Integrated Semantic-Syntactic Video Modeling for Search and Browsing*

Ahmet Ekin, A. Murat Tekalp†, and Rajiv Mehrotra

A. Ekin (ekin@ece.rochester.edu) is with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627-0126.

A. M. Tekalp (tekalp@ece.rochester.edu, mtekalp@ku.edu.tr) is with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627-0126 and with the College of Engineering, Koc University, Istanbul, Turkey.

R. Mehrotra (rajiv.mehrotra@kodak.com) is with the Entertainment Imaging Division, Eastman Kodak Company, Rochester, NY 14650.

*This work is supported by grants from the NSF IIS program and Eastman Kodak Company.

† A. M. Tekalp is the corresponding author of this manuscript.

Abstract

Video processing and computer vision communities usually employ shot-based or object-based structural video models and associate low-level (color, texture, shape, and motion) and semantic descriptions (textual annotations) with these structural (syntactic) elements. Database and information retrieval communities, on the other hand, employ entity-relation (ER) or object-oriented models to model the semantics of multimedia documents. This paper proposes a new generic integrated semantic-syntactic video model to include all of these elements within a single framework to enable structured video search and browsing combining textual and low-level descriptors. The proposed model includes semantic entities (video objects and events) and the relations between them. We introduce a new "actor" entity to enable grouping of object roles in specific events. This context-dependent classification of attributes of an object allows for more efficient browsing and retrieval. The model also allows for decomposition of events into elementary motion units (EMU) and elementary reaction/interaction units (ERU) in order to access mid-level semantics and low-level video features. The instantiations of the model are expressed as graphs. Users can formulate flexible queries that can be translated into such graphs. Alternatively, users can input query graphs by editing an abstract model (model template). Search and retrieval is accomplished by matching the query graph with those instantiated models in the database. Examples and experimental results are provided to demonstrate the effectiveness of the proposed integrated modeling and querying framework.

Keywords

Integrated video model, video objects, events, object motion description, model-based query formation, query resolution by graph matching.

I. Introduction

Content-based video search finds applications in many areas, including video on demand, digital broadcast, and video surveillance. Most applications require the ability to search content at both semantic (e.g., objects, events, and relations) and low (e.g., color and motion) levels in order to satisfy user requirements. Studies in video retrieval systems have concentrated on representing video content in terms of either textual descriptors, such as keywords and structured annotations, based on entity-relationship (ER) or object-oriented modeling (in the database and information retrieval communities) or low-level image features such as color, texture, shape, and motion descriptors based on structural modeling (in the image processing and computer vision communities). Colombo et al. [1] refer to these visual information systems as the first and second generation systems, respectively. The former approach provides the ability to perform semantic level search; however, an-

notations must be entered manually and may become subjective. The latter approach enables automatic computation of features and use of objective similarity metrics; however, it is difficult to relate these features with semantics. There are very few works that aim to bridge the gap between the two approaches. These works can be classified as those which incorporate semantics into a primarily low-level feature based framework and those which incorporate low-level features into a database-modeling approach.

Video database models using keywords and structural annotations have been introduced for structured representation and semantic search of textual multimedia data. Oomoto et al. [2] suggest an object-based schemaless model for their video database system, OVID. A video object in OVID refers to a meaningful scene in terms of object identifier (oid), an interval, and a collection of attribute-value pairs. The authors propose an SQL-based querying language, VideoSQL, and define a generalization hierarchy of atomic values and objects to form general subjects and video objects. Hjelsvold and Midstraum [3] develop a generic video model that captures the video structure at various levels to enable specification of video structure, the annotations, and sharing and reuse in one enhanced-ER model. The thematic indexing is achieved by annotations defined for video segments and by specific annotation entities corresponding to persons, locations, and events. Adali et al. [4] introduce AVIS with a formal video model in terms of interesting video objects. A video object in AVIS refers to a semantic entity that attracts attention in a scene. The model includes events as the instantiation of activity types and the roles of objects in the events. AVIS is supported by an SQL-like language for semantic queries. Koh et al. [5] introduce a layered structure consisting of five levels to model video. They extend four-layered video hierarchy consisting of frame, shot, scene, and video levels, by decomposing the shot level into chunk and sequence levels. Al Safadi and Getta [6] propose a schemaless semantic video model to express various human interpretations. Their conceptual model constitutes semantic units, description of semantic units, association between semantic units, and abstraction mechanisms over semantic units. Smith and Benitez [7] provide a conceptual model with structural entities in terms of region, shot, frame, video, segment, and image, and semantic (video content) entities, such as object and event. Various relationships between the structural and semantic entities are defined in their model. The aforementioned

models, in general, are based on textual annotations and do not include low-level features, such as motion of video objects using trajectories and motion parameters.

The second-generation systems provide automatic tools to extract low-level features, such as color, texture, shape, and motion. In these systems, objective similarity measures are used to find the similar images/video to the query that is usually expressed by an example image/video. The users are also allowed to change the feature weights, paint, sketch, and give feedback. These systems include, but are not limited to, QBIC [8], Virage [9], VisualSEEk [10], VideoQ [11], VIOLONE [12], and MARS [13]. For a more complete review, the reader is referred to excellent survey papers, more recently [14], [15], and earlier ones [16]-[19], on different aspects of multimedia systems. There is a consensus among researchers on the need to incorporate semantic-level representation into these and similar systems [20]-[28]. Photobook [20] describes images in terms of a small set of perceptually significant coefficients. Appearance, shape, and texture are used as the object, "thing," or "stuff" descriptions. Smoliar and Zhang [21] seek an environment for interacting with visual objects. In their application, they use color, texture, and shape as basic image features and study the news video content parsing by using domain-based methods. The visual and textual indexing schemes are discussed; while the former is automatically obtained from the parsing process, the latter is manually entered. Colombo et al. [1] suggest a semantic level of representation by organizing low-level features into expressive and emotional levels. ViBE [22] is developed to enable semantic browsing of the video content by integrating shot-boundary detection, shot-tree representation, and pseudo-semantic labeling. Rui et al. [23] use relevance feedback to capture high-level subjectivity by dynamically updating feature weights. PicHunter [24] also employs relevance feedback to predict what the users search for in a Bayesian framework. The framework includes an entropy-minimizing display algorithm to maximize the information obtained from the user at each iteration. Similarly, Santini and Jain [25] propose to adapt the similarity metrics to support semantics. They give the returns of a famous image search engine to a sample query and show the lack of semantics in the returns, although any one of the returned image is related to the example image in terms of low-level aspects. Going from this observation, they propose a solution, staying within the domain of low-

level features, by adapting the similarity metrics according to the user feedback. In [26], multiple semantic visual templates are generated from a single query to capture the user-defined semantics. The proposed framework employs a Bayesian relevance feedback as in [24] to improve the retrieval performance. Naphade and Huang [27] use Bayesian belief networks to extract high-level semantic labels. They explicitly define the interactions of semantic concepts by the conditional probabilities between the network variables. In general, the novelty about the above techniques is their extension of the second-generation visual information systems with some level of semantics. However, these systems try to accomplish the incorporation of semantics by pure low-level features without using an explicit and generic semantic model.

In a recent study, Hacid et al. [28] adopt a database modeling approach and extended the model to include low-level features. Their model consists of two layers: 1) feature and content (audiovisual) layer that contains low-level features, and 2) semantic layer that provides conceptual information. The first layer is based on the information obtained from QBIC [8]. The queries are answered by a declarative, rule-based, constraint-query language that is based on the temporal cohesions. This work is a good starting point for a system to handle mixed-level queries, but their framework does not support object-based motion descriptions and mid-level semantics for motion. Since object motion is arguably the most important cue to characterize events in a video, we propose a new framework to address this need.

