Semantic Extraction and Semantics-based Annotation and Retrieval for Video Databases

Yan Liu (liuyan@cs.columbia.edu) and Fei Li
(fl200@cs.columbia.edu)
Department of Computer Science, Columbia University

March 15th, 2001

Abstract. Digital video databases have become more pervasive and finding video clips quickly in large databases becomes a major challenge. Due to the nature of video, accessing contents of video is difficult and time-consuming. With content-based video systems today, there exists a significant gap between the user's information and what the system can deliver. Therefore, enabling intelligent means of interpretation on visual content, semantics annotation and retrieval are important topics of research. In this paper, we consider semantic interpretation of the contents as annotation tags for video clips, giving a retrieval-driven and application-oriented semantics extraction, annotation and retrieval model for video database management system. This system design employs an algorithm on objects' relation and it can reveal the semantics defined with fast real-time computation.

Keywords: multimedia database system, extraction, semantics annotation, semantics-based retrieval

1. Introduction

The rapid growth and wide application of video databases lead to challenge of fast video data retrieval upon user's query. Through surveying a variety of users of multimedia databases systems, Rowe et al. (L. A. Rowe, J. S. Boreczky and C. A. Eads, 1994) characterized the types of video queries and identified the following "indexes" that should be associated with the video data in order to satisfy the queries:

1. Bibliographic data: This category contains information about the entire video (e.g. title, source, abstract, subject and genre etc.) and the individuals involved with the production of the video.

2. Structure data: Structure data is hierarchical description of the video, such as *movie*, *segment*, *scene* and *shot*.

3. Semantics and content data: Video contains visual content and audio content. Users of video retrieval system want to find video clips with query on the semantic content of the video. Or given a sketch with description of color, shape, etc, the video database system can retrieve the video with visual similarity.

© 2005 *Kluwer Academic Publishers. Printed in the Netherlands.*

In current video database systems, videos are often queried based on visual similarity, which is calculated from low-level features on images, such as color, shape, texture, motion, histogram, etc. Some recent research work is using spatial and temporal information or working on specific categories of videos to provide content-based retrieval. (Chang, Chen, Meng, etc, 1997)(Yeung, Yeo, and Liu, 1996)(Huang, Liu, and Rosenberg, 1999). Those methods are successful in providing visual content of the video. However, in most situations, they can not reflect enough semantic information of the video, which the video database users are more concerning about.

The semantic description is the corresponding data abstractions to represent the video, instead of visual content, as textual descriptions *a car is running by a tree*. Ideally, the video will be automatically annotated as a result of machine interpretation. However, such data abstractions may not be feasible in practice in that: for a special video clip, different users may have different semantics interpretations because of varieties in education backgrounds and objects interested. At the same time, these abstractions are determined manually. Manually annotating semantics on video clips, with low successful ratio and low retrieval speed, limits applications of video databases. Automatic generation of such descriptions assists not only in building query languages that enable efficient storage and retrieval based on visual content of video, but also in managing and manipulating individual video clips for multimedia applications. Consequently, when larger databases of video are involved, automatic extraction of video semantic information becomes crucial.

In this paper, a video database system design for automatic semantics extraction, semantics-based video annotation and retrieval with textual tags is proposed. The semantic extraction is retrieval-driven, which means new tags are generated through user's query. Considering the case that different people may have various understanding and descriptions on the same video clip, we develop this model with flexibility in giving different textual descriptions for the same content of one video clip. This information is derived from objective description, instead of personal feelings. Annotation with multiple tags also avoids possible real-time computation on low-level image features, which fastens the query procedure.

The paper is organized as the following: in Section 2, background and some related work is given. In Section 3, first an image/video retrieval prototype is described, from which our system design is derived. Then, we introduce the structure of our system design for semantics-based annotation and retrieval and how its components interact with each other. To evaluate our system, as an example, an algorithm for track detection is proposed in Section 4. Our methodology and procedures of solving the problem are also illustrated. The experiment shows good performance in Section 5. At the end of this paper, we mention some issues about this system design and future research work in Section 6.

2. Related work

Before exploring semantic analysis on video, it is worthwhile to define some terms used here.

1. *Video shot*: an unbroken sequence of frames recorded from a single camera.

2. *Key frame*: the frame which can represent the salient content of a shot. Depending on the content complexity of a shot, one or more key-frames can be extracted.

3. *Video scene*: a collection of semantically related and temporally adjacent shots, depicting and conveying a high-level concept of story.

2.1. Clustering and browsing

In video and film, a *story* is told by the presentation of the images of time. It is built of shots and groups of shots are concatenated to form a depiction of the three-dimension event. To reflect the semantics inside video, global browsing and analysis among video shots are required. The first step in achieving concatenating shots is to label shots with a description of the content of the shot.

