen, each concept is represented, in a left-side panel, with a labeled button that shows the shape that can be used to annotate the concept (figure 5.5). The tool is also provided with facilities for viewing, updating, modifying, and deleting semantic annotations that have already been stored.

For enabling exact region-oriented annotations of the video frames, each descriptor is associated with the convenient shape that fits well in general with 2D-projection of the object to annotate. For example, balls are annotated using spheres, goal-boxes and a playfield zones are annotated using polygons, players are annotated using rectangles, etc.

The annotation is totally made using the mouse which simplifies the process and allows for fast annotation. Since from a frame to the successor frame, most objects remain appearing in the screen, the tool is provided with the facility to copy the annotation of the previous frame,



Figure 5.3: Real time multi-players tracking



Figure 5.4: Evolution of the real-time multi-player tracker

and then resize or move the objects remaining from the precedent frame. If an object appeared or disappeared from the previous to the current frame, the user can freely create new objects or delete existing ones. Some screen shots of the tool are showed figures 5.6,5.7,5.8.

Object annotations are similarly stored in an XML file following the syntax shown in section 5.6.2.

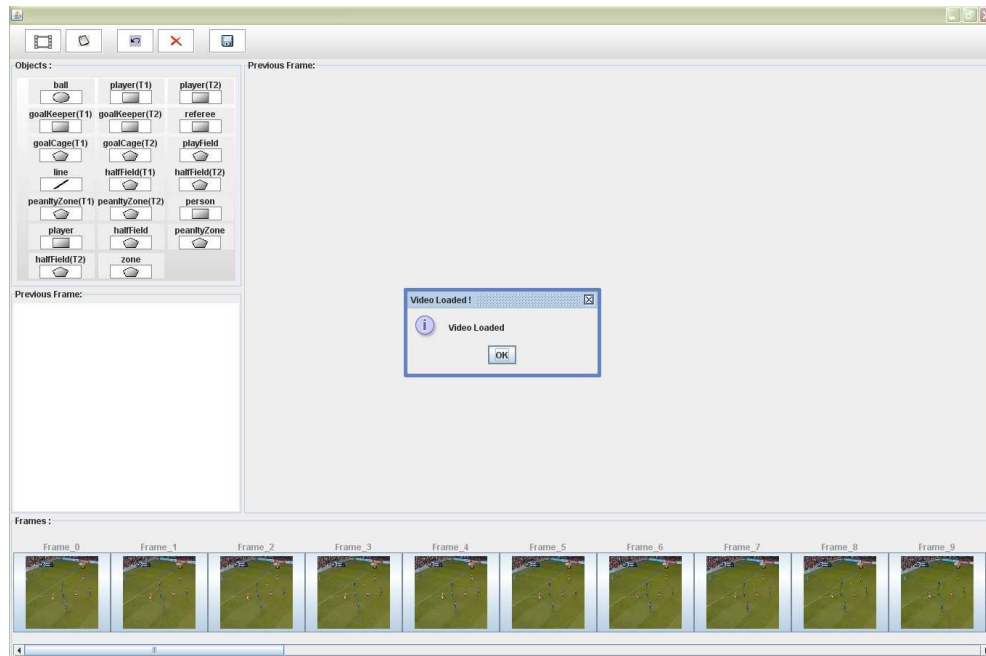


Figure 5.5: Loading the frames of the video to be annotated to the annotation tool.

5.4 Model Editor

Instead of giving the user the ability to only select the desired event to retrieve, this module of the framework offers the ability to describe exactly the event she or he wants to retrieve. Using some mid-level concepts provided by the framework, the user composes the structure of his event by a drawing finite state machine, and then she associates to each state the adapted conceptual graph which is drawn also in a separate panel. The constructed model is then stored in an event models base. Some screen shots of the tool are shown figures 5.9,5.10,5.11. The left panel displays the concepts of objects that can be used to compose the event model, the spatial relations that can be used to link objects within the event model, and the event models already declared and stored in the event model base. The user starts by drawing the automata on the upper area. Then she or he clicks on each state in order to create the conceptual graph corresponding to it in the lower area. The created event model is then stored in an XML file containing all the declared event models see section 5.6.

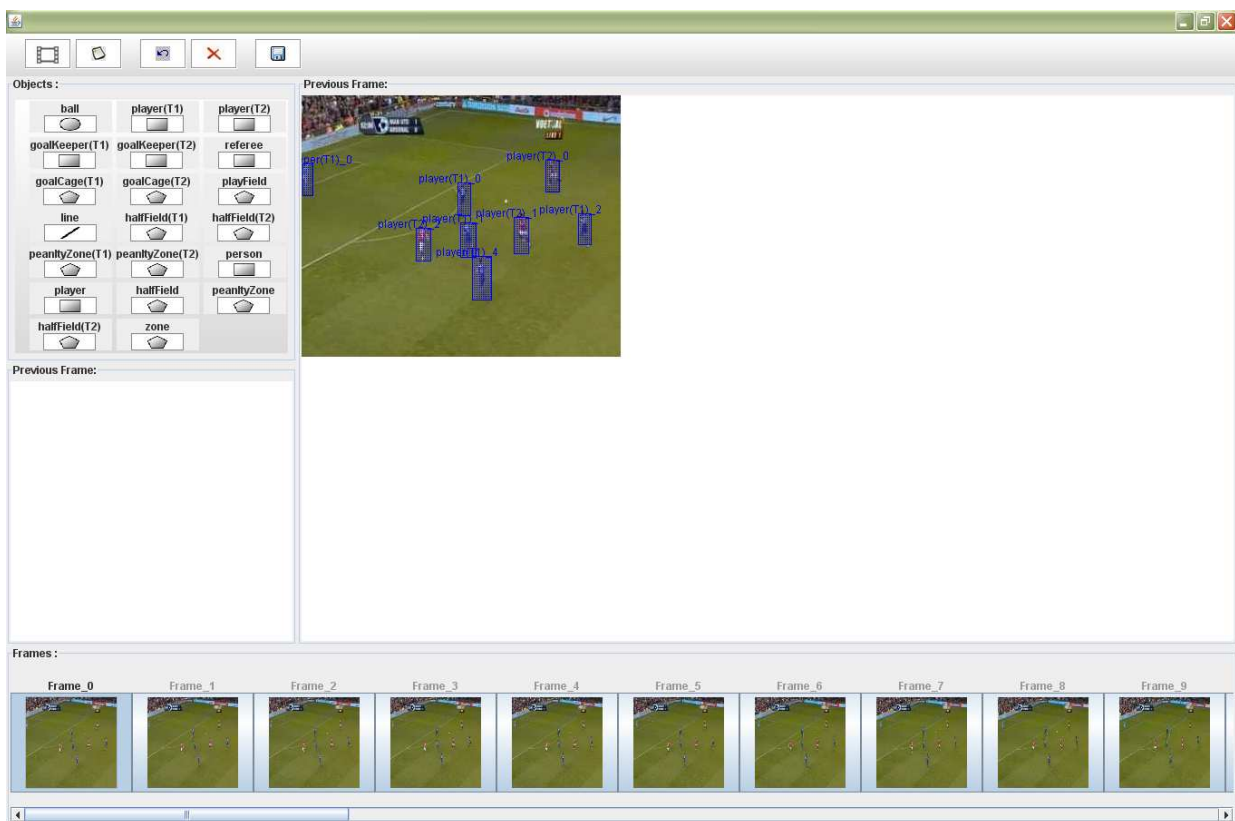


Figure 5.6: Annotating a single frame of the video.



Figure 5.7: Exporting descriptions of a frame to the next frame in order to facilitate its annotation.

