Spatio-Temporal Querying of Video Content Using SQL for Quantizable Video Databases

Vani Jain and Ramazan Aygun
Computer Science Department, University of Alabama in Huntsville, Huntsville, US
{vjain,raygun}@cs.uah.edu

***Abstract*-Multimedia database modeling and representation play an important role for efficient storage and retrieval of multimedia. Modeling of semantic video content that enables spatiotemporal queries is one of the challenging tasks. A video is called as "quantizable" if the instants of a video are enough for a person to imagine the missing scenes properly. A semantic query for quantizable videos can be defined in a more flexible way using spatio-temporal instants. In this paper, we provide a semantic modeling and retrieval system, termed as G-SMART. Firstly, the videos are quantized according to semantic events. Then semantic instants and events of the video that include objects, events, and locations are provided as a grammar-based string. This linear string representation enables both the spatial and temporal retrieval of the video using Structured Query Language (SQL). The redundancy in this linear representation is reduced by using data reduction properties such as removal of implied information. Various types of queries such as event-object-location, event-location, object-location, event-object, current-next event, projection and semantic event are supported by G-SMART. A graphical user interface is designed to build queries and view the query results. G-SMART enables multimodal presentation by displaying the query results in the form images and videos. We show our results on a tennis video database.**

***Index Terms*-Video databases, semantic modeling, multimodal information retrieval, spatiotemporal queries**

I. INTRODUCTION

Multimedia information retrieval and modeling has gained great consideration in the last two decades. The interest on modeling and representation of data has increased recently [27] [11] [3], after it reached its peak for multimedia database modeling in 1990s. Towards the end of 1990s, it was realized that the updates to multimedia databases are seldom and then multimedia information retrieval and indexing had become a challenging research topic [15] [2]. Later on, researchers focused on the low level media feature extraction and analysis [16] [4]. Currently, a lot of efforts have been made to fill the gap between the low level media features and high level semantics [17] [6].

High bandwidth internet, high speed processors and large storage devices have made videos popular and easily available to everyone. Video conferencing, online videos, advertisements, sports, movies and news have gained popularity among the users of all ages and professions. The increase in use of videos has drawn our attention to various concerns like efficient storage and fast retrieval. The most successful video retrieval tools such as YouTube [26], Google videos [10] are keyword-based. This means that each video is described as a set of keywords and videos are retrieved using these keywords. In YouTube, when a user uploads a file, the user needs to enter the metadata description of the video.

Semantic Query Engine. A semantic query engine can perform queries beyond keyword-based queries. Assume that instead of a computer-based query engine, a human query expert answers user queries. How would a user explain the query to the query engine? If the query engine can execute the query that can be performed by a human query expert, it is called as a semantic query engine. In this context, all keyword-based systems do not have a semantic query engine since keywords are only combined with Boolean operators without any semantics.

Quantizable Video. A video is quantizable if the instants of a video are enough for a person to imagine the missing scenes properly. For example, basketball player Nowitzki passes the ball to another player Kidd. If the beginning of the action (i.e., the basketball player throws the ball) and the ending of the action (i.e., the other player receives the ball) are enough to imagine what happened in between the beginning and ending, the video is quantizable. There are 4 important concepts for this example: "Nowitzki", "Kidd", "ball", and "pass". The query can be specified in natural language as "retrieve all videos where *Nowitzki passes* the *ball* to *Kidd*". It can also be specified that the ball first hits the court before Kidd receives the ball. This query could be specified as "retrieve all videos where *Nowitzki passes* the *ball* to *Kidd* with a *bounce-pass*". The locations of the players can also be provided to enhance the query such as "retrieve all videos where *Nowitzki* at the *center circle passes* the *ball* to *Kidd* at the *freethrow circle* with a *bounce-pass"*. This query expression can be simplified as "(Nowitzki @ Center Circle) → bounce-pass → (Kidd @ Freethrow Circle)".

The previous example shows an example of cases where a query can be expressed as a series of spatial instances that are linked by semantic events. This can be represented as a single dimension or linearly. The only missing component is the logic of ordering these spatial instances. If the content of a video is represented with respect to a grammar, the querying can be performed

using pattern matching modules of powerful querying languages such as Structured Query Language (SQL). We propose a novel video modeling and retrieval method, called *Grammar-Based Semantic Modeling and Retrieval Tool* (*G-SMART*) to represent the semantic contents of the videos. The objective of *G-SMART* is to provide a representation for these contents in the order of their occurrences without requiring any changes on traditional database systems and SQL. The semantic contents mainly consist of the events occurring in the video, main objects of the video and location of the objects. For example, in a soccer game *free kick, corner kick* and *penalty kick* are the events; players are the objects; and the soccer field is the location. These contents are represented in a linear string with the help of a grammar in *G-SMART*. The linear string preserves the temporal order of the video. The grammar defines the rules for representing the videos. The representation can be made compact or reduced by applying the data reduction properties.

The grammar-based representation simplifies maintaining the semantics of a video especially if the video content follows some rules as in sports games. These types of videos have three types of contents: objects, events and locations. The grammar helps us determine the connections between these components. The grammar can even further help us embed the rules of a game as part of the database. Since each game has a different set of rules, this requires differences especially when creating queries or applying semantic data reduction. In terms of modeling and representation, different types of videos can be represented with different grammars. For a new type of game, our system just requires a new grammar and a set of rules based on this grammar. This shows the flexibility of *G-SMART*.

A visual query language [30] is developed to express SQL queries that include joins with no emphasis on spatio-temporal content. BilVideo [32] extends SQL for spatio-temporal queries. Nevertheless, SQL-like query languages or extensions to SQL complicate querying [31]. One of the key points of *G-SMART* is the storage and retrieval of semantic data as a linear string in traditional relational database systems. This alleviates the use of powerful SQL for spatiotemporal queries. The linear string-based representation enables us to have a powerful video retrieval tool. This tool allows retrieval of videos on the basis of occurrence of events, the objects of the clip and locations of these objects. Some of the sample queries that are possible for the tennis video are:

- Retrieve all the clips where Oscar Hernandez gives a back hand shot,
- Retrieve all the clips where ball goes outside the court,
- Retrieve all the clips where player Roger Federer plays with Rafael Nadal.
- Provide the projection for the player Lleyton Hewitt for the given clip.

The visualization of many video query results at the same time is actually cumbersome, if the video clips are not short. The view of many clips also requires the streaming of multiple clips from the video database server. G-SMART generates summaries of videos in the form of still images. These summaries are based on the number of events in the video clip and are provided to the user before actually playing the clip. The summaries provide a small sized presentation of the entire clip that can be easily downloaded or displayed. This makes them more suitable for small handheld devices like PDAs. Therefore, G-SMART enables presentation of different media information like animated images and video.

This paper is organized as follows. The following section provides related research on spatiotemporal modeling, retrieval, and extraction of low-level features. Section 3 explains the architecture, modeling, and representation of G-SMART. The application of G-SMART on tennis videos is explained in Section 4. Section 5 explains how spatiotemporal queries are achieved in G-SMART. Multimodal presentation and user interface is discussed in Section 6. The last section concludes our paper.

