A Semantic Model for Video Description and Retrieval*

Chia-Han Lin, Andro H. C. Lee and Arbee L.P. Chen

Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.
E-mail:alpchen@cs.nthu.edu.tw

Abstract. In this paper, a semantic video retrieval system is proposed based on the stories of the videos. A hierarchical knowledge model is used to express the semantic meanings contained in the videos, and a video query language is also provided. The terms of Object, Action and Relation are used to specify rich and complex semantic meanings in a query. Based on the proposed knowledge model, the retrieval system is able to make inferences on the terms appearing in a query, and determine whether a video semantically matches the query conditions. The semantic similarity measurement is also proposed for processing approximate queries.

1 Introduction

Due to the improvement of computer power and the growth of storage space, videos can now be stored in digital formats on computer platforms. Recently, these digital videos are massively distributed and users may want to search for a desired video efficiently. This demand leads to the research of video retrieval.

In the video retrieval system, some approaches [5][9][10] use the text annotation to describe the video content. However, the complicated video content is difficult to describe just by text. Other approaches [3][17] of video retrieval are based on visual features of videos such as the color of the video frame or the shape and the motion trajectory of the objects in videos. Venus [12] is a video retrieval system, which considers the spatial-temporal relationship of objects. Moreover, a video query language is provided for users to specify queries. The hybrid method proposed in [8] combines both the text annotation and visual features to describe the content of video.

In addition to visual characteristics, a video is usually associated with a story, which is more meaningful to humans. Lilac et al. [6] expresses this information in terms of objects, actions and associations. The associations represent the relationships between the objects and the actions. Aguis and Angelides[1] proposed a semantic content-based model for semantic-level querying including objects, spatial relationships between objects, events and actions involving objects, temporal relationships between events and actions. However, it is hard to describe a complex

* This work was partially supported by the Program for Promoting Academic Excellence of Universities in the Republic of China under the Contract No. 89-E-FA04-1-4

story based on the proposed model. Some other researches focus on designing a semantic video query language[2]. More discussions on defining the query language in formal grammar can be found in [5][7]. However, these researches focus on syntax definition rather than the description and organization of semantic meanings.

Some other approaches capture the semantic meaning with some concepts[14] [16]. A video is segmented into several video clips which is described by the associated concepts. In [11], the semantic meaning is captured in concepts with a concept degree to represent the intensity of the concept in a video segment. However, the structure used to express the spatial-temporal relationships of objects is not included.

Both the spatial-temporal relationships and the semantic meaning are important components of a video. A new video retrieval system, which expresses the video content in a semantic way, is proposed in this paper. The events happening in a video such as “moving”, “fighting”, or "talking" are recorded. We also provide a definable knowledge model to represent the semantic meanings. Finally, a semantic video query language is designed for users to specify queries.

The organization of this paper is as follows. Section 2 presents the modeling of videos. A semantic video query language (SVQL) is proposed in Section 3. Section 4 presents query processing and the semantic similarity measurement. Section 5 concludes this paper and presents future work.

2 Video Modeling

2.1 Knowledge Models

The background knowledge is used to realize the semantic meaning of the video content. In our system, the knowledge can be built as a hierarchical semantic model based on the characteristics of the applications. In this model, three types of knowledge, *Object*, *Action*, and *Relation*, are defined. “Object” represents the objects appearing in the video. “Action” represents an action, such as “moving”, of an object. “Relation” is used to describe a relationship between objects, which can be a spatial relation or an implicit semantic relation such as “father of”. Each of the three types of knowledge has a corresponding hierarchical semantic tree, which is organized with the object-oriented inheritance relationship to express the relation between different semantic meanings. Fig. 1. is an example of the hierarchical semantic tree for objects.

2.2 Metadata Structures

Some information of a video, such as title and names of actors, can be simply recorded in the metadata. However, some other information, such as the appearance and disappearance of objects, the attributes of objects, or the events happening in the video, may change frequently when a video is played. Such information is recorded with additional duration information.

In the proposed approach, the metadata of videos are recorded in tables. The description for each video is recorded as a row in a table. The format of the tables is defined as follows.