After giving labels to video shots, time-constrained clustering has been developed to compute the hierarchical structure of certain types of video content (Yeung and Yeo, 1996). Typically, a video that tells a story is composed of a sequence of *story units* denoted by U_1, U_2, ..., U_n. Story units are the interesting objects extracted from the shots. The story takes place in a small number of locales: d_1, d_2, ..., d_l, which are the backgrounds of the shots. Then, if fact that story unit U_i takes place in the locale d_j is denoted by U_i 2 d_j, this expression can be replaced by a character, for example A, the structure of the video may look like as the following:

$$U_1\ 2\ d_3\, ; U_2\ 2\ d_1\, ; U_3\ 2\ d_4\, ; : : : ; U_n\ 2\ d_3 :$$

Therefore, an algorithm can find meaningful story units by examining the sequence of labels of shots and identifies the sequences of labels. Different expressions like U_i 2 d_j have different labels unless they have the same story units and locales. Consider a video sequence of 10 shots labeled as follows:

$$A\, ; B\, ; A\, ; C\, ; D\, ; F\, ; C\, ; G\, ; D\, ; F :$$

The first story unit has to contain the first shot as well as the last shot which have the same label as the first shot. Furthermore, it may contain the intermediate shots. This process of inclusion can be recursively applied to successive shots in the story unit until the last shot in the first story unit has been reached. The algorithm operates in $O .S/$time, where S is the number of shots. Based on the concatenation of a story unit, users can browse the video easily. In (Y. Rui, T. S. Huang and S. Mehrotra, 1999), a semantic structure for video browsing and retrieval: table-of-content is proposed. The table-of-content is like the content of a book, which is at the scene-level.

The current research achievements in video browsing give the users a semantic structure of the video analysis. However, the interpretation of the video at the shot level is left for the users themselves. At the same time, how to concatenate shot to form story unit and how to find locales are also problems under investigation.

2.2. Modeling of temporal events

In addition to segmenting a video into larger units, such as story units, labeling sequences can be used to recognize *dialogue* and *action* events. Using the degree of repetition or the lack of repetition in a sequence of labels, one can classify the video sequence into one of three categories: *dialogues, actions*, and *others* (Yeung and Yeo, 1996).

For dialogue video clip, models can be constructed to capture the repetitive nature of two dominant shots while incorporating the possibility of *noise* label. A noise label could represent a shot in the local area where the dialogue takes place, but it could also represent some other shot. Consider an example of a video sequence of 22 shots with the following label sequence:

$$A ; B ; A ; X ; Y ; Z ; A ; B ; A ; B ; A ; B ; C ; D ; E ; F ; E ; D ; E ; G ; H ; I :$$

Here the labels are derived from visual data content of the shots. Hence, shots with the same label are likely to contain the same object and background. The label sub-sequence $A ; B ; A ; B ; A ; B$ characterizes a dialogue in which there is no noise label. The sub-sequence $D ; E ; F ; E ; D ; E$ also characterize dialogue in which label F is a “noise” label.

An action event represents exciting action sequences in action movies. An action sequence in motion pictures or video is characterized by a progressive presentation of shots with contrasting visual data content to express the sense of fast movement and to achieve a strong emotional impact. This type of sequence of shots would most likely to be found in a scene where there is a rapid unfolding of the story. It would also happen where the camera is not fixed at a location or in the following events, there is a significant amount of object movement. In such a sequence, there is typically little or no recurrence of shots taken from the same camera or the same person or background locale.

However, since *dialogue* and *action* events constitute about 50% to 70% of each video in general, differentiate only dialogue or action or others is coarse to video queries from users.

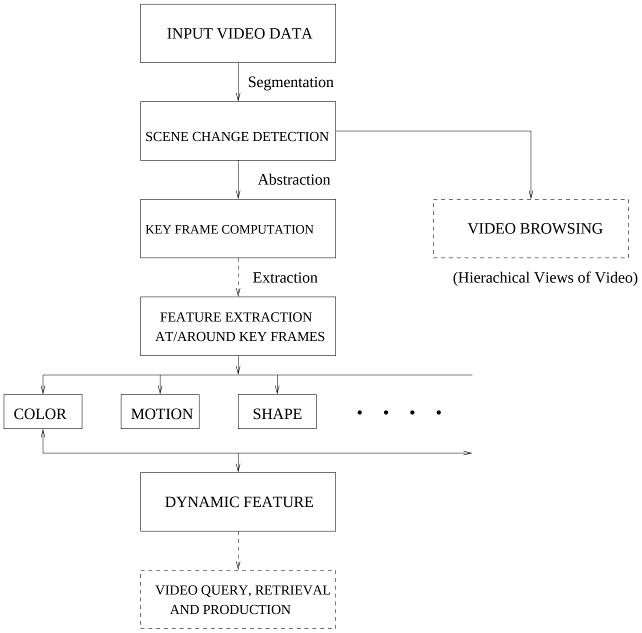


Figure 1. The function and structure of VIMS system

3. Structure of our model

3.1. An prototype for image/video retrieval

The current technique on retrieving a video is to divide the video hierarchically and index terms to the video clips, based on image matching techniques. Indexing are the low-level features extracted from the key-frames in the video clip, such as: shape, color, motion, etc. Through the comparison of features extracted with the video query, the system would retrieve the corresponding images or video clips as queried. Fig. 1 illustrates the VIMS (Video Information Management System) prototype as the core of Grand Challenge application (Lee, Li and Xiong, 1997).

