Multimedia Systems 7: 409–423 (1999)

Multimedia Systems

A multi-level abstraction and modeling in video databases*

Young Francis Day[1], **Ashfaq Khokhar**[2], **Serhan Dağtaş**[3], **Arif Ghafoor**[4]

[1] Multimedia Documentation Program, Siemens Corporate Research, 755 College Road East, Princeton, NJ 08540, USA
[2] Department of Electrical Engineering and, Department of Computer and Information Science, University of Delaware, Newark, DE 19716, USA
[3] Philips Research-USA, 345 Scarborough Rd, Briarcliff Manor, NY 10510, USA
[4] Distributed Multimedia Systems Laboratory, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA; e-mail: ghafoor@ecn.purdue.edu

Abstract. In this paper, we propose a multi-level abstraction mechanism for capturing the spatial and temporal semantics associated with various objects in an input image or in a sequence of video frames. This abstraction can manifest itself effectively in conceptualizing events and views in multimedia data as perceived by individual users. The objective is to provide an efficient mechanism for handling content-based queries, with the minimum amount of processing performed on raw data during query evaluation. We introduce a multi-level architecture for video data management at different levels of abstraction. The architecture facilitates a multi-level indexing/searching mechanism. At the finest level of granularity, video data can be indexed based on mere appearance of objects and faces. For management of information at higher levels of abstractions, an object-oriented paradigm is proposed which is capable of supporting domain specific views.

Key words: Semantic modeling – Video databases – Content-based retrieval – Spatio-temporal logic – Object-oriented modeling

1 Introduction

Multimedia databases have recently been the subject of intensive research. A number of Web-based emerging applications such as telemedicine, digital libraries, distance learning, tourism, distributed CAD/CAM, GIS, etc., are expected to use general-purpose multimedia database systems. Unlike traditional relational databases, multimedia databases allow direct manipulation of multimedia objects consisting of text, images, graphics, audio, music, and full-motion video data.

Many Web-based multimedia applications require digitizing large archives of image and video data for interactive retrieval including searching, browsing, selective replay, editing, etc. Due to shear volume of such data, these capabilities require efficient computer vision/image-processing algorithms for automatic abstractions and indexing of images and video clips. In addition, there are two prominent issues associated with video/image data modeling and management.

- *Development of formal techniques for semantic modeling of multimedia information, especially for video and image data.* These models should be rich in their capabilities for abstracting multimedia information and capturing semantics. They should be able to provide canonical representations of complex images, scenes, and events in terms of objects and their spatio-temporal behavior. These models need to be compared and evaluated, in case of their varied theoretical bases and complexities.
- *Design of powerful indexing, searching and organization methods for multimedia data.* Search in multimedia databases can be computationally intensive, especially if content-based retrieval is needed for image and video data stored in compressed or uncompressed form.

The key characteristic of video data is its spatial/temporal semantics that makes it unique from other types of data such as text, voice, and image. A user of video database can generate queries containing both temporal and spatial concepts. However, considerable semantic heterogeneity may exist among users of such data due to differences in their pre-conceived interpretation or intended use of the information provided in a video clip. *Semantic heterogeneity* has been a difficult problem for conventional databases [7], and even today this problem is not clearly understood. Consequently, providing a comprehensive interpretation of video data is a challenging problem.

In an effort to address these issues in an organized manner, we view the video data modeling at two levels of abstraction as depicted in Fig. 1.

1. *Low level modeling.* The identification of objects, their relative positions and movements, segmentation and grouping of video data using image/video-processing techniques fall into this category. The major challenge at this level is accurate recognition and tracking the movements of objects at an intra- and inter-frame level. At this level, recognition of objects of interest in each frame is performed by automatic or manual techniques and

* This research has partially been supported by an NSF grant, IRI-9619812
Correspondence to: A. Ghafoor

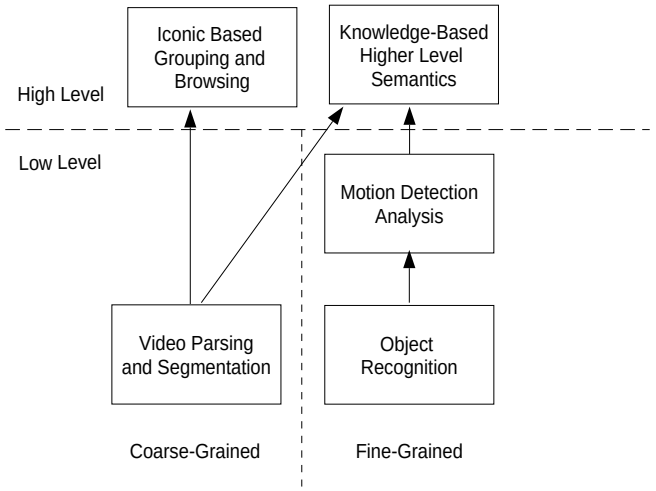

Fig. 1. Semantic modeling of video data

intermediate-level data indexes are created for subsequent, higher level analyses. One such data-indexing technique is the model Video Semantic Directed Graph (VSDG) proposed in [8]. VSDG is used to maintain temporal information of objects once they are identified by object recognition techniques. This model forms the basis for developing a higher level abstraction and indexing mechanism.

2. *High-level modeling.* The high-level semantics can be formulated by the users to construct different views of the video data. There has been a growing interest in developing efficient theoretical foundations to represent high-level semantics and event specifications. Several methods have been proposed in the literature on this topic. The essence of these formalisms is the temporal modeling and specification of events present in video data. Semantic operators, including logic, set, and spatio-temporal operators, are extensively used to develop such formalisms. Logical operators include the conventional boolean connectives such as *not*, *and*, *or*, *if-then*, *only-if*, and *equivalent-to*. Set operators like *union*, *intersection*, and *difference* are mostly used for event specification, as well as for video composition and editing. Spatio-temporal operators, based on temporal relations, are employed for event specification and modeling. This leads to a high-level characterization of information in video data, and subsequently an object-oriented, conceptual model as presented in the following sections.

1.1 Background

Most of the existing video database systems to a limited extent address the spatio-temporal semantics either by employing primitive image-processing techniques for indexing of video data or using traditional database approaches based on keywords or annotated textual descriptors [4, 10, 21, 22]. For indexing, keywords and textual descriptions have also been suggested in an object-oriented realm [18, 20]. Video segments can be joined or concatenated based on their semantics. However, these approaches are very tedious, since the perception of video contents is done manually by users, not through an automatic image-processing/computer-vision-based mechanism.

A number of systems that automate the indexing mechanism have also been proposed, such as [19] that automatically parses video data into scenes using a color histogram comparison routine. This method has limited capability, since only semantics associated with scene changes are captured. In [24], a hierarchical video stream model is proposed that uses a template- or histogram-matching technique to identify scene changes in a video segment. In this system, a video stream is parsed, and the information is stored in the database. However, this system is limited to specific types of videos and uses only manual indexing mechanism. Additionally, there is no modeling of temporal events within a shot.

An approach based on spatio-temporal logic is presented in [6], which is used to describe the content of an image or a sequence of images. A prototype image sequence retrieval system is developed, where images are processed and represented by spatio-temporal logic, and query is input by using example images, which is then translated into spatio-temporal logic. Query processing is done by matching spatio-temporal logic representations of query images and images stored in the database. This work represents a significant progress in content-based retrieval of video data. However, due to the limitation of the methodology used in this approach, the modeling of higher level concepts of spatio-temporal events is not addressed, nor is the grouping of information across video clips.

In [28], a multimedia database system for content-based retrieval is presented. An object-oriented data model and a query language are used. The database schema is represented through a hierarchy with *is_a* and *part_of* relationships among classes. A class is associated with a domain knowledge to represent a certain concept. Retrieval is done by matching the query and the domain knowledge stored in classes. Video data is associated with textual annotation-like knowledge.

Another video database system based on an algebraic video model is presented in [26]. The proposed *video algebra* provides functionalities for creating video presentations which include nested structures, temporal composition, and multiple views. It also allows users to assign multiple coexisting interpretations to same video segment, and provides associative access based on the information content. The contents of a video segment can be arranged in a hierarchy. Yet, there is no spatio-temporal modeling of the video data itself.

In [9], a three-level motion analysis methodology is proposed. Starting from the extraction of trajectory of a macro-block in an MPEG video, followed by averaging all trajectories of the macro-blocks of the objects, and finally relative position and timing information among objects, a dual hierarchy (spatial + temporal) is established for representing video.

1.1.1 A classification of existing models for spatio-temporal specification

Above mentioned techniques for *high-level modeling* can be classified into two categories; object-centered models and event-centered models.

- In **object-centered models**, the coordinates of the centers of objects are used as a sequence that describes the spatio-temporal behavior of an object [8, 9]. This description can be relative to the starting point of the trajectory or a fixed point (origin) on the screen. Similarly, relative position of two objects can be represented as the difference vector between them. While trajectory-based methods offers such generality, they lack the flexibility to correctly categorize similar events that may be represented by a wide variety of trajectory descriptions.
- According to the **event-centered models**, a set of spatial relations [1] are used to determine the relative positions of two objects at each time instance based on their spatial projection intervals on each axis [3, 6]. For a video sequence, this translates into a series of symbols which are generally handled by algorithmic methods with a polynomial computational complexity. A conversion from symbolic to numerical representation, however, allows analytical methods to be used for categorization purposes and eliminates the computational burden. Symbolic descriptions are suitable only for multiple object events by design, but offers a more flexible and effective way to describe events.

In summary, most of the existing video database systems lack the ability to provide a general-purpose abstraction mechanism which otherwise is needed to handle semantic heterogeneity that may exist across a large population of users.

1.2 Our approach

This paper deals with the issues related to *user-independent view* and semantic modeling of image/video data. We emphasize that a general-purpose multimedia database system should provide an environment for users to express and for the system to process semantically heterogeneous queries. Toward this goal, we propose a model that captures spatio-temporal aspects of information associated with objects (such as persons, buildings, and vehicles) present in video data. This provides a somewhat *semantically unbiased abstraction* of video data. For each input video clip, using a database of known objects, we first suggest to identify the corresponding objects, their sizes and locations, their relative positions and movements, and then encode this information in a spatio-temporal model. The encoded data potentially can be used to develop a semantically rich *information space*. The proposed model helps in avoiding extensive computation on raw data during on-line query processing.