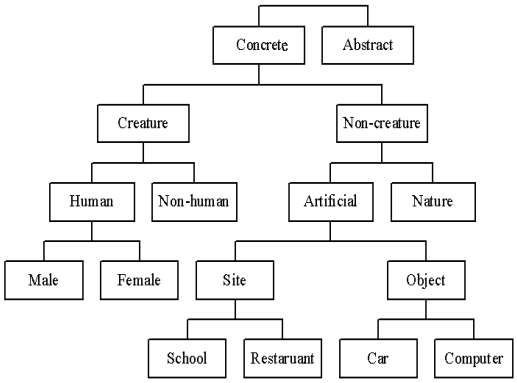

Fig. 1. An example of the hierarchical semantic tree for objects.

Film_Info_table records the production information of a video, such as title and names of actors.

Scene_Concept_table records the values of the concept degree of scene concepts. The scene concept proposed in our previous work[11] is used to express the semantic information of a video segment.

Object_Attribute_table records the values of the object attributes. Since the values may be changed when the video is played, the duration information will be also recorded with the values.

Relation_table records relationships between objects. The object's id and the duration information where the relations hold will be recorded.

Action_table records the actions of a video object or objects. The object's id and the duration information where the actions hold will be recorded.

3 Semantic Video Query Language

Based on the proposed video model, SVQL is proposed for users to specify queries. The variables are declared to represent different objects, actions and relations. Furthermore, these variables can be grouped with a function to form an expression representing an action or a relation. Finally, a sequence of expressions is built to represent a complicated "story", which can be regarded as the query condition. The video clips containing the story will be found and the specified information of the video clips will be returned as the result.

3.1 Syntax

A simplified form of syntax is listed as follows. The formal and detailed definitions in the format of BNF can be found in[13].

```
Find        <target>
Which       Has-
            Object: <var_declaration>
            Action: <var_declaration>
```

Relation: <var_declaration>
Semantic-
<object_expression>,
<doing_expression>,
<relation_expression>
→
next description block…

The clause of "Find" expresses the form of the query result while the clause of "Which" represents the query conditions.

3.2 The <target> field

There are several ways to display different types of the query result based on the specification of the <target> field. The <target> field has the following possible values:

1. *Clip*: It indicates that the system should return the video clips matching the expressions in the "Semantic-" part.
2. *Attribute of FilmInfo*: It indicates that the system should return the value of FilmInfo's attributes for each matched video clip.
3. *SceneConcept*: Specifying SceneConcept indicates that the system should return the concept degrees of each matched video clip.
4. *Variables or some attributes of variables*: Specifying attributes of variables indicates that the system should return the values of the attributes. Specifying variables indicates that the system should return the values for each variable's attributes.

3.3 The "Has-" part

The variables used in the <target> field and the "Semantic-" part are declared in this part as one of the three types: Object, Action and Relation.

3.4 The "Semantic-" part

This part is used to describe the whole "story." Several types of expressions are provided to describe various semantic meanings. Multiple expressions can be specified and separated by a comma. All expressions in this part are verified to find a suitable query result containing the corresponding semantic meanings. The three different types of expressions are introduced as follows:

<*object_expression*>

This type of expressions is used to set the constraint of an Object variable's attribute, or to specify a variable as an instance of some class in the corresponding hierarchical semantic tree. Some operators that can be used in this type of expressions are defined as follows:

1. IS-A operator: "∈"
   The IS-A operator is used to specify that a variable is an instance of some class in the corresponding hierarchical semantic tree.
2. Attribute operator: "."

Attribute operator is used to specify the attribute of an object. The syntax of using this operator is "<obj>.<attribute>." "<obj>" is a variable name and "<attribute>" is one of the object's attributes.

3. Comparison operator "=", "<", ">", "≠",
   Comparison operator is used to set constraints between attributes and values.
4. Alternative operator: "|"
   This operator is used in the right-hand side of an expression to indicate the "OR" logic, which means if one of these values satisfies the expression, the whole expression holds.

Here is an example of a sequence of object expressions:

x ∈ "car", y ∈ "man,"
x.color = "red" | "blue,"
y.age = "42"

In this example, x is a car, and y is a man. The color of the car is red or blue. The man is 42 years old.

<*doing_expression*>

This type of expressions is a function using Object and Action variables as its parameters. It is used to express an action occurring in a video. The two types of functions are introduced below. The parameters of *object1* and *object2* are two variables declared as Object. *Action* is a variable declared as Action.

1. DoingTo(*object1*, *object2*, *action*)
   It expresses "*object1* is doing *action* to *object2*."
2. Doing(*object1*, *action*)
   It expresses "*object1* is doing *action.*"

Here is an example following the previous example:

m = "drive," DoingTo(y, x, m)

In this example, m is an action of "drive," which means "a man is driving a car."