The first task to be done on a raw video is to partition it into units, which facilitates later retrieval. The partition process involves boundary detection between uninterrupted segments – shots and scenes, and selection of ideal key-frames or construction of a key-frame, such as mosaic (Irani, Anandan and Hsu, 1995) to present the shots. We extract important features from key-frames and denote them as indexes, such as motion, color, shape, texture and color histogram, etc. Content-based image retrieval can be carried out by flexible image matching between the linear sketch and the abstract images in the pictorial index. As image matching for image/video database and content-based retrieval are based on visual similarity, they are meaningless in some

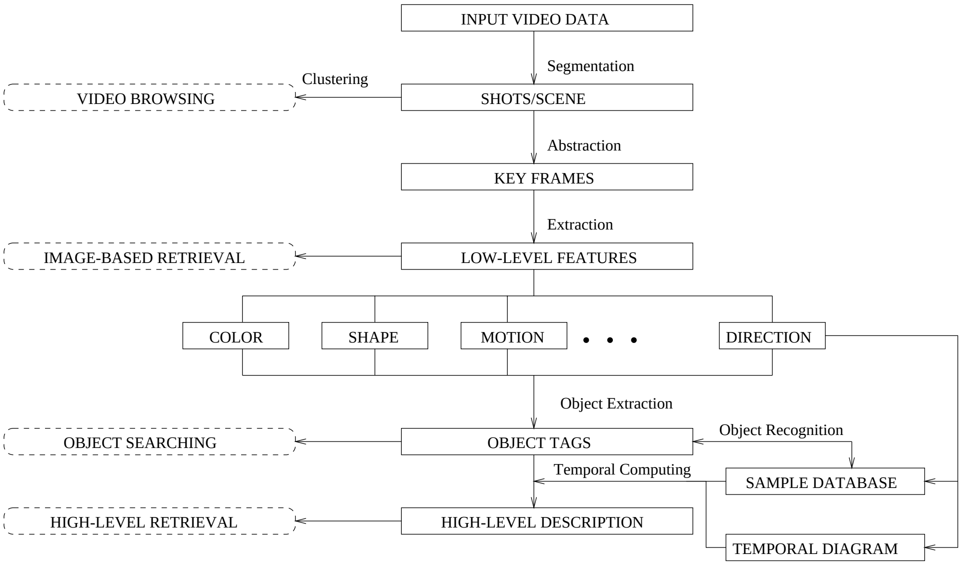


Figure 2. Semantic description model with flexibility

sense to users. For example, if the end user gives a text query: *A red car is running by a tree*, we can simply view those videos in the video databases one by one to find the video clips according to the query.

3.2. STRUCTURE DESIGN

Currently, textual tags on the content are attached to the shots and they have to be done manually with the aid of a computer, which is time-consuming and inaccurate, especially for a large digital library or a special video with more than one meanings. How to automatically analysis the content is crucial. It is also important that the text description of the attributes should at least partially reflect the characteristics of non-text multimedia data types. Therefore, we formalize the following problem and design a system to solve this problem.

Problem 1. How to find a procedure for automatic semantics extraction of video and use this semantic description as an index for retrieval?

The obstacle of constructing a system of dealing with semantic description annotation and retrieval is the temporal information hidden in the videos. To address flexibility without the burden of preprocessing steps, we develop the method on low-level features and object extraction/recognition techniques. Fig. 2 illustrates the structure of our model.

To reveal the temporal information, we use the temporal diagram with different information on the arcs for different hierarchical video. Refer to Fig. 3 for detailed information.