II. RELATED WORK

In this section, we provide an overview of approaches on video representation, modeling and retrieval. These approaches mainly vary upon the aspect of video to be used for modeling. Since we show the application of G-SMART on tennis videos, we also provide related work for the extraction of high-level features from low-level features for tennis videos.

Semantic content: A video can be represented in terms of alphabets. The sequence of alphabets represents the features of video. In [27], the video data is transformed and represented as a stream of alphabets. A video is represented in terms of four streams: court field, camera motion, scoreboard and audio events. A mapping table has been suggested where the symbols are mapped to video data. For example, court stream can be court or non-court. Here, symbol ‘A’ can represent court and symbol ‘B’ can represent non-court. A clip can be represented with the help of these symbols such as “B, A, D, G……..”. This representation provides an alphabetical representation to the video clip. However, there is no representation for the main objects of the video such as the players or the ball in sports video. Representation of the events like ‘kicking ball’ and ‘defending goal’ is missing. They have also not provided the queries that can help in retrieval of information, since they target video data mining. In [22], a video management and application processing framework (VideoMap) is proposed. The main component of VideoMap is query-based video retrieval mechanism. The architecture of VideoMap has a specification language processing component that allows the user to define the event/action semantics with the help of an activity model. It also allows querying with the help of query language processing component that uses CAROL/ST query language instead of using a well known and commonly used query language, SQL. There has also been research on the representation of an image as a string [5].

Affective perception of video: In [11], the videos are divided with respect to video content perception as

affective and cognitive. Cognitive perception represents the facts that are present in the videos such as news and events, whereas affective perception represents the effects the video creates on the viewers such as thrilling scenes. A graph-based model has been proposed for the modeling of affective video content. A 2D emotion space with arousal and valence as dimensions is defined. Valence is the intensity of emotion; and arousal is the type of emotion. The low level features of the video that are extracted by processing tools are mapped to 2D emotion space.

Modeling fuzzy information: Fuzzy information basically represents the uncertain and imprecise values where exact values are not available. In [25], a conceptual model called $ExIFO_2$ is used to model the fuzzy information. In [3], $ExIFO_2$ is utilized for multimedia database applications. This conceptual model is then mapped to logical object-oriented model called fuzzy object oriented data (FOOD). Various mapping algorithms are proposed to map from conceptual model to logical model. Experiments were carried out on soccer videos. The querying interface also permits fuzzy values in setting up the queries.

Spatiotemporal data modeling: There has been a lot of research in the area of modeling spatial and temporal contents of video. In [14], the trajectory of a moving object is modeled. The spatial representation for each object is provided with the help of minimum bounding rectangles and temporal interval algebra is used for representing the temporal relationships. A trajectory matching algorithm is provided for matching the moving objects. The model has been integrated with Object Oriented Database Management System (OODMBS) and uses Object Query Language (OQL). Pissinou et al. propose a Topological-Directional Model for the spatiotemporal contents of the video [19]. The model represents an object by a minimum bounding rectangular parallelepiped (mbrp). Mbrp is used to enclose an object. A key frame is described with the help of relative positions of objects. The relative positions are specified using the relationships between the projections of mbrp of objects. The temporal contents are described by the set of these mutual spatial relationships. These representations do not model the semantic contents of the videos such as the events and do not provide a very efficient retrieval method. In [8], SQL is extended to describe various spatiotemporal queries. However, the representation of data is not very efficient and the queries where the events are one after the other are also missing. In [18], the authors suggest a model called Content Based Retrieval (COBRA) for mapping low level features to high level features. This model supports stochastic techniques such as Hidden Markov Models (HMMs) for mapping purposes. They provide an object grammar, an event grammar and algebraic operators. However, their grammar defines the components and domain of objects and events rather than how they should be represented in the database. There is also no hierarchical representation of video content.

High-level feature extraction from low-level features: There has been significant research in the extraction of low level features of sports videos and mapping it to high level concepts [20][24][18][28][12]. Neil et al. [20] propose human behavior recognition method by combining non parametric classification technique with parametric representation. The position, velocity and local motion are represented by the data driven human recognition method. Further, these three databases are searched and Bayesian network is used to fuse their results. Lastly, the behavior is encoded as the sequence of actions. They conducted experiments on the tennis videos and promising results were obtained. The events such as *service, backhand at net* and *walking at net* can be recognized. In [28], a novel action recognition algorithm is proposed. This algorithm is based on motion analysis. The experiments were performed on the tennis game where left swing and right swing were recognized. The accuracy for this method is very high. The authors also propose a multimodal framework where the action recognition can be integrated with audio analysis and real-world trajectory computation. This provides a mid level representation and allows high level analysis such as video indexing and annotation, highlighting, ranking, tactic analysis and statistics. The tennis ball can also be tracked even by using low quality single camera video [24]. The algorithm finds out the ball's trajectory. In the algorithm, the foreground moving objects are detected, then the foreground blobs are divided into tennis ball candidates and non candidates; and finally, particle filter is used to track these candidates. This information can be combined with the player's position to detect the key events such as hit and bounce in a specific region of the tennis court.

The focus of our research is the modeling of high-level information as a linear string following a grammar to enable various types of queries such as event-object-location, event-location, object-location, event-object, current and next event, projection and semantic event. Further, the research supports multimodal presentation by returning images in the form of animated gif files and video, as an output of the query. The queries will be explained in section 5.

III. VIDEO MODELING and REPRESENTATION

Videos can be classified into two types with respect to its semantic content [27]. The first type covers videos having content structure such as movies and the second one includes events such as sports videos. Our method provides a new approach for representing, modeling and querying of semantic features of the sports videos.

A. Architecture of G-SMART

G-SMART provides an architecture that maintains and stores the videos, their representation and summaries. It also facilitates the querying of these videos with the help of user interface. A popular video description standard, MPEG-7 [13][1] specifies the syntax and descriptors for the description of multimedia content. A video in MPEG7