<*relation_expression*>

By using this type of functions, the relationships between the variables of Object are described. Since the relation represents the relationship between two objets, the syntax of the function can be expressed as follows:

RelationTo(*object1*, *object2*, *relation*)
It expresses "*object1* has a relation of *relation* to *object2*."

Here is an example following the previous example:

n = "owner_of," RelationTo(y, x, n)

In this example, n is a relation of "owner_of" The expression indicates a relation between x and y, which means "a man y is the owner of a car x."

Description Block

The Transition operator "→" is used to connect the expressions in the "Semantic-" part from several groups. Each group of the expressions is considered as a Description Block, which describes one "story." The Transition operator "→" indicates that these stories happen in order.

Here is an example following the previous one:

p = "walk," DoingTo(y, x, p)
→
y ∈ "man, x ∈ "car," m = "drive," DoingTo(y, x, m)

The story of these expressions is "At first, a man y walks to a car x, then he drives

that car."

3.5 A Complete Query Example

Find a video clip containing the story: "Tom stands to the left of Michael and Michael attacks Tom, Tom asks Michael, "why are you attacking me?" but Michael just runs away.

Find Clip
Which
Has-
Object: Tom, Michael, dialog
Action: attack, say, run
Relation: left
Semantic-
Tom ∈ "man", Tom.name = "Tom",
Michael ∈ "man", Michael.name = "Michael",
left = "left_to", RelationTo(Tom, Michael, left),
attack = "attack", DoingTo(Michael, Tom, attack),
→
say = "say", dialog ∈ "dialog",
dialog.content = Why are you attacking me? ,
DoingTo(Tom, say, dialog)
→
run = "run", Doing(Michael, run)

4 Query Processing

4.1 Evaluating Expressions

The description for each video is recorded as a row in the metadata table. To process a query, the expressions are evaluated one by one. The matched descriptions can be found after the expression evaluation. If the matched description does not exist for an expression, no result will be found for this query. The video clips containing all matched descriptions will be added to the solution set. After all expressions have been evaluated, the desired target information for each video clip in the solution set will be returned as the query result.

Description Blocks can be connected by the Transition operator. During the query processing, each Description Block is evaluated to find a solution set. The video clips can be selected from the solution sets in order by verifying the temporal relationship of these video clips.

4.2 Reasoning in Semantic Hierarchy

In SVQL, users can declare three types of new variables, Object, Action, and Relation. With the IS-A operator, a variable can be specified as an instance of some class of the

corresponding hierarchical semantic tree. A reasoning rule can be defined as follows:

Semantic Class Matching Rule:
Assume A is a variable in a query and B is a candidate description in the database. A and B are declared as of same type. $A \in A'$ and $B \in B'$. Then B is a matched answer to A if and only if A' = B' or A' is an ancestor class of B'.

For example, based on the hierarchical semantic tree of Object shown in Figure 2, if a variable in the query is "Creature" the description of "Male" in the database will be matched since "Creature" is an ancestor class of "Male."

4.3 Semantic Similarity Measurement

Each variable can be specified to be an instance of some classes in the three semantic hierarchical trees. By Semantic Class Matching Rule, whether the candidate object matches a query variable can be determined. However, in order to allow approximate queries, another approach is designed to calculate the similarity between unmatched classes.

The hierarchical semantic tree is organized with an IS-A relation from top to down. The upper classes are more general and the lower classes are more specific. When considering the issue of semantic similarity measurement, two cases are discussed based on the paths from the root to the two unmatched classes. The first case is that the paths are different. Therefore, if the paths branch earlier, the two classes are more dissimilar. In order to calculate the dissimilarity between two classes, a weight is set for each branch in the hierarchical semantic tree. The weight of the upper branch is higher. The dissimilarity between two classes can be defined as the number of edges from the two classes to the branch plus the weight of that branch.

The other case is the reverse situation described in the Semantic Class Matching Rule: the candidate class and the query class are at the same path, but the candidate class is the ancestor of the query class. The similarity can be measured as the number of edges between the two classes.

Based on these two cases, the Dissimilarity Index Building Algorithm is proposed to calculate the dissimilarity weight of the branches in the hierarchical semantic tree while building the hierarchical semantic model. Moreover the Dissimilarity Measuring Algorithm is proposed to measure the dissimilarity between two classes in a hierarchical semantic tree. By using the algorithm, the dissimilarity degree for each query result can be calculated and can be used to rank the query results. Both algorithms can be found in[13].