In addition, we propose a two-pronged approach for modeling image/video data as illustrated in Fig. 2. We introduce an object-oriented model to store and retrieve these spatio-temporal events and semantics associated with a video. The top flow corresponds to spatio-temporal modeling and characterization of events present in the video data. This modeling deals with information at the level of a video clip. The bottom flow corresponds to the users' view of the data, where grouping/linking of information across clips is supported using an object-oriented paradigm. Integration of both intra- and inter-clip modeling leads to an efficient indexing mechanism for on-line content-based query processing. For most of the queries, the proposed framework avoids performing computation on raw data during query processing, since such computations can be quite extensive and should be carried out off-line. Also, this framework allows conceptualization of video data using both bottom-up and top-down object-oriented data abstraction approaches. In the bottom-up approach, a user can build complex events using simple events, while in the top-down approach, a user can integrate/group events with shared semantics. We also discuss an overall description of a database architecture for implementing an indexing scheme for video data. Throughout this paper we will use a hypothetical sports video database as a running example to illustrate various concepts.

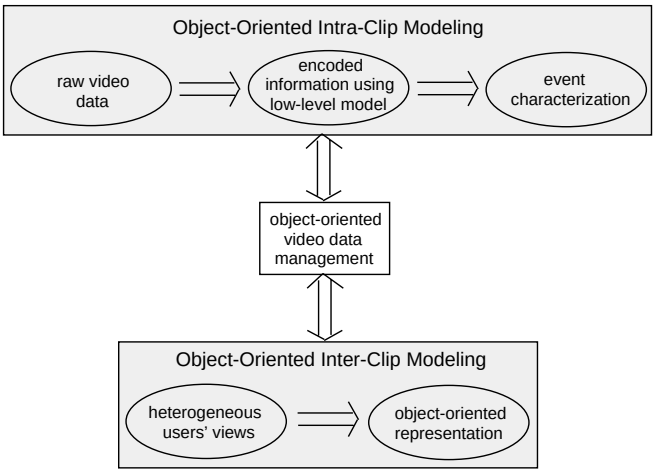

Fig. 2. An object-oriented approach to data abstraction of video data

The organization of this paper is as follows. In the next section, we present a model for capturing spatial and temporal relationships among salient objects present in a video clip. Section 3 outlines a methodology for formally characterizing "events" embedded in video data. For this purpose, *generalized n-ary relations* are introduced in this section. An object-oriented paradigm is proposed in Sect. 4 to categorize events into classes and to provide powerful abstraction tools to users for indexing of video data. We also discuss an overall description of the proposed database architecture for implementing the object-oriented indexing scheme. The conclusion section summarizes the paper.

2 Low-level modeling of image/video

Generally, most worldly phenomena can be expressed in the form of knowledge by describing the interplay among physical objects in the course of their relationship in space and time. Physical objects may include persons, buildings, vehicles, etc. Video is a typical replica of such a worldly environment. In conceptual modeling of video data, it is therefore

important that we identify physical objects and their relationships in space and time. Subsequently, we can represent these relations in a suitable structure that is useful for users to manipulate.

Various models have been proposed in the literature to specify temporal relations among objects, including the well-known model of temporal-interval [1, 13, 17]. For spatial relations, most of the modeling techniques are based on projecting objects onto a 2D or 3D coordinate system. Very little attempt has been made to formally express spatio-temporal interactions of objects in a single framework. Though, in [14], spatial/temporal meta-data for video database is defined, no detailed approach has been provided for data modeling and information management. In this section, we present a model to capture both spatial and temporal semantics of video data. An important feature of this model is that it allows a low-level unbiased representation of video information. The key concept of the model is that such a representation of a video database can provide a stable and unified reference framework for specifying complex spatio-temporal events, and hence allows users to construct a wide range of *views*.

Formally, a video sequence can be viewed as a structure $\mathscr{V}$ = ($\mathscr{F}$, $<$, $\mathscr{D}$, h), where

- $\mathscr{F}$ is a set of video frames (f_i), also called a sequence,
- $<$ is a binary, transitive, irreflexive relation on $\mathscr{F}$. ($\mathscr{F}$, $<$) is called the flow of video sequence,
- $\mathscr{D}$ is the domain of frames. Each frame f_i has a domain d_i which consists of extractable features (e.g., salient physical objects),
- h is a map such that, for any predicate p, there exists a possible set $\{s_i\}$, s_is are disjoint subsequences of $\mathscr{F}$ such that $h(p,s_i)$ is true.

By applying a set of functions to a video clip, a collection of tuples representing the appearance of salient physical objects are generated.

2.1 Spatio-temporal modeling over a sequence of frames

The spatial attribute of a salient physical object present in a frame can be extracted as a bounding volume V, that describes the spatial projection of an object in three dimensions. It is a function of $Bounding\_Rectangular(L)$, $centroid$, and $depth$ information related to the object. The bounding rectangular is computed with reference to a coordinate system with an origin at the lower left corner of each frame. The pair (x, y) represents the coordinate of the lower left corner of rectangular L. Both V and L are expressed as

$$Bounding\_Rectangular\ (L) = (width, height, x, y), \quad (1)$$

$$Bounding\_Volume\ (V) = (Bounding\_Rectangular, centroid, depth). \quad (2)$$

Temporal information of objects can be captured by specifying the changes in the spatial parameters associated with the bounding volume (V) of objects in a given sequence of frames. At the finest level of granularity, these changes can be recorded at each frame. Although such a fine-grained temporal specification may be desirable for frame-based indexing of video data, it may not be required in most of the applications. Also, the overhead associated with such detailed specification may be formidable. Alternatively, a coarse-grained temporal specification can be maintained by only analyzing frames at δ distance apart. This skip distance (δ) in terms of number of frames will depend upon the complexity of episodes. There is an obvious tradeoff between the amount of storage needed for temporal specification and the detailed information maintained by the model.

2.2 The proposed model

An object appearing in the video clip can be represented by a tuple containing a set of spatio-temporal descriptions. It is assumed that a video clip (VC) is first parsed into segments using histogram comparison [19] and a sequence of segments (S_is) are identified. Within each segment, motion tracking of identifiable domain objects is performed. Then the model can be formally described as follows.

1. For each segment S_i of VC, $1 \leq i \leq m$ (assume that VC consists of m segments)

For each identified domain object o_{ij} in S_i, record the following information,

ϱ = ($oid, d\_d, \{s\_t\}$), where,

- oid: object identifier assigned by the system;
- d_d: descriptive data, e.g., object type, name (for human beings, whenever possible);
- $\{s\_t\}$: an ordered set of spatio-temporal description, s_t = (π, τ, m_v), where
  - π: the starting frame number of the object's appearance;
  - τ: the duration (in frames) of the object's appearance; $\tau = n\delta + 1$, $n \geq 0$ and is an integer.
  - m_v: a motion vector associated with o_{ij} during the interval starting at π with duration τ; m_v = ($Z_1, \ldots, Z_{\frac{\tau-1}{\delta}+1}$). The element Z_i ($\forall i\, 1 \leq i \leq \frac{\tau-1}{\delta} + 1$) of m_v is the *bounding volume* at *i-th* sampled frame. In other words, Z_i = ($Bounding\ Box_i$, $depth_i$, $centroid_i$), and $Bounding\ Box_i$ = ($width_i$, $height_i$, x_i, y_i), where τ represents the number of frames associated with the object o_{ij} in a certain subinterval of S_i, and δ is the time granularity for tracking motion of every object in a video segment.

2. Perform concatenation across segments as follows.

- If an oid is unique across segments, then put the corresponding tuple ϱ in the object collection VO of VC; otherwise, perform concatenation as follows.
- If $\varrho_i.oid$ = $\varrho_j.oid$, then create a new tuple ϱ_k, where $\varrho_k.oid$ = $\varrho_i.oid$, $\varrho_k.d\_d$ = $\varrho_i.d\_d \cup \varrho_j.d\_d$, $\varrho_k.\{s\_t\}$ = $\varrho_i.\{s\_t\} \cup \varrho_j.\{s\_t\}$. Put ϱ_k in VO. Note that within $\varrho_k.\{s\_t\}$, if s_t_r = (π_r, τ_r, m_v_r) , s_t_w = (π_w, τ_w, m_v_w), and $\pi_r + \tau_r = \pi_w$, then create s_t_u = (π_u, τ_u, m_v_u), where $\pi_u = \pi_r$, $\tau_u = \tau_r + \tau_w$, $m\_v_u = m\_v_r \cup m\_v_w$, put s_t_w in $\varrho_j.\{s\_t\}$, remove s_t_r and s_t_w from $\varrho_j.\{s\_t\}$.

Conceptually, the model can be illustrated by Fig. 3, where the *x*-axis represents frame numbers. Within each segment, an identified domain object is represented by a set of intervals in which it appears. In Fig. 3, there are two segments. Objects O_1, O_2, O_3, and O_4 are identified in segment

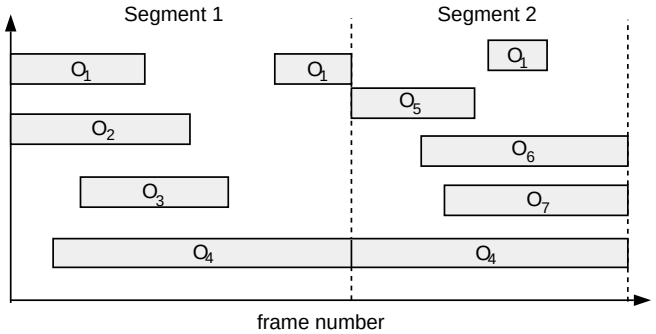

Fig. 3. Conceptual representation of low-level modeling

Table 1. n-ary relations

Relation name	Symbol	constraints, $\forall\ i,\ 1 \leq i < n$	τ
before	B	$\tau_e^i < \tau_s^{i+1}$	$\sum_{i=1}^{n-1} {\tau_\delta}^i + \tau^n$
meets	M	$\tau_e^i = \tau_s^{i+1}$	$\sum_{i=1}^{n} \tau^i$,
			$\sum_{i=1}^{n-1} {\tau_\delta}^i + \tau^n$
overlaps	O	$\tau_s^i < \tau_s^{i+1} < \tau_e^i < \tau_e^{i+1}$	$\sum_{i=1}^{n-1} {\tau_\delta}^i + \tau^n$
contains	C	$\tau_s^i < \tau_s^{i+1} < \tau_e^{i+1} < \tau_e^i$	τ^1
starts	S	$\tau_s^i = \tau_s^{i+1} \wedge \tau_e^i < \tau_e^{i+1}$	τ^n
completes	CO	$\tau_s^i < \tau_s^{i+1} \wedge \tau_e^i = \tau_e^{i+1}$	τ^1
equals	E	$\tau_s^i = \tau_s^{i+1} \wedge \tau_e^i = \tau_e^{i+1}$	τ^1

τ_s^i = starting coordinate of object τ^i; τ_e^i =ending coordinate of object τ^i; ${\tau_\delta}^i = \tau_s^{i+1} - \tau_s^i$

1, O_1 disappeared for a number of frames and then reappeared. The two intervals where O_1 appears are related by a *before* relation (see next section). Similarly, five domain objects are identified from segment 2, where two objects O_1 and O_4 have appeared in segment 1.

