Flexible Querying Using Structural and Event Based Multimodal Video Data Model

Hakan Öztarak[1] and Adnan Yazıcı[2]

[1] Aselsan Inc, P.O. Box 101, Yenimahalle, 06172, Ankara, Turkey
hoztarak@aselsan.com.tr
[2] Department of Computer Engineering, METU, 06531, Ankara, Turkey
yazici@ceng.metu.edu.tr

Abstract. Investments on multimedia technology enable us to store many more reflections of the real world in digital world as videos so that we carry a lot of information to the digital world directly. In order to store and efficiently query this information, a video database system (VDBS) is necessary. We propose a structural, event based and multimodal (SEBM) video data model which supports three different modalities that are visual, auditory and textual modalities for VDBSs and we can dissolve these three modalities within a single SEBM model. We answer the content-based, spatio-temporal and fuzzy queries of the user by using SEBM video data model more easily, since SEBM stores the video data as the way that user interprets the real world data. We follow divide and conquer technique when answering very complicated queries. We give the algorithms for querying on SEBM and try them on an implemented SEBM prototype system.

1 Introduction

Since multimodality of the video data comes from the nature of the video, it is one of the important research topics for the database community. Videos consist of visual, auditory and textual channels, which bring the concept of multimodality [1]. Modelling, storing and querying the multimodal data of a video is a problem, because users want to query these channels from stored data in VDBS efficiently and effectively. In [5], a **s**tructural and **e**vent **b**ased, **m**ultimodal (SEBM) video data model for VDBSs is proposed with querying facilities. SEBM video data model supports these three different modalities and we propose that we can dissolve them within a single SEBM video data model, which makes us find the answers of multimodal queries easily.

Definition of multimodality is given by Snoek et. al. as the capacity of an author of the video document to express a predefined semantic idea, by combining a layout with a specific content, using at least two information channels, [1]. Moreover they give the explanations of the modalities that we use in SEBM as:

- *Visual modality*: contains everything, either naturally or artificially created, that can be seen in the video document;

H. Legind Larsen et al. (Eds.): FQAS 2006, LNAI 4027, pp. 75 – 86, 2006.

- *Auditory modality*: contains the speech, music, and environmental sounds that can be heard in the video document;
- *Textual modality*: contains textual resources that can be used to describe the content of the video document.

Nowadays researches are concentrating on efficient and effective ways of querying the multimodal data, which is integrated with temporal and spatial relationships. Modelling is as important as querying, because it is an intermediate step between data extraction and consumption. In general, researchers propose their querying algorithms with their data models. Snoek et. al. give the definition of multimodality and focus on similarities and differences between modalities in [1]. They work on multimodal queries in [18]. They propose a framework for multimodal video data storage, but only the semantic queries and some simple temporal queries are supported. They define collaborations between streams when extracting the semantic from the video. Oomoto et. al. don't work on multimodality but investigate the video object concept which is a base for spatio-temporal works [7]. Day et. al. extend the spatio-temporal semantic of video objects [17]. Ekin et. al. introduce object characteristics, and actors in visual events [4]. Köprülü et. al. propose a model that defines spatial and temporal relationships of the objects in visual domain which includes fuzziness, [3]. Durak in [2], extends the model proposed in [3]. She introduces a multimodal extension of the model and gives two different structures for visual, auditory and textual modalities. BilVideo is a good example for a VDBS, which considers spatio-temporal querying concepts, [8].

Main contributions of our work can be summarized as follows: In this study, we work on querying features of SEBM, which is based on human interpretation of video data. This interpretation is like telling what is happening in videos. If one can express information in digital world as human does in real world, then we think that all of the queries coming from a user can be handled more accurately and effectively. So we can bypass the problem of handling the models in different data structures and handle them separately as done in [2]. In SEBM, actor entities that are only defined for visual domains in [4] are modelled for multimodal domains. These entities give us the ability to express and query the structure of events in multimodal domains. Moreover object characteristics that involve a particular feature of an object or relation of an object with other objects are also introduced in SEBM for multimodal domains different than [7] which considers only visual domain. We propose some algorithms to query these stored relationships of video objects and events and follow divide and conquer approach in query processing to answer complex, nested, conjunctive, spatial, temporal, content-based and possibly fuzzy video queries. This approach gives us the ability to deal with much more complex and compound multimodal queries different than ones in [2] and [3]. We support these algorithms with an implemented querying prototype system that uses SEBM while modelling the data.