As a sign of a certain level of maturity reached in the field of content-based retrieval, the ISO MPEG-7 standard (formally Multimedia Content Description Interface) provides normative tools to describe multimedia content by defining a normative set of descriptors (D) and description schemes (DS). One of these DSs, the Semantic DS, introduces a generic semantic model to enable semantic retrieval in terms of objects, events, places, and semantic relations [29], [30]. MPEG-7 also provides low-level descriptors, such as color, texture, shape, and motion, under a separate Segment DS. Thus, in order to perform mixed-level queries with spatio-temporal relations between objects, e.g., "Object #1 is to the left of Object #2, and it participates in Event A," one needs to instantiate both a Semantic DS and a Segment DS with two separate graph structures. This is a drawback

of the MPEG-7 model that unduly increases representation complexity, resulting in inefficiency in query resolution and problems due to the independent manipulations of a single DS. Furthermore, there is no context-dependent classification of object attributes in the MPEG-7 Semantic DS. For example, if an object appears in multiple events, all attributes (relations) of the object related to all events are listed within the same object entity. This aggravates the inefficiency problem.

In this paper, we propose a generic integrated video model that combines low-level attributes with semantic-level entities in the most efficient and flexible (expressive) manner. The main contributions of our work are as follows:

1. We propose a generic, integrated model to describe video events *in terms of motions of video objects*. We decompose (temporally segment) events into elementary motion units (EMUs) and elementary reaction units (ERUs) in order to incorporate low-level object motion features and their spatio-temporal relationships into the event description.

2. We introduce *the actor entity* to store event-specific roles of an object. This context-dependent grouping of object roles not only provides a better design philosophy, but also significantly improves search efficiency.

3. We propose a graph-based query formation and matching framework based on the proposed model, which provides broad flexibility in terms of expressive power of the queries. The mixed-level queries, that is, the semantic queries with low-level features, require matching both the graph structure and node attributes corresponding to low-level features.

The rest of the paper is organized as follows: We describe the proposed generic video-event model in the next section. Section III provides examples for the instantiation of the model in a soccer video. In Section IV, we present graph-based query formation and resolution framework. Experimental results are given in Section V, and Section VI provides conclusions.

II. Integrated Semantic-Syntactic Model

In this section, we present a generic, integrated semantic-syntactic model that allows efficient description of video events and the motion of objects participating in these events. The model is *generic* in the sense that it is domain independent, and it is integrated in

the sense that it employs both high- and low-level features. Our model is an extension of ER models [31] with object-oriented concepts. The goal of the model is to describe video events. Thus, the main entities in the model are events, objects that participate in these events, and "actor" entities that describe object roles in the events. For example, consider a player object with name, age, and all other event-independent attributes. The same player assumes different roles throughout a video, e.g., it becomes "scorer" in one event, "assist-maker" in another, and so on. These context-specific object roles form separate actor entities, which all refer to the same player object. Low-level object motion and reactions are also context-specific roles; hence, they are described as attributes of "actor" entities and actor-actor relations, respectively. To describe low-level features, we define a "segment" descriptor that may hold multiple object motion units (EMUs) and reaction units (ERUs), to include low-level object motion information (e.g., trajectories) and interactions (e.g., spatio-temporal relations), respectively. In addition to segment-level relations, we also define semantic relations between the entities in the model. In the following, we formally define model entities and relationships.

• **Video Event:** Video events are composed of semantically meaningful object actions, such as walking and standing up, and interactions among objects, such as passing and tackling. In order to describe complex events, an event may be considered to be the composition of several subevents that can be classified as actions and interactions. Actions generally refer to semantically meaningful motion of a single object; whereas interactions take place among multiple objects. Events and subevents can be associated with semantic time and location.

Formally, a video event is described as $e = \{eventID,\ name,\ L,\ S_{location},\ S_{time}\}$ tuples where *eventID* is a unique id of type ID, *name* is the name of the event, L is one or more media locators of the event life span, $S_{location}$ is the semantic location, and S_{time} is the semantic time of the event.

• **Video Object:** A video object refers to a semantically meaningful spatio-temporal entity. Objects have attributes that are either event-independent, e.g., name, or event-dependent, such as the semantic role of an object in an event. Only event-independent attributes are used to describe an object entity. The event-dependent roles of an object

are stored in actor entities (defined later). In our model, we allow generalizations and specializations of video objects by the class hierarchy of objects. Formally, a video object can be defined as $o = \{oid, V, L\}$ where:

– oid is a unique object identifier of type ID.

– $V = \{A_1 : v_1, A_2 : v_2, ..., A_N : v_N\}$ refers to N event-independent, attribute-value pairs.

– L is one or more media locators of the object.

• **Actor:** Video objects play roles in events; hence, they are the actors within the events. As such, they assume event-specific semantic and low-level attributes that are stored in an actor entity. That is, the actor entity enables grouping of object roles in the context of a given event. At the semantic level, a video object carries a linguistic role and a semantic role. Both usually vary from one event to another. We adopt the linguistic roles that are classified by Semantic DS of MPEG-7 [29] as: *agentOf, patientOf, experiencerOf, stimulusOf, causerOf, sourceOf, destinationOf, beneficiaryOf, themeOf, objectResultOf, instrumentOf, locationOf, pathOf,* and *accompanierOf*. Semantic roles also vary with context, such as a single person may assume a *driver* role in a traffic-monitoring video and a *captain* role in a soccer game. At the low-level, we describe object motion by elementary motion units (EMUs) as segment-level actor attributes.

Formally, an actor entity is described as follows: $a = \{id,\ Roles\{linguistic,semantic\},\ E\}$ where *id* is a unique identifier of type ID, *Roles* contain linguistic and semantic roles. Finally, E is the list of EMUs (defined below) of the video object in the event.

• **Video Segment:** We define temporal video segments corresponding to actions and interactions as action units and interaction units, respectively. In general, motion of objects within an action unit and their interactions within an interaction unit may be too complex to describe by a single descriptor at the low level. Thus, we further subdivide action units into elementary motion units (EMU) and interaction units into both elementary motion units (EMU) and elementary reaction units (ERU), in order to integrate low-level descriptors into the event description. They are defined in detail in the following:

– **Elementary Motion Units (EMUs):** The life span of video objects can be segmented into temporal units, within which their motion is coherent and can be described by a single descriptor. Each EMU is represented with a single motion descriptor, which

can be a Trajectory descriptor or a ParametricMotion descriptor. Formally, an EMU is represented as $emu = \{L, MBR, M, T\}$ where:

$*$ $L = \{t[start : end] : v_i\}$ is the temporal interval of the emu life span and N $(i = 1 : N)$ media file locations corresponding to it.

$*$ MBR is the bounding box of the object in the representative frame and is described by the upper-left and lower-right corner points.

$*$ M is the motion parameters (the number of parameters uniquely specifies the selected motion descriptor).

$*$ T is the list of trajectory points of the object within the EMU and is represented as in MPEG-7 [32].

– **Elementary Reaction Units (ERUs):** ERUs are spatio-temporal units that correspond to object-object, low-level interactions. The interactions may be temporal reactions, spatial reactions, and/or motion reactions.

$*$ Temporal Reactions: We consider the "coexistence" of two objects within an interval and describe temporal object relations by Allen's interval algebra [33], which consists of 13 relations: *equal,* and *before, meets, overlaps, starts, contains, finishes,* and their inverses.

$*$ Spatial Reactions: Spatial reactions are divided into two classes: *Directional* and *Topological* relations. Directional relations include *north, south, west,* and *east* as strict directional relations, *northeast, northwest, southeast, southwest* as mixed-directional relations, and *above, below, top, left, right, in front of, behind, near,* and *far* as positional relations. The topological relations include *equal, inside, disjoint, touch, overlap,* and *cover.* Most of the above relations are due to Li et al. [34] and their revision of Egenhofer's work [35]. The matching of spatial relations is considered in [36].