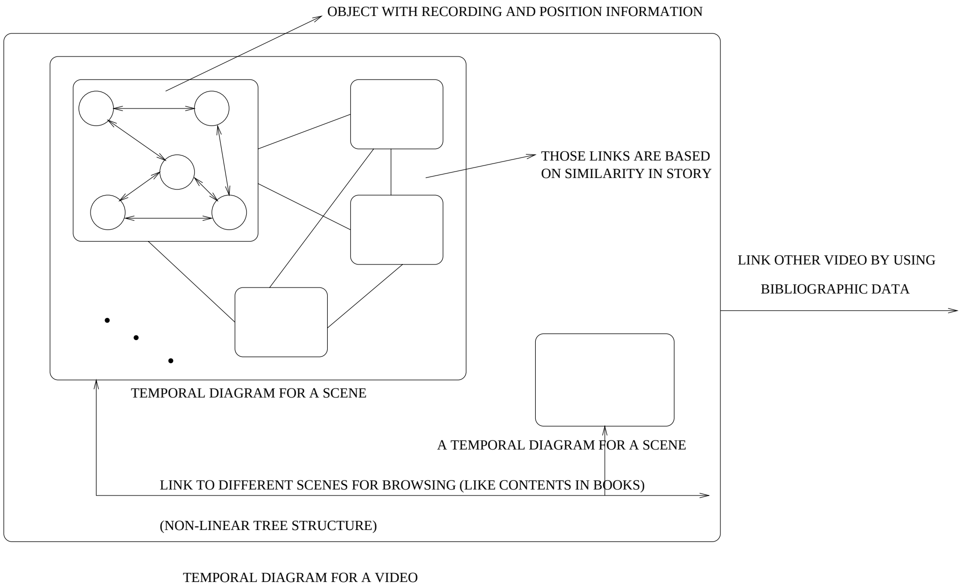

OBJECT WITH RECORDING AND POSITION INFORMATION
THOSE LINKS ARE BASED ON SIMILARITY IN STORY
LINK OTHER VIDEO BY USING BIBLIOGRAPHIC DATA
TEMPORAL DIAGRAM FOR A SCENE
A TEMPORAL DIAGRAM FOR A SCENE
LINK TO DIFFERENT SCENES FOR BROWSING (LIKE CONTENTS IN BOOKS)
(NON-LINEAR TREE STRUCTURE)
TEMPORAL DIAGRAM FOR A VIDEO

Figure 3. Temporal diagram for hierarchical video with different information

We build one temporal diagram for the entire video with links to other videos who have the same or similar bibliographic data. In this temporal diagram, each component represents one temporal diagram for a scene, where the arcs between two scenes represent the relationship between these two scenes in one cluster. For each scene, we also have a temporal diagram, where every components is a shot in this scene. For every shot, the components represent objects in the shot. Depending on the assumption in the following subsection, we construct the temporal diagram for a shot and give the direction of recording and position relation on the arcs. Using such low-level features and relative temporal position diagram, we can add textual description to this shot/scene. Since different people have different descriptions on one shot/scene, we use an array to store the descriptions. Searching video with textual description can be converted to searching text tags in the arrays if the tags are available. Otherwise, the video system will run the procedure again to find video clips according to the new video query and save the query to the corresponding shots/scene's text description array for future references.

4. Running procedure and characteristics

Refer to Fig. 2, part of the system design is the image retrieval model based on the VIMS prototype (Lee, Li and Xiong, 1997). Building on this, we gain the semantic description system by adding structures to reveal temporal information: *text description array* and *relative position diagram*.

4.1. Running procedure

For a video, the system is running as follows. From the results of the scene change phase (segmentation), shots and scenes are got. Then, we get low level features extracted from the key-frames in the shots. With the techniques of object recognition (Xu, Wu and Ma, 1999) (Yang and Wu, 1997), we can extract the objects according to the template in the image sample database and the viewing direction to this object. The corresponding tag for a shot will consist of a list of objects' names and visual attributes in this shot.

From the informal definition of shots, which is a sequence of images captured by a camera within one move, we know that in one shot, the possibility of appearance of two or more objects in the same classification and same low-level features is much small. Therefore, one assumption is got here first.

Assumption 1. In one shot, if two objects are recognized as the same kind of objects and they have the same low-level features after rotating camera position, and if they do not appear in one frame at the same time, we can assume that the objects are the same with the same name.

At first, we go one way like the VIMS. Then, we have the low-level features extracted from the key-frames. Based on techniques of object recognition and the assumption we made before, object tags, object boundary and viewing directions are recorded.

Secondly, for the retrieval part, user queries can be divided into three categories. If the query is about low-level image feature or global viewing of the video, the system can automatically extract the attributes from previously saved tags or present the structural information. If the query is about a named object, this information can also be extracted quickly from object name tags, as what relational database does in text matching. The low-level description retrieval can be executed by part of the model using the same techniques and algorithms. For query with semantics, the following procedure is employed. Every time when a user queries a video clip with text, the system will first go to the textual description array to find whether those text descriptions of this shot/scene will satisfy the query. If yes, the appropriate video clip is found. If no such tag is found, the system to run again to find appropriate video clip. Since the tag information is built from users' query, the system is self-adaptive because those tags queried frequently are retrieved frequently and they can be given higher priority. Every time, the system will record the query after computation into the array for future retrieval.

From the interfaces among the blocks of our model, we provide Fig. 4 to illustrate the global view.