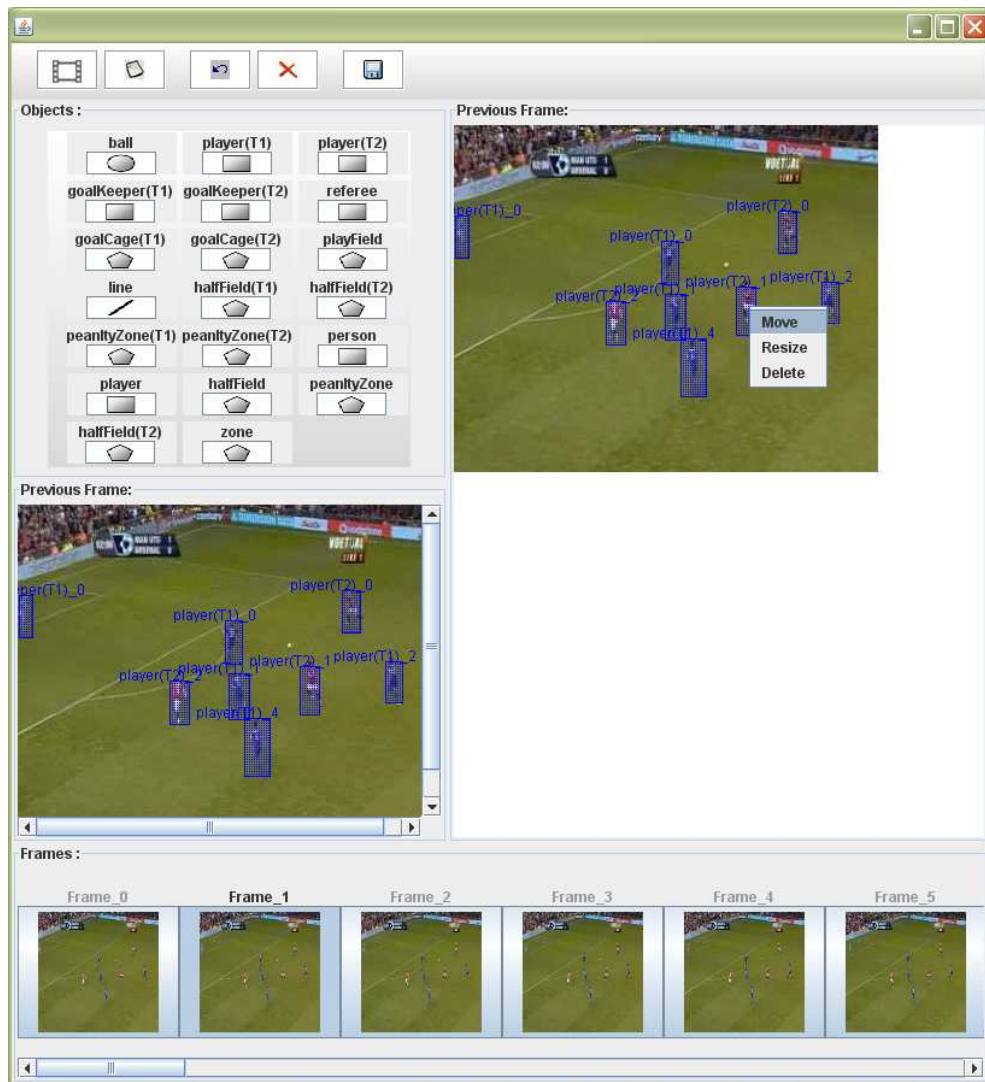


Figure 5.8: Modification of existing descriptions.

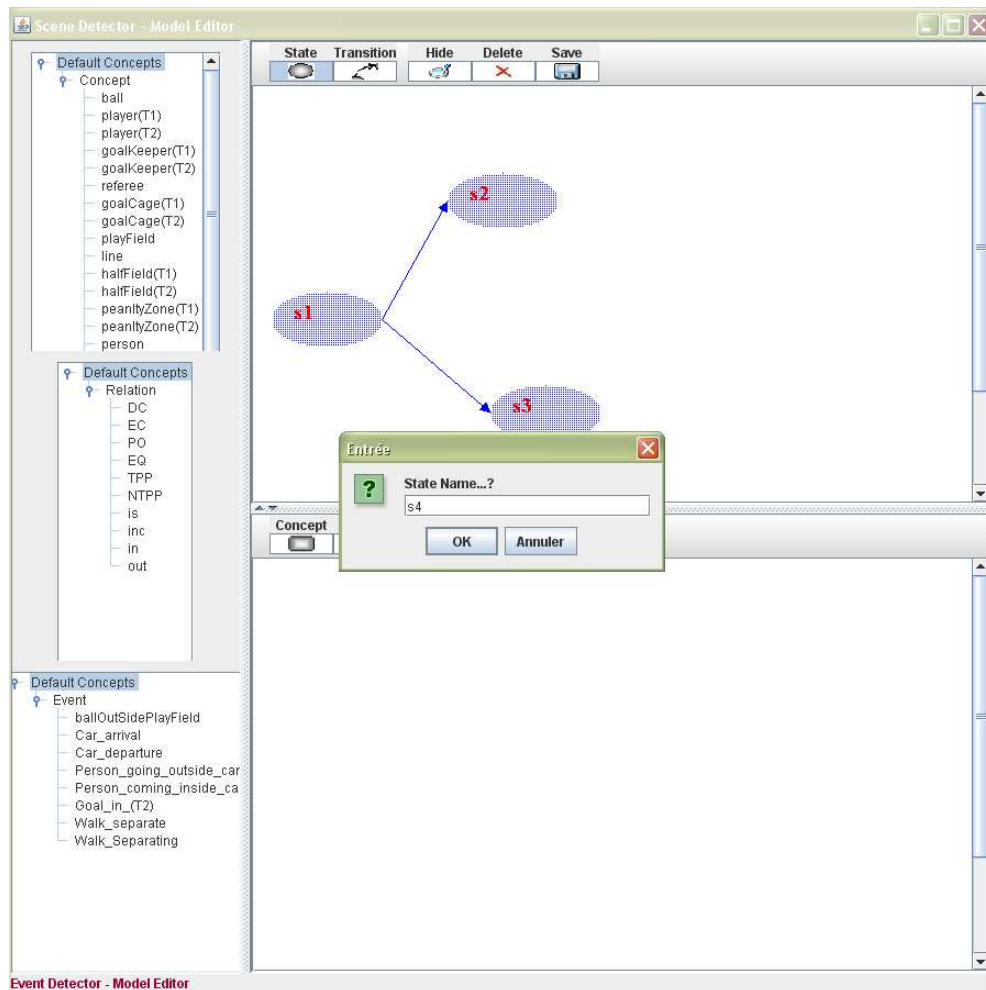


Figure 5.9: Building a finite state machine that represent the temporal composition of an event.

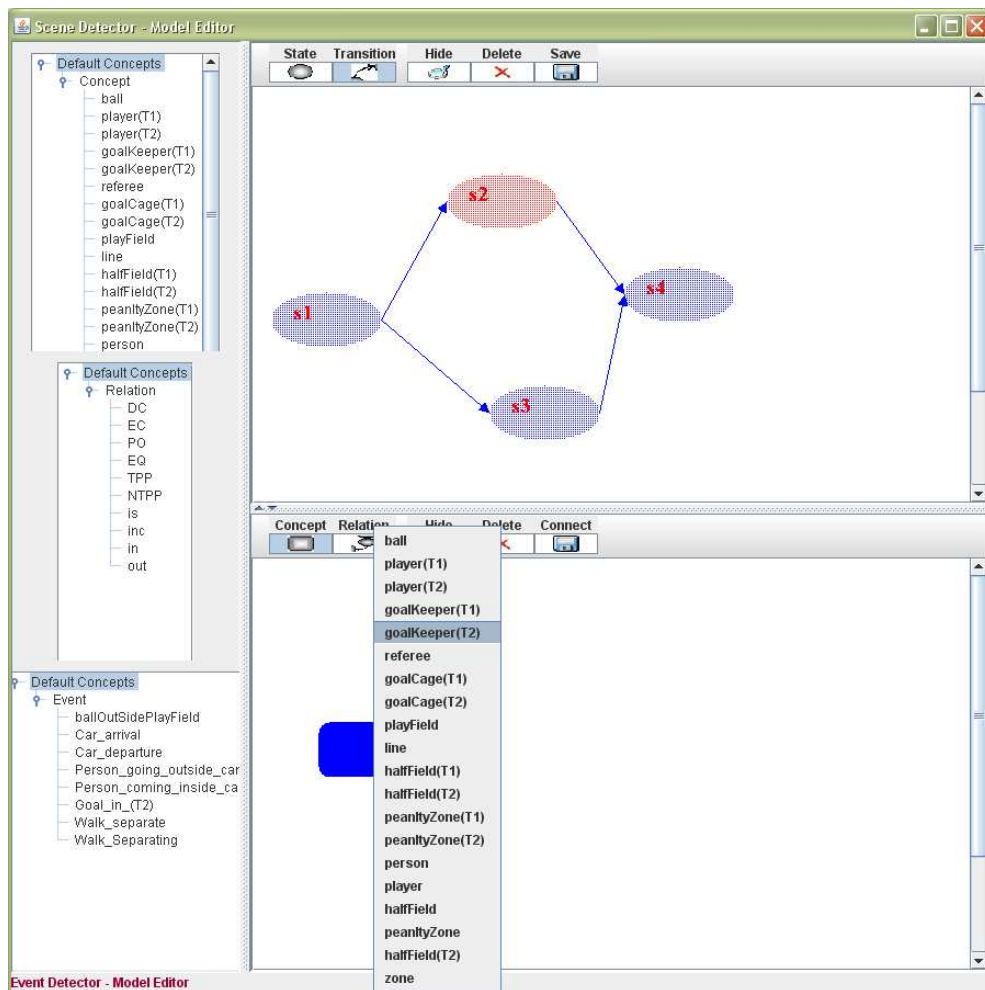


Figure 5.10: Constructing the conceptual graph associated to a selected state of the event model. Selecting the type of concepts from a list.