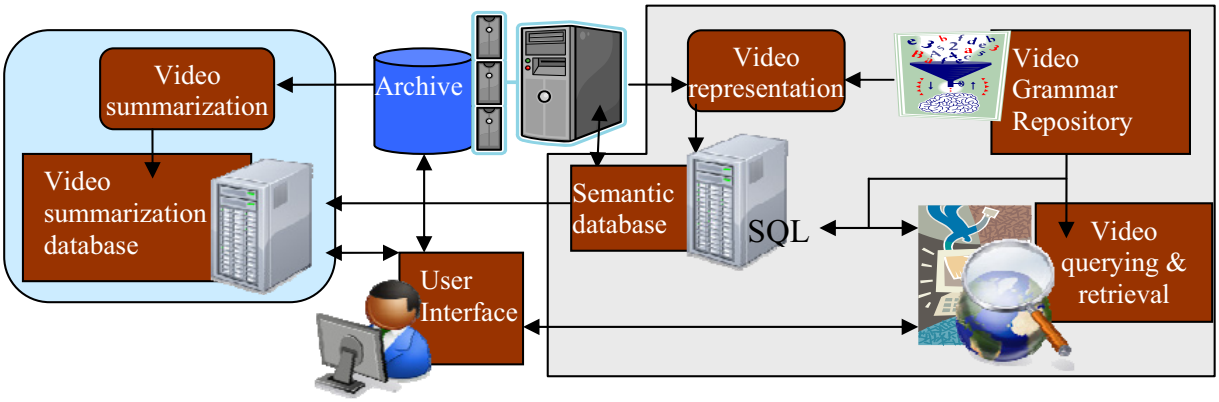


Fig 1: Block diagram of the architecture of G-SMART

Fig 2 A discretization of a tennis video: V_8

[13] [1] can be mapped into linear string and spatiotemporal queries can be performed on a traditional relational database system using G-SMART. The block diagram in Fig 1 shows the system architecture. The components of the architecture are explained in the subsequent subsections. The main components of the architecture are: Video archive, Video grammar repository, Video representation, Semantic database, Video summarization, Video summarization database, Video querying and retrieval engine, and User interface. By using the grammar repository, the video content is converted to string representation, and these are maintained in the semantic database. The video querying engine responds to user queries using the grammar from the repository and applying pattern matching on the semantic database. Video summarization database maintains a summary of the videos by applying video summarization based on the number of semantic events retrieved from the semantic database. The video summaries are first presented to the user after querying. We now explain the components of this architecture.

- **Archive:** Video archive maintains the collection of all videos. It stores the data in a raw format without any processing. Video archive provides the unprocessed clips to video representation and video summarization modules that maintain the metadata and provides the summary of a video, respectively. The semantic database maintains the pointers to videos in the video archive. The videos are retrieved from the archive for displaying to the user.
- **Video grammar repository:** The grammar repository maintains the syntax of data representation and is used to represent the contents of the video and then storing the semantic strings in the semantic database.
- **Video representation:** Video representation enables representation of semantic features of the sports videos as string data that are compliant with the proposed grammar. This allows spatiotemporal queries using query language, SQL.
- **Semantic database:** Semantic database stores all the semantic features of the video clips that are maintained by the video representation module.
- **Video querying and retrieval engine:** Video querying and retrieval module constructs the query from the information provided by the user interface, executes the query and retrieves the result. The query engine performs the pattern matching according to the available grammar from the video grammar repository.
- **Video summarization:** During the querying and retrieval process, G-SMART returns the summaries of the clips as an initial output of the queries. These summaries allow the user to understand the clip without viewing the entire clip. This saves the user time as he understands the clip by looking at the summary and can view the entire clip according to his interest.
- **Video summarization database:** Video summarization database stores and maintains all the video summaries in the animated gif format. This database is created as an output of video summarization process. The video query and retrieval engine retrieves the summaries from this database and provides them to the user interface.
- **User interface:** User interface accepts the queries and provide them to video querying and retrieval engine. After the query is executed, the user interface displays the results of the query to the user.

B. Semantic Video Quantization

The temporal and spatial domains can be modeled as continuous or as discrete instants. For example, a

trajectory of a ball can be represented with a continuous function. The trajectory of the ball can be discretized and the sampled data at the discrete instants can be used to represent the trajectory of the ball. This sampling or discretization should be achieved in a way that the actual trajectory can be generated accurately.

Definition 1. A discrete video is a sequence of n linearly ordered instants and represented as $V_n=<I_0, I_1, ..., I_{n-1}>$ where I_i is the i^{th} instant in the video (Fig 2). Each I_i is further called as a *spatial instance.*

Definition 2. A period of a video V_n that has a starting instant with p and ending instant with q is represented as $V_n[p,q]=<I_p, I_{p+1}, ..., I_q>$ where $0 \le p \le q \le n$. A period of a video is further called as *spatiotemporal instance*. This type of representation conforms to the time quanta definition in [27]. A video can also be represented as a sequence of spatiotemporal instances: $Vn=<P_0, P_1, \ldots, P_m>$ where P_i is period.

Definition 3. The actual duration or length of a period $V_n[p,q]$ is represented as $Dur(V_n[p,q])$.

C. The Components of Video

In our model, we identify the main objects, events, locations and cameras (or camera views) in the videos and the temporal information, τ. These can be briefly explained as follows:

- **Objects:** An object in a video is a region that has a semantic meaning and its spatial properties change over time. Object represents the main entity that performs some action. Objects are of prime interest to the viewer of the clip. For example, 'players' in a sports video are objects. Each object *O* in a video is represented with an alphabet from domain $\Sigma_O = \{O_1, O_2, O_3, \ldots O_n\}$.
- **Events:** Events represent the main occurrence in the video. Event is a happening of some action by an object. When an object does something at a given location and time and attracts the viewer, it can be termed as an event. For example, 'serving' is an event in a tennis game. Each event E is represented with an alphabet from the event domain $\Sigma_E = \{E_1, E_2, E_3 \ldots E_m\}$.
- **Locations:** Spatial information is represented with the locations (or positions). It represents the space occupied by the objects. For example, soccer field is the location for players and ball. The location can be determined semantically with respect to the regulations of the sports. Each region L can be represented from the location domain $\Sigma_L = \{L_1, L_2, L_3, \ldots, L_p\}$.
- **Cameras:** In each video, various cameras (or camera views) provide different footage. For example, in a video first camera provides court view, second camera provides audience view and another camera provides zoom coverage of players. Each camera C is represented with an alphabet from the camera domain $\Sigma_C = \{C_1, C_2, C_3, \ldots, C_q\}$.

In the videos each event occurs one after the other. Thus, we can represent the videos as a sequence string $S \in \{O_n, E_m, L_p, C_q\}^*$. The length of S, |S|, provides information on the length of the sequence in the temporal dimension, τ.

D. Grammar for Spatiotemporal Representation

Since we represent spatiotemporal content of a video as a string, the grammar is required to parse and extract the spatiotemporal information from the string. We can define the grammar for the representation of the sports videos as:

```
<video> :: = <sequence of clips>

<sequence of clips> :: =<clip>|
            <sequence of clips>

<clip> ::= <camera>
            "["<spatiotemporal>"]"

<spatiotemporal> ::= <spt>
            | <spatiotemporal>

<spt>  :: =  [<event>] <obj> <loc>
```

where *video* is a sequence of clips; *clip* has a camera view and a spatiotemporal instances; and a spatiotemporal instance (*spt*) is a sequence of spatial instances with an object (*obj*), location (*loc*) and an optional *event*. For example, consider the subsequence $S_t = \{S\ G\ 10\}$. In S_t, S represents the event saving goal, G represents the goal keeper object and 10 represents the location of the goal keeper. Here, we first decompose a video into clips to emphasize the use of different cameras.

E. Reduction of Data

If the spatiotemporal and action information for every object is represented, the redundancy becomes inevitable. Such redundancy can be reduced by using the semantics of the game, removing unnecessary information and combining properties of spatiotemporal activities. We summarize them with the following two properties.