The rest of the paper is organized as follows: Section 2 presents how SEBM models the video data with exploring video segmentation, video entities and video actions. In Section 3, query processing on SEBM is investigated and content based, spatio-temporal, hierarchical and fuzzy queries are explored. Throughout the parts 2 and 3 the usage of SEBM is also explored. The last section provides conclusion with some future extensions of our model.

2 Modelling the SEBM Video Data Model

Single video is composed of sequential frames, which are individual images. Each frame has individual image properties like color or shape. Every image can contain objects, positioned on the image. However, when we arrange these images sequentially, we can see that these objects can be told to do something semantically and are part of some events. In [5], it is explained that human interprets the three modalities of the video data by using an event structure and we have developed our SEBM video data model by considering this fact. We position the video events at the core part of the SEBM video data model and propose that we can model every reality in videos by expressing them as video events or relating them with video events. SEBM is a kind of translation of human sentences, which s/he uses while interpreting the video data, to the video database model. We developed the SEBM video data model as a combination of five different sub-models. Fig.1 shows the hierarchical structure of these sub-models that we use while constructing the SEBM:

1. Video Sequence Sub-Model (segments the video according to the meaning as shown with the link-c)
2. Video Shot Sub-Model (segments the sequences according to low-level features of the them as shown with the link-d)
3. Video Object Sub-Model (stores the objects globally and access the spatio-temporal information of the objects through events, as shown with the link-a and f)
4. Video Object Characteristics Sub-Model (stores the objects features and their relationships, as shown with the link-b)
5. Video Event Sub-Model (stores the events under corresponding video segment with spatio-temporal information and associate their object structure, as shown with the link-e and f)

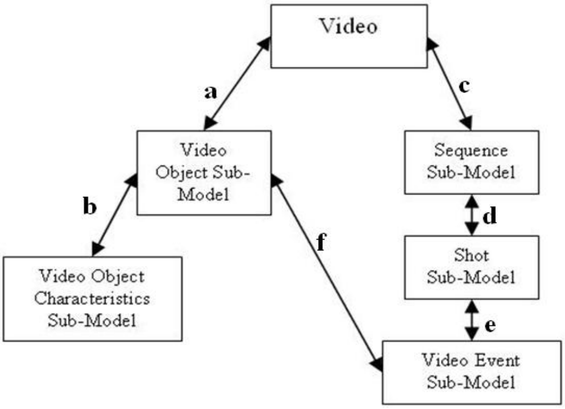

Fig. 1. Hierarchical Structure of SEBM Sub-Models

In [6], it is proposed that successful content-based video data management systems depend on three most important components: key-segments extraction, content descriptions and video retrieval. In our model, we firstly segment the whole video into meaningful intervals temporally according to the semantic; for example "party", "in house conversation" or "escape from prison". We call each of these meaningfully different segments as "scene" or "sequence". Zhang et. al. define video shot as a video sequence that consists of continuous video frames for one camera action and explain how camera changing or motions and special edit effects cause shot boundaries, [9]. By using this definition, we further temporally divide the sequences into smaller segments called "shots" according to the physical changing in parts, like colour. For example, if the camera or background changes at a particular point in a particular sequence, we split the sequence from that point. Since this splitting is done according to physical changing, automatization is much easier in that part than sequence segmentation. On the other hand, sequence segmentation requires much more artificial intelligence, since it is done according to the semantic. In our prototype system, while segmenting the video data, semi-automatic approach is used. Firstly, video is segmented according to the background changes automatically by using IBM MPEG-7 Annotation Tool, [10]. These segments are called shots and expressed in video shot sub-model. Then, these shots are manually grouped into sequences according to their semantics and expressed in video sequence sub-model. From shots, sequences are created. Every shot is a member of exactly one sequence. At the end of grouping, if the video has N number of shots and M number of sequence, N.M sequence-shot pairs are created. Video is a set of sequence-shot pairs.

Specific entities that are visible or tangible are called Video objects. Ekin et. al. call them as action units or interaction units, [4]. If we see or touch the entity in real world, we can declare it as a video object. For example; John, Kylie, t-shirt, hamburger etc. are video objects and expressed in video object sub-model. In SEBM, every video object has a list of roller event list which one can access the spatio-temporal information of the object directly through the object itself. This list is created automatically while creating the events. While objects are used to structure a particular event, the ID of the event is added to the roller event list of that object. The object is sequentially searched through the present objects and found in O(n) time. Since only the created objects can be added to the events, there is no possibility of not to find the object in present video objects.