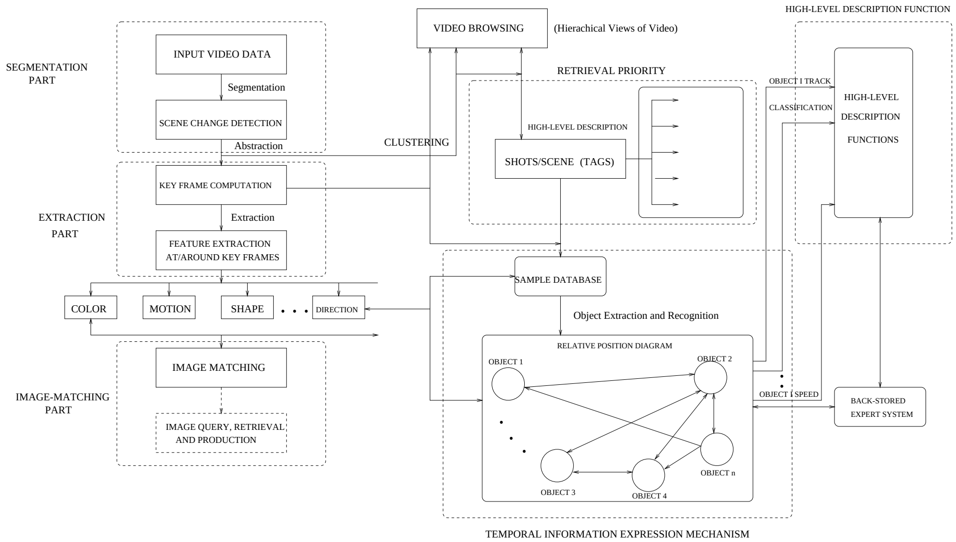


Figure 4. Interfaces of different blocks in our model

4.2. Object tracking techniques with semantic description

The key techniques of this system are to define the semantics and the information needed to construct it.

From normal knowledge, we know that a still object, like tree, hill or building can not move, therefore, we can give the motion definition like this:

Definition 1. An object is in motion means that there is a change of the relative positions between this object with a still object.

Thus, we can use change of position to detect the track of one object or the motion information. The procedure of getting such semantics description for the video requires the shape information like rectangular boundaries of objects and relative positions of the edges of rectangular boundaries for all the objects in a video shot. Here, we use the direction information and object recognition to reveal the object's track. Refer to Fig. 5 as an illustration.

For every object in a frame, we can construct a grid based on its rectangular boundaries. For object i, we divide the frame into 9 rectangular areas. Here, we use four values to present its four rectangular boundaries: left boundary (lb), right boundary (rb), top boundary (tb) and bottom boundary (bb). Therefore, if we use a ordered set to represent the position information for object A in frame i,

$$A_i = \{lb_A, rb_A, tb_A, bb_A\}.$$

For object B in frame i, we have

$$B_i = \{lb_B, rb_B, tb_B, bb_B\}.$$

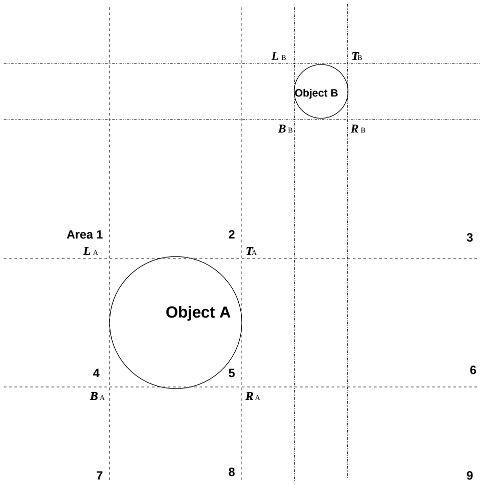


Figure 5. Revealing track and motion vector by direction information

There are three possibilities of relative positions of A and B:

1. A is in B or B is in A,

2. A and B do not cover each other,

3. A and B have some common areas.

We can use set operation to classify the relative positions: the first case is held if
$lb_A < lb_B < rb_B < rb_A$ and $bb_A < bb_B < tb_B < tb_A$
or
$lb_B < lb_A < rb_A < rb_B$ and $bb_B < bb_A < tb_A < tb_B$
The second case holds when

1. B is above A: $tb_A < bb_B$,

2. A is above B: $tb_B < bb_A$,

3. A is left to B: $rb_A < lb_B$,

4. B is left to A: $rb_B < lb_A$.

While for the third case, the unequal equations are used to describe the relative positions. For the areas divided by object A, we describe relative position of object B by using such information. However, the last table of relative position for object A and B can be simplified by using Karnaugh Map method:

1. B is in area 1: $tb_A < bb_B$ and $rb_B < lb_A$,

2. B is in area 2: $tb_A < bb_B$, $lb_A < lb_B$, and $rb_B < rb_A$,

3. B is in area 3: $tb_A < bb_B$ and $rb_A < lb_B$,

4. B is in area 4: $tb_B < bb_A$, $bb_A < bb_B$, and $rb_B < lb_A$,

5. B is in area 5: $lb_A < lb_B < rb_B < rb_A$ and $bb_A < bb_B < tb_B < tb_A$

6. B is in area 6: $bb_A < bb_B < tb_B < tb_A$ and $rb_A < lb_B$

7. B is in area 7: $tb_B < bb_A$ and $rb_B < lb_A$,

8. B is in area 8: $lb_B < lb_A < rb_A < rb_B$ and $tb_B < bb_A$

9. B is in area 9: $tb_B < bb_A$ and $rb_A < lb_B$.

While for other occasions when object B covers more than one areas divided by object A, we also have such inequalities to describe it. For the object we are interested in, we convert the viewing directions and record the relative position information to the arcs in the temporal diagram of the shot. Such that, if we view object A in a certain direction in frame i, while for frame $i + 1$, the viewing direction is changed by β angle. Therefore, for the frame $i + 1$, we convert all the position information by multiplexing $\cos\beta$. And then, we record the new temporal information in the temporal diagram.