There are three choices for representing the data structure for the proposed model. In the first choice, an object's appearances are represented by a tuple $(clip\#, segment\#, oid, *d\_d, \{\pi, \tau, m\_v)\})$. Note, that $segment\#$ may be a list. For the second choice of data structure, each appearance of an object is represented as $(clip\#, segment\#, oid, *d\_d, \pi, \tau, m\_v)$. The third choice is to use the VSDG model presented in [8].

From motion vectors (m_vs) we can perform inter-object motion analysis to determine the relative movements among objects. For any sampled frame, the relative position between objects O_i and O_j can be evaluated by applying the spatial relationship between their projections on each coordinate axis, x, y, and z.

We would like to point out that extraction of features such as motion of an object, bounding volumes, etc., directly from raw video data is computationally tedious. The current state-of-the-art techniques in image understanding/computer vision are not robust enough to handle complex scenes in real time. However, our formalization for composing spatial/temporal events does not depend on any particular feature extraction or recognition technique. We believe that realization of advanced/robust image-understanding engines bounds to exist due to its vital importance in commercial and defense applications. Currently, we perform these operations in a semi-automatic way through our implementation programs.

3 Framework for characterizing events

As mentioned in the previous section, the proposed model is a low-level spatio-temporal representation of video data. However, it provides a mechanism that allows specifications of more complex spatio-temporal events, based on the spatio-temporal operations discussed next.

3.1 Generalized spatial and temporal operations

The generalized operators are extensions of our earlier work on temporal relations [17]. The reason for introducing the generalization to both spatial and temporal domains is to express spatial/temporal events in a unified format. We first give a definition for an *interval*, then present a definition for a *generalized n-ary relation*.

Definition 1. Interval
Let $[S\_T, \leq]$ be a partially ordered set, and let a, b be any two elements of S_T such that $a \leq b$. The set $\{x | a \leq x \leq b\}$ is called an interval.

Definition 2. Generalized *n*-ary relation
A generalized n-ary relation $R_n^G(\tau^1, \ldots, \tau^n)$, $n \geq 2$, is a permutation among n intervals, τ^i, $i = 1, \ldots, n$ which reside on an axis L with an origin o, R_n^G satisfies one of the condition s in Table 1.

The relation is represented by the corresponding name and symbol. The operands of the relations, $\tau^i, i = 1, \ldots, n$, are either the projections of the bounding volumes of the physical objects on an axis (spatial domain) or time span of a certain event (temporal domain).

The generalized *n*-ary relations are shown in Fig. 4, where $\tau_\triangle^i$ represents the inter-interval delay between interval i and i+1. Same relations can be used either in space or time domains, since the one-dimensional spatial axis is conceptually equivalent to the time axis, which is one-dimensional by definition. The only difference between spatial and temporal *n*-ary operations is that they apply to different domains. In the spatial domain, operands represent the physical position of the objects, whereas in the temporal case they represent the duration of a certain phenomenon. Such generality allows a formal representation of both spatial and temporal events by the seven fundamental *n*-ary relations shown in Fig. 4. Constraints associated with each relation that must be satisfied to uniquely define the corresponding relation are shown in Table 1. The aggregate duration (τ) of each relation is also listed.

3.2 Symbols and definitions

Before introducing the object-oriented model for spatial and temporal events, we first describe constructs of a language for presenting the model. Similar constructs have been introduced and followed in [2, 11, 12]. Here, we provide only the syntax for the language, the semantics are introduced in the event definitions. We first need to introduce the following symbols and definitions in order to define the syntax of the language:

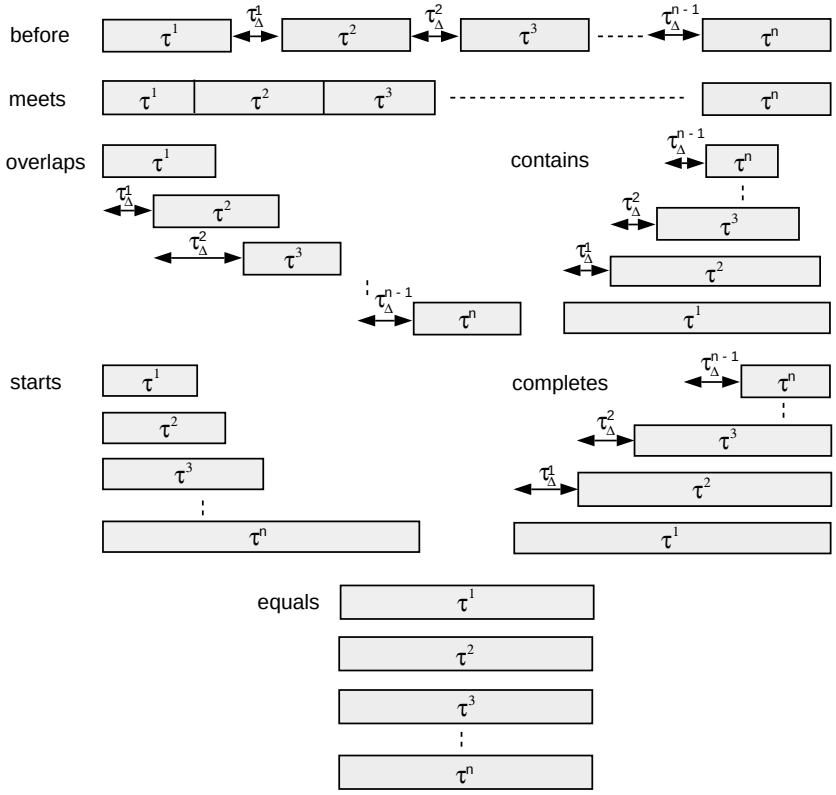

Fig. 4. n-ary relations

– a set DOM of domain symbols;
– a countable collection *VAR* of variable symbols;
– a collection FN of function name symbols;
– a collection R_n^G of *n*-ary predicate symbols;
– a collection EN of event name symbols;
– a set of Types = $\{object, interval\}$;
– a set of terms, including DOM, VAR, $FN(t_1, \ldots, t_n)$, $EN(t_1, \ldots, t_n)$, where $t_1, \ldots, t_n$ are terms;
– the set of closed terms, being those terms in which variable symbols do not appear;
– atomic formula, being of the form $t_1 = t_2$, $t_1 < t_2$, $t_1 > t_2$, $t_1 \in t_2$ (for terms t_1 and t_2), $\top$, or $R_n^G(t_1, \ldots, t_n)$ for terms $t_1, \ldots, t_n$ and *n*-ary predicate symbol R_n^G;
– the set of formulae, being the smallest set which includes all the atomic formula and also $\neg\alpha$, $\alpha \wedge \beta$, $\alpha \vee \beta$, $\forall\, x\alpha$, $\exists x\alpha$, and $\alpha \rightarrow \beta$ for any formulae α and β;
– the usual concepts of the free variables, bound variables, and substitution.

3.3 Spatial events

The information provided by the bounding volumes of objects in a frame can be used to describe more meaningful semantic information present in a frame. As it provides the most fundamental information about a frame, such as the locations of individual objects, it can be used to construct higher level content in the frame. Such detailed information contents in a single frame can be termed as *spatial events*.

For example, *presiding* a meeting attaches a meaning to some spatial area in a scene. For this event, a person in a frame needs to be identified such that he/she is either standing or sitting on a chair in the center or front of a meeting room. Similarly, a person may be sitting on a chair or some physical object. In this case, we have a conceptual spatial object 'sitting' with attributes 'a physical object which sits' and 'a physical object being sat on', and they are related by the 'sitting' relationship.

In order to express such events in a precise manner, we now present a formal definition of a spatial event based on the spatial operations discussed in the previous section.

Definition 3. *Spatial Event.*
A spatial event $\Phi_s(so)$ can be defined as an assertion (τ, Θ_s, μ, η), where

– *Φ_s is the name of the class of event;*
– *so is a tuple (α_1,α_2,. . .,α_k), where α_i is a variable representing a physical object in the domain;*
– *τ is the collection of projections of α_is on x-, y-, z-axes, $\tau = \{\tau_x^{\alpha_i}, \tau_y^{\alpha_i}, \tau_z^{\alpha_i}\}$, $1 \leq i \leq m$;*
– *Θ_s is a spatial assertion which specifies the spatial relationships among τs using n-ary operators. Formally,*

$$\Theta_s = R_1(\tau_1^{1_1}, \ldots, \tau_1^{1_{n_1}}) \Diamond_1 R_2(\tau_2^{2_1}, \ldots, \tau_2^{2_{n_2}}) \Diamond_2 \ldots \Diamond_{m-1} R_m(\tau_m^{m_1}, \ldots, \tau_m^{m_{n_m}}), \quad (3)$$

where R_j, $j = 1, \ldots, m$ is a generalized n-ary relation, $\Diamond_k$, $k = 1, \ldots, m-1$ is one of the logical operators ($\wedge$ or $\vee$) and $\tau_j^{j_i}$ is the projection of object j_i in relation j on x-, y-, or z-axis. A term $R_i(\tau_i^{i_1}, \ldots, \tau_i^{i_{n_i}})$ may be substituted by $\Phi_s^i(so_i)$, where Φ_s^i is the name of a spatial event, and $so_i \subseteq so$;
– *μ is the frame number of the spatial event;*

– *η is the duration of the spatial event and is default to one frame.*

Each spatial event instance is stored in the database in the following format:
(TAG, $clip\#$, $segment\#$, $class$, oid, $participating\_physical\_object$($oidlist$), $component\_events$, $starting\_frames$, $duration$($in\ frames$)).

TAG is used by the identification algorithm given later; $class$ represents Θ_s; oid is the system-assigned object id of the spatial event; $participating\_physical\_object$($oidlist$) stores the object ids of α_i, $1 \leq i \leq k$, $component\_events$ represents $\Phi_s^i(so_i)$. Other terms are self-explanatory.