Objects may have features like "John is 25 years old", "Kylie has blue eyes" or may have relationships with other objects like "Kylie is the sister of John", "John is a pilot of the plane". These relationships and features are expressed in video object characteristics sub-model in SEBM. Since video objects are stored directly under the video in hierarchy (Fig.1), all of the objects can be used to create a relation independent than their spatio-temporal information. For example, object *John* may have a relation *brotherof* with *Kylie*. So a video object named John will have a relation VOC={brotherof, Kylie}in SEBM. If *ball* has a *color blue*, then the object named *ball* will have a feature VOC={color, blue}.

Specific events that occur in a certain place during a particular interval of time are called video events. Video events stored under a particular shot of the video in SEBM and expressed in video event sub-model. As a result, particularly, every event belongs directly to some specific shot and indirectly to some specific sequence (Fig.1). Video events have event structure, which stores the subject, object, semantic place where the event occurs, accompanied object and directed object. For example in the sentence “John kills Kylie with a knife in the park”, “John” is the subject, “Kylie” is the object, “Knife” is the accompanied object, “park” is the semantic place where the event occurs. Every event has start and end times labels, which is the temporal information for that event. Every event has visuality and auditorility flags that correspond their modality information. Moreover every event has keywords part to store extra information of an event or the words that can be heard in the auditory events. Keywords are free texts. If some audio event such as *John said “Hello”* must be declared, the word *hello* must be put in the keywords. The spatial information is stored in temporal and spatial region list (TSRL). The members of this list are minimum bounding rectangles (MBR) labelled by a specific time in a video. All the objects that belong to the event are positioned in defined MBRs. Textual information in the video is embedded into the model as making a new event named “isWritten” and putting the written text into the keyword field of the sub-model. The spatio-temporal information of the text is also included in the TSRL of the event.

3 Flexible Querying Using the SEBM Video Data Model

Semi-automatic extracted information from video(s) is stored in a video database and then queried and accessed, [5]. However, there are some issues while considering these processes. First of all, how the information is extracted and stored in a database? Then, when you store all information that you need about videos in a database, which types of queries are supported? How are these queries processed? Since there is no standard querying language or query models that you can use in video databases, querying the video database is a challenging problem. One possible solution is to develop a video data model that fits into the area of interest. We try to solve the problem of modelling the multimodal video data by using SEBM. Then we try to find the answers of the queries like; what is going on in the video, who are the people in the videos, what are the relations between them and what is happening when and where? In our developed prototype system, the database is queried about visual, auditory and textual contents. Spatio-temporal relations between events and objects are also queried. Moreover, hierarchical and conjunctive contents are also queried. Besides these, we handle the structural queries about objects and events. Fuzzy queries are also solved on SEBM prototype system.

3.1 Content Based Queries (Simple, Complex and Hierarchical)

Content Based Queries are about the content of the information that we extracted from the videos. With these queries, we can retrieve events, objects and their relationships in the video.

Definition 1 (Simple Content Based Queries). The content based queries, which are about only one of the SEBM sub-models. For example, “What information we store in a specific model entity (sequence, shot, object, characteristic or event)?” or “which model entity has the specific information that we supply, like name or timestamps (start or end points)?" can be two example queries. Since we have five different sub-model we also have five different SQs:

1. Queries about Video Sequence Sub-Model (SQ1)
2. Queries about Video Shot Sub-Model (SQ2)
3. Queries about Video Event Sub-Model (SQ3)
4. Queries about Video Object Sub-Model (SQ4)
5. Queries about Video Object Characteristics Sub-Model (SQ5)

Definition 2 (Complex Content Based Queries). The content based queries derived from simple queries and about the relationship information between sub-models. They are formatted by using the relationships between sub-models from simple queries as shown in Table.1.

Table 1. Complex Query Structures. Every complex query (CQ1-4) is constructed by using the relationships that are given in the column of formation structure. For example in order to query the relation between SQ2, SQ4 and SQ5, firstly the relation of (SQ2-CQ2) is used, and then CQ2 can be replaced by the structure of (SQ3-CQ1). The query becomes SQ2-SQ3-CQ1 and then CQ1 can be replaced by the structure of (SQ4-SQ5). The query becomes SQ2-SQ3-SQ4-SQ5. The result shows us that the relation should also consider the answer of SQ3, which is about video event sub-model. This complex query formation is shown in Fig.2.