$*$ Motion Reactions: General motion reactions include *approach, diverge,* and *stationary.* Each relation can be extended by the application-specific attributes, such as velocity and acceleration.

Formally, an ERU is defined as $eru = \{type, V, L\}$, and it is stored as an attribute for the actor-actor relationship. V is the attribute-value pairs depending on the ERU type, and L is defined similar to EMUs.

• **Relations:** We define relations between various entities:

– **Event-Event Relations:** An event may be composed of other events, called subevents. Causality relationship may also exist between events; for instance, an object action may cause another action or interaction. Furthermore, the users may also prefer to search the video events from a temporal aspect by using temporal relations of the events. Therefore, we consider event-event relations in three aspects: *composedOf, causal,* and *temporal*. *ComposedOf* relation type assumes a single value *composedOf,* while the *causality* relation may be one of *resultingIn* and *resultingFrom* values. The *temporal* relations follow Allen's temporal algebra [33]. The event-event relations are described by a number of directional relation links, defined as $\{type, name, sourceID, destinationID\}$ tuples where:

∗ *type* refers to the type of the relation, such as causal.

∗ *name* refers to the name of the relation, such as resultingIn and resultingFrom.

∗ *sourceID* is the reference to the source entity; in this case, it is a reference to an event.

∗ *destinationID* is the reference to the destination entity; in this case, it is a reference to an event.

– **Object-Object Relations:** Similar to events, an object may be *composedOf* other objects. *ComposedOf* relationships for objects assumes one of the *partOf, componentOf, memberOf,* and *substanceOf* values. Another relationship of objects is meta-relations defined as those relations that are not visually observable from video content. The formal representation of all object-object relations is easily obtained from the event-event relation links by recalling that the references *sourceID* and *destinationID* refer to objects instead of events.

– **Actor-Actor Relations:** A single actor entity contains only one object; therefore, actor-actor relations are defined to keep semantic and low-level, object-object relations in the event life span. Therefore, actor-actor relations are defined as $aa = \{act1id, act2id, R, E\}$ where *act1id* and *act2id* are the actor references, R is the semantic level object-object relation, and E is the list of segment-level object reactions, ERUs.

• **Semantic Time and Location:** Semantic time and location refer to the world time and location information, respectively. Semantic time may be specified by its widely known name, such as "Gulf War," or by its calendar attributes. Therefore, we keep time information as $\{semantic\ name, time\ interval\}$ pairs by associating a semantic name to a time

interval. Similar to semantic time, location is described as $\{semantic\ name, address\}$.

• **Media Locator:** Video objects, events, EMUs, and ERUs contain a set of media locators to keep their media life spans. Each distinct temporal interval may be represented by one or more media files. The media files may be video clips of the same event recorded by different camera settings, still images as keyframes or they may be in other formats, such as document and audio. A media locator contains $medloc = \{t[start : end], v_i\}$ corresponding to a temporal interval and N $(i = 1 : N)$ media files.

In Fig. 1, the graphical representation of the model is shown with the following notations: A rectangle refers to an entity, a diamond is a relationship with the relationship name written next to it, and an oval represents an attribute of an entity or a relationship. The model is instantiated by graphs due to the following reasons: 1) graph-based representation and manipulation tools have been widely employed, and several of them are also shown to be computationally complete, e.g., [37], 2) graph-based systems inherently enable the specification of queries by graphs, efficient menu structures, and tractable number of windows for the systems with graphical user interfaces [38]. In the remainder of the paper, we will also use graph concepts, *G = (V,E)*, vertices, *V,* and edges, *E,* to refer to model entities and relationships, respectively. The entities in Fig. 1, video objects, actors, and events, become graph vertices with the corresponding attributes, and object-object, event-event, event-actor, actor-object, and actor-actor relationships form graph edges.

III. Model Examples

In this section, we describe a video clip of a soccer goal by using model entities and relationships. The graphical notations that are used in the instantiations (and queries in Section V) extend the notations used in Fig. 1 in the following aspects: i) an event entity description is labeled as "*name:***EVENT,**" such as "*Free Kick:***EVENT,**" ii) an actor is shown as "*semantic role:***ACTOR,**" such as "*Kicker:***ACTOR,**" iii) a video object is specified either by "*class:***OBJECT,**" if the described object belongs to a specialized object class, e.g., player, or by "*name:***OBJECT,**" if the object has only standard attributes, such as name and media locators.

The *goal* event in the example clip is composed of three subevents: a *free kick,* a *header,* and a *score* event. As the first description example, we present the description of *free kick*

concept and the instantiation of its low-level descriptors. Three key frames of the event are shown in Fig. 2. In Fig. 3, conceptual description of the *free kick* subevent is shown where a *free kick* event *has* two actors with roles, *kicker* and *kicked object.* They *interact* during the event and form a set of segment-level relations as ERUs. Each actor carries event-specific linguistic roles and low-level object motion attributes as a set of EMUs. The kicker *is* a *player,* and the kicked object *is* a *ball.* The event-independent attributes of objects, such as *name* and *position* of the player, are stored in video object vertices. Fig. 3 is a conceptual description of the *free kick* event meaning that it does not refer to a specific media segment; therefore, many attributes are not instantiated (shown as "...") except those that stay the same in every *free kick* event, such as *kicker* is always *agentOf* the event.

In Fig. 3, low-level, spatio-temporal attributes of the player and the ball are represented as EMU and ERU attributes of actor vertices and actor-actor edges, respectively. The detailed description of the object motion and object reaction segments for the example scene in Fig. 2 is presented in Fig. 4. For simplicity, we assume that the *free kick* subevent starts at frame #0 of the corresponding video. The player has a media life span from frame #0 to frame #25 where its motion is described by two EMUs, while the ball appears in the whole event life span, and its motion attributes create three EMUs. (The reader is referred to [39]-[42] for automatic low-level descriptor extraction algorithms.) Segment relations between the two objects are valid only in the interval of their coexistence. The temporal interval of the player-media life span *starts* the temporal interval of the ball, meaning that their life span intervals start together, but the player-media life span ends earlier. Motion reactions of the two objects are "approach" before the time point of "kick," and "diverge" after it. The detail of motion descriptions in the model can be adjusted to the requirements by adding attributes to the motion ERU. For instance, the stationary feature of the ball during the "approach" relationship is described by zero velocity. Topological and directional spatial object relations are also shown in Fig. 4. The bounding boxes of the player and the ball are *disjoint* starting at frame #0 (Fig. 2 (a)), they *touch* each other from frame #12 to frame #15 (Fig. 2 (b)), and they are *disjoint* after frame #16. Although two objects always have topological relationships, they may not have a directional relationship

for every time instant. That situation is illustrated for the time interval (frame #12, frame #15).

In our second example, we present the complete description of the example video clip that starts with the *free kick* subevent in the previous example. The key frames for the other subevents, i.e., *header* and *score,* in the clip are shown in Fig. 5. *Header* and *score*, defined as the entering of the ball to the goal, occur after the *free kick* subevent. Since all of the subevents have descriptions similar to Figs. 3 and 4, we do not explicitly describe *header* and *score* events. In Fig. 6, the temporal relationship between the above three subevents are described by using *before* relationship in Allen's interval algebra. Next, the *goal* event is composed of *free kick, header,* and *score.* Three players act in the composite *goal* event as the assist maker, the scorer, and the goalie.

