A Hierarchical Video Annotation System

ChengZhi Xu[1], HuiChuan Wu[1], Bo Xiao[1], Philip C-Y Sheu[1,2], Shu-Ching Chen[1,3]

[1]State Key Laboratory of Software Engineering, Wuhan University, China
xcz911@gmail.com, rivrain@gmail.com, oboaix1985@163.com
[2]Department of EECS, University of California, Irvine, CA, USA
psheu@uci.edu
[3]School of Computing and Information Sciences
Florida International University, Miami, FL, USA
chens@cs.fiu.edu

***Abstract*—Semantic search service is generally considered to include text search, audio search, and video search. The annotation of a video is crucial to the success of video search and retrieval. With the precise annotation of video files and their semantic reasoning, more accurate search results according to the users' query intention can be obtained. In this paper, a prototype of such a video annotation system, called Semantic Video Annotation System (SVAS), is presented. SVAS uses a three-level annotation architecture and a semantic video search language called Semantic Query Description Language for Video (SQDL-V). SQDL-V engine based on SVAS is able to return more accurate search results in comparison to the formal video search method.**

Keywords-semantic search engine; video annotation; semantic reasoning; SVAS; SQDL-V.

I. INTRODUCTION

With the development of computer and network technologies, people increasingly depend on the use of search engines to look for interesting information. At present, Google, Yahoo, MSN, Baidu and other search engines have been able to solve the issue of massive text search. However, most of the existing search engines for video data still do not have effective methods. The main problem is the lack of effective means of establishing suitable index for video retrieval. At the same time, large-capacity storage devices and digital equipments, as well as the widespread use of multimedia technology, have generated huge amounts of video data. As a result of the excessive growth of data quantity and the lack of processing capability, it has become an urgent demand as how to effective utilize relevant video content. To address these problems and promote the development of new video applications, techniques to index, browse, and retrieve these massive amounts of video data, and to classify their corresponding semantic content are required.

Content-based video annotation which classifies the objects embedded in videos by their concepts or semantic meaning seems to be promising for video indexing, browsing, and retrieval [1][2][3]. Some studies separate the annotation data from the video data, and the annotations are labeled along the video timeline to form a linear structure. In [3], all these labels are stored in some independent XML documents. However, the annotations data is a single-layer structure, which does not have sufficient capacity to express the wealth of the information in the video, and which will disable the search engine to satisfy user's demand for video.

As the annotations are separated from the video, a user can tag the personalized annotations without changing the video file. Such services have been provided by, for example, the YouTube. The users of YouTube can tag their own comments to the video, voice-over information, even to link the video to other comments or videos. The main approach is to provide a number of annotation types for the user to choose. These types include the character dialog box (Speech Bubble), voice-over box (Note), and hint (Spotlight). This classification lets the users to annotate video in details. However, this approach still cannot accurately reflect all of the video information.

In this paper, a hierarchical annotation system is proposed and the annotation is stored into a database. A hierarchical annotation system has different size particle tags. A user can annotate video from macro- and micro- levels. A semantic video search language called Semantic Query Description Language for Video (SQDL-V) is also proposed. As being a structured query language, SQDL-V asks a user what he/she wants to search, not how to search, and it has abundant of semantic meaning, through which user can search and retrieve video from time relation, spatial relation and social relation.

The rest of this paper is organized as follows. Section II provides the related work which introduces main video annotation methods and some technology in business domain and academic world. In Section III, our proposed SVAS is discussed. In Section IV, we introduce SQDL-V and give some examples. The conclusions are given in Section V.

II. RELATED WORK

At present, there are three main video annotation methods: manual annotation, rule-based annotation, and machine learning method [4]. Traditional manual annotation is a single-layer structure, whose ability of semantic expressions is far from abundant. This method is not only

time-consuming and laborious, but also easier to be misunderstood by the users. At the same time, different people having different criteria will also destroy the consistency of the annotations. Though manual annotation is the worst method when efficiency and speed are considered, it has the best performance in the veracity field. Rule-based annotation methods use expert knowledge to classify the annotation [5][6]. The category of the annotations is an embryonic form of hierarchical annotations. Generally, a single-level category of the annotations cannot cover all the semantic content, and therefore it cannot meet the general-purpose requirements of video annotations. Machine-learning methods, based on studying the samples, establish all kinds of semantic models and extend these models to the entire video [7][8]. Although machine-learning methods reduce the manual operations, they rely on learning samples. These samples are provided by humans and annotated manually. Selecting inappropriate study samples will result in study failure. At the same time, machine learning methods rely on artificial intelligence. Only when there is a breakthrough development in artificial intelligence, machine learning methods are able to give its full potential in video annotations.