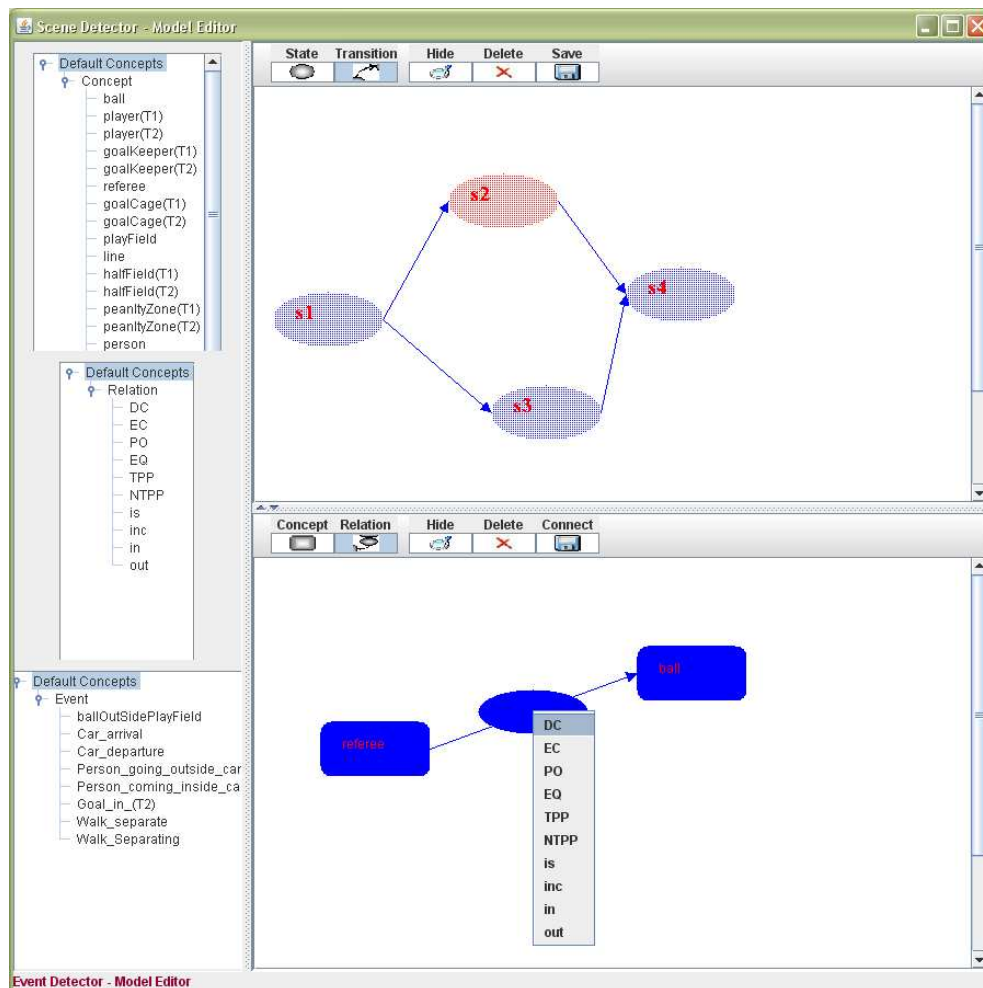


Figure 5.11: Semi automatic selection of relations between concepts.

5.5 Event Detector

This module is the heart of the framework. The current release of the prototype implements an event detector based on the event model and the algorithms presented in Chapter 2. The program can be launched on from the command line using the command:

```
EventDetector File_Path Event_Name [from_frame] [until_frame]
```

The program takes as input:

- *File_Path*: the video document path,
- *Event_Name*: the name of the event whose the model is stored the event (see section 5.6),
- *from_frame*: in option, the user can specify a frame from which the detection will begin,
- *until_frame*: similarly, in option, the user can specify a frame till which the detection will be performed.

If an event occurrence is found, this occurrence is added in the occurrence file (see section 5.6) following the syntax presented in section 5.6.2.

5.6 Representation Syntax

We use three different XML files that constitute the base of our prototype.

- A concept file in which the lattice of concepts (object types) used to annotate the video documents is described.
- An event file in which the models of events (Chapter 2) are described. Each event model is described using an automata and several conceptual graphs associated to the automata states.
- An occurrence file in which the occurrence of objects and events are stored.

5.6.1 Concept syntax

Concepts are organized in a lattice where element are organized in a hierarchy using the relation "is-a". The top element is considered to be of type "thing". The syntax of the file is the following :

```
<Concepts>
  <Element>
    <ID>ball</ID>
    <type>Basic</type>
    <shape>oval</shape>
    <parent>thing</parent>
  </Element>
```

```
...
</Concepts>
```

5.6.2 Occurrence syntax

Occurrences are the information produced by manual annotation or automatic extraction of objects and events occurring in video documents. The following example shows an XML description occurrence of a player of the first soccer team (T1). The player appears at frame "1" of the video document "foot.mpg" in the screen area whose MBR (minimum boundary rectangle) coordinates are $x_0 = 276, y_0 = 54, dx = 19, dy = 36$.

```
<Occurrences>
  <Element>
    <ID>player(T1)_0-Frame_1-File_foot.mpg</ID>
    <media>
      <type start="1" end="1">Frame</type>
      <location file="_foot.mpg">Video</location>
    </media>
    <content>
      <type referent="player(T1)_0" value="player(T1)"
        certainty="0.8">concept</type>
      <form type="concept" coords="276 54 19 36">rectangle</form>
    </content>
  </Element>
  ...
</Occurrences>
```

The following example shows an XML description occurrence of an event occurrence. An occurrence of the event "ballOutSidePlayField" is detected in file "foot.mpg" from frame 15 to frame 19.

```
<Occurrences>
  <Element>
    <ID>ballGoesOutSidePlayField0-Frame_15_19-File_foot.mpg</ID>
    <media>
      <type start="15" end="19">Frame</type>
      <location file="_foot.mpg">Video</location>
    </media>
    <content>
      <type referent="ballGoesOutSidePlayField0"
        value="ballGoesOutSidePlayField" certainty="0.5">event</type>
      <form type="event"></form>
    </content>
  </Element>
```


...
</Occurrences>

5.6.3 Event model syntax

An simple example of the description of an event model is the following. The automata of the event "ballGoesOutsidePlayField" is composed of two states "ballInPlayField" and "ballOffPlayField". Each state is associated with an XML element *graph* that describes the concepts composing that graph and the relations linking those concepts. A transition "after" is declared from the state "ballInPlayField" to the state "ballOffPlayField".

```
<Events>
  <Element>
    <ID>ballGoesOutsidePlayField</ID>
    <description>detection of the ball when it goes outside
      the playfield</description>
    <inputs>
      <item></item>
    </inputs>
    <outputs>
      <item>timeInterval</item>
    </outputs>
  <Automata>
    <state>
      <ID>ballInPlayField</ID>
      <description>ball in playfield</description>
      <final>false</final>
      <initial>true</initial>
      <graph>
        <concept referent="b1">ball</concept>
        <concept referent="p1">playField</concept>
        <relation referent="r1">
          <object>b1</object>
          <subject>p1</subject>
          <predicate>in</predicate>
        </relation>
      </graph>
    </state>
    <state>
      <ID>ballOffPlayField</ID>
      <description>ball out playfield</description>
      <final>true</final>
      <initial>false</initial>
      <graph>
        <concept referent="b1">ball</concept>
        <concept referent="p1">playField</concept>
```

```

        <relation referent="r2">
            <object>b1</object>
            <subject>p1</subject>
            <predicate>out</predicate>
        </relation>
    </graph>
</state>
<transition>
    <message>after</message>
    <source>ballInPlayField</source>
    <target>ballOffPlayField</target>
</transition>
</Automata>
</Element>
...
<Events>

```

5.7 Conclusion

This chapter presents an overview of the framework developed for detecting objects and annotating videos, and retrieving events within video documents.

We have tested our framework with 10 soccer video clips issued from TV broadcasts and Web pages. The objective was to detect events such as "Penalty", "Goal", and "Ball going outside playfield". Each video clip was about 60 seconds length. Frame resolution varies from a video clip to another. It ranges from 352×288 to 720×576 measures in pixels. We have manually annotated objects contained in 5 video clips and we have left 5 video clips with no annotation so they will be proceeded automatically for detecting objects before detecting complex events.

For the first set of video clips (the annotated set), the detection of events is performed perfectly. However the time processing depends on the complexity of conceptual graphs contained in the finite state machine associated to the event. In fact when the graph describing a situation contains multiple objects to be detected (more than 10 objects), the graph matching can last few second for each key frame. However, when the graph is simple and contains less than 10 objects, the detection of events can be done in almost real time.