Complex Query Type	Formation Structure	Example Queries
CQ1 Relations between objects and characteristics	SQ4 – SQ5	• Which kind of characteristics has the object named John? • Who is the brother of sister of John? • Who is the brother of the person seen in the event of eating between timestamps [3.0, 30.0]?
CQ2 Relations between events and objects, or events and characteristics	SQ3 – SQ4 SQ3 – SQ5 SQ3 – CQ1	• When does John give the book to the brother of Jimmy? • When is the text "Happy Birthday" seen on the screen? • When does John say "Hello" to the sister of Jimmy? • When does John crash to the chair where Jimmy sits?
CQ3 Relations between shots and events, objects or characteristics	SQ2 – SQ3 SQ2 – SQ4 SQ2 – SQ5 SQ2 – CQ2	• In which shot John gives the Hamburger to the sister of Jimmy?
CQ4 Relations between sequences and shot, event, object or characteristic	SQ1 – SQ2 SQ1 – SQ3 SQ1 – SQ4 SQ1 – SQ5 SQ1 – CQ3	• Which sequences have dinner shots? • In which sequence does John drive the car? • Give me the timestamps of the sequence where John fights with Kylie's brother.

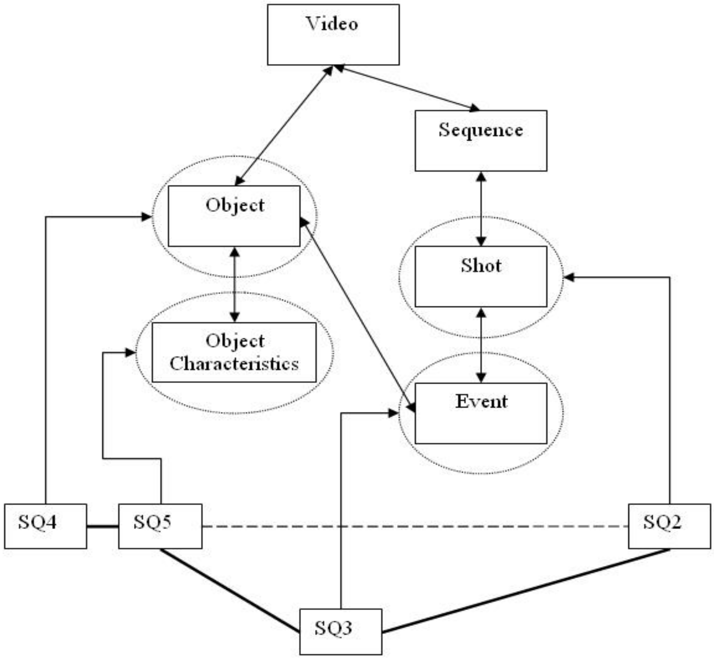

Fig. 2. Example Relation between CQs (formatted from SQs) and SEBM. (Video event sub-model is a bridge sub-model between Video shot sub-model and video object sub-model.)

These four types of CQs and five types of SQs can be used to compose much more complicated compound queries. Queries can be built to create conjunctive or disjunctive queries. For example, “When does John give Kylie’s brother a hamburger while (and) Jimmy is saying hello?”, “What are the sequences where the text “Happy birthday” appears on the screen or “strange noise” is heard at the background?”. Since each of CQs is composed of SQs and possibly other CQs, the SQs are processed first and then the results of these SQs are merged to find the answer of CQs. That is, the divide and conquer technique is used. For example, assume that we have a query like “When does John give Kylie’s brother a hamburger while (and) Jimmy is saying hello?” Firstly, we find Kylie’s brother, Tom. Next, we find the intervals where John gives Tom a hamburger for example [t1,t2]. Next, we find when Jimmy says hello, for example [t3, t4]. Lastly, we merge (intersect) both intervals and find the common time interval [t5, t6], which is the result.