When we give the relative position of the objects in frame i and frame $i + 1$, we can derive the motion vector. Therefore, if we know the relative position is changed between object i and object j, and object j is a still object, we will conclude that object i is moving past object j with motion vector: $V = \{V_x, V_y\}$

Besides retrieval, this technique can be applied in real-time computation to detect the objects track to find which one is in usual action and which one is not. It can be applied in monitoring system.

Another retrieval method works as the following. If we denote the object that we are concerned with as the main object, we can denote other objects in this shot as background. Therefore, since for every shot we have chosen the main object and the background, we can use this information to describe whether this shot is similar to another shot or not. Labels can be assigned for the shot. We add more information to classify videos. We consider that each shot i has a set S_i containing its objects.

$$S_i = \{Object_{1i}, Object_{2i}, \ldots, Object_{ni}\}.$$

If the number of the same objects in two shots, that is, the number of the objects of the intersection of the object sets of two shots, exceeds a threshold defined, we can say that these two shots are in the same scene.

That is, for

$$S_j = \{Object_{1j}, Object_{2j}, \ldots, Object_{mj}\}.$$

The intersection is:

$$S_i \bigcap S_j = \{Object_1, Object_2, \ldots, Object_k\}.$$

If k is more than the threshold, cluster S_i and S_j are put into the same scene. In a special case, a dialogue can be recognized if the exclusive object set is: $\{A\}, \{B\}, \{A\}, \{B\}$ etc, that is,

$$S_i - S_i \bigcap S_j = \{A\}.$$

and

$$S_j - S_i \bigcap S_j = \{B\}.$$

We know that this kind of video is dialogue while the two persons are A and B. Using set operation, for consequent shots, we use intersection to find which object appears and which object disappears. For example, S_i and S_{i+1} are two sets for two consequent shots.

$$S_i = \{Object_{1i}, Object_{2i}, \ldots, Object_{ni}\}.$$

$$S_{i+1} = \{Object_{1i+1}, Object_{2i+1}, \ldots, Object_{ni+1}\}.$$

Then, the objects appear are in the set of:

$$S_i - S_i \bigcap S_{i+1}.$$

while the objects disappear are in the set of:

$$S_{i+1} - S_i \bigcap S_{i+1}.$$

Any functions on semantics description for video defined can be added to this model. Such as: object i track, classification of a certain video clip, object i speed, object i appears, etc. Indeed, we can get the semantic description by real-time computation on these information got. We will see that this model not only reveals the information such as dialogue, action, but also other information such as motion, moving, object track, etc.

4.3. System properties

1. Because the system design is based on traditional image searching model, it can be compatible to traditional image searching and matching. The system supports different levels of query, from image matching to semantics retrieval. Semantics can vary with more definitions.

2. The system is also application-oriented. The model allows additions of any semantics description function to calculate solutions.

3. The automatic semantics-based annotation and retrieval provide efficiency to users than manual annotation. For manual annotation, only one or few tags are added and it is error-prone for user's query.

4. The model we give here is a retrieval-driven system design. The system can derive different information from the same video clip with different video queries. The semantics extraction is derived from computation on relation among objects. For different queries, object can be varied in the same shot, even in the same frame. The computation in the system is done only when it is needed.

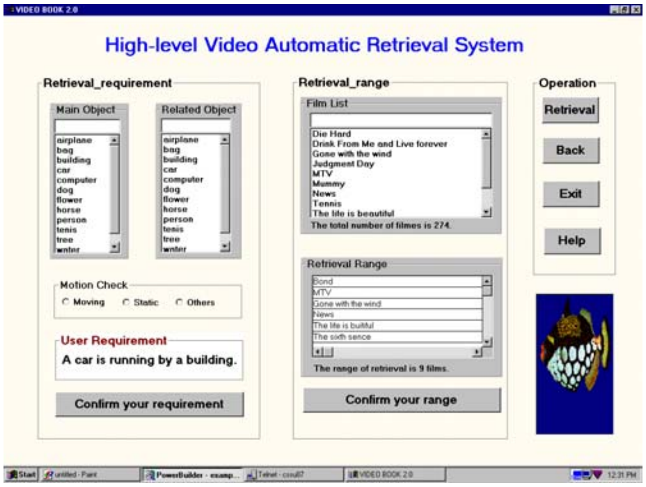
5. The computation results are saved after each query. With the same query, tag can be retrieved quickly without computation on relative positions among interesting objects, which save much time. Due to the application property, the system keeps the semantic tags for video with its popularity. For the most accessed tag, it is saved in the array and will be matched first for user's query.

5. Experimental results

¡ The pre-processing part: We showed part of system to illustrate semantics annotation and retrieval. Pre-processing extracts key-frames, index tags and record viewing direction and the objects' rectangular boundaries.

¡ The performance: A user interface is shown in Fig. 6. To avoid trouble of directing correct query, for example, "A car is running by a tree" and "Car is moving around a tree" have the same semantics interpretation, the system normalizes the user query by listing the main object, relative object and then asking the user to choose one of the semantic descriptions on the interaction of these objects.