An object's projection on an axis may appear in more than one relation (R_j). Note that the definition allows more complex spatial events to be constructed by relating several spatial events using logical operators. Also, the separation of a physical object and its projections is made. The interval (τ) for the result of an *n*-ary (R_i) as well as of a logical operation, is the aggregate interval, i.e., the spatial interval between the smallest starting coordinate and the largest ending coordinate of the objects involved.

As an example of a spatial event, consider a player *holding* the ball in a basketball game. To simplify the characterization of this situation, we assume, when the bounding rectangles of the objects *player* and *ball* are in contact with each other in a frame, the event "player holding the ball" is asserted. This is characterized by a set of six *n*-ary relations between ${\tau_x}^p$ (${\tau_y}^p$), the projection of the bounding rectangular associated with object player p on the x (y-)-axis and ${\tau_x}^b$ (${\tau_y}^b$), which is the projection of the bounding rectangular associated with the object ball on the x (y-)-axis. Their relation is as follows:

$\exists p \in player, \exists b \in ball,$

$$\begin{aligned} E^s_{hold}(p,b) &= (\tau^1, \Theta^1_s, \mu^1, \eta^1) \\ &= ((\tau^p_x, \tau^p_y, \tau^b_x, \tau^b_y), (M.O.C.S.CO.E(\tau^p_x, \tau^b_x) \\ &\quad \wedge (M.O.C.S.CO.E(\tau^p_y, \tau^b_y)), \mu^1, 1). \end{aligned} \tag{4}$$

If the specified condition is satisfied for a specific frame, the event function E_s is said to be valid in that frame. Note that $R_1.R_2.\ldots.R_m(\tau_1, \ldots, \tau_n) = R_1(\tau_1, \ldots, \tau_n) \vee R_2(\tau_1, \ldots, \tau_n) \vee \ldots R_m(\tau_1, \ldots, \tau_n)$. In this example, $player$, $ball \in DOM$; p, b, τs $\in VAR$; M, O, C, S, CO, $E \in R^G_n$; $E^s_{hold} \in EN$; and $player$ is of $object$ type, while τ^p_x is of $interval$ type.

As another example, the expression for "*a is to the left of b* "(in the observer-centered view) is

$\exists a, b \in physical\_object,$

$$\begin{aligned} E^s_{a\_left\_b}(a,b) &= (\tau^2, \Theta^2_s, \mu^2, \eta^2) \\ &= ((\tau^a_x, \tau^b_x), B.M.O(\tau^a_x, \tau^b_x), \mu^2, 1). \end{aligned} \tag{5}$$

Spatial events can be used as the low-level (fine-grain) indexing mechanisms for video data where information contents at the frame-level are generated. Modeling more complex information contents, such as *gloomy weather*, that can be extracted via image/vision-processing techniques is a more challenging problem and may require color-based content-modeling technique [14].

Algorithm SE-Identification: identification of a spatial event within a scene

Spatial-Event-Identification(ε,α_1,. . .,α_k)
There is a set of intervals I_i (maybe an empty set) for each α_i in which the physical object represented by α_i appears.
Perform Intersection($I_1, I_2, \ldots, I_k$) as follows:
  Intersection($I_1, I_2, \ldots, I_k$)
    = Intersection(Intersection($I_1, \ldots, I_{k-1}$),I_k)
  $I_1 = \{I^1_1, \ldots, I^u_1\}$
  $I_2 = \{I^1_2, \ldots, I^v_2\}$
  Intersection(I_1, I_2) = $\cup_{i=1}^{u}{}_{j=1}^{v}$ intersection(I^i_1, I^j_2)
Perform concatenation on I_r = Intersection($I_1, I_2, \ldots, I_k$) so that the member of I_r are related by before relation.
for each member I^i_r of I_r
  do for each sampled frame f_s
    if assertion associated with ε is true in f_s
      then an instance ϵ of $\varepsilon(\alpha_1, \ldots, \alpha_k)$ is identified
        Perform concatenation on ϵ to form a simple temporal event if necessary and record corresponding information.

To facilitate identification of spatial events, union and intersection of two intervals, within the domain of video sequence consisting of frames, are defined as follows:

$$union(I_1, I_2) = \{x | x \in I_1 \vee x \in I_2\},$$
$$intersection(I_1, I_2) = \{x | x \in I_1 \wedge x \in I_2\}.$$

Similarly, the union and intersection of *n* intervals can be defined recursively as follows:

$$union(I_1, I_2, \ldots, I_n) = union(union(I_1, I_2, \ldots, I_{n-1}), I_n),$$
$$intersection(I_1, I_2, \ldots, I_n) = intersection(intersection(I_1, I_2, \ldots, I_{n-1}), I_n).$$

We now present an algorithm (Algorithm SE-Identification) for identifying a spatial event from a segment. We assume that low-level processing as proposed in Sect. 2.2 has been performed and stored in the data structure given earlier.

As an example of this algorithm, consider the hypothetical segment consisting of three players and a basketball, as shown in Fig. 5, Suppose the event $E^s_{hold}(player1, ball)$ needs to be identified. The above algorithm first finds the intersection of the existence intervals between $player1$ and $ball$, i.e., $I_{P_1} = \{I^1_{P_1}\}$, $I_{ball} = \{I^1_{ball}\}$, and $I_r = Intersection(I_{P_1}, I_{ball}) = intersection(I^1_{P_1}, I^1_{ball})$. Since the result I_r contains only one interval, no *concatenation* of intervals is needed. Next, for each sampled frame existing in I_r, we check whether or not it satisfies the assertion given earlier. If it does, an instance of $E^s_{hold}(player1, ball)$ is identified. Many such instances may be present in I_r. In this case, a continuous sequence of instances correspond to a simple temporal event. It is assumed that all the frames in interval I_r' satisfy the assertion. Here I_r' corresponds to a simple temporal event $E^t_{hold}(player1, ball)$. Similarly, to identify spatial event $E^s_{hold}(player2, ball)$, intersection on existence intervals of player2 ($I_{P_2} = \{I^1_{P_2}, I^2_{P_2}\}$) and ball $I_{ball} = \{I^1_{ball}\}$ is performed. In other words, we compute $Intersection(I_{P_2}, I_{ball}) = intersection(I^1_{P_2}, I^1_{ball}) \cup intersection(I^2_{P_2}, I^1_{ball})$. Suppose the result consists of two intervals, I_{r_1} and I_{r_2}, related by *before* relation as shown in the figure. For each such interval, identification and concatenation procedures are performed. This results in an interval I_{r_2}' that corresponds to the specified event of interest $E^t_{hold}(player2, ball)$.

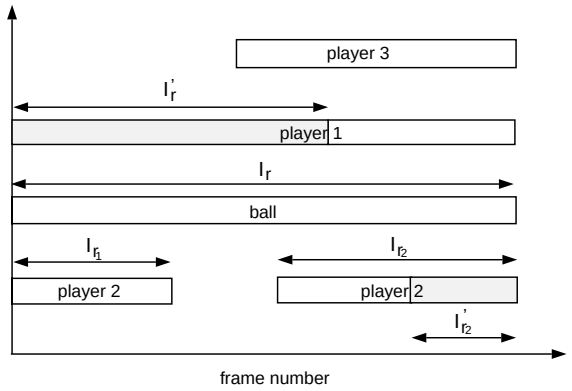

Fig. 5. Example of algorithm SE-Identification

It is important to note that, if a spatial event *persists* over a number of contiguous frames it can be considered as a simple temporal event. Not all persistent spatial events can be meaningfully transformed into temporal events, as will be seen in the next section.

3.4 Temporal events

The next level of video data modeling involves the temporal dimension. Temporal modeling of a video clip is important for users to ultimately construct complex views or to describe events in a clip. Events can be expressed by interpreting collective behavior of physical objects. In a simplistic manner, the behavior can be described by observing the total (or partial) duration during which an object appears in a given video clip. Its relative movement with respect to other objects over the sequence of frames in which it appears is also observed. For example, occurrence of a *slam-dunk* in a sports video clip can be an event in a user's specified query. Modeling of this event requires occurrence of at least two temporal *sub-events* which include tracking the motions of the player involved in the slam-dunk and of the ball in a careful manner, especially when the ball approaches the hoop. The overall process of composing a slam-dunk event requires a priori specification of multiple temporal sub-events. It is noted that a simple temporal event can be expressed formally as a logical expression consisting of various spatial events that span a number of frames. Subsequently, more complex temporal events can be defined recursively in terms of other temporal events related by the *n*-ary relations. We now formally give the definition of temporal events as follows.

Definition 4. Temporal event (composite)
A temporal event $\Phi_t(so)$ can be defined as an assertion (ϕ,Θ_t,η), where

- *Φ_t is the name of the class of event;*
- *so is a tuple (α_1,α_2,...,α_k), where α_i is a variable representing a physical object in the domain;*
- *ϕ is a set of temporal events $\{\phi_i(so_i)\}$, and $so_i \subseteq so$;*
- *Θ_t is a temporal assertion which specifies the temporal relationships among members of ϕ using n-ary operators. Formally,*

$$\Theta_t = R_1(\tau_1^{1_1}, \ldots, \tau_1^{1_{n_1}}) \Diamond_1 R_2(\tau_2^{2_1}, \ldots, \tau_2^{2_{n_2}}) \Diamond_2 \ldots \Diamond_{m-1} R_m(\tau_m^{m_1}, \ldots, \tau_m^{m_{n_m}}), \quad (6)$$

where R_j, $j = 1, \ldots, m$ is a generalized n-ary relation, $\Diamond_k$, $k = 1, \ldots, m-1$ is one of the logical operators ($\wedge$ or $\vee$) and $\tau_j^{j_i}$ is the duration for the j_ith temporal event (a member of ϕ) in relation i. A term $R_i(\tau_i^{i_1}, \ldots, \tau_i^{i_{n_i}})$ may be substituted by $\Phi_t^i(so_i)$, where Φ_t^i is the name of a temporal event, and $so_i \subseteq so$. $\Phi_t^i(so_i)$ may or may not be a member of ϕ;

- *μ is the starting frame of the temporal event;*
- *η is the aggregate duration (in frames) of Θ_t.*

Each temporal event instance can be represented as follows:
(TAG, $clip\#$, $segment\#$, $class$, oid, $participating\_physical\_object(oid\ list)$, $participating\_events$, $starting\_frames$, $duration(in frames)$).
The meaning of the individual field is similar to those in the data structure of spatial events.