For the second set of video clips (the non annotated set), the time processing of event detection within each video clip depends primely on its resolution. In fact, this parameter influences directly the quality of object detections. This makes event detection closely dependent to the quality of the low level features extractions. A reliable event detection requires powerful objects detectors.

The implemented version of the framework is a preliminary release of the integrated software. It enables describing and detecting events following the event model defined in the chapter 2 that represents events using Finite State Machine combined with Conceptual Graphs. However, this version does not implement the event model presented in the chapter 3, nor the database management approach described in the chapter 4. Those works would be implemented in the next release of the framework.

6

Conclusion and Future Works

6.1 Summary and Contributions

In this thesis, we investigated several issues related to the field of semantic analysis and understanding of video documents, especially in relation with the higher abstraction level that is event-based querying. While major works focus on one specific issue, the objective of the thesis was to produce an integrated video model enabling for description, detection, and retrieval of events within video documents.

Our main contributions in this work were to propose, at a first stage, a semantic representation language enabling for spatiotemporal specification and modeling of events in video documents. Then, the produced event models are used as queries to detect events. The formalism presented is extended in order to enable for more expressive description of events by supporting fuzzy spatiotemporal reasoning and handling uncertainty in object classifications. Moreover, new similarity measures and matching algorithms are proposed to assess the degree of match between the video document and the event model of video. Finally, a deductive based approach is proposed to enable for advanced modeling and retrieval of semantic data within video databases. Its main innovation is the ability to locate events and objects with regards to several spatial and temporal references.

To summarize, we tried to provide preliminary answers to the following questions:

- How can we classify video content into different semantic abstraction levels?
 - We have chosen to classify video content into three abstraction levels that are, features level, object level and a high semantic level. The high semantic level is decomposed itself into three abstraction levels that are situation level, event level, and cognitive activity level (Chapter 0.1).
- What are the techniques used to extract and represent video data at the abstraction levels?
 - The chapter 1 cites multiple techniques used to analyze and explore video content following the different semantic. We have cited techniques such as WBIR, TBIR, CBIR,

learning techniques, logical-based representation techniques, annotation-based data models and object-relational data models.

- what is the benefit of using object-relational video data models?
 - As cited in section 1.4.2, in contrast to annotation based data-models that associate an annotation to a segment of the video, object-relational data-models provide users with more sophisticated capabilities. They enable to represent the real world objects appearing in a video, and the events, and even relationships between objects.
- how to model and detect an event in video documents?
 - We have proposed two event models that can be used to describe and to detect an event within video documents. The first is described in Chapter 2 and the second is described in chapter 3.
- how to handle uncertainty due to imprecise event description and uncertainty produced by error-prone object classification algorithms?
 - Chapter 3 discusses the uncertainty issue by doing a distinction between uncertainty due to imprecise event description (query) and uncertainty due to classification algorithm (detection). It proposes then new model for handling those two sources of uncertainty and some event detection algorithms that take uncertainty into account.
- how to define fuzzy spatial and temporal relations?
 - Section 3.3.2 proposes an extension of Allen relations and RCC8 relations for representing respectively temporal and spatial relations. Instead of verifying whether two intervals satisfy a fixed relation we propose a method for calculating a confidence degree of the satisfaction of that relation by the two intervals.
- why deductive approach can be convenient for semantic content analysis in video databases?
 - Deductive approach would enable, as demonstrated in Chapter 4, to store semantic information concerning objects and events using facts and then to use a declarative language to specify rules and infer new information. It enables for using more expressive queries such as recursive queries and rules that can not be expressed within basic relational databases.
- how to locate event according to multiple spatial or temporal environments?
 - Most former video data models enable to locate objects and event occurring in videos only with regards to their positioning in the video stream or to the time. In our object relational data-model proposed in Chapter 4 we provide the user with facilities to define new temporal and spatial frames of reference that would enables for locating video content simultaneously with regards to different spatiotemporal environments (see 4.3).
- how to infer new semantics within video databases?
 - In addition to using the basic data model syntax provided in Chapter 4, users are able to extend that syntax with additional predicate symbols they need in order to represent their domain-specific data. Rules can then be defined to combine data model and

user defined predicates to answer users queries and to infer new semantic information not stored in the video databases.

6.2 Future Work

The main technical perspective of this work is the production of an advanced version of the integrated framework. Few more technical components should be implemented to provide the complete software. Then, a formal evaluation of the tools and techniques used can be performed. A future research direction of this work, will be the extension of the architecture of the framework with components enabling the fusion of low level visual features with audio and textual data. It would be interesting to experiment how could fusion improve detection of events in video documents. Another future work is to enable manual annotation both at the level of low level features and at the level of events. A future version of the framework should also enable to answer user queries dealing with low level visual features. An architecture that represents the future framework is shown figure 6.1. The new components and connections to be added are highlighted in red color in this figure.

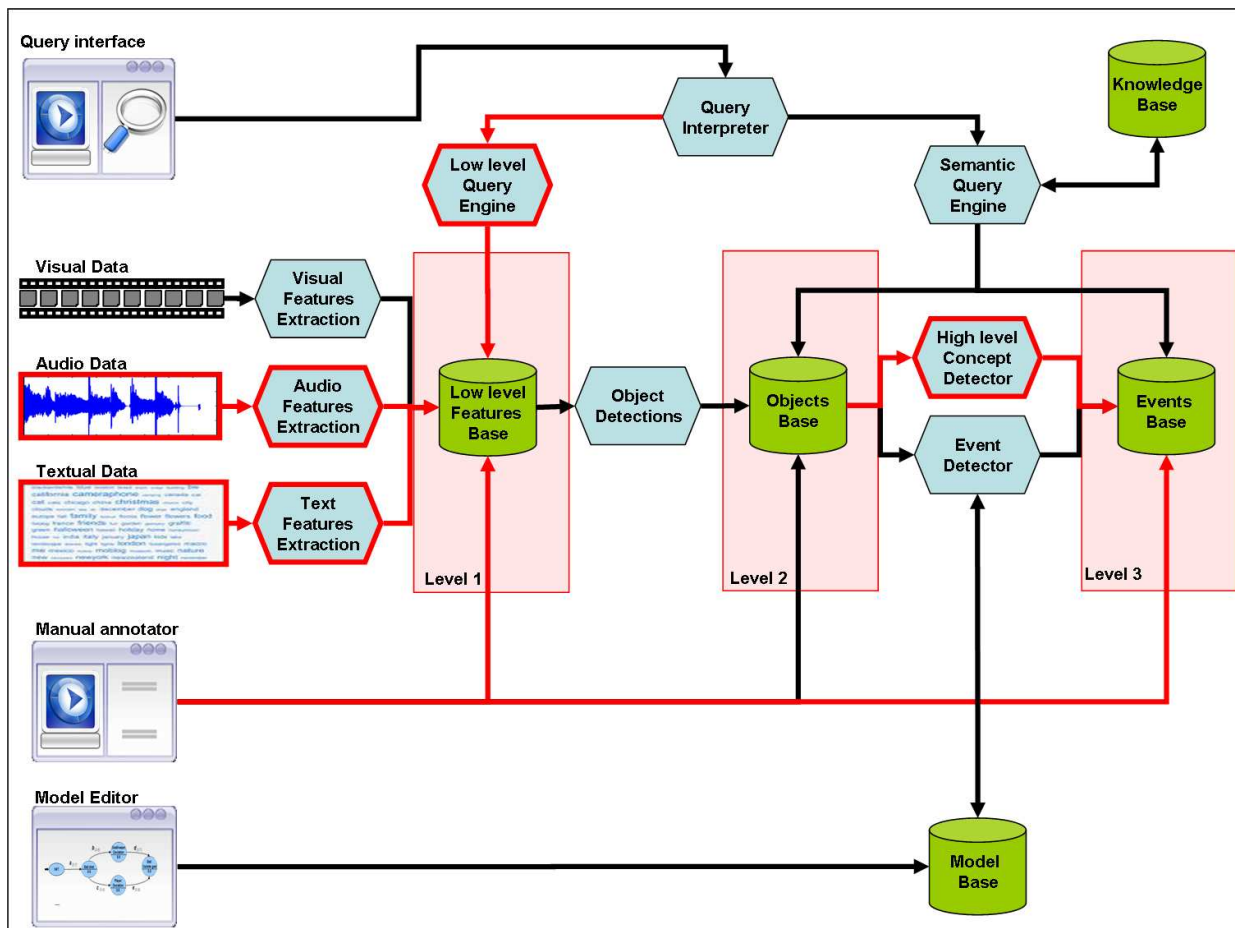


Figure 6.1: Future framework for cross-media semantic analysis and retrieval of video documents.