Many research groups have started to develop techniques for video search and chosen a multi-method to approach video search. A semantic video search engine which includes two search methods, namely concept-based and visual example-based was developed in [9], and an interactive video search system which is based on visual paradigm was proposed in [10]. Before searching video, a user needs to draw an image to describe the search conditions by a special tool, and then the system uses this image as an example data to match the video from the Web. This method is interesting, but drawing is too complex to a general user, and drawing an image cannot precisely express the user's demand.

The Carnegie Mellon University Informedia group proposed an interactive video search system based on merging storyboard strategies. This system uses four queries to stepwise refine the search result, such as query-by-text, query-by-image, query-by-concept, query-by-best-of-topic [11]. A multi-method strategy is more precise than the single-method, but a user needs to put more effort into the interactive search process, and it is more time consuming.

A new trend is to introduce the ontology into video search. The research group in the City University of Hong Kong proposed a model, called ontology-enriched semantic space (OSS) [12]. It matches user's keywords to ontology concepts, and induces these concepts to other relevant concepts for video search. However, the maintenance and update of ontology model are difficult.

Some video search systems use Graphic User Interface (GUI) to deal with search work, while some other systems use video search language to do this job. The research group in University of Michigan proposed a temporal query language for video data. Many temporal relationships have been defined, such as before, after, meet, start, during, finish [13]. Time relationship just one of important relationship in the video, but there are many other relationships that have not been mentioned by this language. In Hanoi University of Technology, a SQL-based query language combining object features and semantic events for video retrieval was proposed [14]. Its syntax is: SELECT <output> FROM <database> WHERE<condition>. However, it only outputs the frame of a video, and its condition part only compares an object attribute with other attribute or constant.

III. SEMANTIC VIDEO ANNOTATION SYSTEM (SVAS)

As mentioned earlier, though manual annotation is considered the worst method in the aspects of efficiency and speed, it has the best performance in the veracity field. Since veracity is one of the most important aspects in semantic search and our proposed SVAS (Semantic Video Annotation System) is developed for semantic video retrieval, the concerns have been on how to improve the efficiency and annotation speed in the precondition of high veracity. Therefore, SVAS adopts the manual annotation method.

SVAS is able to supply the video annotation data for the semantic video search engine. The users of the semantic video search engine can use the Semantic Query Description Language to search the video information from the database indirectly.

A. Objectives

Most of the current existing video annotation systems are video scenario based. Notes can be added to the time segments on a video timeline. A user can also view the video clip, mark a time segment, playback the segment, or attach his/her written notes to the segment. All of the annotation information is in the video level and will be mixed together, which makes it very difficult on semantic video retrieval. That is, the users cannot effectively and easily get what they want. Resolving this problem is our main objective. In addition, a semantic video annotation tool at least should support the following functionality:

- Annotate a video effectively;
- Divide a video into a number of scenes;
- Divide a scene into a number of frames;
- Develop a unified schema for semantic video annotation;
- Import & export the semantic video annotation information;
- Annotate a scene and a frame solely;
- Support as many video formats as possible; and
- Provide convenience on annotation data transfer.

B. Architecture

Almost all of the existing system architectures are only video-level based. In our proposed SVAS, an innovative multi-level annotation strategy is developed, which are shown in Fig. 1. All the videos have their general architecture. Below the video-level, there are triple-level structures including scene level, shot level, and frame level. In SVAS, the scene and shot are integrated as one concept to form the triple-level architecture.

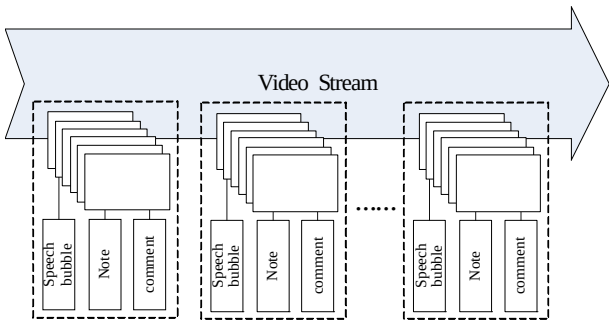

Figure 1. Multi-level structure.