At the lowest level, simple temporal events are first constructed from spatial events using the above definition, with a condition that the n-ary operators are of type *meets* and all operands of a certain operation belong to the same spatial event. This allows us to represent the "persistence" of a specified spatial event over a sequence of frame. This also corresponds to a simple temporal event that is valid for the corresponding range of frames with duration ℓ_t. If the event starts at frame #α and ends at frame #β, then $\ell_t = \beta - \alpha + 1$. At higher levels where operands themselves are also temporal events (in the case of composite temporal events), the duration of an n-ary/logical operator is the aggregate duration of its operators τ^js, that are associated with corresponding temporal events.

An important property of temporal events is *concatenability* [2] , which is the foundation for constructing a simple temporal event. Before presenting this property, we introduce the notion of predicates. A predicate (event) $P(a_1, \ldots, a_m)(e)$ if true during an interval i can be represented as $P(a_1, \ldots, a_m, i)(e(i))$. Concatenability means that, if an event is true in intervals I and J, and I *meets* J, then the event is true in an interval $K = M(I, J)$. Formally, $\forall i, j\ e(i) \wedge e(j) \wedge M(i, j) \rightarrow e(M(i, j))$. An example of concatenability is that if person A is walking during intervals I and J, and I meets J, then A is walking during an interval K=$Meets(I, J)$. On the other hand, if person A performs *slam-dunk* once in each interval I and J, and I meets J, then A is not performing slam-dunk once in the interval K = $M(I, J)$, instead, A performs slam-dunk twice.

An example for a temporal event consisting of two spatial events is "passing of a ball between two players". This event can be characterized by relating two similar spatial events $E^s_{hold}(u, b)$, "holding of the ball by player u" and $E^s_{hold}(v, b)$, "holding of the ball by player v", which can be described as in the previous section.

A pass event can be composed of these events joined with two predicates. The first predicate is that both $E^s_{hold}(u, b)$ and $E^s_{hold}(v, b)$ should persist for a finite duration. In other words, the ball should be in contact with each player for a period of time for each event to be considered "holding". The second predicate specifies that these events should follow each other with a certain delay bounded by some specified value. The first predicate regarding persistence can be

formally described as a temporal event that uses a *meets* operation with occurrences of $E^s_{hold}(u,b)$ or $E^s_{hold}(v,b)$ over ℓ_t number of frames as its operands:

$\exists p \in player, \exists b \in ball,$

$$\begin{aligned} E^t_{hold}(p,b) &= Persistent(E^s_{hold}(p,b), \mu_t, l_t) \\ &\equiv \forall l'_t, IN(l'_t, l_t) \rightarrow Persistent(E^s_{hold}(p,b), l'_t) \\ &\equiv ((E^s_{hold}{}^{(1)}(p,b), \ldots, E^s_{hold}{}^{(l_t)}(p,b)), \\ &\quad M(l^{(1)}, \ldots, l^{(l_t)}), \mu_t, l_t). \end{aligned} \quad (7)$$

Finally, we can express the *pass* event using *before* n-ary operation between $E^t_{hold}(u,b)$ and $E^t_{hold}(v,b)$ as:

$\exists u, v \in player, \exists b \in ball,$

$$\begin{aligned} E^t_{pass}(u,v,b) = ((&E^t_{hold}(u,b), E^t_{hold}(v,b)), \\ &B(l_1 : \tau^1_\triangle, l_2 : \tau^2_\triangle), \mu, \eta). \end{aligned} \quad (8)$$

Here $\tau^1_\triangle$ and $\tau^2_\triangle$ are the inter-interval offsets between the temporal events.

As discussed earlier, temporal events can be specified in a more general way by assigning ranges to interval lengths and inter-interval offsets instead of exact values. In our example, for instance, one can allow l_1 and l_2 to vary between 15 and 90 frames and $\tau^1_\triangle$ between 6 and 45 frames for the temporal event to be considered as a pass (actual time depends on the frame rate of the video). This means that, at 30 frames/s, we require a holding persist for 0.5–3 s and the period when the ball is in the air be between 0.2 and 1.5 s. For example, the event $((E^t_{hold}(X,b), E^t_{hold}(Y,b)), B(1.0 : 0.3, 2.0 : 0), \mu_1, \eta_1)$ which specifies that "player X holds the ball for 1.0 s and after 0.3 s player Y holds the ball for 2.0 s" is considered to be a pass event whereas $((E^t_{hold}(Z,b), E^t_{hold}(W,b)), B(1.5 : 2.5, 2.0 : 0), \mu_2, \eta_2)$ is not valid, since $\tau^1_\triangle$ (which equals 2.5 s) does not lie within the specified range. Note that inter-interval offset $\tau_\triangle$ is always zero for the last operand of a relation.

The pass event example can be recursively used to describe more complex temporal events such as "two successive passes". The expression for such an event is as follows:

$\exists u, v, w \in player, \exists b \in ball,$

$$\begin{aligned} E^t_{2-pass}(u,v,w,b) = ((&E^t_{pass}(u,v,b), E^t_{pass}(v,w,b)), \\ &B(l_1 : \tau^1_\triangle, l_2 : \tau^2_\triangle), \mu, dur). \end{aligned} \quad (9)$$

Figure 6 summarizes the whole process of event specification for the pass example. Note that the spatial events E^ss are as described in Sect. 3.3.

An algorithm (Algorithm TE-Identification) for identifying a temporal event within a video segment is given next. This algorithm can also be used to identify a temporal event within a clip, except in this case the search scope is the whole video clip instead of a segment.

As an example of this Algorithm, suppose four instances of *holding a ball* event exist in a segment as shown in Fig. 7. To identify the temporal event $E^t_{pass}(player1, player2, ball)$, we need to find an instance of event $E^t_{hold}(player1, ball)$ and $E^t_{hold}(player2, ball)$. Subsequently, we need to check whether these events are related by $before$ with an inter-interval delay less than a specified value. The algorithm for this example works as follows. We have $\Theta_t = P_1 = t_{11} = B(l_1 : \tau^1_\triangle, l_2 : \tau^2_\triangle)$. First, unmark TAG field of all four temporal events in the segment. Assume that the events are sorted in an ascending order of starting frames. Then, the algorithm finds an unmarked event corresponding to $E^t_{hold}(player1, ball)$ (e_1); in this case, it is event *A*. Next, it finds an unmarked event corresponding to $E^t_{hold}(player2, ball)$ (e_2), such that this event is related to event *A* with the $before$ relation, and it has the earliest starting frame number among all instances of $E^t_{hold}(player2, ball)$ existing in the segment; let it be event *B*. Then the condition for a *pass* event between player 1 and player 2 is tested. The duration of this event can be calculated. The process is repeated and the algorithm finds an unmarked event *C* corresponding to e_1, the first component event of the pass event. However, e_2, which is the second component event of the pass event, cannot be found. Therefore, only one instance of pass is found, which is designated as $t^{(1)}_{11}$. Finally, it sets $P^{(1)}_1 = t^{(1)}_{11}$, and finds duration($P^{(1)}_1$) = duration($t^{(1)}_{11}$).

In summary, the proposed framework of generalized n-ary operators and the encoded information of the model

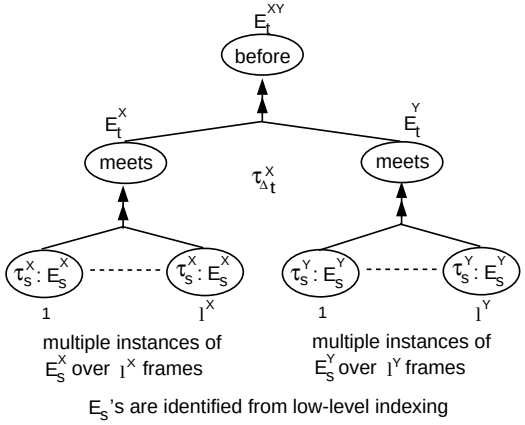

Fig. 6. An example of composition hierarchy of 'pass'

Algorithm TE-Identification: identification a temporal event within a scene

Temporal-Event-Identification(Θ_t)
Input: $\Theta_t = P_1 + P_2 + \ldots + P_k$, where $P_i = t_{i1}t_{i2}\ldots t_{ij}$, and each t_{ij} is an *n*-ary expression
for each P_i
  do if P_i is an *n*-ary operator
    then for each t_{ij}
      do Unmark all temporal events in the segment
        repeat
          Find e_1 (unmarked) corresponding to τ_1 of t_{ij}
          Search all $e's$ $\pi(e) \geq \pi(e_1)$ to find events corresponding to $\tau_2 \ldots \tau_n$
          if search is successful
            then create a sub-event instance $t^{(l)}_{ij}$ of t_{ij}
              record its component events corresponding to $\tau_1, \ldots, \tau_n$
              calculate its aggregate interval using Table 1
        until no unmarked events corresponding to e_1 is found
      $P^{(q)}_i = t^{(l)}_{i1}t^{(m)}_{i2}\ldots t^{(n)}_{ij}$,
      $1 \leq l \leq \#(t_{i1}),\ 1 \leq m \leq \#(t_{i2}),\ \ldots,\ 1 \leq n \leq \#(t_{ij})$
      $\tau(P^{(q)}_i)$ = minimum interval containing all τ s of $t^{(l)}_{i1}t^{(m)}_{i2}\ldots t^{(n)}_{ij}$
    else Find the corresponding event in the collection for temporal events

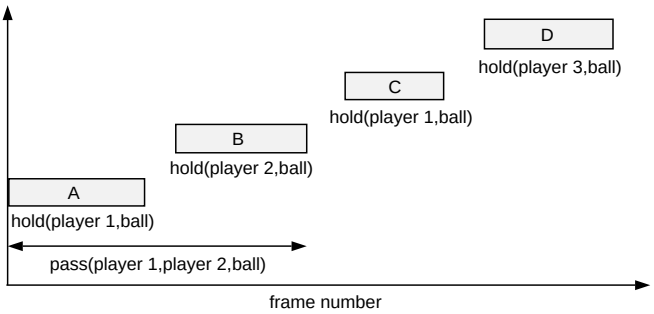

Fig. 7. Example of algorithm TE-Identification

can provide a mechanism for constructing and characterizing events. More importantly, we can develop hierarchical relationships among different types of complex events that inherit properties of simple events. Such a hierarchical structure can help in indexing and searching events of interest.