6.3 Academic and Industrial Results

6.3.1 Publications

All the contributions provided in the chapters of this thesis have been reviewed, approved and published in major conferences. A summary of those publication is presented in the next table.

6.3.2 Technical tools

First release of some technical tools were also product, especially :

- version beta of an automatic annotator tool of video documents.
- version beta of soccer player detectors in video documents.
- version beta of event model editor.
- version beta of event detector.

Publication Type	Thesis Chapter	Publication Reference
Conference	Chapter 4	A. Azough, A. Delteil, M. Hacid, F. De Marchi. A Database Approach for Expressive Modeling and Efficient Querying of Visual Information. The International Conference on MultiMedia Modeling Conference (MMM 2010), Chongqing, China. January 2010.
	Chapter 3	A. Azough, A. Delteil, M. Hacid, F. De Marchi. Fuzzy Conceptual Graphs for Handling Uncertainty in Semantic Video Retrieval. IEEE International Symposium on Multimedia (ISM 2009), San Diego, California, USA. December 2009.
	Chapter 2	A. Azough, A. Delteil, F. De Marchi, M. Hacid. Intuitive Event Modeling for Personalized Behavior Monitoring. IEEE International Conference on Pattern Recognition (ICPR 2008), Tampa, Florida. December 2008.
Book Chapter	Chapter 2	A. Azough, A. Delteil, F. De Marchi, M. Hacid. Semantic Language for Description and Detection of Visual Events. Advances in Semantic Media Adaptation and Personalization - Volume 2, 2009.
Workshops	Chapter 2	A. Azough, A. Delteil, F. De Marchi, M. Hacid. Description and Discovery of Complex Events in Video Surveillance. IEEE International Workshop on Semantic Media Adaptation and Personalization (SMAP 2008). Prague, Czech Republic. December 2008.
	Chapter 2	Azough, A. Delteil, F. De Marchi, M. Hacid. A Representation Language for Multimedia Web Semantic. IEEE International Workshop on Semantic Media Adaptation and Personalization (SMAP 2007). London, United Kingdom. December 2007.
	Not included in the thesis	M.Z. Maala, A. Delteil, A. Azough. A Conversion Process From Flickr Tags to RDF Descriptions. International Conference on Business Information Systems - Social Aspects of the Web (SAW 2007). Poznan, Poland. 2007.
	Chapter 2	A. Azough, A. Delteil, F. De Marchi, M. Hacid. A Representation Language for Semantic Description of Multimedia Contents. Summer School on Multimedia Semantics (SSMS 2007), Glasgow, UK. July 2007.

Bibliography

- [1] Mpeg-7: Overview ofmpeg-7 description tools, part 2. *IEEE MultiMedia*, 09(3):83–93, 2002. 24, 48
- [2] Google hot trends. Technical report, Google, <http://www.google.fr/trends/hottrends>, July 2009. 87
- [3] A. N. A. A. Aydin Alatan and W. Wolf. Temporal structure analysis of broadcast tennis video using hidden markov models. In *Incorporating audio cues into dialog and action scene detection. In Multimedia Tools and Applications, volume 14*, pages 137–151, 2001. 21
- [4] A. F. B. A. D. Wilson and J. Cassell. Temporal classification of natural gesture and application to video coding. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 948–954, 1997. 21
- [5] A. J. H. Z. A. Vailaya, M.A.T. Figueiredo. Image classification for content-based indexing. In *IEEE Transaction Image Processing 10 (1)*, pages 117–130, 2001. 18
- [6] S. Adali, S. K. Candan, S. S. Chen, K. Erol, and V. S. Subrahmanian. The advanced video information system: Data structures and query processing. *Multimedia Systems*, 4(4):172–186, 1996. 32, 33
- [7] T. G. Aguierre-Smith and G. Davenport. The stratification system: A design environment for random access video. *Proc. Third Int’l Workshop Network and Operating System Support for Digital Audio and Video*, Nov. 1992. 30
- [8] J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, Nov. 1983. 66, 107
- [9] R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994. 37
- [10] A. Z. Arifin and A. Asano. Image segmentation by histogram thresholding using hierarchical cluster analysis. *Pattern Recogn. Lett.*, 27(13):1515–1521, 2006. 15
- [11] R. Arndt, R. Troncy, S. Staab, L. Hardman, and M. Vacura. COMM: Designing a Well-Founded Multimedia Ontology for the Web. In *ISWC*, 2007. 24
- [12] J. Assfalg, M. Bertini, C. Colombo, A. D. Bimbo, and W. Nunziati. Semantic annotation of soccer videos: automatic highlights identification. *Comput. Vis. Image Underst.*, 92(2-3):285–305, 2003. 21
- [13] Y. W. A.Y. Ng, M.I. Jordan. On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems, vol. 14*, MIT Press, Cambridge, MA, 2002. 20

- [14] R. S. Aygün and A. Yazici. Modeling and management of fuzzy information in multimedia database applications. *Multimedia Tools Appl.*, 24(1):29–56, 2004. 32
- [15] S. Badaloni and M. Giacomini. A fuzzy extension of allen’s interval algebra. In *AI*IA ’99*, London, UK, 2000. Springer-Verlag. 69
- [16] J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994. 17
- [17] M. Bertini, A. D. Bimbo, and W. Nunziati. Model checking for detection of sport highlights. In *MIR ’03: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 215–222, New York, NY, USA, 2003. ACM. 22
- [18] M. Bertini, A. D. Bimbo, and C. Torniai. Automatic annotation and semantic retrieval of video sequences using multimedia ontologies. In *MULTIMEDIA ’06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 679–682, New York, NY, USA, 2006. ACM Press. 27, 28
- [19] M. Bertini, R. Cucchiara, A. D. Bimbo, and C. Torniai. Video annotation with pictorially enriched ontologies. In *ICME*, pages 1428–1431. IEEE, 2005. 28
- [20] M. Bertini, A. Del Bimbo, and C. Torniai. Automatic annotation and semantic retrieval of video sequences using multimedia ontologies. In *MULTIMEDIA ’06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 679–682, New York, NY, USA, 2006. ACM. 23
- [21] J. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, 1998. 16
- [22] A. F. Bobick and A. D. Wilson. A state-based technique to the representation and recognition of gesture. In *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pages 1325–1337, Dec. 1997. 21
- [23] A. Bonzanini, R. Leonardi, and P. Migliorati. Event recognition in sport programs using low-level motion indices. *icme*, 00:255, 2001. 21
- [24] M. Brand. Understanding manipulation in video. In *Proc. Int. Conf. Automatic Face and Gesture Recognition*, pages 94–99, 1996. 21
- [25] F. Bremond and G. Medioni. Scenario recognition in airborne video imagery. In *Proc. Int. Workshop Interpretation of Visual Motion*, pages 57–64, 1998. 21
- [26] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: image segmentation using expectation-maximization and its application to image querying. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(8):1026–1038, 2002. 16
- [27] S. Chang and S. Liu. Picture indexing and abstraction techniques for pictorial databases. 6(4):475–484, July 1984. 13
- [28] M. Charhad and G. Quenot. Semantic video content indexing and retrieval using conceptual graphs. In *ICTTA*, Damascus, Syria, apr 2004. 28
- [29] C. W. Chen, J. Luo, and K. J. Parker. Image segmentation via adaptive k-mean clustering and knowledge-based morphological operations with biomedical applications. *IEEE Transactions on Image Processing*, 7(12):1673–1683, 1998. 15