C. Components

SVAS includes three modules: the video control and video information collection module, the video annotation module, and the database module. The first two modules are connected with the video stream, while the last one stores the video annotations. The Eclipse platform is used as the development environment using java programming language and MySQL for the database. Java Media Framework (JMF) is adopted as the video processing framework so SVAS can well support the AVI format. The general architecture of SVAS is presented in Fig. 2.

The video control and video information collecting module can control the video states in real-time. These include play, pause, stop, and replay. Moreover, this module can get the related information of the current video file, such as video name, video duration, and video frame timestamps.

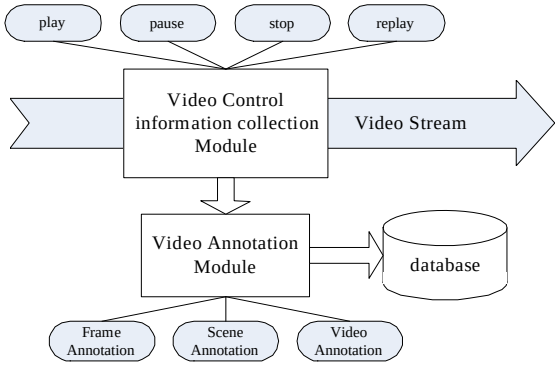

Figure 2. Architecture of SVAS.

D. Annotation Process

The video annotation module can annotate the video file based on the previous module. Thanks to the analysis of the video architecture, the annotation module works in three steps.

In the first step, it starts to play the video, and chooses a frame by pausing the video. At this time, the related information such as the video name and current frame number will be returned by the first module. If that frame is the key frame in the video or scene, the user can also add some alias information to distinguish each other. Then the user can annotate the location of the frame where it occurs. Also, if a user has some additional information to annotate, the user can add them into the description field. This is just the general annotation of the frame. The tool can even annotate the object in the frame, and the relationships of the objects, and the events in the frame. First, the object annotation includes the object name and object concept, and the object relationship includes the spatial relationship and logic relationship. The spatial relationship describes the location relationship of the two objects, and the logic relationship defines the social relation of the two objects. The events in the frame can be annotated in the form of event definition and event relationship definition. An event definition includes two objects and an action, which these three elements constitute an event. An event relationship can point out the order of two events.

In the second step, after the annotation of a serial of frames, a user can choose a number of successive frames to construct a scene, and then the tool can annotate the new constructed scene. First, the definition of a scene includes three elements, namely the key frame, start frame, and end frame. Of course, the video level information can be restored in the frame level (i.e., we can ignore the video information here). If a user has some additional information to annotate, the user can add them into the description field. This is just the general annotation of the scene. In the next step, the tool can annotate the object in the scene, and the relationships of the objects, and the events in the scene. Like first step, the tool annotates the object, object relationship, and events once again. However, this is not a simple repeat. In this step, the object will be the scene level object, and the events will be the scene level events. User can choose the kernel objects and kernel events as the scene objects and events.

In the third step, after the scene level annotation, the tool will deal with the top level annotation—video level annotation. As we all know, the video level annotation should be very obvious and usual. However, in our system, the video annotation is very similar to the scene annotation. First, a user can make a general annotation for the video, which includes the video name and video category. Then, we can do the annotation job once again, which has done in the above step. The difference is that everything is on the video level. At the same time, the relationship of the objects in the scene level and video level is only a logic relationship. The spatial relationship does not exist any more.

E. Database Design

In our tool, all the annotation data can be stored in the database in real-time. Therefore, the database of the tool is also designed to be triple-level. In order to be compatible with other annotation system, our tool can export annotation from the database to an XML file. Table I, Table II, and

Table III show the frame-level, scene-level, and video-level tables in the database.

TABLE I. Tables at the frame level

Table name	Description
frame	*Frame basic information*
frameobjects	*Frame object information*
frameevents	*Frame event information*
frameeventrelations	*Frame event relationship, sequence, parallel, etc.*
framelogicrelations	*Frame object logic relationship, social relationship*
framespatialrelations	*Frame object spatial relationship, location relationship*

TABLE II. Tables at the scene level

Table name	Description
scene	*Scene basic information*
sceneobjects	*Scene object information*
sceneevents	*Scene event information*
sceneeventrelations	*Scene event relationship, sequence, parallel, etc.*
scenelogicrelation	*Scene object logic relationship, social relationship*

TABLE III. Tables at the video level

Table name	Description
video	*Video basic information*
videoobjects	*Video object information*
videoevents	*Video event information*
videoeventrelations	*Video event relationship, sequence, parallel, etc.*
videologicrelation	*Video object logic relationship, social relationship*

F. User Interface

Our annotation tool has a human-base interface, whose screenshot is displayed in Fig. 3. The main interface has been divided into two parts. The left part is the video player panel, and the right part is the triple-level annotation panel.