Video sequence-shot-event-object hierarchy is also queried in content-based queries and called as hierarchical queries. The hierarchy of video parts is shown in Fig.1. From the users point of view, either sequences or shots can be seen as big video parts containing small video parts i.e. events. Hierarchical queries are some kind of content-based queries but containing the hierarchy of video entities. For example:

- What is happening in the Party? (Assume “Party” is a sequence name and user wants to know which events take place in this sequence)

- Which songs are heard in song contest? (Assume that “song contest” is a sequence and every song time interval is labelled as a shot. In this example sequence-shot hierarchy is queried)

Hierarchical queries are handled by matching the correct links of SEBM video data model shown as a, b, c, d etc. in Fig.1 with query structure. Every link is searched to find the correct correspondence of the hierarchy.

3.3 Spatial, Regional and Trajectory Queries

While annotating the video region, we store the spatial information in video event sub-model, i.e. where a particular event occurs on the video screen. MBRs are used here. Since video events occur not only on a single frame but also on continuing frames, we take more than one particular rectangle information where each rectangle includes the event and related objects. Adjacent two rectangle, give us the trajectory information of the event in the screen. If there is a trajectory change, we add another rectangle information.

Region information is discrete. But when we have queries like “Find the events that occur in a given rectangle on the screen” or “Give me the intervals where the plane passes the rectangle that I draw on the screen”, we take the rectangle area and look if it contains a rectangle that a particular event has.

SEBM prototype system runs the following algorithm for the regional queries.

```
1. For every event,
    a. For every region information of the event
        i. If the region that user enters while creating the query is equal to or contains the region of the event, add the event to the result list. For example in Fig.3, first region of the trajectory contains the region of the event
```

SEBM prototype system runs the following algorithm for the trajectory queries:

```
1. For every event,
    a. For every adjacent two region information of the event,
        i. If the first region that user enters while creating the query is equal to or contains the i$^{th}$ region of the event and the second region that user enters while creating the query is equal to or contains the i+1$^{th}$ region of the event as shown in Fig.3, add the event to the result list.
```

Spatial queries consists keywords for comparing two events TSRL information such as left, top, top-left or top-right. Other relations such as right or below can be thought as the inverse of left or top. Spatial relations consider the timestamps of TSRL information different than regional and trajectory queries, because the queried events must happen at the same time in the same video. Since the SEBM video data model stores the TSRL information not storing whole trajectory but only beginning, ending and direction changing locations, the whole trajectory of the events are created in run time of SEBM prototype system. Trajectory creation process considers time with width and height changing ratios through time. After trajectory creation process is done, the TSRL information consist not only beginning and ending locations but also intermediate locations.

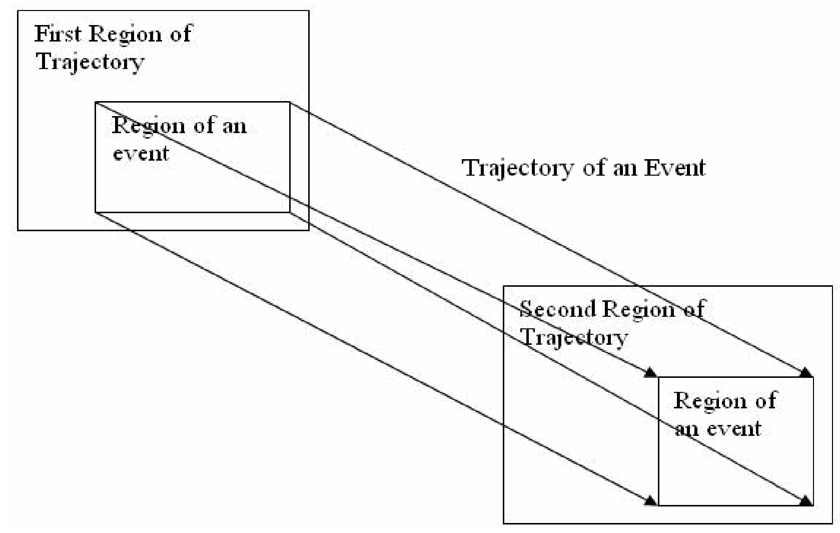

Fig. 3. Trajectory Query Relationship

SEBM prototype system runs the following algorithm for spatial queries:

```
1. For every two event,
   a. For every region of the first and second event that has same timestamp (or
      approximate timestamps)
      i. If the centers of regions provide the directional relations defined in [2]
         (left, right, top or bottom) then the timestamp is added to the result list.
```

3.4 Temporal Queries

Like spatial queries temporal queries are for querying the comparison. But they are for comparison of timestamps between two video entities. Durak uses temporal relationships as “before”, “meets”, “overlaps”, “during”, “starts”, “finishes”, “equal” which are defined formally by Allen in [19]. In order to make comparison between timestamps, some operations on intervals must be done, for instance union or intersection. Pradhan et. al. define these interval operations in [13]. For example (comparison keywords are written in italic):

Assume that the user wants to solve the following query:

- Find me the intervals when John says “Hello” to his brother before Kylie is seen on the screen and “Happy Birthday” is written on the board.