- **Property 1**: **Removal of implied data.** In sports videos some information is implied or can be predicted from the spatial information of other objects. We may decrease redundancy by removal of implied data. For example, in a soccer game the location of goal keeper during the penalty kick is fixed. Thus, we can remove this information from the representation as this information is already available from the regulations of the game.
- **Property 2: Removal of redundant data.** Any data that is useless for spatiotemporal content from the viewer's point can be removed from the representation. For example, information gathered by the camera focusing on commentator is not actually part of the game. Such information is generally not of much use for spatiotemporal queries and can be removed.

The purpose of these properties is not to make querying complex and difficult but to remove the redundant data.

IV. MODELING AND REPRESENTATION OF TENNIS VIDEO

We explain the application of the above approach on the tennis videos. The representation, modeling and data reduction of tennis video are described in the following subsections.

A. Representation of Tennis Video

We give brief information on tennis videos while explaining the representation of the tennis video content.

- **Objects:** There are 3 main objects identified in the tennis game. The objects are identified as Σ_O = { U, V, b} where U is the first player, V is the other player and b is the ball.
- **Events:** The main events are identified in the alphabet Σ_E ={F, B } where F is the forehand shot and B is the backhand shot
- **Location:** The tennis court is divided into regions by line segments to apply the rules of tennis game as in Fig 3. For representation of locations and for semantic retrieval, we divide the court into partitions in the same way and apply the numbering in Figure 2.

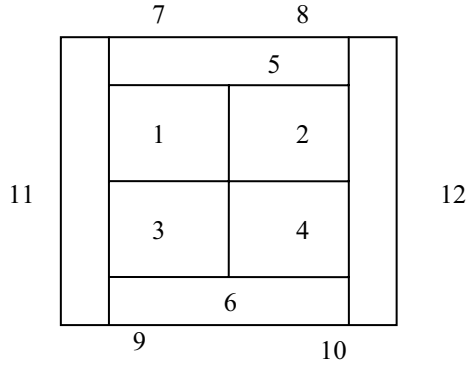


Fig 3 Tennis court segmentation

Cameras Views. We identify six types of camera views in a tennis game:

A – Gives a close view of the player at location 7 & 8 in Fig 3
B - Gives a close view of the player at location 9 &10 in Fig 3
C - Court view
D - Action Replay
R - Rest time
Com – Commentators

Figures 4, 5, and 6 show sample images by cameras A, B and C.

Grammar for Tennis Video Database. We now extend the grammar, mentioned in section 3.D for tennis videos.

```
<obj > ::= <player>|b
<player> ::= U|V
<event>::= F|B
<location> ::=
     1|2|3|4|5|6|7|8|9|10|11|12|N
<camera> ::= <close-view camera>|C|D
<close-view camera> ::= A|B
<spt>   :: =  <obj> <loc>
         |<event> <player> <loc>
<spatiotemporal> ::= <spt>
         |<spatiotemporal>
<sequence of clips> :: =<clip>
         | <sequence of clips>
<clip> ::= < close-view camera >
       "[" <player> "]"
      | C"[" <spatiotemporal> "]"
      |D"[""]"
<video> :: = <sequence of clips>
```

Note that it is also possible to enforce some constraints based on this grammar. For example, events forehand and backhand shots are only eligible for player objects; the players cannot pass the other side of the net during the play; a player cannot hit the ball twice; and so on. We do not mention them here since we provide data reduction and not to confuse the reader by extending the grammar.

B. Modeling

We now apply the above grammar for the following video clip. Consider the video sequences in Fig 7. We can represent this video sequence using the grammar for tennis videos as:

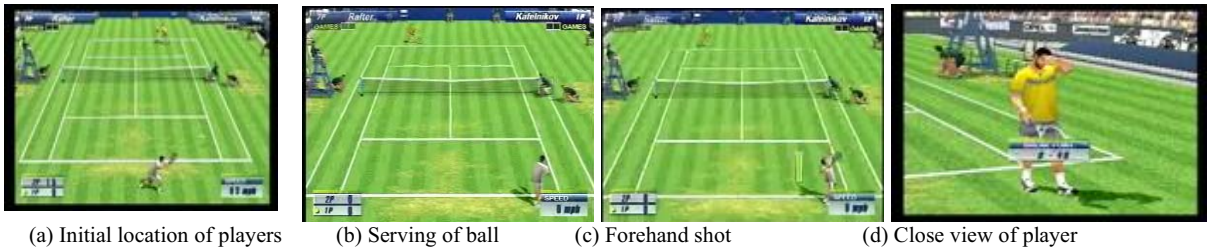
T_1={A [U] C [U7b7V10 b4 F_V10 b8]}

The above sequence indicates that the camera *A* captures the close view of the first player and then court view captures that first player; ball is at location 7; and player 2 is at location 10. The first player serves; ball hits location 4; the player at location 10 gives a forehand shot; and ball goes out of the court at location 8.

Fig 4 Close view by camera A [33]

Fig 5 Close view by camera B [33]

Fig 6 Court view by camera C [33]



(a) Initial location of players (b) Serving of ball (c) Forehand shot (d) Close view of player

Fig 7 Different spatial locations and camera views [33]

(a) Serving of ball (b) Back hand shot (c) Back hand shot

(d) Forehand shot (e) Close view of player

Fig 8 Sample sequences from a tennis video [33]

Fig 8 provides another clip from a tennis video. The representation of this clip is:

T_2 = {A [U] C [U8 b8 V9 b3 Bv9 b5 B_U8 b4 F_V10 b5] D[ ]}

The sequence represents that camera *A* captures the close view of the first player and then the court view captures that the first player is at location 8 and the second player is at location 9. The first player serves and ball goes to location 3 and then the second player gives a back hand shot at location 9. As a result, the ball goes to location 5 and then the first player gives a backhand shot at location 8. The ball goes to location 4 and the second player gives a forehand shot at location 10. Finally, the ball goes to location 5. This sequence is replayed by camera D.

C. Data Reduction

We can now apply the data reduction properties that are explained in section 3.B.

Removal of implied data. According to this property, we may eliminate all the implied information. Consider the above 2 sequences T_1 and T_2.

a) While serving, the ball is with the player. The initial ball location is removed as it is the same as that of player1.

T_1'= {A [U] C [U7~~b7~~V10 b4 F_V10 b8]}
= **{A [U] C [U7V10 b4 F_V10 b8]}**
T_2'= {A [U] C [U8 ~~b8~~ V9 b3 Bv9 b5 B_U8 b4 F_V10 b5] D[ ]}
= **{A [U] C [U8 V9 b3 Bv9 b5 B_U8 b4 F_V10 b5]D[ ]}**

b) The players change their locations alternatively. For example, if the initial position of players are {7, 10} then in the next serve the players position are {8,9}. This information can be easily predicted with the help of first sequence.

T_2''= {A [U] C [~~U8 V9~~ b3 Bv9 b5 B_U8 b4 F_V10 b5]D[ ]}
= **{ A [U] C [b3 Bv9 b5 B_U8 b4 F_V10 b5] D[ ] }**

Here, the information from T_2'' is removed as it is implied from T_1'.