4 Object-oriented modeling of video data

Considerable semantic heterogeneity may exist among users of video data due to the differences in their pre-conceived interpretation or intended use of the video information. Semantic-based integration of different views may be required for a large number of users. Data management and efficient query processing, in the process of such integration, is a complex and challenging problem. Conventional data-modeling techniques lack the ability of managing complex events of video data and supporting heterogeneous views of the data. For example, the relational model has a drawback of losing semantics, which can cause erroneous interpretation of views and events.

The object-oriented technology, on the other hand, can provide a powerful paradigm to meet the requirements of our semantic modeling and management of complex video data. Its data and computational encapsulation features offer elegant data-modeling capabilities at various levels of information granularity in a video database system. The paradigm can allow users to combine multiple views of the data into a single comprehensive view. The basic concept that data is associated with procedures manipulating it is especially appealing in modeling video data, where the raw data needs to be processed and spatio-temporal contents of the information needs to be combined using rather complex logic. Object-oriented modeling allows such complexity to be independently managed and linked together via communication among objects using messages. The *class* concept in object-oriented paradigm is especially suitable for semantic-based grouping of events.

In this section, we discuss a modeling process of multiple and heterogeneous views of users in an object-oriented environment and describe how multiple events in video data can be integrated into a single framework. The fundamental premise here is that we can establish a correspondence between hierarchical relationship of video events (e.g., Fig. 6) discussed in the previous section and different classes of objects using various object-oriented abstractions. In object-oriented environment, objects, classes, and meta-classes can be defined recursively at arbitrary levels. By embedding an event in a class, not only necessary information of an event can be recorded as attributes, but most importantly, the identification of an event can be treated as method(s) of a class. Also, if an attribute is another event (class), the current class can invoke the methods of that class by sending message(s) to it.

Table 2. Generic spatial event

CLASS *Generic (Persistent) Spatial Event*
ATTRIBUTE
object_identifier (oid)
event_definition_expression* /* class method */
BOOLEAN TAG, $IS\_TEMPORAL$
clip#, segment#, starting_frame#, duration
oid_list_of_participating_object*
oid_list_of_component_event*
METHOD
identification_procedure() /* class method */
(
For each tuple-collection representation of a video clip in the video DB (or a group of clips)
Perform Algorithm 1 to identify the (persistent) spatial events
)
return_single_value_attribute(attribute_name)
return_participating_object_id()
return_component_event_id()

4.1 From events to classes

For mapping events to classes, events can be categorized into two generic classes. A *generic spatial event* class and a *generic temporal event* class are defined for spatial events and temporal events, respectively. These two classes are called *perspectives* [15, 23] since they do not have instances of their own, rather they are used for generating new classes.

In Table 2, we provide a generic template for declaring a *spatial* class. Along with the attributes, such as object id, pointer-to-object definition, object id list of participating physical objects, etc., the main component of the class is the *class method* to identify the given spatial event. The actual events identified are the instances of a spatial event class. Since a spatial event definition has parameters, the identification procedure is performed for each combination of parameters. The system is updated with new instances and event types as they are identified, during the archiving/retrieval process. Note that the identification procedure is only associated with the class definition, i.e., it is a class method, not a method of an instance of a class. Additionally, the *duration* attribute is used to record the *persistence* of a spatial event. As a result, it is not necessary to record a lot of redundant information. A spatial event persisting for a number of frames can be called a simple temporal event if it is meaningful to do so. That is, not all persistent spatial events represent meaningful temporal events. The users have to make the decision if a persistent spatial event is a simple temporal event. Attributes TAG and $IS\_TEMPORAL$ are used for identification procedures and for denoting temporalness, respectively. The method $return\_single\_value\_attribute$ is used to return any attribute with a single value, e.g., clip#.

Table 3 provides a template for declaring a *temporal* class. For an instance of this class, a component can be

Table 3. Generic temporal event

CLASS *Generic Temporal Event*
ATTRIBUTE
object_identifier (oid) event_definition_expression* /* class method */ BOOLEAN Spatial_Component, TAG clip#, segment#, starting_frame#, duration oid_list_of_participating_object* oid_list_of_component_event*
METHOD
identification_procedure() /* class method */ (For each clip in the video DB (or a group of clips) Perform Algorithm 2 to identify the temporal events) return_single_value_attribute(attribute_name) return_component_event_id() return_participating_object_id()

an instance of another temporal event class or may be an instance of a persistent spatial class with the property that instances of the spatial class are related with the temporal relation $meets$. The structure of the temporal class is similar to that of *generic spatial* class, except that the identification procedure is different.

4.2 Abstraction of video data in object-oriented paradigm

We now consider spatial and temporal events as object classes to show how existing object-oriented abstractions can be used to define new classes and how inheritance can be used to construct complex views. An important aspect of these abstractions is that they allow grouping and merging of information entities which may not have any temporal or spatial relationship among them, but some general semantics. This is not possible, otherwise, with the use of simple *n*-ary spatial or temporal relations.

In Sects. 3.3 and 3.4, spatial/temporal events are constructed using *n*-ary relations and 'AND' and 'OR' logical operators. To represent these relations in the object-oriented environment, one possible approach is the mapping shown in Fig. 8. An *n*-ary relation among *n* events can be modeled using aggregation abstraction (*n*-ary), where the superclass is the result of the *n*-ary relation, and the subclasses are related by one of the *n*-ary relation. The logical operators 'AND' and 'OR' are modeled through aggregation (IS-PART-OF) and specialization/generalization (IS-A), respectively. The graphical representations of the mapping is shown in Fig. 9. Note that an event definition expression is defined either through product of sum (POS) or sum of product (SOP), where each term in the SOP or POS is, in turn, a POS or SOP. However, in Algorithm 2 presented in Sect. 3.4, the event definition expression is assumed to be a SOP expression, since a POS expression can be translated into a SOP expression. For the composition of a spatial class, the parent node represents the result of an *n*-ary spatial operation and all the children nodes are also the spatial classes. Similarly, for the temporal classes, the parent node represents an *n*-ary temporal operation and the child nodes are either spatial or temporal classes.

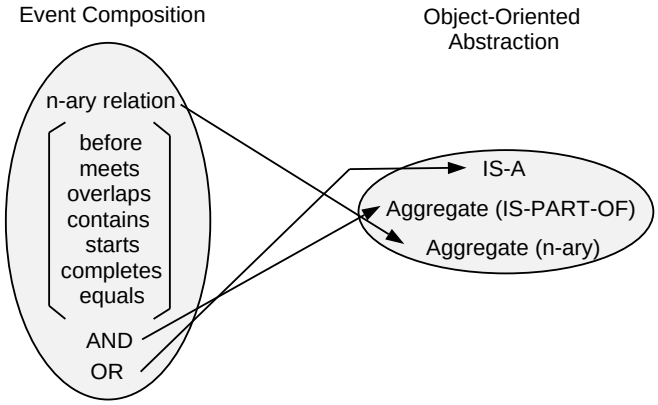

Fig. 8. Mapping between event composition constructors and object-oriented abstractions(1)

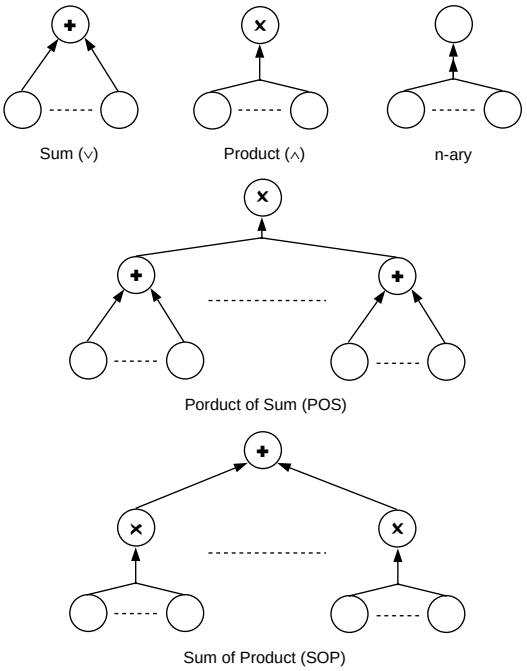

Fig. 9. Mapping between event composition constructors and object-oriented abstractions(2)

For developing an abstraction of a video database, we can use either a top-down (specialization) or bottom-up (generalization) approach [16]. Irrespective of the abstraction used, we can utilize three types of semantic relationships between classes. These include, generalization (IS-A), aggregation (IS-PART-OF), and aggregation (*n*-ary), and are used together, depending on the grouping requirements of users. A four-level object-oriented abstraction of the framework is shown in Fig. 10. Level one consists of identified physical objects. Spatial objects (events), identified from spatial relationships among objects at level one, constitute level-two abstraction. Temporal objects (simple or complex) constructed using *n*-ary relations (aggregate) form level-three abstraction. Objects generated using generalization (IS-A) or aggregation (IS-PART-OF) constitute level-four abstraction. Temporal inheritance property exists among subclasses and superclass of a generalization abstraction. However, there are some rules that must be observed while using these abstrac-

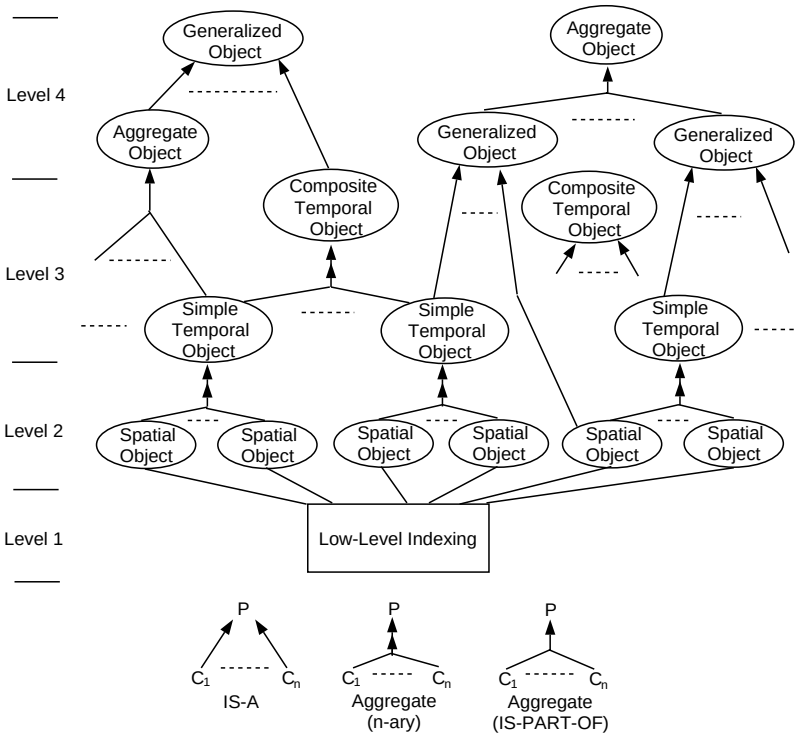

Fig. 10. Object-oriented abstraction of video databases

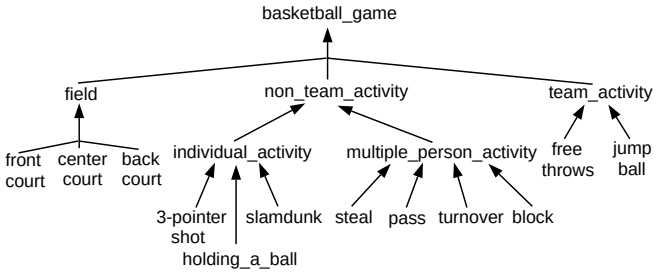

Fig. 11. An object-oriented view of basketball events

tions for modeling of video data. These rules are discussed later in the section. We proceed by defining the view hierarchy (a video database abstraction from users' point of view) and present associated constraints.