The video player has many methods to play video, such as play, pause, forward frame play, and backward frame play. Through this player, a user can get the detailed information from the video. The annotation panel has three property sheets according to the three-level annotations. A user can tag the video in the sequence accordance with the three steps introduced earlier, or any sequence the user needed. Meanwhile, the user can store the annotations into the database.

IV. SQDL-V AND QUERY METHOD

Most systems use GUI to input user demand, this is convenient. But if we want assign a search plan or batch process, then GUI cannot do anything about it. Our annotations store in relational database, so a simple approach is use SQL to search information in the database directly. But our annotation is a three-level structure; the database contains too many tables, the query of SQL must be very complex, and complex language is too hard for user to use. So we proposed a special search language for hierarchical video annotation, whose name is Semantic Query Description Language for Video (SQDL-V). In this section, we introduce SQDL-V and explain how SQDL-V converts to SQL.

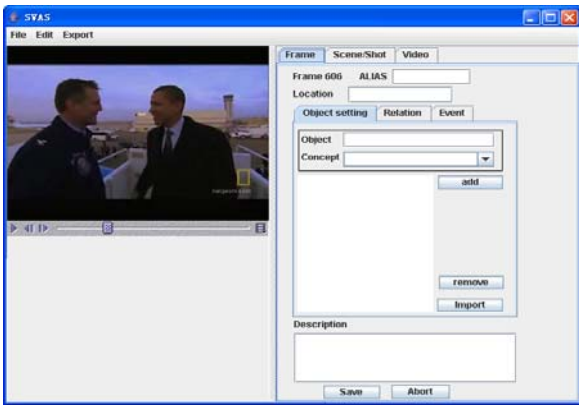

Figure 3. Graphic user interface.

Like SQL, SQDL-V is a structured query language. SQDL-V contains many semantic phrases, and so the user can expediently search information at each annotation level. As the information is stored in a database, ultimately SQDL-V will convert to SQL to search the database.

A. SQDL-V grammar

SQDL-V expands the basis SQL by introducing the semantic representation service environment and variables to represent all kinds of relational. SQDL-V grammar is similar with SQL grammar with almost the same basic structure. The SQDL-V grammar is shown in Table IV. As can be seen from Table IV, SQDL-V Basic architecture consists of three parts: *WITH Clause*, *WHERE Clause*, and *SELECT Clause*.

TABLE IV. SQDL-V grammar

SQDL-V_expression::=With_Clause Where_Clause Select_Clause
*With_Clause ::= <WITH> DataSource (, DataSource)** *DataSource ::= Input_Statement | Variable_Statement* *Input_Statement ::= <INPUT> DataType ObjectName "=" String* *Variable_Statement ::= DataType ObjectName ["= (" ParamName ("," ParamName)* ")"]* *ParamName ::= VariableName | Constant* *VariableName::= ObjectName ("." AttributeName)** *Constant ::= String | Integer | Float*
*Where_Clause ::= <WHERE> Where_Item ("," Where_Item)** *Where_Item ::= Condition | Function* *Condition ::= ParamName Relation_Op ParamName* *Relation_Op ::= "=" | ">" | ">=" | "<" | "<=" | "!="* *Function::=FunctionName "[" ParamName ("," ParamName)* "]"*
Select_Clause ::= <SELECT>VariabletName

In the *WITH Clause*, the search level (frame, scene, or video) and the variables in that level are defined. If needed, the user input can also be defined using the *INPUT* phrase. In the *WHERE Clause*, the restriction is expressed in the object-relational concept. An object is a hierarchical

structure, and the dot operator is used to refer to an object attribute. Hence, the form of search restriction will look like "*f.e1.object1=x*", where *f* (frame) are three-level hierarchies. In the expression *m.n*, *m* is the father hierarchy or an object, and *n* is the child hierarchy or the attribute of *m*. Moreover, the relationship between two different objects can be expressed with *relation[object1, object2]*, too. Finally, in the *SELECT Clause*, the search target is specified. The target is always *id* of a level, such as frame *id*, scene *id*, or video *id*.

B. SQDL-V function and predicate

In order to make SQDL-V more precise and standardized user inputs, some functions and predicates about time, spatial, and logical relations are designed as shown in Table V.