- [30] C.-Y. Chen, J.-C. Wang, J.-F. Wang, and Y.-H. Hu. Event-based segmentation of sports video using motion entropy. In *ISM '07: Proceedings of the Ninth IEEE International Symposium on Multimedia*, pages 107–111, Washington, DC, USA, 2007. IEEE Computer Society. 22
- [31] T.-S. Chua and L.-Q. Ruan. A video retrieval and sequencing system. *ACM Trans. Inf. Syst.*, 13(4), 1995. 30
- [32] D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 750, Washington, DC, USA, 1997. IEEE Computer Society. 16
- [33] D. S. C.P. Town. Content-based image retrieval using semantic visual categories. *Society for Manufacturing Engineers, Technical Report MV01-211*, 2001. 18, 19
- [34] R. R. D. Srivastava and P. Revesz. Constraint objects. In *Proc. Second Int'l Workshop Principles and Practice of Constraint Programming (PPCP '94)*, pages 218–228, 1994. 90
- [35] I. S. D. Stan. Mapping low-level image features to semantic concepts. In *Proceedings of the SPIE: Storage and Retrieval for Media Databases*, pages 172–179, 2001. 19
- [36] S. Dasiopoulou, V. Mezaris, I. Kompatsiaris, V. K. Papastathis, and M. G. Strintzis. Knowledge-assisted semantic video object detection. In *IEEE Transactions on Circuits and Systems for Video Technology*, pages 15(10):1210–1224, Oct. 2005. 28
- [37] M. Davis. Media streams: an iconic visual language for video representation. pages 854–866, 1995. 23
- [38] Y. Deng and B. S. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(8):800–810, 2001. 16
- [39] M. Doerr. The cidoc conceptual reference module: an ontological approach to semantic interoperability of metadata. *AI Mag.*, 24(3):75–92, 2003. 29
- [40] M. E. Dönderler, E. Şaykol, U. Arslan, O. Ulusoy, and U. Güdükbay. Bilvideo: Design and implementation of a video database management system. *Multimedia Tools Appl.*, 27(1):79–104, 2005. 32, 110
- [41] T. D'Orazio, N. Ancona, G. Cicirelli, and M. Nitti. A ball detection algorithm for real soccer image sequences. In *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 1*, page 10210, Washington, DC, USA, 2002. IEEE Computer Society. 22
- [42] D. Dubois, H. Fargier, and H. Prade. Possibility theory in constraint satisfaction problems: Handling priority, preference and uncertainty. *Applied Intelligence*, 6:287–309, 1996. 69
- [43] D. Dubois, H. Fargier, and H. Prade. Possibility theory in constraint satisfaction problems: Handling priority, preference and uncertainty. *Applied Intelligence*, 6:287–309, 1996. 69
- [44] D. J. Duke, I. Herman, T. Rist, and M. Wilson. Relating the primitive hierarchy of the premo standard to the standard reference model for intelligent multimedia presentation systems. *Comput. Stand. Interfaces*, 18(6-7):525–535, 1997. 23
- [45] S. T. E. Chang. Svmactive -support vector machine active learning for image retrieval. In *Proceedings of the ACM International Multimedia Conference*, pages 107–118, October 2001. 18

- [46] L. O. E. Kijak and P. Gros. Temporal structure analysis of broadcast tennis video using hidden markov models. In *Electronic Imaging: Science and Technology: Storage and Retrieval for Media Databases, volume 5021*, pages 277–288, 2003. 21
- [47] A. Ekin, A. M. Tekalp, and R. Mehrotra. Automatic soccer video analysis and summarization. *IEEE Transactions on Image Processing*, 12(7):796–807, 2003. 22, 32
- [48] N. Fatemi and P. Mulhem. A conceptual graph approach for video data representation and retrieval. *IDA'99, LNCS 1642*, pp. 525–536, page 525, 1999. 33, 34, 40
- [49] H. Feng and T.-S. Chua. A bootstrapping approach to annotating large image collection. In *MIR '03: Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 55–62, New York, NY, USA, 2003. ACM. 16
- [50] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: The qbic system. *Computer*, 28(9):23–32, 1995. 13, 31
- [51] J. Geurts, J. van Ossenbrugen, and L. Hardman. Requirements for practical multimedia annotation. In *Multimedia and the Semantic Web, 2nd European Semantic Web Conference*, 2005. 23, 33
- [52] Y. Gong, T. Lim, and H. Chua. Automatic parsing of tv soccer programs. In *IEEE International Conference on Multimedia Computing and Systems*, pages 167 – 174, May 1995. 21
- [53] A. Gupta and R. Jain. Visual information retrieval. *Commun. ACM*, 40(5):70–79, 1997. 13
- [54] T.-S. C. H. Feng. A bootstrapping approach to annotating large image collection. In *Workshop on Multimedia Information Retrieval in ACM Multimedia*, pages 55–62, November 2003. 18, 22
- [55] V. Haarslev and R. Moller. Description of the racer system and its applications. In *Proceedings International Workshop on Description Logics (DL-2001)*, pages 131–141, Stanford, USA, August 2001. 27
- [56] M.-S. Hacid, C. Decleir, and J. Kouloumdjian. A database approach for modeling and querying video data. *IEEE Trans. on Knowl. and Data Eng.*, 12(5):729–750, 2000. 32, 33
- [57] I. Haritaoglu, D. Harwood, and L. S. Davis. W⁴: real-time surveillance of people and their activities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):809–830, August 2000. 17
- [58] V. Harmandas, M. Sanderson, and M. D. Dunlop. Image retrieval by hypertext links. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 296–303, New York, NY, USA, 1997. ACM. 13
- [59] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *Systems, Man and Cybernetics, Part C, IEEE Transactions on*, 34(3):334–352, 2004. 17, 37
- [60] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *Systems, Man and Cybernetics, Part C, IEEE Transactions on*, 34(3):334–352, 2004. 17

- [61] K. A. Hua, K. Vu, and J.-H. Oh. Sammatch: a flexible and efficient sampling-based image retrieval technique for large image databases. In *MULTIMEDIA '99: Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 225–234, New York, NY, USA, 1999. ACM. 16
- [62] Y. Huang, J. Llach, and S. Bhagavathy. Players and ball detection in soccer videos based on color segmentation and shape analysis. In *MCAM07*, pages 416–425, 2007. 116
- [63] J. Hunter and F. Nack. An overview of the mpeg-7 description definition language (ddl) proposals. 16(1-2):271–293, September 2000. 52
- [64] I. C. I.K. Sethi. Mining association rules between low-level image features and high-level concepts. In *Proceedings of the SPIE Data Mining and Knowledge Discovery, vol. III*, pages 279–290, 2001. 3, 18, 19
- [65] N. Ikizler and D. A. Forsyth. Searching for complex human activities with no visual examples. *Revue International Journal of Computer Vision, Springer Netherlands*, 2008. 20
- [66] A. Isaac and R. Troncy. Using several ontologies for describing audiovisual documents : A case study in the medical domain, workshop on multimedia and the semantic web, second european semantic web conference (eswc 2005), heraklion, crete. 29
- [67] Y. A. Ivanov and A. F. Boblic. Recognition of visual activities and interactions by stochastic parsing. In *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pages 852–872, 2000. 21
- [68] A. D. B. W. N. J. Assfalg, M. Bertini and P. Pala. Soccer highlights detection and recognition using hmm's. In *Proc. IEEE ICME*, 2002. 21
- [69] M. G. J. Eakins. Content-based image retrieval. In *Technical Report, University of Northumbria at Newcastle*, 1999. 3
- [70] A. S. J. Luo. Indoor vs outdoor classification of consumer photographs using low-level and semantic features. In *International Conference on Image Processing (ICIP), vol II*, pages 745–748, October 2001. 18
- [71] J. M. J. Shi. Normalized cuts and image segmentation. In *IEEE Trans. Pattern Anal. Mach. Intell. (PAMI)* 22 (8), pages 888–905, 2000. 20
- [72] A. Jaimes and J. R. Smith. Semi-automatic, data-driven construction of multimedia ontologies. In *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo*, pages 781–784, Washington, DC, USA, 2003. IEEE Computer Society. 28
- [73] F. Jing, M. Li, L. Zhang, H. Zhang, and B. Zhang. Learning in region-based image retrieval. In E. M. Bakker, T. S. Huang, M. S. Lew, N. Sebe, and X. S. Zhou, editors, *CIVR*, volume 2728 of *Lecture Notes in Computer Science*, pages 206–215. Springer, 2003. 16
- [74] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. In *Image Vis. Comput.*, vol. 14, no. 8, pages 609–615, 1996. 21
- [75] M. J. Jones and J. M. Rehg. Statistical color models with application to skin detection. *Int. J. Comput. Vision*, 46(1):81–96, 2002. 116
- [76] C. R. Jung. Unsupervised multiscale segmentation of color images. *Pattern Recogn. Lett.*, 28:523–533, March 2007. 16