IV. Query Formation and Resolution

The expressive power of the model enables the users to form various types of queries. Among these, a query may be related to only high-level entities, may involve both high-level entities and relations, or may require matching a low-level criteria in one part of the scene and high-level constraints in another. We employ a single graph-based query representation and retrieval framework for the above queries. The queries are represented by graph patterns that can be formed by editing an example description or the database scheme. In order to facilitate query formation by the user, we further define abstract models (model templates) as special graph patterns for certain events in a specific domain. The user can also form queries by editing an abstract event model from the model database. The relevant sections of the database can be retrieved by matching query graph patterns with the graphs of the descriptions in the database. The similarity of the query graph patterns to each of the matching subgraphs in the database is calculated by matching both high- and low-level attributes.

A. Graph-Based Query Formation

The formation of model graphs from model entities and relationships was explained at the end of Section II. To recall, graph vertices correspond to video objects, events, and actors, and directed edges with attributes stand for the relationships between the graph

vertices. In this section, we define the formation of queries by graph patterns, introduce abstract models (graphs) as specialized graph patterns, and explain how browsing is related to the graph pattern concept.

A graph pattern is defined as an instance of the database model with the exception of null or undefined values for some attributes [38]. For instance, Fig. 7 is a graph pattern over the database model specified in Fig. 1. The graph pattern in Fig. 7 may be used to search for the events where two players act as agents. Graph patterns can be obtained by editing the database scheme (or an example description) or by using abstract models, which is explained below. Editing the database scheme refers to modifying the model diagram in Fig. 1 by copying, identifying, and deleting some part of the scheme. Therefore, graph patterns conform to the database scheme, and syntactically incorrect queries cannot be formulated. In certain structured domains, the number of popular queries may be limited, and model templates, called abstract models, as special graph patterns, can be defined for specific events. An abstract model is the abstraction of a specific model instantiation from specific object, location, and time instances (Fig. 3 is the abstract model of a *free kick* event). An abstract model conforms to the database scheme, therefore, the queries based on abstract models are also syntactically correct. A similar concept of abstraction has been included in the MPEG-7 standard, and it is called "formal abstraction" [29].

Browsing is a special case of querying with graph patterns where the query is limited to one or more specific vertices [38]. In most cases, browsing is composed of obtaining the attributes of a specific vertex as shown in Fig. 8 (a). However, in some cases, browsing may be related to more than one vertex and may require matching graph edges to obtain the results in terms of other vertices. An example of this kind of browsing is shown in Fig. 8 (b) where the graph pattern may be used to browse the events; object A and object B act together.

B. Typical Queries in Video

In this section, we classify the typical queries in our model. The classification is based on the structure of the query and the retrieval process.

1. **Single Vertex Queries:** A single vertex query is a query for video objects or video events satisfying the given attribute-value constraints. A single vertex query is answered

by checking all vertices of the specified type about the satisfiability of the constraints. The results of the query can be displayed by key frames representing the media life span of the queried vertex. The queries, "find objects in time interval [a,b]," "find all collision events," and "find players who are younger than 25," are single vertex queries.

2. **Actor-Related Queries:** Actor is a vertex that keeps event-specific object roles. The queries involving Actor vertices are described below:

(a) High-level actor attributes as linguistic and semantic roles of objects may be the constraints for object-event relation queries. The queries "show the objects of event E" and "show the events where object A acts as agent and object B acts as patient" are examples of these types of queries.

(b) Queries related to low-level object features in a specific event also involve actor vertices, since each actor vertex contains low-level object motion attributes as a set of EMUs. The queries, "show all speeding violations where a car exceeds the limit by 5 mph" in traffic monitoring domain, and "display all tennis serves faster than 90 mph" for a tennis game, are examples of low-level queries in a specific event.

3. **Edge (Relation) Queries:** The queries in this class involve event-event, object-object, and actor-actor relations. In event-event and object-object edge queries, the name of the semantic relation is given, and satisfying vertices or vertex pairs are acquired. If any one of the vertices is explicitly specified, the query becomes an example of browsing as in Fig. 8. Actor-actor relations constitute the low-level and semantic-level object relations in an event. As a graph pattern, they can be specified by either keeping only actor entities and specifying the semantic and/or segment relations between them, such as "find the scenes where two objects are approaching to each other," or by adding object and event links to the actors, hence further limiting the context of the relation, e.g., "find all goal events where the scorer and the goalie are approaching each other within a distance of 5 ft."

C. Query Resolution by Graph Matching

Query resolution requires matching the query graph with the subgraphs of the descriptions in the database. The subgraph isomorphism has been shown to be an NP-complete problem [43]. In our application, we use the following model-based constraints to reduce

the search space in query resolution: 1) the type of vertices in the description is known to be an event, an object, or an actor with each type having distinguishing features from the others, and 2) the directed edges correspond to different types of relations with type-specific semantic-level and/or low-level attributes.

A query as a graph pattern consists of a set of constraints that may be related to vertices, edges, or both. In our implementation of graph matching, we start the search from the vertices and find a set of matching vertices for each query vertex in each distinct database description. Starting with the most constrained query vertex, assumed to be inversely proportional to the number of matching vertices, we check for the edge constraints. That is, we look for the combinations of the resulting set of vertex matches for the edge constraints. The steps of the recursive search for graph matching are as follows:

Graph Matching Algorithm:

1. For each query vertex, find the initial set of matching vertices using the query constraints.

2. Rank the query vertices from the one having the least number of matches to the most.

3. Check the edge constraints to find the combinations of the vertices that match the query graph. For this purpose, use a recursive search from the most constrained vertex to the least constrained one, and return whenever an edge constraint fails.

We use the vertex and edge attributes defined below to find the isomorphism or similarity between two graphs:

• **Vertex Matching:** When the graph element is a vertex, we have three choices: i) an event, ii) an object, or iii) an actor vertex. Therefore, the most distinguishing attribute for a vertex in finding a match is its type. Next, we evaluate the other specific attributes defined below for each vertex type:

1. **Event:** Event vertices are evaluated by the equivalence of *name* attribute. As explained in Section II, *name* is the text associated with the event, such as *goal* in soccer and *left turn* in traffic monitoring.

2. **Actor:** The match between two actor vertices is found by comparing two high-level attributes: *linguistic role* and *semantic role.* Low-level descriptors for object motion can also be used for the same purpose.

3. **Object:** Object vertices are compared by the equivalence of the *object class* and the specified *attribute-value* pairs.

• **Edge Matching:** The match between two edges is defined by the equivalence of semantic and segment relations.

We compute two different cost values to find and rank the matching graphs:

1. **Semantic Cost:** In our model, semantic cost of a match is determined as the cost due to the mismatches in event structure. That is, the graphs that differ in temporal structure of events will be penalized by the cost of the mismatch.

We calculate the semantic cost value to match each query event vertex as the sum of the cost of insertions and deletions of event vertices and temporal event edges:

$$C_{sem} = \sum_{i=1}^{N} \{C_{insertion}(V_i, E_i) + C_{deletion}(V_i, E_i)\} \tag{1}$$

2. **Syntactic Cost:** The syntactic cost is defined as the dissimilarity of two graphs due to their low-level features. As explained in Section II, in our model, low-level features are defined for actor vertices and actor-actor links. Low-level vertex features are either object motion parameters or object trajectories while actor-actor links contain low-level spatio-temporal object reactions. The cost function for object motion descriptor (ParametricMotion) is given in Eq. 2, where N is the number of parameters and w_i are the parameter weights. The metrics for the computation of the similarity of trajectories are well-defined and available in the literature [11], [44]. The normalized spatio-temporal similarity of two trajectories is computed by Eq. 3, where M is the length of the shorter trajectory. Binary decisions, as matching or unmatching, for spatio-temporal relations are given by the equality of the corresponding reactions in the specified time interval.