TABLE V. SQDL-V functions and predicates

Time relation 1. *Sequence*[event1,event2] Description: Event1 occurred before event2. 2. *Parallel*[event1,event2] Description: Event1 and event2 occurred at the same time.
Spatial relation 1. *Left*[object1,object2] Description: Object1 is to the left of object2. 2. *Down*[object1,object2] Description: Object1 is under object2. 3. *Include*[object1,object2] Description: Object2 is in object1, or object1 includes object2. 4. *Front*[object1,object2] Description: Object1 is to the front of object2.
Logic relation 1. *f.logicRelation = relation, f.logicRelation.object1 = x, f.logicRelation.object2 = y* (relation, x, and y are defined in the WITH clause.) Description: It is the frame-level logic relation between x and y. 2. *s.logicRelation = relation, s.logicRelation.object1 = x, s.logicRelation.object2 = y* (relation, x, and y are defined in the WITH clause.) Description: The scene-level logic relation between x and y. 3. *v.logicRelation = relation, v.logicRelation.object1 = x, v.logicRelation.object2 = y* (relation, x, and y are defined in the WITH clause.) Description: The video-level logic relation between x and y. 4. Logic relation can be defined by the user freely. Description: The logic relation can be defined by the user.

C. SQDL-V examples

Here, several SQDL-V examples in each level are presented as follows.

Example [1]: A frame-level query: The select condition is about spatial relation of two objects. This example is to search a frame which contains two objects, namely President Obama and the First Lady, and these two objects have a spatial relationship of side by side.

WITH frame f
INPUT String x="President Obama", String y="First Lady"
WHERE left[x, y]
SELECT f.id

Example [2]: A frame-level query: The select condition is about time relation of two events. This example is to search a frame which contains two events, namely the First Lady shaking hand with the Queen and the journalist taking the pictures, and these two events have a time relationship of occurring at the same time.

WITH frame f, event e1, event e2
INPUT String x="First Lady", String y="Queen", String action1="shake hands" String m="Journalist", String action2="take picture"
WHERE f.e1.object1=x, f.e1.object2=y, f.e1.action=action1, f.e2.object1=m, f.e2.action=action2, parallel [e1, e2]
SELECT f. id

Example [3]: A scene level query: To search a scene which contains two objects, namely Obama and Air Force One, and these objects form an action which Obama gets off Air Force One.

WITH scene s, event e
INPUT String x="Obama", String y="AirForceOne", String action="get Off"
WHERE s.e.action=action, s.e.object1=x, s.e.object2=y
SELECT s.id

Example [4]: A video-level query: To search a video which includes two objects, Obama and Jason, and these two objects have a logic relation of colleagues.

WITH video v, event e
INPUT String x="Obama", String action="Jason", String relation="colleague"
WHERE v.logicRelation=relation, v.logicRelation.object1= x, v.logicRelation.object2 = y,
SELECT v. id

D. SQDL-V conversion

In the above examples, the SQDL-V sentence is based on an object-relational database. However, our database is not based on object-relational database. Therefore, we have to convert the object relation into the none-object relation. Take Example [1] as an instance. Before the conversion, we parse and extract SQDL-V expression features, and organize them as a form, which is displayed in Table VI.

TABLE VI. SQDL-V expression features

WITH CLAUSE	**VARIABLE**	**TYPE**	**VALUE**
	X	String	President Obama
	Y	String	First Lady
WHERE CLAUSE	**PREDICATE**	**PARAM1**	**PARAM2**
	left	x	Y
SELECT CLAUSE	select	f.id	

In this process, the SQDL-V sentence is divided into three parts: *WITH* clause, *WHERE* clause, and *SELECT* clause. These parts map the basic elements of the SQL sentence and are used for the conversion.

Step1: Recognize the search level from the variable type in the *WITH* clause;
Step2: Extract the restriction from the *WHERE* clause;
Step3: Match the predicate with one of the relations, and decide which table needs to be searched; and
Step4: Integrate these factors to the SQL sentence.

After these four steps, the SQDL-V sentence can be translated into an SQL statement as shown below. After executing the SQL statement, the result data can be obtained.

SELECT frameID FROM framespatialrelations
WHERE spatialRelation='left' AND object1='President Obama' AND object2='First Lady';

V. THE CAPABILITY OF SVAS & SQDL-V

SQDL-V is designed for SVAS. Comparing with the traditional video search method, the combination of SVAS and SQDL-V provides a much more effective mechanism for video index. In the traditional condition, the video file is annotated with single level, and the search process is also based on the single level. The video search results are limited and not accurate. However, with the annotation of the SVAS and the query based on SQDL-V, the video search results are much more accurate. Moreover, in SVAS, the user can not only query for videos, but also query for frames and scenes, which can match the user's requirement in a video.