The system checks the existing tags for the query. If there exists the same tag, relevant video clips are retrieved out. Otherwise, the system will use tags with "car" and "tree" to check motion on our departmental digital library. The semantics description for video clip used is: "*A car is running by a tree*". Fig. 7 lists films found matching the query. Fig. 8 shows frames found in one of the candidate films.



High-level Video Automatic Retrieval System

Main Object:
airplane
bag
building
car
computer
dog
flower
horse
person
tenis
tree
winter

Related Object:
airplane
bag
building
car
computer
dog
flower
horse
person
tenis
tree
winter

User Requirement
A car is running by a building.

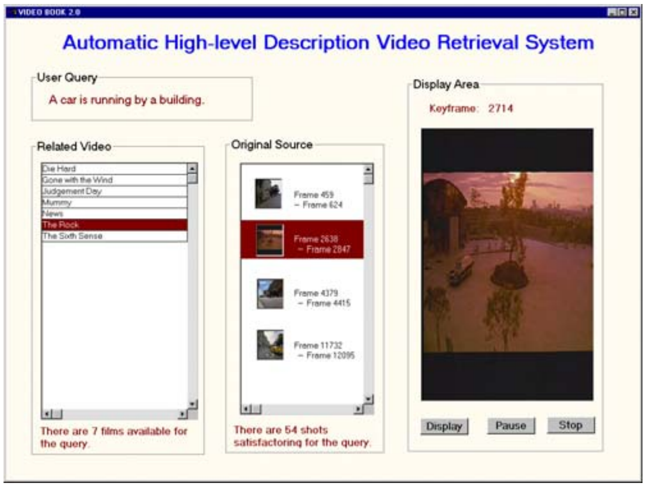
Film List:
Die Hard
Drink From Me and Live forever
Gone with the wind
Judgment Day
MTV
Mummy
News
Tennis
The life is beautiful

The total number of filmes is 274.

Retrieval Range:
Bond
MTV
Gone with the wind
News
The life is buitful
The sixth sence

The range of retrieval is 9 films.

Figure 6. User interface design



Automatic High-level Description Video Retrieval System

User Query: A car is running by a building.

Related Video:
Die Hard
Gone with the Wind
Judgement Day
Mummy
News
The Rock
The Sixth Sense

There are 7 films available for the query.

Original Source:
Frame 459 – Frame 624
Frame 2638 – Frame 2847
Frame 4379 – Frame 4415
Frame 11732 – Frame 12095

There are 54 shots satisfactoring for the query.

Keyframe: 2714

Figure 7. Retrieving result

6. Conclusion

Semantic description model for video contents can be of tremendous commercial value. In this paper, we propose a general semantic-based annotation, retrieval system design for video databases. It provides easier retrieval and objective results for users to query. And the real-time computation can be negligible and acceptable for a large digital library.

Further research could be on how to retrieve those video clips required by semantic description and transfer video through the Internet. On-line providing semantics structure of video in streaming VoD service for browsing is more attractive to users under limited bandwidth. We will apply our tech-

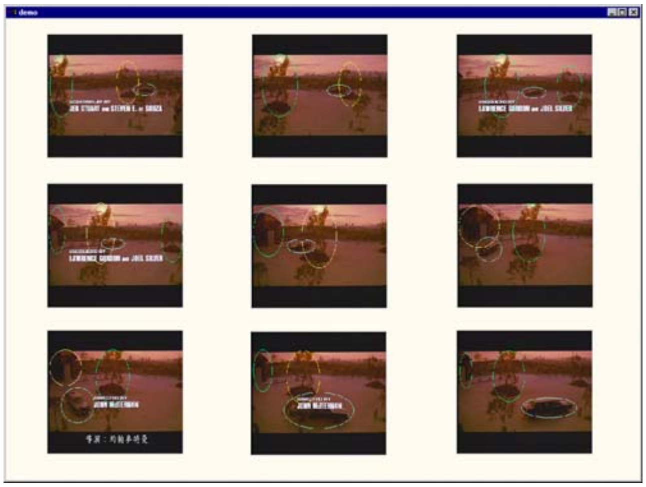


Figure 8. Searching result

niques (Li, Liu and Ahmad, 1999)(Li. etc, 1999)(Liu, Lee and Li, 1999) to this problem, combining the techniques mentioned in this paper.

References

R. M. Bolle, B. L. Yeo, and M. M. Yeung, "Video query: research directions", IBM Journal of Research and Development, March 1998.

S.-F. Chang, W. Chen, H. J. Meng, H. Sundaram, and D. Zhong, "VideoQ-an automatic content-based video search system using visual cues", in Proc. of ACM Multimedia 1997.

W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee, "Using adaptive tracking to classify and monitor activities in a site", in Proc. of Conf. on Computer Vision and Pattern Recognition 1998.

Q. Huang, Z. Liu, and A. Rosenberg, "Automatic semantic structure reconstruction and representation generation for broadcast news", in Proc. of SPIE, Storage and Retrieval for Image and Video Databases VII, 1999.

M. Irani, P. Anandan, and S. Hsu, "Mosaic based video compression", in Proc. of SPIE Conf. on Electronic Imaging, Digital Video Compression: Algorithms and Techniques, Vol. 2419, Feb. 1995.