- [77] H. K. K. Takahashi, S. Seki and R. Oka. Recognition of dexterous manipulations from time varying images. In *Proc. IEEE Workshop Motion of Non-Rigid and Articulated Objects, Austin, TX*, pages 23–28, 1994. 21
- [78] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *J. ACM*, 42(4):741–843, 1995. 100
- [79] R. Klapsing, G. Neumann, and W. Conen. Semantics in web engineering: Applying the resource description framework. *IEEE MultiMedia*, 8(2):62–68, 2001. 23
- [80] F. Kokkoras, H. Jiang, I. Vlahavas, A. K. Elmagarmid, E. N. Houstis, and W. G. Aref. Smart videotext: a video data model based on conceptual graphs. *Multimedia Syst.*, 8(4):328–338, 2002. 31
- [81] M. Köprülü, N. K. Çiçekli, and A. Yazıcı. *Spatio-Temporal Querying in Video Databases*. 2002. 32
- [82] O. Kucuktunc, U. Gudukbay, and O. Ulusoy. A natural language-based interface for querying a video database. *IEEE MultiMedia*, 14(1):83–89, 2007. 32
- [83] B. Z. L. Zhang, F. Liu. Support vector machine learning for image retrieval. In *International Conference on Image Processing*, pages 7–10, October 2001. 18
- [84] C. Lagoze and J. Hunter. The abc ontology and model. *Journal of Digital Information*, 2001. 29
- [85] R. Leonardi and P. Migliorati. Semantic indexing of multimedia documents. *IEEE MultiMedia*, 9(2):44–51, 2002. 21
- [86] W. Leow and S. Lai. Scale and orientation-invariant texture matching for image retrieval, 2000. 17
- [87] M. Levene and G. Loizou. *A Guided Tour of Relational Databases and Beyond*. Springer-Verlag, London, UK, 1999. 87
- [88] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, 2006. 3
- [89] Q. Li and L. S. Huang. A dynamic data model for a video database management system. *ACM Comput. Surv.*, 27(4):602–606, 1995. 31
- [90] W.-H. Lin, R. Jin, and A. Hauptmann. Web image retrieval re-ranking with relevance model. In *WI '03: Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, page 242, Washington, DC, USA, 2003. IEEE Computer Society. 13
- [91] S. Linckels and C. Meinel. A simple application of description logics for a semantic search engine. In *IADIS AC*, pages 306–311, 2005. 29
- [92] A. J. Lipton, H. Fujiyoshi, and R. S. Patil. Moving target classification and tracking from real-time video. In *WACV '98: Proceedings of the 4th IEEE Workshop on Applications of Computer Vision (WACV'98)*, page 8, Washington, DC, USA, 1998. IEEE Computer Society. 17
- [93] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recogn.*, 40(1):262–282, 2007. 13

- [94] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recogn.*, 40(1):262–282, 2007. 16
- [95] C. Lutz, C. Areces, I. Horrocks, and U. Sattler. Keys, nominals, and concrete domains. In G. Gottlob and T. Walsh, editors, *IJCAI*, pages 349–354. Morgan Kaufmann, 2003. 29, 62
- [96] R. M. M. Bilenko, S. Basu. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, pages 81–88, July 2004. 19
- [97] W. Y. Ma and B. S. Manjunath. Netra: a toolbox for navigating large image databases. In *ICIP '97: Proceedings of the 1997 International Conference on Image Processing (ICIP '97) 3-Volume Set-Volume 1*, page 568, Washington, DC, USA, 1997. IEEE Computer Society. 13
- [98] M. Mancas, B. Gosselin, and B. Macq. B.: Segmentation using a region-growing thresholding. In *Proceedings of the SPIE 5672 (2005) 388398*, pages 12–13. 15
- [99] D. L. McGuinness and F. van Harmelen. OWL web ontology language overview. W3C recommendation, W3C, February 2004. 23
- [100] S. Mckenna. Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56, October 2000. 17
- [101] M. Mechkour. Emir2: An extended model for image representation and retrieval. In *DEXA '95: Proceedings of the 6th International Conference on Database and Expert Systems Applications*, pages 395–404, London, UK, 1995. Springer-Verlag. 28
- [102] B. T. Messmer and H. Bunke. A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1998. 78, 79
- [103] D. Meyer and J. Denzler. Model based extraction of articulated objects in image sequences for gait analysis. In *ICIP '97: Proceedings of the 1997 International Conference on Image Processing (ICIP '97) 3-Volume Set-Volume 3*, page 78, Washington, DC, USA, 1997. IEEE Computer Society. 17
- [104] B. Micusik and A. Hanbury. Steerable semi-automatic segmentation of textured images. In *IN: PROC. SCANDINAVIAN CONFERENCE ON IMAGE ANALYSIS (SCIA*, pages 35–44, 2005. 16
- [105] P. Mulhem, W. K. Leow, and Y. K. Lee. Fuzzy conceptual graphs for matching images of natural scenes. In B. Nebel, editor, *IJCAI*. Morgan Kaufmann, 2001. 62, 78
- [106] B. R. N. M. Oliver and A. P. Pentland. A bayesian computer vision system for modeling human interactions. In *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22,, pages 831–843, Aug. 2000. 21
- [107] W. C. Naidoo and J. R. Tapamo. Soccer video analysis by ball, player and referee tracking. In *SAICSIT '06: Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, pages 51–60, Republic of South Africa, 2006. South African Institute for Computer Scientists and Information Technologists. 118
- [108] E. Oomoto and K. Tanaka. Ovid: Design and implementation of a video-object database system. *IEEE Trans. on Knowl. and Data Eng.*, 5(4), 1993. 32

- [109] J. Owens and A. Hunter. Application of the self-organizing map to trajectory classification. In *Proc. IEEE Int. Workshop Visual Surveillance*, pages 77–83, 2000. 21
- [110] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases, 1994. 13
- [111] K. Petridis, D. Anastasopoulos, C. Saathoff, N. Timmermann, Y. Kompatsiaris, and S. Staab. M-ontomat-annotizer: Image annotation linking ontologies and multimedia low-level features. In *KES (3)*, pages 633–640, 2006. 31
- [112] T. N. Phyu. Survey of classification techniques in data mining. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I (IMECS 2009)*, Hong Kong, March 18 – 20, 2009. 18
- [113] P. R. Pietzuch, B. Shand, and J. Bacon. A framework for event composition in distributed systems. In *Middleware*, pages 62–82, 2003. 40
- [114] R. Rahmani, S. A. Goldman, H. Zhang, J. Krettek, and J. E. Fritts. Localized content based image retrieval. In *MIR '05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 227–236, New York, NY, USA, 2005. ACM. 13
- [115] Y. Raimond, S. Abdallah, M. Sandler, and F. Giasson. The music ontology. Vienna, Austria, September 2007. 29
- [116] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *Proceedings 3rd Int. Conf. KR*, 1992. 72
- [117] J. Renz and B. Nebel. On the complexity of qualitative spatial reasoning: a maximal tractable fragment of the region connection calculus. *Artif. Intell.*, 108(1-2):69–123, 1999. 42
- [118] K. Rodden, K. R. Wood, and K. R. Wood. How do people manage their digital photographs? In *SIGCHI*, NY, USA, 2003. 30, 59
- [119] H. T. S. Rui, Y. and S. Chang. Image retrieval: Current techniques, promising directions and open issues. In *Journal of Visual Communication and Image Representation*, Vol. 10, March, 1999. 13
- [120] Y. Rui, T. S. Huang, and S.-F. Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(1):39–62, March 1999. 13
- [121] J. C. SanMiguel, J. M. MartÁñez, and Álvaro García. An ontology for event detection and its application in surveillance video. *Advanced Video and Signal Based Surveillance, IEEE Conference on*, 0:220–225, 2009. 28
- [122] C.-R. S. S.D. MacArthur, C.E. Brodley. Relevance feedback decision trees in content-based image retrieval. In *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries (CBAIVL 00)*, pages 68–72, June 2000. 19
- [123] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, Inc., Orlando, FL, USA, 1983. 116
- [124] L. G. Shapiro, G. C. Stockman, L. G. Shapiro, and G. Stockman. *Computer Vision*. Prentice Hall, January 2001. 116

- [125] R. Shaw, R. Troncy, and L. Hardman. Lode: Linking open descriptions of events. In A. Gómez-Pérez, Y. Yu, and Y. Ding, editors, *ASWC*, volume 5926 of *Lecture Notes in Computer Science*, pages 153–167. Springer, 2009. 29
- [126] H. T. Shen, B. C. Ooi, and K.-L. Tan. Giving meanings to www images. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, pages 39–47, New York, NY, USA, 2000. ACM. 13
- [127] R. Shi, H. Feng, T. Chua, and C. Lee. An adaptive image content representation and segmentation approach to automatic image annotation. pages 545–554, 2004. 16
- [128] R. Shi, H. Feng, T.-S. Chua, and C.-H. Lee. An adaptive image content representation and segmentation approach to automatic image annotation. In *CIVR*, pages 545–554, 2004. 18
- [129] J. R. Smith and S.-F. Chang. Visualseek: a fully automated content-based image query system. In *MULTIMEDIA '96: Proceedings of the fourth ACM international conference on Multimedia*, pages 87–98, New York, NY, USA, 1996. ACM. 13
- [130] J. R. Smith and C. S. Li. Decoding image semantics using composite region templates. In *CBAIVL '98: Proceedings of the IEEE Workshop on Content - Based Access of Image and Video Libraries*, page 9, Washington, DC, USA, 1998. IEEE Computer Society. 17
- [131] J. R. Smith and B. Lugeon. A visual annotation tool for multimedia content description. In *Proceedings of SPIE Photonics East, Internet Multimedia Management Systems, Vol. 4210*, pages 49–59, 2000. 26
- [132] J. F. Sowa. *Conceptual structures: information processing in mind and machine*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1984. 37
- [133] J. F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley Longman Publishing Co., Inc., 1984. 40
- [134] J. F. Sowa. Top-level ontological categories. *Int. J. Hum.-Comput. Stud.*, 43(5-6):669–685, 1995. 27
- [135] J. F. Sowa. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Course Technology, August 1999. 59
- [136] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. volume 2, page 252 Vol. 2, 1999. 17
- [137] G. Stoilos, N. Simou, G. Stamou, and S. Kollias. Uncertainty and the semantic web. *IEEE Intelligent Systems*, 21(5):84–87, 2006. 62
- [138] U. Straccia. Reasoning within fuzzy description logics. *J. Artif. Intell. Res. (JAIR)*, 14:137–166, 2001. 29, 62
- [139] U. Straccia. A fuzzy description logic for the semantic web. *Capturing Intelligence: Fuzzy Logic and the Semantic Web. Elsevier (2005)*, 2005. 29, 62
- [140] B. Sumengen and B. S. Manjunath. Graph partitioning active contours (gpac) for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):509, 2006. 16
- [141] N. Sumpter and A. Bulpitt. Learning spatio-temporal patterns for predicting object behavior. In *Image Vis. Comput.*, vol. 18, no. 9, pages 697–704, 2000. 21