A view hierarchy H_V consists of the following part : H_G, H_A, and H_N;

- H_G is the set of generalization abstraction axioms, each of the form : 'E_1 *IS-A* E_2' iff $\forall x, E_1(x) \rightarrow E_2(x)$, E_1, $E_2 \in Node(H)$, and $\rightarrow$ means 'imply'.
- H_A is the set of aggregation (IS-PART-OF) abstraction axioms, each of the form '$E_1, \ldots, E_n$ *IS-PART-OF* E_0' iff $\forall x\ E_0(x) \rightarrow E_1(f_1(x)) \wedge E_2(f_2(x)) \wedge \ldots E_n(f_n(x))$, $E_1, \ldots, E_n \in Node(H)$, and f_i is a function that specifies E_i being part of E_0 in some way. E_1 through E_n are called direct components of E_0.
- H_N is the set of aggregation (*n*-ary) abstraction, each of the form: $E_0 = R_n^G(E_1, \ldots, E_n)$ iff duration of (E_0) = aggregate duration of $R_n^G(l_1, \ldots, l_n)$ and $\forall x\ E_0(x) \rightarrow E_1(f_1(x)) \wedge E_2(f_2(x)) \wedge \ldots E_n(f_n(x))$, $E_1, \ldots, E_n \in Node(H)$, and f_i is a function denoting the i-th component of an *n*-ary relation.

The interconnection of H_G, H_A, and H_N are defined through the following constraints. Given a superclass C_{super}, a number of its subclasses C_{sub_i}, i = 1, ..., m, m a positive integer ≥ 1, and an abstraction $\mathscr{A}$, we list the following rules. Assume that, for a class, $\mathscr{RA}$ represents the set of abstractions it can participate as a subclass.

$\mathscr{A}$ = IS-A

1.1 If $\forall i$, C_{sub_i} is a spatial event, then C_{super} represents a *category* and is called a *derived spatial class*. C_{super}'s $\mathscr{RA}$ is equal to {IS-A, IS-PART-OF, *n*-ary (persistence)}.

1.2 If $\forall i$, C_{sub_i} is a temporal event, then C_{super} represents a *category* and is called a *derived temporal class*. C_{super} has $\mathscr{RA}$ equal to {IS-A, IS-PART-OF, *n*-ary}.

1.3 If $\exists i, j, k$, $i \neq j \neq k$, $1 \leq i, j, k \leq m$, such that C_{sub_i} is a spatial event, C_{sub_j} is a temporal event, C_{sub_k} is a non-spatio-temporal class, then C_{super} represents a generalization and has $\mathscr{RA}$ = {IS-A,IS-PART-OF}.

1.4 If $\forall i$, C_{sub_i} is either a spatial or temporal event, and #($C_{sub_i} \in spatial$) ≥ 1, #($C_{sub_i} \in temporal$) ≥ 1, then C_{super} represents a generalization and $\mathscr{RA}$ is equal to {IS-A,IS-PART-OF}.

$\mathscr{A}$ = IS-PART-OF.

2.1 If $\forall i$, C_{sub_i} is a spatial event, then C_{super} is a composite spatial event, and it has $\mathscr{RA}$ = {IS-A, IS-PART-OF, *n*-ary (persistence)}.

2.2 If $\forall i$, C_{sub_i} is a temporal event, then C_{super} is an aggregation of temporal events where no temporal relation is specified among C_{sub_i}s. C_{super}'s $\mathscr{RA}$ is equal to {IS-A,IS-PART-OF}.

2.3 If $\exists i, j, k$, $i \neq j \neq k$, $1 \leq i, j, k \leq m$, such that C_{sub_i} is a spatial event, C_{sub_j} is a temporal event, C_{sub_k} is a non-spatio-temporal class, then C_{super} represents an

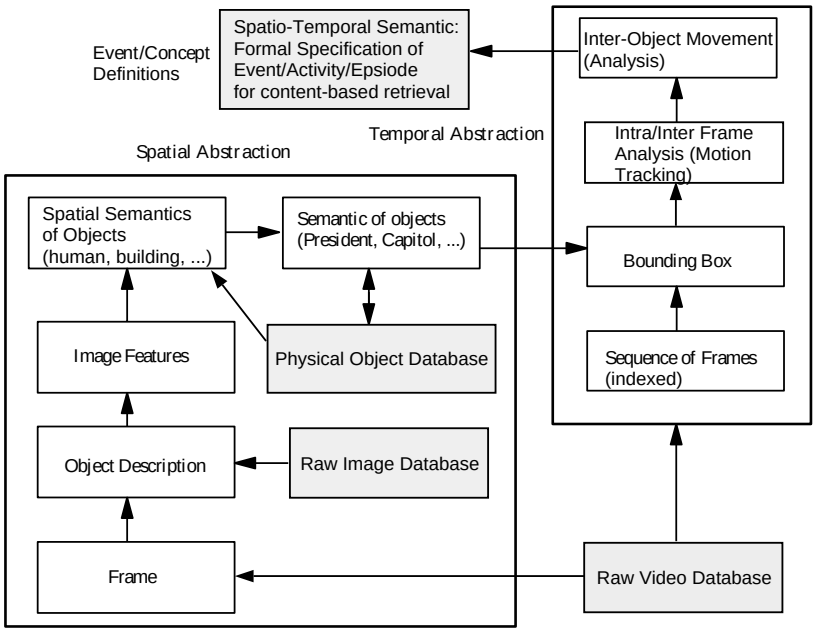

Fig. 12. System abstraction for the proposed image/video database

aggregation of heterogeneous events and its $\mathscr{PA}$ is {IS-A,IS-PART-OF}.

2.4 If $\forall i$, C_{sub_i} is either a spatial or temporal event, and #($C_{sub_i} \in spatial$) $\geq$ 1, #($C_{sub_i} \in temporal$) $\geq$ 1, then C_{super} represents an aggregation of spatio-temporal events and has $\mathscr{PA}$ = {IS-A,IS-PART-OF}.

$\mathscr{A}$ = *n*-ary.

3.1 If $\forall i$, C_{sub_i} is a spatial event, all C_{sub_i} are of the same class with the same parameters, and the *n*-ary relation is *meets*, then C_{super} is a simple temporal event and its $\mathscr{PA}$ = {IS-A,IS-PART-OF,*n*-ary}.

3.2 If $\forall i$, C_{sub_i} is a temporal event, then C_{super} is a composite temporal event and it has $\mathscr{PA}$ = {IS-A,IS-PART-OF,*n*-ary}

An example of video abstraction for a sports database is given in Fig. 11. Class *basketball_game* is the aggregation of three subclasses, namely, *field*, *team_activity*, and *non_team_activity*. The derived spatial class *field* is composed of three spatial subclasses, namely, *front court*, *center court*, and *back court*. All these subclasses may not be promoted, i.e., they may not be considered as simple temporal events if they persist. Classes *individual_activity* and *multiple_person_activity* are related to *non_team_activity* class by a IS-A relation. *team_activity* is a temporal event with two temporal subclasses *free throws* and *jump ball*. From a user's point of view, grouping of object classes can be purely based on shared semantics without any spatial or temporal relationship among them. An example is *individual_activity* class, which is a generalization of two temporal classes ($3-pointer\ shot$ and $slam-dunk$) and one spatial class ($holding\_a\_ball$). This class is neither temporal nor spatial in its characteristic. On the other hand, *multiple_person_activity* class is a derived temporal class, since all its children are temporal subclasses. It can be noticed that all the leaf nodes are spatial or temporal classes where objects operate on the low-level model in Sect. 2.

In the proposed model, *inheritance* in the IS-A relation is realized through a bottom-up approach called *generalization*. A new class (parent) is formed by extracting the common structure from existing classes (children). For aggregation abstraction, the parent class does not possess anything from its children classes.

4.3 System architecture and management of objects for content-based retrieval

Based on the discussion of the previous sections, we envision four levels of indexing of video databases to facilitate content-based query processing. The first level maintains low-level spatial/temporal information about salient physical objects using the proposed model in Sect. 2. At higher levels, the indexing is mainly based on user-specified semantics/contents. For example, at the second level, spatial events are constructed by processing the spatial information maintained by the low-level model. The third level maintains indices for temporal events (objects) using the information available at the second level and also temporal information at the same level through recursive formulations of complex events. The fourth level manages complex objects and provides indexing mechanism to groups of events with related contents, where those objects and class objects are generated through generalization and aggregation as shown in Fig. 10. Accordingly, a system architecture for a video database is shown in Fig. 12.

As mentioned earlier, the first-level indexing is based on a low-level model (such as VSDG), where the relevant information such as the identities of the salient objects of interest and their bounding volumes have been extracted from raw video data. This constitutes a challenging problem even for today's advanced computer vision technology. Although discussion of this problem is not the main theme of this paper, it's worthwhile to mention some issues related to the initial processing.

First, for easier and more efficient object recognition, physical objects should be grouped into classes (*Physical Object Database* in Fig. 12). This enables pre-defined object models to be used and simplifies the recognition problem through appropriate matching techniques. Since the identities of human objects, which are obviously of special interest in video databases, are determined by their faces, their recognition should be given a special treatment [25, 27]. The second important problem is to obtain information about the bounding volume for any recognized object, a process which can be carried out through well-established feature extraction algorithms and can be used in the later steps to construct the entire database. This processing corresponds to the *Bounding Box* module of the architecture shown in Fig. 12. The processed information is maintained via the low-level model and constitutes the first-level indexing. The second, third, and fourth level of indexing (as shown in Fig. 10) is maintained in the module labeled as *Event/Concept Definitions*.