5 Conclusion

In this paper, a semantic video retrieval system is proposed. A hierarchical semantic model is designed to express the background knowledge for different applications. Based on the model, the semantic meanings contained in videos are described and recorded in metadata for query processing. A semantic video query language is proposed for users to specify queries. Variables can be declared as objects, actions, or relations, which can be used to describe an event as the query condition. The video

clips or desired information can be found from the metadata. During the query processing, a reasoning process is used to parse the semantic meaning of the query and the approximate results can be found based on the proposed similarity measure.

A simple metadata structure is proposed in this approach. Our future work is to design an index structure for the metadata to enhance the query processing performance.

References

[1] Harry W. Agiuo and Marios C. Angelides, "Modeling Content for Semantic-Level Querying of Multimedia," *Multimedia Tools and Applications*, Vol.15, No.1, 2001.

[2] Edoardo Ardizzone and Mohand-Said Hacid, "A Semantic Modeling Approach for Video Retrieval by Content," *Proc. IEEE International Conference Multimedia of Computing and Systems*, 1999.

[3] E. Ardizzone, M. La Cascia and D. Molinelli, "Motion and Color-Based Video Indexing and Retrieval," *Proc. IEEE Pattern Recognition*, pp.135-139 1996.

[4] T. Chua and L. Ruan, "A Video Retrieval and Sequencing System," *ACM Transaction on Information Systems*, 1995.

[5] Cyril Decleir and Mohand-Said Hacid, "A Database Approach for Modeling and Querying Video Data," *Proc. IEEE 15th International Conference on Data Engineering*, 1999.

[6] Lilac A.E. Al Safadi and Janusz R. Getta, "Semantic Modeling for Video Content-Based Retrieval Systems," *Proc. IEEE 23th Australasian Computer Science Conference*, 2000.

[7] M.-S Hacid, C. Decleir, and J. Kouloumdjian, " A database approach for modeling and querying video data," *IEEE Transactions on Knowledge and Data Engineering*, Vol.12, No.5, pp.729-750, Sept.-Oct. 2000.

[8] Mi Hee, Yoon Yong Ik and Kio Chung Kim, "Intelligent Hybrid Video Retrieval System supporting Spatio-temporal correlation, Similarity retrieval," *Systems, Man, and Cybernetics*, 1999.

[9] R. Hielsvold and R. Midtstraum, " Modeling and Querying Video Data," *Proceedings of the 20th International Conference on VLDB*, 1994.

[10] Haitao Jiang, Danilo Montesi and Ahmed K. Elmagarmid, "VideoText Database Systems," *Proceedings of the International Conference on Multimedia Computing and Systems*, 1997.

[11] Jia-Ling Koh, Chin-Sung Lee and Arbee L.P. Chen, **"**Semantic Video Model for Content-Based Retrieval," *Proc. International Conference on Multimedia Computing and Systems*, 1999.

[12] Tony C.T. Kuo and Arbee L.P. Chen, "Indexing, Query Interface and Query Processing for Venus: A Video Database System," *Proc. International Symposium on Cooperative Database Systems for Advanced Applications*, 1996.

[13] Andro H. C. Lee, "A Semantic Model for Video Description and Retrieval," *Master Thesis, Dept. of Computer Science, National Tsing Hua University*, Taiwan, 2001

[14] Suieet Pradhan, Keishi Tajima, Katsumi Tanaka, "Querying Video Databases based on Description Substantiality and Approximations," *Proceedings of the IPSJ International Symposium on Information Systems and Technologies for Network Society*, September 1997.

[15] T. G. A. Smith and G. Davenport, "The Stratification System: A Design Environment for Random Access Video," *Workshop on Networking and Operating System Support for Digital Audio and Video*, 1992.

[16] Mitsukazu Washisaka, Toshihiro Takada, Shigemi Anyagi and Rikio Onai, "Video/Text Linkage System Assisted by a Concept Dictionary and Image Recognition," *Proceedings of the International Conference on Multimedia Computing and Systems*, 1996.

[17] D. Zhong and S.-F. Chang, "Video Object Model and Segmentation for Content-Based Video Indexing," *Proc. of IEEE International Symposium on Circuits and Systems*, 1997.