We have compared the traditional video query method with the SVAS SQDL-V query method. First, the query for frames and scenes is the particular function of SVAS SQDL-V method. Second, when we search for a video, the traditional method can only index the results on video level. However, the SVAS & SQDL-V method can index the result on three levels (frame, scene and video). Therefore, more useful results can be returned. In one of the experiments, with the same database and the same keywords, the traditional method can return 10 videos, and 6 of them are not what the user need. On the other hand, the SVAS SQDL-V method can return 5 results, and only 1 of them is not accurate. Many experiments were conducted, and the outcomes demonstrate that SVAS SQDL-V method achieves more effective performance.

VI. CONCLUSIONS

In this paper, the SVAS semantic video annotation system is proposed. SVAS stores the annotation information in a database, rather than in an XML file, which is efficient and provides a parallel processing mechanism to facilitate synchronous annotations. Many people can cooperate to annotate a video at the same time. Meanwhile, SVAS annotation is a hierarchical structure, which is tagged according to the physical structure of the video, and represents the essence information of the video. At same time, a Semantic Query Description Language for Video (SQDL-V) is proposed to search a video at any level. SQDL-V is upper level language to SQL. It converts complex SQL to a simple and unified form so that the users can accept SQDL-V more naturally.

ACKNOWLEDGMENT

This research has been partially supported by National Science Foundation of China (NSFC) under grant No. 60873007 and the 111 Project of China under grant No. B0737.

REFERENCES

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

[2] C.-W. Ngo, H.-J. Zhang, and T.C. Pone, "Recent Advances in Content-Based Video Analysis," International Journal of Image and Graphics, p 445- 468, 2001.

[3] N. Dimitrova, H.-J. Zhang, B. Shahraray, M.I. Sezan, T.S. Huang, and A. Zakhor, "Applications of video content analysis and retrieval," IEEE Multimedia, vol. 9, no. 3, pp. 42-55, July-Sept, 2002.

[4] H.-J. Zhang, "Content-based video analysis, retrieval, and browsing," Book Chapter of Readings in Multimedia Computing and Networking, Academic Press, 2002.

[5] Y. Rui, T. Huang, and S. Mehrotra, "Constructing table-of-content for videos," ACM Journal of Multimedia Systems, vol. 7, no. 5, p 359-368, 1999.

[6] M. Yeung and B. Yeo, "Time-constrained clustering for segmentation of video into story units," Proc. of IEEE International Conference on Pattern Recognition, 1996, pp. 375-380.

[7] L. Xie, P. Xu, S.-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with domain knowledge and hidden markov models," Pattern Recognition Letters, vol. 25, no. 7, pp. 767-775, 2004.

[8] J. Yuan, J. Li, and B. Zhang, "Learning concepts from large scale imbalanced data sets using support cluster machines," Proceedings of the 14th Annual ACM International Conference on Multimedia, MM 2006, p 441- 450, 2006.

[9] Zavesky, Eric, Chang, Shih-Fu, "Columbia University's Semantic Video Search Engine," CIVR 2008 - Proceedings of the International Conference on Content-based Image and Video Retrieval, p 545-546, 2008.

[10] SF Chang, W Chen, HJ Meng, H Sundaram, D Zhong, "A fully automated content-based video search engine supporting spatiotemporal queries," IEEE Transactions on Circuits and Systems for Video technology, vol. 8, No. 5, september, p602-615, 1998

[11] Christel Michael G, Yan Rong, "Merging storyboard strategies and automatic retrieval for improving interactive video search," Proceedings of the 6th ACM International Conference on Image and Video Retrieval, CIVR 2007, p 486-493, 2007.

[12] Wei Xiao-Yong, Ngo Chong-Wah, "Ontology-enriched semantic space for video search," Proceedings of the ACM International Multimedia Conference and Exhibition, p 981-990, 2007.

[13] Hibino. S, Rundensteiner E.A, "A visual query language for identifying temporal trends in video data," Multi-Media Database Management Systems, Proceedings., International Workshop on, On page(s): 74-81, 1995.

[14] Le Thi-Lan, Thonnat Monique, Boucher Alain, "A query language combining object features and semantic events for surveillance video retrieval," Proceedings. 14th International Multimedia Modeling Conference, MMM 2008, p 307-317, 2008.