$$D(EMU_1, EMU_2) = \sum_{n=1}^{N} w_n \left| a_{1n} - a_{2n} \right| \tag{2}$$

$$D(T_q, T_{db}) = \frac{1}{M} \sum_{i=1}^{M} [(T_q.x_i - T_{db}.x_i)^2 + (T_q.y_i - T_{db}.y_i)^2 + (T_q.z_i - T_{db}.z_i)^2]^{1/2} \tag{3}$$

V. Results

To evaluate the expressive power of the querying in the proposed framework, we have constructed a database of soccer goals that contains the model descriptions of 35 MPEG-2

video clips. Each clip in the database contains 80 to 150 frames making a combined 3857 frames in the database. To instantiate the model descriptors as in Sec. III, each video clip has been manually annotated. At first, high-level object and event descriptors were instantiated, and the temporal boundaries of the events were determined. Next, segment-level object attributes in the form of EMUs and ERUs were found within event boundaries. The trajectory descriptor is selected to describe object motion due to the characteristics of long shots. To extract trajectory descriptors, we developed a manual authoring tool, a snapshot of which is shown in Fig. 9. Using the authoring tool, the locations of the soccer ball and soccer players were found and registered for each frame. The viewpoint registration onto the standard soccer field was completed by manually selecting at least 3 corner points in the scene and by calculating global mapping parameters from the current frame to the standard field using the selected field points and their respective positions on the standard field model. The global motion between the scene and the standard field model is described by 6-parameter affine transformation. Furthermore, approximate 3-D coordinates of the ball were also found by employing a physics-based approach. The low-level descriptions are integrated into the high-level descriptions by using the "Actor" vertices.

The clips in the database form five major semantic concepts as shown in Table I. That means each video in the database contains at least 3 subevents of the composite *goal* event, a *score* event that is preceded by a *header* or a *shooting* event, and a *long-pass* event. A *long-pass* may be due to a *free kick,* a *cross,* or a *center-pass* event. Several of the clips also include other events, such as *controlling the ball* and *short-pass,* in addition to those in Table I. For each semantic concept, we provide abstract models in XML form, which is editable by the users, to form their queries. In the following, we present the experiments using two different querying methods: i) query by example and ii) query by description.

A. Query by Example

In this experiment, we showed three example video clips in the database with different semantic concepts to five users and asked each of the users to select the clips in the database with the same concept. If a clip was selected by the majority of the users for the same example video, it is marked to be in the ground truth of that query. Next, we

searched for the similar clips to the queries by using two different methods: i) only low-level ball trajectories were used to find similar clips, ii) High- and low-level descriptors in the proposed model were used for matching and ranking purposes, respectively. That is, high-level event names and temporal event relations were used to find the matching clips, and the low-level ball trajectories were used to rank those clips for presentation.

The similarity of the query clip to the clips in the database was found by using Eq. 1 and Eq. 3. In Eq. 1, an infinite cost value is assigned for the mismatch of two events in Table I, that is, partial matches in event structures are only allowed if the mismatch involves at least one event that does not appear in Table I, such as *controlling the ball* and *short-pass.* The precision and recall rates for each query are shown in Table II.

The low precision and recall rates for the low-level system resulted from the gap between the low-level features and high-level semantics, and it justifies the need to use a model with high-level descriptors. As expected, the precision and recall rates when the model is used are high. The relatively low precision rate in query #1 is due to the confusion between the definition of *center-pass* and *cross* events. The majority of the users labeled several *cross* events as *center-pass* events. Because of the same reason, the recall rate in query #3 is relatively lower. Nonetheless, either precision or recall rate for these queries is high in both cases.

B. Query by Description

The proposed model contains a rich set of descriptors; however, it is almost impossible and also redundant to instantiate descriptors for every query possibility. In the situations where the model does not contain explicit high-level descriptors, the user is able to form high-level concepts from low-level features. In this experiment, we illustrate the expressive power of querying in our model in a situation where the user wants to search for long passes from the right side of the field. As shown in Table I, there is not an explicit textual event instantiation as "right side long-pass" in the database. Therefore, low-level specification of the query is needed, and a long pass from the right of the field can be specified by a trajectory descriptor. In addition to the low-level features, a query may also be extended with high-level constraints. In Fig. 10, the graph, where ** for long pass event name means any match is valid, stands for the query "goal from header that occurs after a long pass

from the right side of the field." The query uses the EMU attribute of the "Actor" entity to specify event-specific trajectory descriptor. The trajectory in the query in Fig. 10 can be specified by a sketch or by a trajectory descriptor of an existing clip in the database. We use the latter for low-level feature specification in query by description approach, i.e., we edit a description by copying its trajectory descriptor to the corresponding abstract model to form the query graph.

In the experiment, we formed the ground truth of the query in Fig. 10 as in Sec. V-A. The clip numbers selected as the ground truth by the majority of five users are shown in the first row of Table III. In Sec. V-A, it has been mentioned that the low-level features are used for ranking purposes; therefore, we expect all "header, score" clips to match the query and the ground truth clips to appear at higher ranks. The ranked results in the second row of Table III show that the top 7 matches are the ground truth clips. There are a total of 19 matching results (please verify that it is the total number of clips for concepts #2, #3, and #5), but 10 of them are shown in Table III.

The model provides a flexible way of adding constraints on queries. For instance, if the user is only interested in *free kick* events out of all long passes, the name of the left-most event in Fig. 10, shown as **, should be specified as *free kick.* The clip numbers of the top 3 matches of that query are shown in the third row of Table III. They form the subset of the ground truth for the query in Fig. 10. Many other interesting queries can be formed on the same graph. For instance, the graph of the query "header goals against *Team A* from the right-side free kicks" can be formed by specifying the team of the goalie as in Fig. 11. The returns of that query are shown in the last row of Table III, where only 2 out of 7 clips in the ground truth of the query in Fig. 10 satisfy the constraints.

In this experiment, we showed how low-level features can be used to form semantic concepts that do not exist in the database. We also exemplified that the same query graph can be used to form many interesting queries by adding more constraints on the model entities. Finally, the effect of adding additional constraints on graph and its filtering effect on the results was illustrated.

VI. Conclusion

We have presented a novel video event model that integrates high-level semantic features and low-level, object-based video features within a single framework. The queries in the system are formulated and resolved in a single graph-based framework that provides expressive querying of video. We have considered two popular querying methods to show the effectiveness of our system. At first, we input example video clips from different semantic concepts into the system and retrieved the clips with similar descriptions. The results of this experiment illustrate the increased precision and recall rates of the model-based system to the low-level system on a ground truth set formed by multiple users. In the second experiment, we demonstrated the expressive querying of the model by querying a semantic concept for which no explicit descriptors exist in the database.

In the experiments, we assumed either the temporal relations between events follow in one direction or the inverse temporal relations are explicitly instantiated in the database. The reason for this assumption is to simplify the complexity of the directional graph matching due to the limitation in directional graph traversal. For the same reason, actor-event and actor-object edges are stored as nondirectional (bidirectional) edges, but this does not impose any constraints on the descriptions as that of the temporal relations because the actor edges do not have any strictly directional attributes. Another assumption in the experiments is that the queries do not involve any inferencing of spatio-temporal relations, i.e., the system does not utilize inference mechanism that will deduce "object C is to the right of object A" given "object B is to the right of object A" and "object C is to the right of object B." Therefore, we have queried those relationships that are explicitly specified in the database.

A promising future research direction related to the proposed model is the resolution of queries that involve incomplete/imperfect concepts. The resolution of such queries requires the integration of low-level and high-level constraints as exemplified in our second experiment. The development of a generic query resolution framework will enable the users to form their own semantic concepts, and they will not be limited to the querying of hard-wired database concepts. An important result of that research is the increased user satisfaction, which is arguably the most important criterion in the evaluation of multimedia

systems.