Removal of redundant data. By applying property 2, we can remove redundant and useless information. For example, rest time, close-view and commentators camera view are not of interest and can be removed from the representation as follows:

T_2'''= { ~~A [U] C~~ [b3 Bv9 b5 B_U8 b4 F_V10 b5] ~~D[ ]~~ }
= **{ [b3 Bv9 b5 B_U8 b4 F_V10 b5]}**

V. VIDEO QUERYING and RETRIEVAL

G-SMART's representation enables different types of queries. These queries can be written with the help of *G-SMART*'s user interface. The user interface abstracts the lower level representation of videos and allows the user to write queries easily. The output of the queries is the video summaries of the retrieved video clips.

Representing the video content as a string helps the user describe many spatiotemporal queries. Most of the queries can be expressed by using SQL and few complex queries can be written by extending SQL through macro facility provided by *G-SMART*. The following types of queries are allowed in *G-SMART*.

1) Event-object-location
2) Object-location
3) Event- location
4) Event-object
5) Current and next event
6) Projection
7) Semantic event

We explain each of these with the examples on tennis videos in the following subsections. The SQL queries are not built by the user, but they are automatically generated by *G-SMART*.

In *G-SMART* database we maintain a table called scoreboardsnapshot to store the semantic description of the video clips.

A. Event-Object-Location

Event-object-location queries retrieve the clips by specifying an event and location of an object in a video. Event-object-location describes the action and spatial information for the specified object. For example, a player takes a free-kick near penalty area in a soccer game. In this example, 'player' is an object, 'free-kick' is the event and 'near penalty area' is the location. Retrieval of such a clip is called Event-object-location query. In the grammar, spatial instance is represented as

```
<spt> ::= [<event>] <obj> <loc>.
```

Therefore, the pattern to be found is of "event object location" in the sequence. These queries can be written in the following way:

```
SELECT clip
FROM database
WHERE sequence like '%event object location %'
```

The above query does the string matching and looks for the pattern of "event object location" anywhere in the sequence.

Example 1. List all the clips where the first player gives a backhand shot from area outside the court.

```
SELECT clip
FROM Scoreboardsnapshot
WHERE sequence like '%BU7%'
      or sequence like '%BU8%';
```

This query finds out the pattern of 'BU7' and 'BU8' anywhere in the string. 'BU7' represents the backhand shot by player U at location 7; and 'BU8' represents the backhand shot by player U at location 8.

B. Object-Location

Object-location queries retrieve the clips by specifying the location and object of the video. Object-location describes the spatial information for the given object. For example, in a soccer game a defender stands near the goal post. Here, the 'defender' is the object and 'near goal post' is the location. Retrieval of such a clip is called Object-Location query. In the grammar, the spatial instance is represented as

```
<spt> ::= [<event>] <obj> <loc>.
```

The event can be ignored and the pattern to be searched is a sequence of object and location. These queries can be written in the following way:

```
SELECT clip
FROM database
WHERE sequence like '%object location %'
```

The above query does the string matching and looks for the pattern of "object location" anywhere in the sequence.

Example 2. List all the clips where the ball goes outside the court.

```
SELECT clip
FROM Scoreboardsnapshot
WHERE sequence like '%b9%' or sequence like
      '%b10%' or sequence like '%b7%'
      or sequence like '%b8%';
```

This query finds out the pattern of 'b9', 'b10', 'b7' and 'b8' anywhere in the string. 'b9' represents the ball at location 9. Similarly, 'b10', 'b7' and 'b8' represent the ball at location 10, 7 and 8, respectively.

C. Event-Location

Event-location queries are formed by specifying the event and the location of the occurrence of event in the video. Event-location describes the spatial information of the action. For example, in soccer game "throw-in" from outside the field has 'throw-in' as the event and 'outside the field' as the location for the event. Retrieval of clips on this criterion is called Event-location query. In the grammar, the spatial instance is represented as

```
<spt> ::= [<event>] <obj> <loc>.
```

The object needs to be ignored. The character '_' allows ignoring a single character in the pattern in SQL. If objects are represented with a single character, using one '_' is satisfactory. The pattern to be searched becomes 'event_location'. These queries can be written in the following way:

```
SELECT clip
FROM database
WHERE sequence like '%event_ location %'
```

The above query does the string matching and looks for the pattern of 'event_location' anywhere in the sequence.

Example 3. List all the clips of forehand shots from area inside the court.

```
SELECT clip
FROM Scoreboardsnapshot
WHERE sequence like '%F_1%' or sequence like
      '%F_2%' or sequence like '%F_3%'
      or sequence like '%F_4%';
```

This query finds out the pattern of 'F one character 1,' 'F one character 2,' 'F one character 3' and 'F one character 4' anywhere in the string. 'F_1' represents the forehand shot by any player at location 1. Similarly, 'F_2' represents the forehand shot by any player at location 2.

D. Event-Object

Event-object queries retrieve the clips by specifying the event and object of the video. For example, in a soccer game saving the goal by goal keeper, 'goal keeper' is the object and 'saving the goal' is the event. Retrieval of such a clip is called Event-object query. In the grammar, the spatial instance is represented as

```
<spt> ::= [<event>] <obj> <loc>.
```

The location can be ignored and the pattern to be searched is a sequence of event and object. These queries can be written in the following way:

```
SELECT clip
FROM database
WHERE sequence like '%event object %'
```

The above query does the string matching and looks for the pattern of 'event object' anywhere in the sequence.

Example 4. List all the clips where the player U gives a backhand shot.

```
SELECT clip
FROM Scoreboardsnapshot
WHERE sequence like '%FU%' ;
```

This query finds out the pattern of 'FU' anywhere in the string. F represents the forehand shot and U represents the first player.

E. Current-Next Event

Current-next event queries are formed by specifying the events that happen one after the other. Current- next event describes the occurrence of an action followed by another occurrence of action anywhere on the time axis. For example, in soccer game short pass and then long pass. It means the retrieval of a clip that has a short pass followed by a long pass. Retrieval of clips on this criterion is called Current-next query. In the grammar, the spatial instance is represented as

```
<spt> ::= [<event>] <obj> <loc>.
```

Two spatial instances would be written as `[<event>]<obj><loc>[<event>]<obj><loc>.` Two events are separated by an object and location. Hence, two '_' characters need to be used in the pattern: one for the object and one for the location. If the second event is optional, the string will lead to a sequence of object and location. These queries can be written in the following two ways:

1)
```
SELECT clip
FROM database
WHERE sequence like '%event_ _objectlocation %'
```
2)
```
SELECT clip
FROM database
WHERE sequence like '%event_ _ event %'
```

The first query checks cases where an event leads to a specific spatial instance whereas the second query explicitly represents back-to-back events. Two '_' characters are used, since it is not important who performs the event and what the location is. Here, event represents the current event and next event can be either represented by an object or an event itself.