- [142] A. K. T. Berk, L. Brownston. A new color-naming system for graphics language. In *IEEE Comput. Graphics Appl.* 2 (3), pages 37–44, 1982. 18
- [143] J. F. T. Hastie, R. Tibshirani. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2001. 18, 19
- [144] J. W. T. Starner and A. Pentland. Real-time american sign language recognition using desk and wearable computer-based video. In *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pages 1371–1375, Dec. 1998. 21
- [145] Tong, H.-Q. Lu, and Q.-S. Liu. An effective and fast soccer ball detection and tracking method. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4*, pages 795–798, Washington, DC, USA, 2004. IEEE Computer Society. 22
- [146] V. Tovinkere and R. J. Qian. Detecting semantic events in soccer games: Towards a complete solution. *icme*, 00:212, 2001. 22
- [147] R. Troncy, W. Bailer, M. Hausenblas, and M. Höffernig. VAMP: Semantic Validation for MPEG-7 Profile Descriptions. Technical Report INS-E0705, CWI and JRS, April 2007. 52
- [148] R. Troncy, W. Bailer, M. Hausenblas, P. Hofmair, and R. Schlatter. Enabling multimedia metadata interoperability by defining formal semantics of mpeg-7 profiles. In Y. S. Avrithis, Y. Kompatsiaris, S. Staab, and N. E. O'Connor, editors, *SAMT*, volume 4306 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2006. 26
- [149] B. L. Tseng, C.-Y. Lin, and J. R. Smith. Using mpeg-7 and mpeg-21 for personalizing video. *IEEE MultiMedia*, 11(1):42–53, 2004. 53
- [150] R. Tusch, H. Kosch, and L. Böszörményi. Videx: an integrated generic video indexing approach. In *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*, pages 448–451, New York, NY, USA, 2000. ACM. 32
- [151] J. Y. U. Meier, R. Stiefelhausen and A. Waibel. Toward unrestricted lip reading. In *Int. J. Pattern Recognit. Artificial Intell.*, vol. 14, no. 5, pages 571–585, Aug 2000. 21
- [152] V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, September 1998. 18
- [153] N. Vasconcelos. The design of end-to-end optimal image retrieval systems. In *Proceedings of the International Conference on ANN*, Istanbul, Turkey, 2003. 20
- [154] N. Vasconcelos. On the efficient evaluation of probabilistic similarity functions for image retrieval. In *IEEE Trans. Inf. Theory* 50 (7), pages 1482–1496, 2004. 20
- [155] T.-S. C. W. Jin, R. Shi. A semi-naive bayesian method incorporating clustering with pairwise constraints for auto image annotation. In *Proceedings of the ACM Multimedia*, 2004. 18, 19
- [156] T. Wada and T. Matsuyama. Multi-object behavior recognition by event driven selective attention method. In *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pages 873–887, Aug.2000. 21
- [157] J. Wang, C. Xu, E. Chng, K. Wah, and Q. Tian. Automatic replay generation for soccer video broadcasting. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 32–39, New York, NY, USA, 2004. ACM. 22

- [158] J. Z. Wang, J. Li, and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(9):947–963, 2001. 13, 17
- [159] U. Westermann and R. Jain. Toward a common event model for multimedia applications. *IEEE MultiMedia*, 14(1):19–29, 2007. 33, 39
- [160] X. H. W.-Y. M. X. L. X. Zheng, D. Cai. Locality preserving clustering for image database. In *Proceedings of the 12th ACM Multimedia*, October 2004. 19
- [161] T. H. X.S. Zhou. Cbir: from low-level features to high-level semantics. In *Proceedings of SPIE Image and Video Communication and Processing*, pages 426–431, San Jose, CA, January 2000. 3
- [162] C. Xu, J. Wang, K. Wan, Y. Li, and L. Duan. Live sports event detection based on broadcast video and web-casting text. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 221–230, New York, NY, USA, 2006. ACM. 22
- [163] R. Y. Chen, J.Z.Wang. An unsupervised learning approach to content-based image retrieval. In *IEEE Proceedings of the International Symposium on Signal Processing and its Applications*, pages 197–200, July 2003. 3
- [164] M. Yang and N. Ahuja. Extraction and classification of visual motion pattern recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 892–897, 1998. 21
- [165] A. Yoshitaka and T. Ichikawa. A survey on content-based retrieval for multimedia databases. *IEEE Trans. on Knowl. and Data Eng.*, 11(1):81–93, 1999. 87
- [166] X. Yu, C. Xu, and Q. Tian. A ball tracking framework for broadcast soccer video. In *ICME '03: Proceedings of the 2003 International Conference on Multimedia and Expo*, pages 273–276, Washington, DC, USA, 2003. IEEE Computer Society. 22
- [167] L. Z. Yu Wang, Chunxiao Xing. Video semantic models : Survey and evaluation. pages 10–20, 2006. 31, 33
- [168] L. A. Zadeh. *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by L. A. Zadeh*. World Scientific Publishing, 1996. 59
- [169] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets Syst.*, 100:9–34, 1999. 69
- [170] C. Zhang, J. Y. Chai, and R. Jin. User term feedback in interactive text-based image retrieval. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 51–58, New York, NY, USA, 2005. ACM. 13
- [171] G. Zhu, Q. Huang, C. Xu, Y. Rui, S. Jiang, W. Gao, and H. Yao. Trajectory based event tactics analysis in broadcast sports video. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 58–67, New York, NY, USA, 2007. ACM. 22

Titre :

Modèle sémantique de la vidéo pour la description, la détection et la recherche des événements visuels

RESUME en français

Cette thèse consiste à explorer l'usage d'outils de support de la sémantique des données dans le domaine du multimédia. La première contribution concerne la génération de descriptions de haut-niveau. Nous proposons un langage de description de haut-niveau qui permet la définition d'événements et d'objets à partir de caractéristiques de bas-niveau. La deuxième contribution concerne l'exploration de certains types de raisonnement dans le contexte de la multimédia sémantique. Nous proposons un langage sémantique (fondé sur les graphes conceptuels flous) pour la description des vidéos et définissons des mécanismes de raisonnement sous-jacents. La troisième contribution porte sur l'indexation et la recherche sémantique dans les bases de données multimédia. Nous proposons un langage de requêtes issu des bases de données déductives pour l'expression de requêtes spatiotemporelles et sémantiques.

TITRE (en anglais):

Semantic Video Model for Description, Detection and Retrieval of Visual Events

RESUME en anglais

This thesis is about to explore the use of tools to support semantics of data in the field of multimedia. The first contribution concerns the generation of high-level descriptions. We propose a description language that allows high-level definition of events and objects from low-level features. The second contribution is the exploration of certain types of uncertainty reasoning in the context of multimedia semantics. We propose a semantic language (based on fuzzy conceptual graphs) for descriptions of videos and define mechanisms underlying reasoning. The third contribution relates to the semantic indexing and retrieval in multimedia databases. We propose a query language from deductive databases for the expression of spatiotemporal and semantic queries.

DISCIPLINE

Informatique

MOTS-CLES français :

Web sémantique, gap sémantique, multimédia sémantique, indexation vidéo, raisonnement, détection des événements, intelligence artificielle, représentation de connaissances, bases de données déductive, incertitude

KEYWORDS:

Semantic Web, semantic gap, Multimedia semantic, video indexing, video content retrieval, uncertainty reasoning, event detection, artificial intelligence, Knowledge representation, deductive databases.

INTITULE ET ADRESSE DU LABORATOIRE :

Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS) UMR 5205
Université Claude Bernard Lyon 1, Bâtiment Nautibus,
43, bd du 11 novembre 1918, 69622 Villeurbanne cedex