Another interesting future research topic for semantic search and retrieval is to develop video algorithms for automatic instantiation of low-level and high-level descriptors in the model. The automatic extraction of low-level descriptors requires solving two main computer vision problems: segmentation and multi-object tracking. It is widely accepted that these problems require using constraints that may be in the form of domain specification, control over camera positions, etc. Understanding of video events is essential for high-level descriptor extraction, and it is another challenging problem involving multiple disciplines.

References

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

[2] E. Oomoto and K. Tanaka, "OVID: Design and Implementation of a Video-Object Database System," *IEEE Trans. on Knowledge and Data Eng.*, vol. 5, no. 4, pp. 629-643, Aug. 1993.

[3] R. Hjelsvold and R. Midstraum, "Modelling and Querying Video Data," in *Proc. of the 20th VLDB Conf.,* Santiago, Chile, 1994.

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

[5] J.L. Koh, C.S. Lee, and A.L.P. Chen, "Semantic Video Model for Content-based Retrieval," in *IEEE Int'l. Conf. Mult. Comp. and Sys.*, vol. 2, pp. 472-478, 1999.

[6] L.A.E. Al Safadi and J.R. Getta, "Semantic Modeling for Video Content-Based Retrieval Systems," in 23^{rd} *AustralAsian Comp. Science Conf.,* pp. 2-9, 2000.

[7] J.R. Smith and A.B. Benitez, "Conceptual Modeling of Audio-Visual Content," in *IEEE Int'l. Conf. on Mult. and Expo (ICME),* vol. 2, pp. 915-918, 2000.

[8] 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," *IEEE Computer*, vol. 28, no. 9, pp. 23-32, Sept. 1995.

[9] J.R. Bach, C. Fuller, et al., "Virage Image Search Engine: An Open Framework for Image Management," in *Proc. of the IS&T/SPIE Conf. on Storage and Retrieval for Image and Video Databases IV,* pp. 76-87, Feb. 1996.

[10] J.R. Smith and S.F. Chang, "VisualSEEk: A Fully Automated Content-Based Image Query System," in *Proc. ACM Conf. Multimedia,* Boston, MA, Nov. 1996.

[11] S.F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong," A Fully Automated Content-Based Video Search Engine Supporting Spatio-Temporal Queries," *IEEE Trans. on Circuits and Systems for Video Tech. (CSVT),* vol. 8, no. 5, pp. 602-615, Sept. 1998.

[12] A. Yoshitaka, T. Ishii, M. Hirakawa, and T. Ichikawa, "VIOLONE: Video Retrieval By Motion Example," *J. Visual Languages and Computing,* vol. 7, no. 4, pp. 423-443, 1996.

[13] K. Porkaew, M. Ortega, and S. Mehrotra, "Query reformulation for content based multimedia retrieval in MARS," in *IEEE Int'l. Conf. on Mult. Comp. Sys.,* vol. 2, pp. 747-751, 1999.

[14] S. Antani, R. Kasturi, and R. Jain, "A Survey on the use of Pattern Recognition Methods for Abstraction, Indexing, and Retrieval of Images and Video," *Pattern Recognition,* vol. 35, no. 4, pp. 945-965, Apr. 2002.

[15] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of Early Years," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI),* vol. 22, no. 12, pp. 1349-1380, Dec. 2000.

[16] Y. Rui, T.S. Huang, and S-F. Chang, "Image Retrieval: Current Techniques, Promising Directions, and Open Issues," *J. Visual Com. and Image Representation,* vol. 10, no. 1, pp. 39-62, March 1999.

[17] Y.A. Aslandogan and C.T. Yu, "Techniques and Systems for Image and Video Retrieval," *IEEE Trans. on Knowledge and Data Eng.*, vol. 11, no. 1, pp. 56-63, Jan./Feb. 1999.

[18] A.Yoshitaka and T. Ichikawa, "A Survey on Content-Based Retrieval for Multimedia Databases," *IEEE Trans. on Knowledge and Data Eng.*, vol. 11, no. 1, pp. 81-93, Jan./Feb. 1999.

[19] F. Idris and S. Panchanathan, "Review of Image and Video Indexing Techniques," *Journal of Vis. Com. and Img. Rep.*, vol. 8, no. 2, pp.146-166, June 1997.

[20] A. Pentland, R. Picard, and S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," in *Proc. of the IS&T/SPIE Conf. on Storage and Retrieval for Image and Video Databases II,* Feb. 1994.

[21] S.W. Smoliar and H. Zhang, "Content-Based Video Indexing and Retrieval," *IEEE Multimedia*, vol. 1, no. 2, pp. 62-72, 1994.

[22] J.-Y. Chen, C. Taskiran, A. Albiol, C.A. Bouman, and E.J. Delp, "ViBE: A Video Indexing and Browsing Environment," in *Proc. of the IS&T/SPIE Conf. on Storage and Retrieval for Media Databases IV,* pp. 199-207, Jan. 1999.

[23] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," *IEEE Trans. on Circuits and Systems for Video Tech.,* vol. 8, no. 5, pp. 644-655, Sept. 1998.

[24] I.J. Cox, M.L. Miller, T.P. Minka, T.V. Papathomas, and P.N. Yianilos, "The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychological Experiments," *IEEE Trans. on Image Processing,* vol. 9, no. 1, pp. 20-37, Jan. 2000.

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

[26] W. Chen and S-F. Chang, "Creating Semantic Visual Templates for Video Databases," in *IEEE Int'l. Conf. on Mult. and Expo (ICME),* vol. 3, pp. 1337-1340, 2000.

[27] M.R. Naphade and T.S. Huang, "A Probabilistic Framework for Semantic Video Indexing, Filtering, and Retrieval," *IEEE Trans. on Multimedia,* vol. 3, no. 1, pp. 141-151, March 2001.

[28] M.-S. Hacid, C. Decleir, and J. Kouloumdjian, "A Database Approach for Modeling and Querying Video Data," *IEEE Trans. on Knowledge and Data Eng.,* vol. 12, no. 5, pp. 729-749, Sept./Oct. 2000.

[29] ISO/IEC Committee Draft 15938-5 Information Technology - Multimedia Content Description Interface: Multimedia Description Schemes," ISO/IEC/JTC1/SC29/WG11/N3966, March 2001.

[30] ISO/IEC Committee Draft 15938-5 Information Technology - Multimedia Content Description Interface: Multimedia Description Schemes," ISO/IEC/JTC1/SC29/WG11/N4242, Oct. 2001.

[31] P.P. Chen, "The Entity-Relationship Model - Toward a Unified View of Data," *ACM Transactions on Database Systems,* vol. 1, no. 1, pp. 9-36, March 1976.

[32] "MPEG-7 Visual Part of Experimentation Model Version 10.0," MPEG-7 Output Document ISO/IEC/JTC1/SC29/WG11/MPEG99/N4063, March 2001.

[33] J.F. Allen, "Maintaining Knowledge About Temporal Intervals," *Comm. ACM,* vol. 26, no. 11, pp. 832-843, 1983.

[34] J.Z. Li, M. T. Ozsu, and D. Szafron, "Modeling of Video Spatial Relationships in an Object Database Management System," in *IEEE Proc. Int'l. Workshop on Mult. Dat. Man. Sys.*, 1996.

[35] M. Egenhofer and R. Franzosa, "Point-set topological spatial relations," *Int'l J. of Geographical Information Systems,* vol. 5, no. 2, pp. 161-174, 1991.

[36] S.K. Chang, "Iconic Indexing By 2D Strings," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 6, no. 4, pp. 413-428, 1984.

[37] M. Gyssens, J. Paredaens, J. van den Bussche, and D. van Gucht, "A Graph-oriented Object Database Model," *IEEE Trans. on Knowledge and Data Eng.,* vol. 6, no. 4, pp. 572-586, Aug. 1994.