Example 5. List the clips where a player gives a backward shot and the ball goes out of court

```
SELECT clip
FROM Scoreboardsnapshot
WHERE sequence like '%B_ _b10]' or sequence
    like '%B_ _b9]' or sequence like '%B_ _b7]'
    or sequence like '%B_ _b8]' or sequence like
    '%B_ _b11]' or sequence like '%B_ _b12]';
```

This query finds out the pattern like 'B _ _ b10]' at the end of string. B_b10 represents the backhand shot by any player that results in the ball at location 10. B_ _b9, B_ _b7, B_ _b8 and B_ _b12 are described similarly. The pattern uses ']' in the end instead of '%' as it tries to match strictly with the string that ends with ']'.

Example 6. List the clips with two back-to-back volleys. This query can be written with the help of macro facility provided by G-SMART.

```
SELECT clip
FROM Scoreboardsnapshot
WHERE sequence like '%B_ _ F%' or sequence like
    '%F_  _B%'  or sequence like '%B_ _ B%'
    or sequence like '%F_  _F%';
```

This query returns the clips with at least one volley. We can use our macro *find_volley(min, max)* to find out the clips with back-to-back volley. Here, *min* is the minimum number of back-to-back volleys and *max* is the maximum number of back-to-back volleys. For the above query, min = 2 and max = *.

F. Projection

Projection types of queries filter the sequence based on the object. It is represented as

$$\Pi_{(O1,O2....On)}(C)$$

where $O_1,O_2,O_3,....,O_n$ are the objects to be projected in clip C. For example, ball projection for a clip will provide the spatial information of ball in a temporal order. In tennis game, initially the ball was outside the court and then it goes inside the court, etc. Capturing all this information is called projection of an object. These queries can be written in the following way:

```
SELECT object
FROM database
WHERE clip = clipid and object=objectid
```

This query gives the projection of an object for a given clip. Consider the following sequence

T_2''= { A [U] C [b3 Bv9 b5 B_U8 b4 F_V10 b5] D[ ] }. The projection of ball T_2'' is {b3, b5, b4, b5}.

Example 7. Find the projection of ball over clip1:

$$\Pi_{(ball)}(C1).$$

This calls the macro *Projection(object, clipname)* where Object = ball and Clipname = C1. Similarly, projections on players and cameras can be found.

G. Semantic Event

All sports have certain terms and semantics that are specific to a particular game. For example, semantic events "throw-in", "penalty-kick" and "free-kick" are associated with the soccer game. Similarly, tennis game has various semantic events such as "reserve", "fault" and "aces". Retrieval of the clips based on these semantics is called as Semantic-event type of queries. For example, "retrieve all tennis clips with a fault" is such a query.

Currently, G-SMART supports "reserve" and "fault" type semantic queries. We propose the Algorithm 1 that predicts the reserve or fault for the G-SMART string representation.

Using the Algorithm 1, given below, the database can be populated with the semantic events. The status field in the table, scoreboardsnapshot, represents the semantic event. The symbol 'F' represents fault and 'R' represents reserve.

Semantic queries can be written in the following way:

```
SELECT clip
FROM Scoreboardsnapshot
WHERE status ='semantic symbol'
```

Example 8. Find all the clips with the reserve:

```
SELECT clip
FROM Scoreboardsnapshot
WHERE status = 'R'
```

Example 9. Find all the clips with the fault:

```
SELECT clip
FROM Scoreboardsnapshot
WHERE status = 'F'
```

Algorithm 1

Predict the fault and reserve semantics from G-SMART Tennis's string

```
//Pred_ball_loc :- indicates the predicted service box. It is the location where
                   the ball is expected to strike after the serve
//ball_loc      :- indicates the actual location where the ball strikes after the serve
//flip_ball_loc(ball_loc):- indicates the function that flips the ball location from 3 (Fig. 3)
to 4 (Fig. 3) and vice versa, and from 1 (Fig. 3) to 2 (Fig. 3) and vice versa.

 For all g ∈ gsmart strings
// At the beginning of each play if the initial locations of players are given then
the location of expected strike is flipped.
  If (initial locations of players given in g)
    Pred_ball_loc = flip_ball_loc(ball_loc)
    Counter=0
  Else
      If (ball_loc == pred_ball_loc) then // if the ball strikes at the service box
          Pred_ball_loc = flip_ball_loc(pred_ball_loc)
         Counter=0
      Else
          If(length(g) >9) //if the ball does not hit at the predicted location
             If (g = "bNb"+pred_ball_loc) then // if the ball hits the net
               Print Reserve //reserve condition
               Flag=1
             Else //fault condition
               Print Fault
               Counter++
             Endif
// if fault happens twice then player position changes and thus the expected
strike position changes
             If flag = 0 Then
                 If (counter = 2) Then
                     counter = 0
                     pred_ball_loc =flip_ball_location(pred_ball_loc)
                 ElseIf (counter = 0) Then
                     pred_ball_loc = flip_ball_location(pred_ball_loc)
                 End If
             End If
          Else // short play with fault
              Print Fault
             Counter++
          End If
    End If

  End if
End for
```

VI. MULTIMODAL PRESENTATION

Multimodal presentation allows the user to view different type of media information such as image, text, video and graphics as the result of queries. Currently, G-SMART supports presentation of query results as image and video. In this section, we will explain the components of the user interface. The user interface enables building the queries as well as viewing the query results. The query results are viewed in two ways: automatic video summary and actual video content. Video summarization in the form of images helps the user to get an idea of what the clip contains without actually playing the clip.

A. Video Discretization: Image

G-SMART uses Still Image Abstract [9] to provide the summary of all the clips retrieved by the querying process. G-SMART calculates the key frames of a clip and generates an animated gif file that provides a quick overview of the clip. Animated gif files present the gif images one by one with a small delay. These files help in understanding the contents of the clips easily as they represent the key events in the order of their occurrences. Further, these small sized animated gif files require less storage space and low bandwidth for distribution. They also provide faster query response as they can be easily distributed. The video discretization process can be divided into two steps:

- Generation of key frames
- Generation of Animated gif file

Santini [21] studied the comprehension gained by the key frames generated with the help of selection algorithms and uniform sampling. The experiments showed that both the methods produce same understanding if the number of frames is the same. By using the same approach, we generate the key frames based on the number of events in the clip. We assume that each event has an interval and the key frame is selected from the center of this interval.

For a given clip, the key frames are selected using the following method. A video *v* is composed of n clips: $c=\{c_1,c_2,c_3,c_4.....c_n\}$ where c_i represents the i^{th} clip in the video. The duration of clip c_i is represented with d_i. The total number of events in clip c_i is denoted with e_i that represents the number of spatial instances in the clip. For simplicity, we assume that the duration or interval of each event is the same. The interval for an event in clip c_i is represented with $interval_i$ as

$$\text{interval}_i = \frac{d_i}{e_i} \qquad (1).$$

The key frame of j^{th} event in clip c_i is determined by

$$k_j = \frac{\text{interval}_i}{2} + j * \text{interval}_i \qquad (2).$$

Substituting (1) in (2), we get

$$k_j = \frac{d_i}{2 * e_i} + j * \frac{d_i}{e_i} \qquad (3),$$

$$k_j = \frac{d_i}{e_i}(0.5 + j) \qquad (4).$$

For example, consider a video clip c_i of length 1 minute, 10 seconds with 6 events. Here, d_i = 70 secs and e_i=6. We divide the clip into 6 events and number them as in Fig 8.