Query is divided into three sub-queries to solve them individually and after solving to combine the result by the user.

- Find me the intervals when John says “Hello” to his brother.
- Find me the intervals when Kylie is seen on the screen.
- Find me the intervals when “Happy Birthday” is written on the board.

Every query returns a temporal set as an answer.

Definition 3 (Temporal Set). The set of temporal information. Temporal information consists of video identification (video name or video place), start time and end time. Temporal operators (intersection, before, equal, meets, finishes, starts, overlaps, during) are applied according to the given algorithm below:

1. Take the first two temporal sets. Construct a new set with the values explained in 1.a.
   a. Apply corresponding temporal relation explained as in [5] to every member of the first set with every member of the second set. If the relation returns a new temporal value and the intervals belong to the same video add them to the new set.
2. Add the new constructed set to the global temporal set. Apply the first step, if global temporal has more than one set.

The algorithm runs in $O(n^2)$. Because the algorithm runs on every set of global set that have O(n) members. Global set has also O(n) members.

3.5 Fuzzy Queries

The nature of human interpretation of real world is not always discrete. Because there is always some unknown spatial information, the video data querying system should consider possible fuzzy queries. Fuzzy queries are constructed by giving some threshold value about the spatial relationship and they may involve some fuzzy conditions as explained in [3]. Threshold value may be given between 0.1 and 1.0 or by using some keywords such as “just”, ”slightly”, etc.

For the fuzzy spatial relationships, Köprülü et. al. define membership functions by using membership values, [3]. The membership value of the relationship is calculated by using the angle between the line connecting the centres of MBRs and the x-axis. The membership functions are given in Table.2.

Table 2. Fuzzy Spatial Relationships

Relation	Angle	Membership Value
Top	arctan(x/y)	1-(angle/90)
Left	arctan(y/x)	angle/90
Top-Left	arctan(x/y)	1-(abs(angle-45)/45
Top-Right	arctan(y/x)	1-((angle-45)/45

Fuzzy queries are processed according to the given algorithm below:

1. For every two event,
   a. For every two region pair of the first and second event that has same timestamp (or approximate timestamps)
      i. If the centers of regions provide the directional relations given in part 3.3 and the threshold value is smaller than calculated membership value (using Table.2), then the timestamp is added to the result list.

For example:

- Show me the part of the video where the plane passes John just above his head. (Similar to spatial queries, the trajectory information of both plane and John is implicitly found as given in part 3.3. Then the positions of both video objects in a particular time are compared according to given membership value to find the solution by using Table.2)
- Show me the part of the video where the text "Happy Birthday" is seen around upper left corner of the screen with a threshold value 0.5. (Regional querying algorithm given in part 3.3 is applied here considering membership value by using Table.2)

3.6 Compound Queries

All content based, hierarchical, spatial, regional, trajectory, temporal or fuzzy queries may be joined to form more complex queries, which is called compound queries. Even compound queries can form more complex compound queries. This formation can be structured by using temporal and spatial relationships operators explained in previous parts. By solving and combining the partial answers of compound queries, the final answer can be formed. This approach is similar to that one we follow while solving the content based queries. For example:

- Show me the part of the video where John's brother who is the friend of the person seen at the upper left of the screen between 30th and 40th seconds is standing near the car on a chair *and* Kylie is walking through the door. (Complex Content based and trajectory queries are combined. Answers of both queries, which are temporal sets, are found and intersected)

Usage of SEBM gives us the ability to solve much more complex and nested queries different than [2 and 3]. We support this idea by implementing the SEBM prototype system, which uses XQuery [12] on Berkeley DB XML [11]. Since the user can form compound queries and answer them by using querying interface of SEBM prototype system, we have shown that SEBM can alter query intensity. SEBM prototype system shows the success of SEBM on compound querying rather than other systems like [14, 15 and 16].