R. Jain, "NSF workshop on visual information management system", SIGMOD Record, Vol. 22, pp. 57-75, Sep. 1993. (Editor).

R. Jain and A. Hampapur, "Meta-data in video databases", in Proc. of ACM SIGMOD 23, No. 4, Dec. 1994, pp. 23-33.

J. C. M. Lee, Q. Li, and W. Xiong, "VIMS: a video information manipulation system", Multimedia Tools and Applications, Vol. 4, No. 1, 1997, pp. 7-28.

J. C. M. Lee, Q. Li, and W. Xiong, "VIMS: towards an adaptive and versatile video manipulation server", in Proc. of IS&SPIE Conf. on Storage and Retrieval for Image and Video, SPIE, Photonic West 97 Meeting, San Jose, CA, Feb. 1997, pp. 166-174.

F. Li, Y. Liu, and Ishfaq Ahmad, "A lossless quality transmission algorithm for stored VBR video", in Proc. of 4th IEEE Int. Symposium on Consumer Electronics, Vol. No. 2, Malacca Malaysia, Nov. 17-19, 1999. pp. 35-38.

F. Li, Y. Liu, J. Y. B. Lee, and I. Ahmad, "Shortest delay scheduling algorithm for lossless transmission of stored VBR video under limited bandwidth", South Africa Computer Journal, No. 24, 1999. pp. 146-154.

Q. Li and J. C. M. Lee, "Dynamic video clustering and annotation", in Proc. of 18th Int. Conf. on VLDB, 1992, pp. 457-468.

Y. Liu, J. C. M. Lee, and F. Li, "Updating scheduling strategy for stored VBR video transmission under limited bandwidth", in Proc. of 4th IEEE Int. Symposium on Consumer Electronics, Vol. No. 2, Malacca Malaysia, Nov. 17-19, 1999. pp. 31-34.

Y. Liu and F. Li, "High-level semantics extraction model for video retrieval", in Proc. of IASTED 4th Int. Conf. on Internet and Multimedia Systems and Applications, Las Vegas, U.S.A. pp. 442-447. Nov. 2000.

Y. Liu and F. Li, "A high-level semantics extraction model for stored videos", in Proc. of IEEE Int. Symposium on Multimedia Software Engineering, Taiwan, pp. 71-74, Dec. 2000.

W. Niblack, R. Barber, W. Equitz etc, "The QBIC: query in image content using color, texture and shape", in Proc. of SPIE on Storage and Retrieval for Image and Video Databases, 1908, 1993, pp. 13-25.

E. Oomoto and K. Tanaka, "OVID: design and implementation for video object server", IEEE Tran. on Knowledge and Data Engineering, Vol. 5, No. 4, 1994, pp. 629-643.

A. Pentland, R. W. Picard, and S. Sclaroff, "Photo-book: tools for content-based manipulation of image databases", in Proc. of SPIE, Vol. 2185, 1994, pp. 34-47.

L. A. Rowe, J. S. Boreczky and C. A. Eads, "Indices for user access to large video database", in Proc. of SPIE 2185, Storage and Retrieval for Image and Video Database II, Feb. 1994, pp. 150-161.

Y. Rui, T. S. Huang and S. Mehrotra, "Constructing table-of-content for videos" Multimedia System, Vol. 7, pp.359-368, 1999.

W. Xiong and J. C. M. Lee, "Efficient scene change detection and camera motion annotation for video classification", Journal of Computer Vision and Image Understanding, Vol. 71, No. 2, 1998, pp. 166-181.

W. Xiong, J. C. M. Lee, and R. H. Ma, "Automatic video data structuring through shot partitioning and key-frame selection", Machine Vision and Applications, Vol. 10, No. 2, 1997, pp. 51-65.

C. Xu, J. Wu, and S. Ma, "A visual processing system for facial prediction", in Proc. of 3th Int. Conf. on Visual Information and Information Systems, Amsterdam, The Netherlands, June 2-4, 1999, pp. 735-744.

L. Yang and J. Wu, "Towards a semantic image database system", Data and Knowledge Engineering, Vol. 22, No. 2, April, 1997, pp. 207-227.

M. M. Yeung and B. L. Yeo, "Video content characterization and compaction for digital library applications", in Proc. of SPIE on Storage and Retrieval for Still Image and Video Databases V, 3022, Feb. 1997, pp. 45-58.

M. M. Yeung and B. L. Yeo, "Time-constrained clustering for segmentation of video into story units", in Proc. 13th Int. Conf. on Pattern Recognition, Aug. 1996. pp. 375-380.

M. M. Yeung, B. L. Yeo, and B. Liu, "Extracting story units from long programs for video browsing and navigation", in Proc. Int. Conf. on Multimedia Computing and Systems, June. 1996..

D. Zhong, H. J. Zhang, and S. F. Chang, "Clustering methods for video browsing and annotation", in Proc. of SPIE on Storage and Retrieval for Image and Video Databases, 2670, 1996, pp. 239-246.