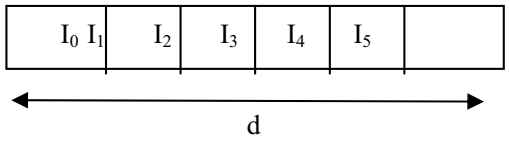


Fig 8: A video clip of length d with 6 events

Using (4), we identify the key frames at the following seconds of the clip.

$k_0 = 70/6(0.5 + 0) = 5.8$
$k_1 = 70/6(0.5 + 1) = 17.5$
$k_2 = 70/6(0.5 + 2) = 29.16$
$k_3 = 70/6(0.5 + 3) = 40.8$
$k_4 = 70/6(0.5 + 4) = 52.5$
$k_5 = 70/6(0.5 + 5) = 64.16$

After the key frames are generated, they are encoded into animated gif. G-SMART provides an interface that enables this. The user interface allows the user to define the list of the key frames, the name of the animated gif file that will be generated and delay. Delay defines the time gap between the appearances of key frames in the animated gif file.

B. User Interface

The web based interface of *G-SMART* (Fig 9) available at http://146.229.232.110/gsmart/gsmart/gsmart-main.aspx provides distributed access of *G-SMART* and makes it platform independent. The home page of *G-SMART* provides a list of all the games. The user can play the entire tennis video of the selected game. The home page also redirects to the spatiotemporal query interface by clicking "Spatiotemporal Queries" button of a particular game. G-SMART derived SQL query page is dynamically populated with the objects based on the selection of the game on the home page.

The Query interface allows creating a query by selecting the type of query from "query template" tab (Fig 9). On selecting the query type, the template appears in the editor box. For example, on clicking *Event-object-location*, *'%eventobjectlocation%'* appears in the box. The interface provides the information and allows the user to select the events, objects and locations of the sports video. On clicking *View,* the SQL query is constructed. This query can be modified by clicking *Clear* and *Add* buttons. After the construction of the query, it can be executed by clicking *Execute* button. This operation queries the database and fetches the clips. The summaries of the retrieved clips are displayed to the user. These summaries are the animated gif images. The user can also play any clip by entering the clip number in the text box and clicking "Submit" and "Play".

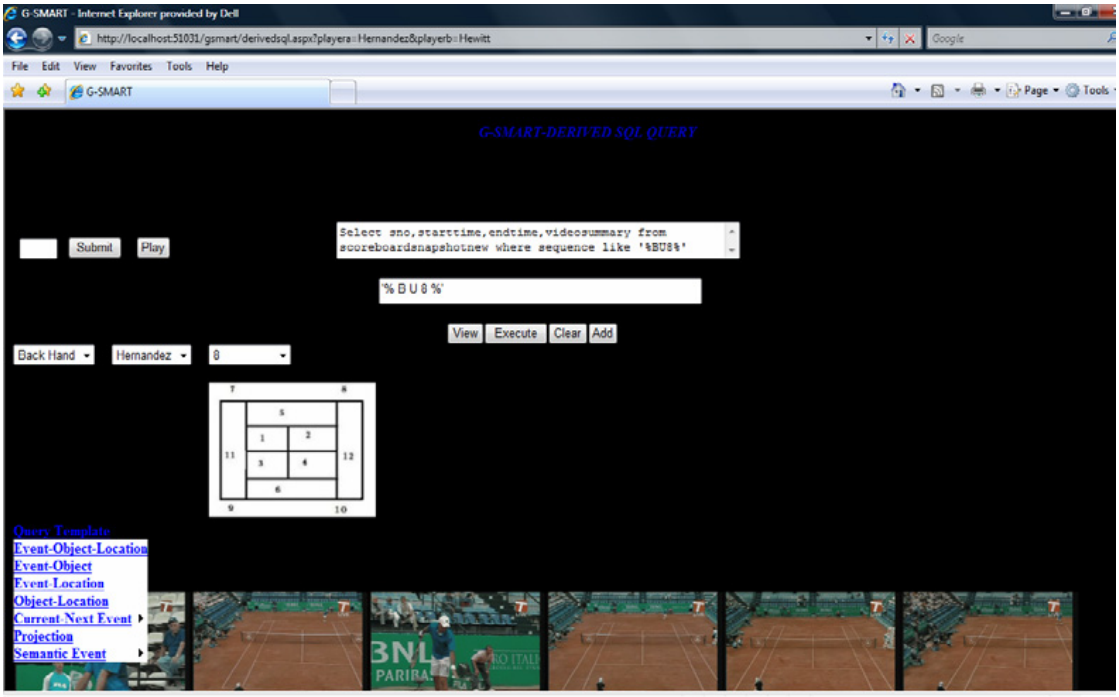


Fig 9: *G-SMART* query interface –User interface of *G-SMART*

G-SMART also provides a user interface that can be used to view the tennis video representations for a specific camera view, such as court view and action replay (Fig 10). This interface parses the G-SMART string and returns the string representation for the selected camera view. By using this interface the user can reduce the data by removing the redundant information, according to data reduction property 2 explained in section IV.

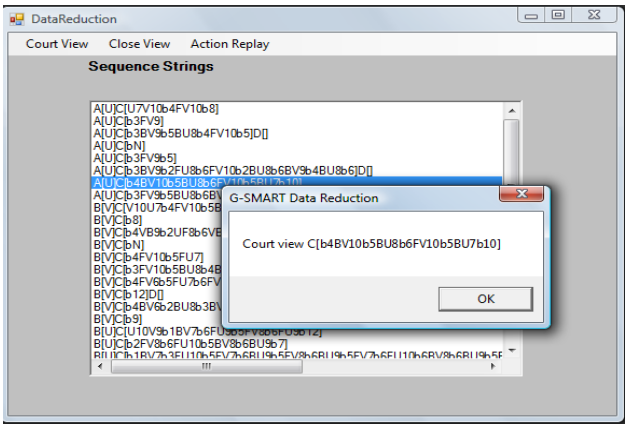


Fig 10: User interface for data reduction of G-SMART

VII. CONCLUSION

In this paper, we have provided a method of video modeling, representation, and retrieval based on a grammar. *G-SMART* defines a grammar for string-based linear representation of the semantic contents of the video. This representation is reduced by applying the data reduction properties as discussed in the paper. Experiments are carried out on tennis videos and we have got promising results. *G-SMART* provides various spatiotemporal queries and video summarization. The strongest points of *G-SMART* is the power of making spatiotemporal queries simple by using SQL (without extending SQL) and allowing multimodal presentation by returning images and videos. Actually, this originates from the powerful representation of semantic video content as a linear string. As future work, we plan to develop indexing strategies for fast retrieval of data.