From a system implementation point of view, there are three major components, namely, *spatial abstraction*, *temporal abstraction*, and *Event/Concept Definitions*, as shown in Fig. 12. In the spatial abstraction component, an image or a frame of a video clip is processed using the *object description* information to obtain features of the image. With the knowledge from *physical object database*, spatial semantics of objects in an image are identified. On the other hand, in the *temporal abstraction* component, *bounding box* information is obtained from the spatial abstraction component, which is then used for intra-/inter frame analysis (motion tracking) of a single object. The next step is inter-object movement analysis, i.e., identifying relative movement between objects. This information is then utilized by component *event/concept definition* to identify events. This component stores the definitions as well as the procedures of identifying events.

The logic of constructing events can be used to derive a content-based query language. A user can specify and store more events, as needed. New classes can also be formed based on the existing classes at lower/same level through *n*-ary operations and class inheritance. The proposed system should be able to support the following types of queries, where any clip, segment, or sub-clip (several segments) satisfying one or more of the following conditions may be returned as an answer to a query:

- appearance of physical objects.
- existence of spatial events.
- existence of user-defined conditions equivalent to spatial events.
- existence of temporal events.
- existence of user-defined conditions equivalent to temporal events.

Note, that these conditions can be logically combined to form more complex conditions.

Queries about the view structure may also be supported. Occasionally, the system may resort to processing of raw video data to identify objects that were not previously identified. We expect the proposed methodology can be used to implement a system with on-line capabilities for query processing.

5 Conclusion

We have presented a framework for semantic-based modeling of video data using generalized *n*-ary operators. The raw video data is processed to extract the spatial information about physical objects. This information can lead to the first-level indexing of spatial events that are expressed formally using spatial relations. Subsequently, higher level indexing of events involving temporal dimension is created using generalized temporal operators. The management of video events can be efficiently carried out using object-oriented technology, since it can provide an elegant paradigm for semantic-based modeling and grouping of information. A video & image database system architecture is then proposed, which consists of three major components to process & manage image & video data and spatio-temporal events.

References

1. Allen JF (1983) Maintaining knowledge about temporal intervals. Commun ACM 26: 832–843
2. Allen JF, Kautz HA, Pelavin RN, Tenenberg JD (1991) Reasoning about plans. Morgan Kaufmann, San Mateo, Calif.
3. Li JZ, Özsu MT, Szafron D (1997) Modeling of moving objects in a video database. Proceedings of IEEE International Conference on Multimedia Computing and Systems, Ottawa, Canada, June 1997, pp 336–343
4. Arman F, Depommier R, Hsu A, Chiu M-Y (1994) Content-based browsing of video sequences. Proc. of Second ACM International Conf. on Multimedia, San Francisco, Calif., pp 97–102
5. Arndt T, Chang S-K (1989) Image sequence compression by iconic indexing. IEEE VL '89 Workshop on Visual Languages, Rome, Italy, pp 177–182
6. Del Bimbo A, Vicario E, Zingoni D (1995) Symbolic description and visual querying of image sequences using spatio-temporal logic. IEEE Trans Knowl Data Eng 7: 609–622
7. Date CJ (1991) An introduction to database systems, vol 1, 5th Ed., Addison Wesley
8. Day YF, Dagtas S, Iino M, Khokhar A, Ghafoor A (1995) Object-oriented conceptual modeling of video data. Proc. of IEEE International Conference on Data Engineering '95, Taipei, Taiwan, pp 401–408
9. Dimitrova N, Golshani F (1994) $\mathscr{R}_X$ for semantic video database retrieval. Proceeding of the ACM Multimedia'94, San Francisco, Calif., pp 219–226
10. Flicker M, Sawhney H, Niblack W, Ashley J, Huang Q, Dom B, Gorkani M, Hafner J, Lee D, Petkovic D, Stelle D, Yankee P (1995) Query by image and video content: the QBIC system. IEEE Comput 28: 23–31
11. Gabbay DM, Hodkinson I, Reynolds M (1994) Temporal logic: mathematical foundations and computational aspects, vol 1, Clarendon Press, Oxford
12. Golshani F, Dimitrova N (1994) Retrieval and delivery of information in multimedia database systems. Inf Software Technol 36
13. Iino M, Day YF, Ghafoor A (1994) An object-oriented model for spatio-temporal synchronization of multimedia information. Proc. IEEE Int. Conf. on Multimedia Computing and Systems ICMCS '94, pp 110–119
14. Jain R, Hampapur A (1994) Metadata in video databases. ACM Sigmod Rec 23: 27–33
15. Korson T, McGregor JD (1990) Understanding object-oriented: a unifying paradigm. Commun ACM 33: 40–60
16. Khoshafian S (1993) Object-oriented databases. John Wiley & Sons, New York
17. Little TDC, Ghafoor A (193) Interval-based conceptual models for time-dependent multimedia data. IEEE Trans Knowl Data Eng 5: 551–563

18. Little TDC, Ahanger G, Folz RJ, Gibbon JF, Reeve FW, Schelleng DH, Venkatesh D (1993) A digital video-on-demand service supporting content-based queries. Proc. ACM Multimedia '93, Anaheim, Calif.
19. Nagasaka A, Tanaka Y (1991) Automatic video indexing and full video search for object appearances. In: 2nd Working Conference on Visual Database Systems, Budapest, Hungary, October 1991, IFIP WG 2.6, pp 119–133
20. Oomoto E, Tanaka K (1993) OVID: design and implementation of a video-object database system. IEEE Trans Knowl Data Eng 5: 629–643
21. Rowe LA, Boreczky J, Eads C (1994) Indexes for user access to large video databases. Proc. of IS&T/SPIE 1994 Int. Symp. on Elec. Imaging: Science and Technology, San Jose, Calif.
22. Smoliar SW, Zhang H (1994) Content-based video indexing and retrieval. IEEE Multimedia 1: 62–72
23. Stefik M, Bobron DG (1986) Object-oriented programming: themes and variations. AI Mag January 1986, pp 40–62
24. Swanberg D, Shu C-F, Jain R (1993) Knowledge guided parsing in video databases. Proc. SPIE 93, San Jose, Calif., pp 3-11–3-22
25. Turk M, Pentland A (1989) Eigenfaces for recognition. J Cogn Neurosci 3: 71–86
26. Weiss R, Duda A, Gifford DK (1995) Composition and search with a video algebra. IEEE Multimedia 2: 12–25
27. Wu JK, Narasimhalu AD (1994) Identifying faces using multiple retrievals. IEEE Multimedia 1: 27–38
28. Yoshitaka A, Kishida S, Hirakawa M, Ichikawa T (1994) Knowledge-assisted content-based retrieval for multimedia database. IEEE Multimedia 1: 12–21

YOUNG FRANCIS DAY is with the multimedia documentation program of Siemens Corporate Reserach (SCR) at Princeton, New Jersey, where he has been a member of technical staff since May, 1997. Dr. Day received his BS in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan in 1989, his MS in computer engineering from Pennsylvania State University (University Park) in 1991, and his PhD in electrical and computer engineering from Purdue University in 1996. Before joining SCR, he worked for the network management group of Alcatel Data Networks in Asburn, Virginia, from 1996 to 1997. His current research interests include interactive multimedia information systems, video data modeling, distributed/networked multimedia documentation systems based on SGML/XML, and multimedia docuemnt management systems. Dr. Day is a member of ACM, IEEE, and ASEE.

SERHAN DAGTAS graduated from Bilkent University, Turkey in 1991 with a B.S. degree in Electrical Engineering. He earned his M.S. and Ph.D degrees in Electrical and Computer Engineering from Purdue University, in 1994 and 1998, respectively. He is currently a Senior Member of Research Staff in Philips Research, USA. His research interests include multimedia data modeling, image and video retrieval, relational and object-oriented databases, image processing and computer vision. Dr. Dagtas was a Fulbright Scholar from 1992 to 1994.

ASHFAQ A. KHOKHAR received his B.S. in electrical engineering from the University of Engineering and Technology, Lahore, Pakistan, in 1985 and his M.S. in computer engineering from Syracuse University, in 1989. He received his Ph.D. in computer engineering from the University of Southern California, in 1993. After his Ph.D., he spent 2 years as a Research Assistant Professor in the Department of Computer Sciences and School of Electrical and Computer Engineering at Purdue University. He joined University of Delaware in 1995, where he is Assistant Professor in the Department of Electrical Engineering and Department of Computer and Information Sciences. His research interests include computational aspects of multimedia systems and image processing, parallel systems.

ARIF GHAFOOR received his B.Sc. in Electrical Enginerring from University of Engineering and Technology, Lahore, Pakistan in 1976. He recieved his M.S. and PhD in Electrical Engineering from Columbia University in 1977 and 1984, respectively. After his graduation he joined the faculty of the Dept. of Electrical and Computer Engineering, Syracuse University, New York. In Spring 1991, he joined the faculty of the School of Electrical and Computer Engineering, at Purdue University, West Lafayette, Ind., where currently he is a Professor, and the Coordinator of Distributed Multimedia Systems Laboratory. This laboratory is a modern research facility to conduct research in multimedia databases, multimedia computing, and broadband multimedia communication. This facility has been funded by various government and private organizations, including the U.S. Dept. of Defense, the National Science Foundation, NYNEX Corporation, AT&T Foundation, IBM, Intel Corp., Fuji Electric Company of Japan, among others. Prof. Ghafoor has been actively engaged in research areas related to multimedia information systems. He has published over 100 technical papers in leading journals and conferences. Prof. Ghafoor has served on various IEEE and ACM Conferences Program Committees. Currently, he is serving on the editorial boards of numerous journals including ACM/Springer Multimedia Systems Journal and the Journal of Parallel and Distributed Databases. He has served as a Guest/Co-Guest Editor for various special issues including a special issue of ACM/Springer Multimedia Systems Journal, Journal of Parallel and Distributed Computing, International Journal on Multimedia Tools and Applications, IEEE Journal on the Selected Areas in Communication, and IEEE Transactions on Knowledge and Data Engineering. He has been consultant to GE, the US Dept. of Defence, and the UNDP. He is a Fellow of the IEEE.