4 Conclusion

The SEBM video data model makes it easy and effective to store the structural semantic information in a video database and then query the database. This video data model can be adapted to various domains, because it is based on understanding of the human about a particular video. Visual, auditory, and textual information in video are all considered in our model. Difficulties on modelling the structure of the video are overcame by using several sub-models. Very tightly coupled relations among these sub-models result in much more information embedded into the model than treating each independently. Query diversity is supported in our model. Content based, fuzzy, spatial and temporal queries are all supported. Automatization in annotation of the video is part of our implementation. In our prototype system, implemented with Java, XML is used to model the information in video and Berkeley XML DBMS is used to store and retrieve video information. Automatization in annotation of the video is part of our implementation. IBM MPEG-7 Annotation Tool is used for this purpose. For querying, XQuery facility of Berkeley XML DBMS is utilized. We plan to improve the fuzzy querying part by including the fuzzy relations and adding fuzzy features to the sub-models of SEBM.

References

[1] C. Snoek, M. Worring, "Multimodal Video Indexing: A review of the State of Art", Multimedia Tools and Applications, 25, pp: 5-35, 2005.

[2] N. Durak, "Semantic Video Modeling And Retrieval With Visual, Auditory, Textual Sources", Ms Thesis, METU, 2004.

[3] M. Köprülü, N.K. Çiçekli, A.Yazıcı, "Spatio-temporal querying in video databases". *Inf. Sci. 160(1-4),* pp.131-152, 2004.

[4] A. Ekin, M. Tekalp, and R. Mehrotra, "Integrated Semantic–Syntactic Video Modeling for Search and Browsing" in IEEE Transactions on Multimedia, VOL. 6, NO. 6, December 2004.

[5] H. Öztarak, "Structural and Event Based Multimodal Video Data Modelling", Ms Thesis, METU, 2005.

[6] D. Tjondronegoro, P. Chen, "Content-Based Indexing and Retrieval Using MPEG-7 and X-Query in Video Data Management Systems", World Wide Web: Internet and Web Information Systems, 5, 207–227, 2002.

[7] E. Oomoto and K. Tanaka, "OVID: Design and implementation of a video-object database system," IEEE Trans. Knowledge Data Eng.,vol.5, pp.629–643, Aug. 1993.

[8] Ö. Ulusoy, U. Güdükbay, M. Dönderler, Ş. Ediz and C. Alper, "BilVideo Database Management System", Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004.

[9] Chengcui Zhang, Shu-Ching Chen, Mei-Ling Shyu, "PixSO: A System for Video Shot Detection", Proceedings of the Fourth IEEE Pacific-Rim Conference On Multimedia, pp. 1-5, December 15-18, 2003, Singapore.

[10] IBM MPEG-7 Annotation Tool Web site, www.alphaworks.com/tech/videoannex, Last date accessed: September, 2005.

[11] Berkeley DB XML Web Site, www.sleepycat.com Last date accessed: September, 2005.

[12] XQuery Web Site, www.w3.org/XML/Query, Last date accessed: September, 2005.

[13] Pradhan S., Tajima K., Tanaka K., "A Query Model to Synthesize Answer Intervals from Indexed Video Units", IEEE Trans. on Knowledge and Data Eng. Vol.13, No.5, pp. 824-838, Sept./Oct. 2001.

[14] S. Hammiche, S. Benbernou, M. Hacid, A. Vakali," Semantic Retrieval of Multimedia Data", MMDB'04, November 13, 2004, Washington, DC, USA.

[15] M.Lyu, E. Yau, S. Sze, "A Multilingual, Multimodal Digital Video Library System", JCDL'02, July 13-17, 2002, Portland, Oregon, USA.

[16] T. Kuo and A. Chen, "Content-Based Query Processing for Video Databases", IEEE Trans. on Multimedia. Vol.2, No.1, March 2000.

[17] Young F. Day, Serhan Dağtaş, Mitsutoshi Iino, Ashfaq Khokhar, Arif Ghafoor, "Object Oriented Conceptual Modeling of Video Data", Proc. Data Eng. (DE '95), pp. 401-408, 1995.

[18] C.G.M. Snoek, M. Worring, "Multimedia event based video indexing using time intervals" Technical Report 2003-01, Intelligent Sensory Information Systems Group, University of Amsterdam, August 2003.

[19] J. Allen, "Maintaining Knowledge about Temporal Intervals", Communications of ACM, 26 (11), pp. 832-843, 1983.