REFERENCES

[1] H. Agius and M. C. Angelides,"MPEG-7 In Action: End user experiences with Cosmos-7 front end systems", *Proceedings of the 2006 ACM Symposium on Applied Computing SAC '06*, pp. 1348-1355,2006

[2] W. Aref, A. C. Catlin, A. Elmagarmid, M. Hammad, I. IIyas and T. Ghanem, "Video query processing in the VDBMS testbed for video database research", *ACM International Workshop On Multimedia Databases,* pp. 25-32,. 2003

[3] R. S. Aygun and A. Yazici, "Modeling and management of fuzzy information in multimedia database application", *Multimedia Tools and Applications*, Vol. 24,No 1, pp. 29-56, 2004

[4] F. I. Bashir, A. A. Khokhar and D. Schonfeld, "Real-time motion trajectory based indexing and retrieval of video sequences", *IEEE Transactions on Multimedia*, Vol. 9, No 1, pp. 58-65, 2007

[5] S. Chang, Q. Shi, and C. Yan, "Iconic indexing by 2-D strings", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9,No. 3, pp. 413-428,1987

[6] L-Y. Duan, M. Xu, Q. Tian, C-S. Xu and J. S. Jin, "A unified framework for semantic shot classification in sports video", *IEEE Transactions on Multimedia*, Vol. 7, No 6, pp. 1066-1083, 2005

[7] R. Elmarsi and S. B. Navathe. *Fundamentals of Database Systems, Fifth Edition*. Addison-Wesley Longman Publishing Co., Inc., 2006

[8] M. Erwig and M. Schneider, “Developments in spatio-temporal query languages”, *Database and Expert systems Applications, 1999. Proceedings. Tenth International workshop*, pp. 441-449, 1999

[9] B. Furht and O. Marques, *Handbook of Video Databases Design and Applications,* CRC Press, 2003

[10] Google Video, http://video.google.com

[11] A. Hanjalic and L-Q. Xu, ”Affective video content representation and modeling”, *IEEE Transactions on Multimedia*, Vol. 7, No 1, pp. 143-154, 2005

[12] Y. Ke, R. Sukthankar and M. Hebert,” Spatio-temporal shape and flow correlation for action recognition”, *Visual Surveillance Workshop*, 2007

[13] H. Kosh, ”Mpeg-7 and multimedia database systems”, *SIGMOD Record*, Vol. 31, No. 2, pp. 34-39, June 2002

[14] J.Z. Li, M.T. Ozsu and D. Szafron, “Modeling of moving objects in a video database”, *1997 International Conference on Multimedia Computing and Systems*, pp. 336**,** 1997

[15] C. Meghini, F. Sebastiani and U. T. Straccia,”A model of multimedia information ietrieval”, *Journal of the ACM,* Vol. 48, No. 5, pp. 909–970,2001

[16] W. C. Naidoo and J. R. Tapamo, “Soccer video analysis by ball, player and referee tracking”, *Proceedings of SAICSIT 2006*, pp. 51-60,2006

[17] M. R. Naphade and T. S. Huang, “A probabilistic framework for semantic video indexing, filtering and retrieval”, *IEEE Transactions on Multimedia*, Vol. 3, No 1, pp. 141-151,2001

[18] M. Petkovic and W. Jonker, “Content-based video retrieval by integrating spatio-temporal and stochastic recognition of events”, *Detection and Recognition of Events in Video, 2001, Proceedings, IEEE Workshop* on Volume, Issue, pp. 75 – 82, 2001

[19] N. Pissinou, I. Radev, K. Makki and W. J. Campbell, “Spatio-temporal composition of video objects: representation and querying in video database systems”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 13, No 16, pp. 1033-1040, 2001.

[20] N. Robertson and I. Reid, “A general method for human activity recognition in video”, *Computer Vision and Image Understanding*, Vol. 104, No. 2, pp. 232-248, 2006

[21] S. Santini, “Who needs video summarization anyway?”,.*ICSC 2007*, pp. 177 – 184, 2007

[22] S.S.M. Chan, Q. Li, Y. Wu and Y. Zhuang, “Accommodating hybrid retrieval in a comprehensive video database management system”, *IEEE Transactions on Multimedia*, Vol. 4, No 2, pp. 146-159, June 2002.

[23] V.S. Subrahmanian, *Multimedia Database Systems*, Morgan Kaufmann Publishers, 1998.

[24] F. Yan, W. Christmas and J. Kittler,” A tennis ball tracking algorithm for automatic annotation of tennis match”, *The British Machine Vision Association*, 2005

[25] A. Yazici and A. Cinar,”Conceptual modeling for the design of fuzzy OO databases”, in Knowledge Management in Fuzzy Databases, O. Pons, A. Vila, and A. Vila and J Kackrzyk (Eds), Physica-Verlag:Heidlberg, New York, Vol 39, 2000, pp.12-35

[26] YouTube, http://www.youtube.com

[27] X. Zhu, X. Wu, A. K. Elmagarmid, Z. Feng and L. W, ”Video data mining: semantic indexing and event detection from association perspective”, *IEEE Transactions on Knowledge and Data engineering*, Vol. 17, No 5, pp. 665-677, 2005

[28] G. Zhu, C. Xu, Q. Huang, W. Gao and L. Xing,” Player action recognition in broadcast tennis video with applications to semantic analysis of sports game”, *Proceedings of the 14th annual ACM International Conference on Multimedia,* pp. 431-440, 2006

[29] J.R. Viqueira and N.A. Lorentzos. “SQL extension for spatio-temporal data”. *The VLDB Journal* 16, 2, pp. 179-200, April 2007

[30] D.A. Keim and V. Lum. “Visual query specification in a multimedia database system”. In *Proceedings of the 3rd Conference on Visualization '92* (Boston, Massachusetts, October 19 - 23, 1992). A. Kaufman and G. Nielson, Eds. IEEE Visualization. pp. 194-201, 1992

[31] G. Erozel, N. K. Cicekli and I. Cicekli. **“**Natural language querying for video databases”. Volume 178, Issue 12, pp. 2534-2552. June 2008

[32] O. Kucuktunc, U. Gudukbay and O. Ulusoy, "A natural language-based interface for querying a video database," *IEEE MultiMedia*, vol. 14, no. 1, pp. 83-89, January-March, 2007

[33] Internet Archive, http://www.archive.org

Vani Jain received the B.S. degree in information technology from University of Pune, India in 206 and the M.S. degree from University of Alabama in Huntsville in 2008.

She currently works as a software analyst at Intergraph Corporation. Her research interests include databases, information retrieval and management.

Ramazan S. Aygün received the B.S. degree in computer engineering from Bilkent University, Ankara, Turkey in 1996, the M.S. degree from Middle East Technical University, Ankara in 1998, and the Ph.D. degree in computer science and engineering from State University of New York at Buffalo in 2003.

He is currently an Assistant Professor in Computer Science Department, University of Alabama in Huntsville. His research interests include multimedia databases, multimedia information, retrieval, multimedia networking, multimedia synchronization, and video processing.

He is a member of IEEE, ACM, IEEE Computer Society, and SIGMM. He has over 30 publications in peer-reviewed conferences and journals. He served on the program committees or as a co-chair more than 20 conferences including ACM Multimedia, International Symposium on Multimedia and International Conference on Semantic Computing.