[38] M. Andries, M. Gemis, J. Paradeans, I. Thyssens, and J. van den Bussche, "Concepts for Graph-Oriented Object Manipulation," *Advances in Database Tech.- EDBT'92*, eds. A. Pirotte et al., pp. 21-38, 1992, Springer-Verlag.

[39] Y. Fu, A. Ekin, A.M. Tekalp, and R. Mehrotra, "Temporal Segmentation of Video Objects for Hierarchical Object-based Motion Description," *IEEE Trans. on Image Processing,* vol. 11, no. 2, pp. 135-145, Feb. 2002.

[40] A. Ekin, A.M. Tekalp, and R. Mehrotra, "Automatic Extraction of Low-Level Object Motion Descriptors," in *Proc. IEEE ICIP,* Thessaloniki, Greece, Oct. 2001.

[41] A. Ekin and A.M. Tekalp, "A Framework for Analysis and Tracking of Soccer Video," in *Proc. of the IS&T/SPIE Conf. on Visual Com. and Image Proc. (VCIP),* Jan. 2002.

[42] A. Ekin, A.M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," accepted for publication in *IEEE Trans. on Image Processing.*

[43] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness,* W.H. Freeman, San Francisco, CA, 1979.

[44] S. Dagtas, W. Al-Khatib, A. Ghafoor, and R.L. Kashyap, "Models for motion-based video indexing and retrieval," *IEEE Trans. on Image Processing,* vol. 9, no. 1, pp. 88-101, Jan. 2000.

Semantic Concept	Number of Clips
1. Cross, Shooting, Score	9
2. Cross, Header, Score	9
3. Free Kick, Header, Score	9
4. Center Pass, Shooting, Score	7
5. Center Pass, Header, Score	1

TABLE I

THE SEMANTIC CONCEPTS IN THE DATABASE AND THEIR DISTRIBUTION

		System Using the Model		Low-Level System	
Query	Query Concept	Precision	Recall	Precision	Recall
Query #1	2	7/9	7/7	1/10	1/7
Query #2	3	9/9	9/9	2/10	2/9
Query #3	4	7/7	7/10	3/10	3/10

TABLE II

THE PRECISION-RECALL RATES OF THE QUERY RESOLUTION USING MODEL AND USING ONLY LOW-LEVEL FEATURES

Description	The Clip Numbers
Ground truth of the query in Fig. 10	21, 22, 23, 26, 30, 31, 33
Query returns (ordered)	33, 22, 31, 30, 26, 23, 21, 6, 20, 18,...
The returns of (Long Pass=Free Kick) query	30, 26, 23, ...
Goals against a specific team	30, 23, ...

TABLE III

THE SEMANTIC CONCEPTS IN THE DATABASE AND THEIR DISTRIBUTION

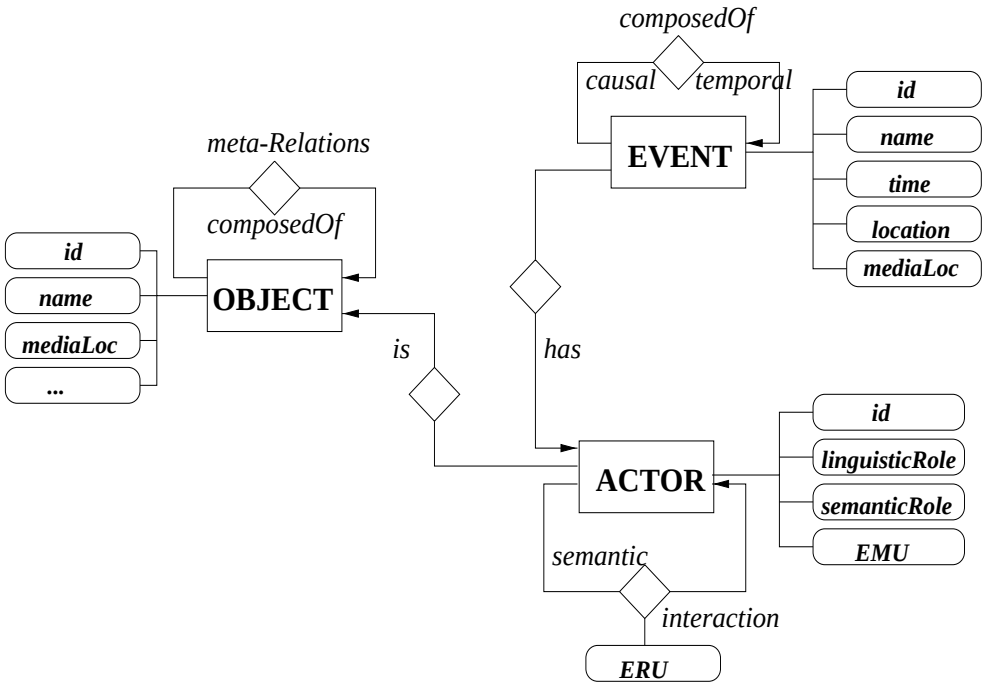

Fig. 1. Graphical representation of the model

(a) (b) (c)

Fig. 2. The key frames of free kick subevent, (a) (frame #0) player is approaching, (b) (frame #14) player touches the ball, (c) (frame #28) ball is on the air and the player is not in the scene

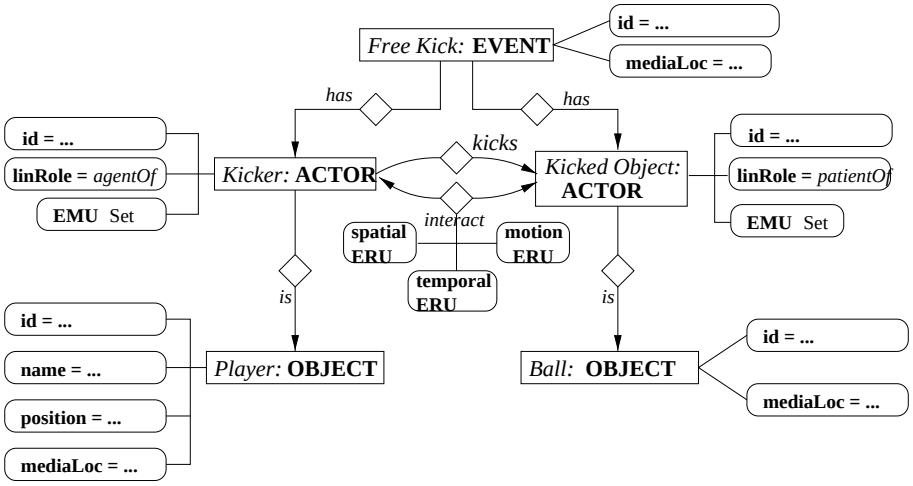

Fig. 3. The description graph of free kick event, its actors, and participating objects

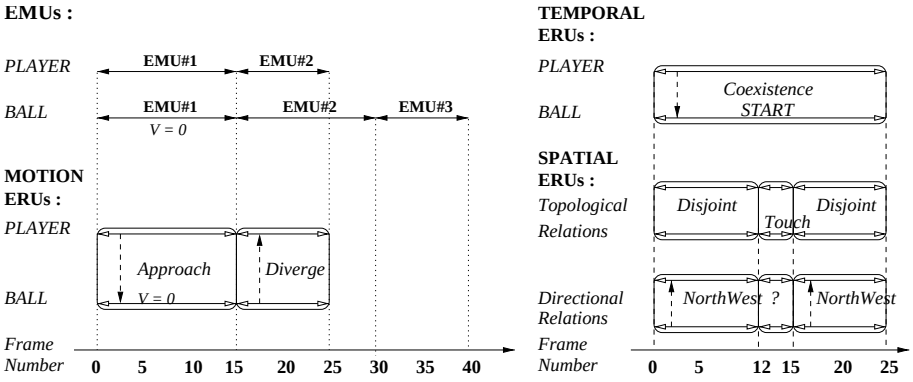

Fig. 4. Low-level video segment relations, EMUs and ERUs, in the free kick event

(a) (b) (c)

Fig. 5. The key frames of header (a-b) and score events

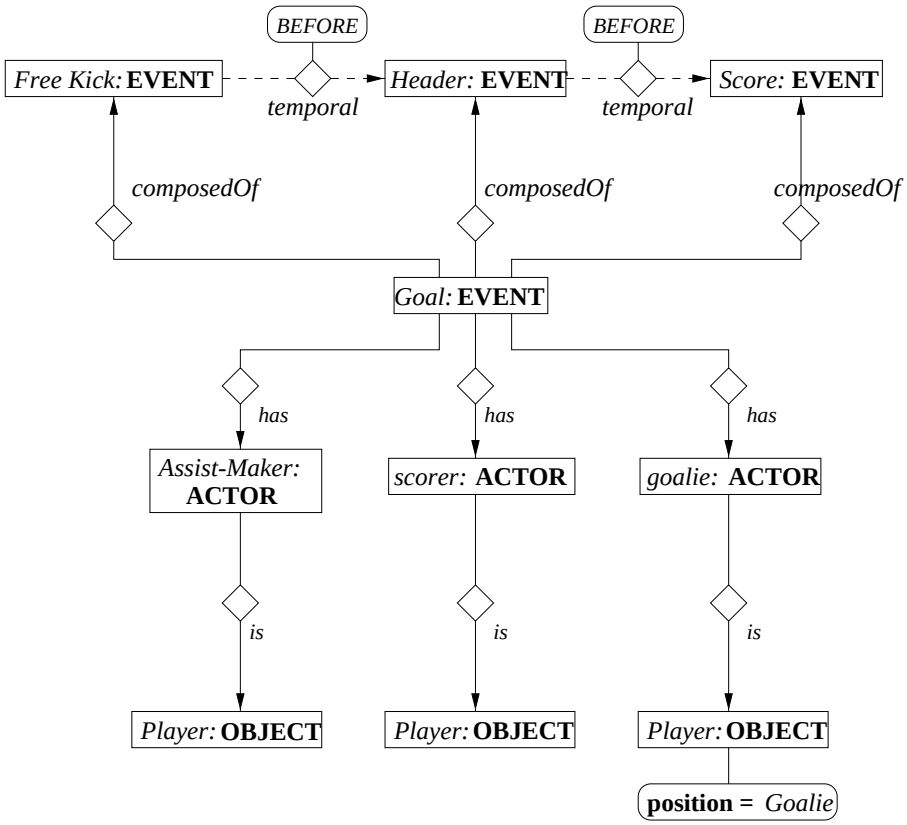

Fig. 6. The description graph of the composite goal event

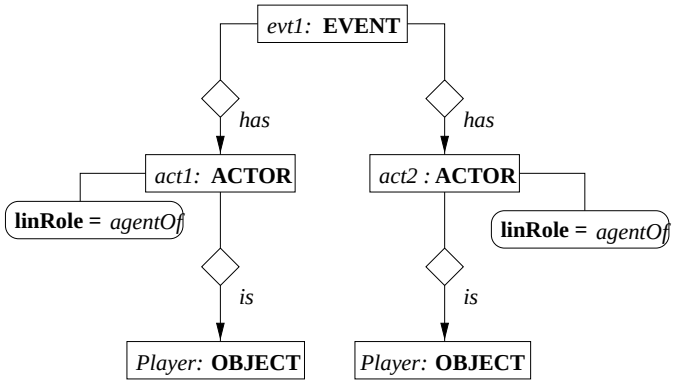

Fig. 7. An example graph pattern for the query "events with 2 player agents"

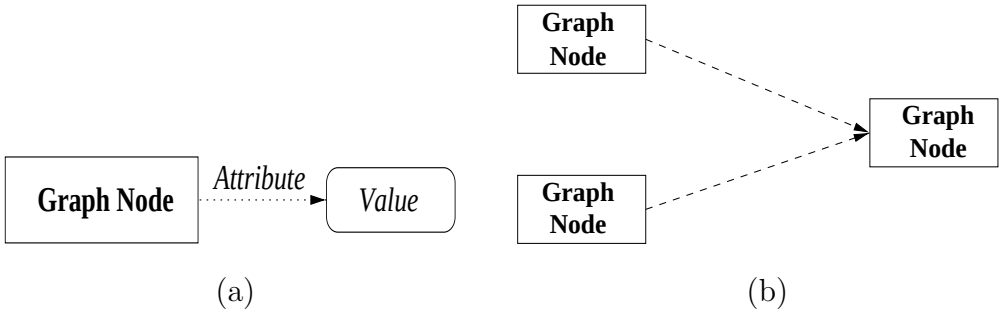

Fig. 8. Browsing patterns (a) from a single node to its attributes, (b) from multiple nodes to another node

Fig. 9. Screen capture of manual authoring tool

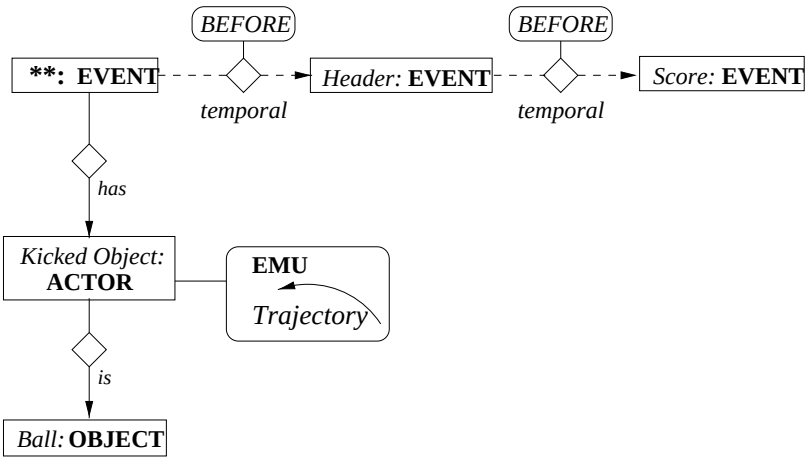

Fig. 10. Query specification by low- and high-level constraints

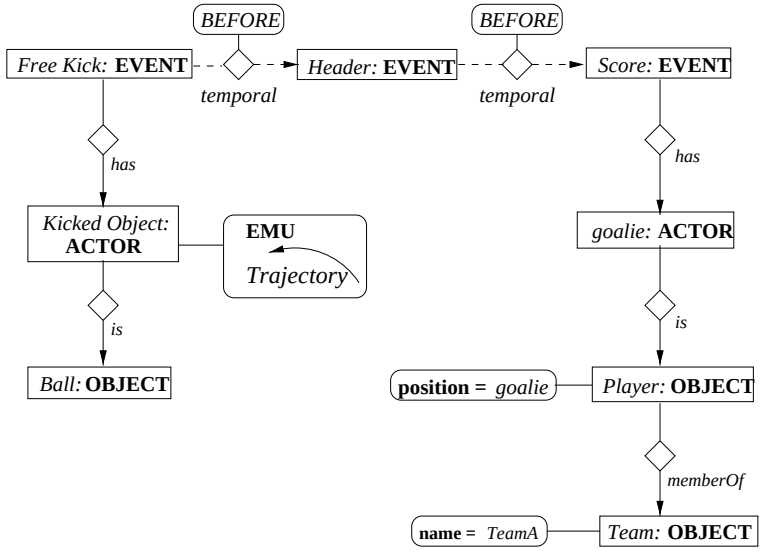

Fig. 11. The header goals against Team A from the right side free kicks