MODELING OF VIDEO OBJECTS IN A VIDEO DATABASES

Lei Chen and M. Tamer Özsu

School of Computer Science
University of Waterloo
{l6chen, tozsu}@uwaterloo.ca

ABSTRACT

In this paper, we present an efficient video data model to represent moving trajectories of video objects and spatio-temporal relationships among the video objects. A video clips is segmented into a set of common appearance intervals (CAI)s. a CAI is a time interval that video objects appear together. Transitions among CAIs record the appearance/disappearance of video objects. Depending on the properties of video objects, they are classified as foreground and background video objects. Foreground video objects are further divided into moving video objects and static video objects. Different models are designed to capture these video objects and spatio-temporal relationships among foreground video objects.

1. INTRODUCTION

There has been considerable research on content-based video modeling and retrieval of video data. Earlier approaches fall into two groups. The first group of techniques are based on low level visual features (e.g., color, shape) [1, 2]. These are are very difficult for naive users who may not be knowledgeable about them. The second class consists of text annotation-based techniques where the retrieval is based on a prepared text description of its contents [3, 4]. These have been criticized as time consuming, subjective and application dependent. Recently, several attempts have been made on modeling video data based on *video objects* [5, 6, 7, 8]. Video objects are the physical objects that appear in the video data. Users of a video database may want to retrieve video data through queries on video objects' properties and spatio-temporal relationships among the video objects. Basically, these queries can be classified into four types:

- Type 1: Queries on video object's movement. Example: "Give me all the scenes in which video objects have similar trajectories as the trajectory of the video object a in the example clip C."
- Type 2: Queries on spatial relationships of video objects. Example: "Give me all the scenes in which actor a appears to the left of actor b."
- Type 3: Queries on simple temporal relationships of video objects. Example: "Give me all the scenes in which actor a appears together with b and a leaves the scene first."
- Type 4: Queries on spatio-temporal relationships of video objects. Example: "Give me all the scenes in which actor a appears on the left of building c before actor b appears to the left of building c."

In order to answer above queries, an efficient video data model needs to be developed that describes the trajectories of video objects and spatio-temporal relationships among these objects. Several approaches have been proposed to model video data based on video objects. Li et. al [5] model pairs of moving objects and spatio-temporal relations among them. However, reducing all the queries to those based on pairs of moving objects limits range of query ability and loses the motion information. In order to deal with semantic heterogeneity of video data, Day et.al [6] propose a Video Semantic Directed Graph (VSDG), which is used to construct users' heterogeneous views of the video data. VSDG is created to model spatio-temporal interactions between video objects. From point of view of multimedia presentation, another two approaches have been proposed to model the spatio-temporal relationships among the video objects. HPNs [7] uses Hierarchical Petri-nets to capture multi-level content of video data, which includes motion trajectories of moving objects and spatio-temporal relationships among video objects. Chen et.al. [8] use augment transition network (ATN) to represent the spaito-temporal relationships among the video objects. VSDG and ATN have difficulties to capture the trajectories of moving objects, and all of three approaches (VSDG, HPNs, ATN) may store redundant information. In this paper, we propose a video data model that represents video in terms of video objects and which overcomes shortcomings of the above models. According to the different properties of video objects, they are classified into different classes and different data structure are used to represent them. The trajectory of a moving object is represented as a dynamic attribute and a motion vector is created to act as the update function to derive the values of this attribute.

2. VIDEO DATA MODEL

We view a video database as consisting of a sequence of clips, where a *clip* refers to an abstraction of a *shot* or a *scene* [9]. The model that is described in this paper is based on capturing the video content in terms of *video objects* (VO) and representing the properties of these objects, their spatio-temporal relationships with each other, and their movement. Consequently, *video objects* are the building blocks of our model and how we model them and their properties is described in Section 2.1. We focus on the representation of the spatio-temporal relationships among VOs in Section 2.2. Finally, in Section 2.3, we complete the model definition by formally defining clips and video.

2.1. Video Object Representation

In order to index and retrieve the *video object*s (VOs), some approximation methods are used to represent them. A *minimum bounding rectangle* (MBR) has been used extensively to approximate a video object since it only needs two points or one point together with width and height. In this paper, we use a MBR describe the spatial layout of a video object , which is represented as a 4-tuple:

$MBR =< x, y, height, width >$

where (x, y) is the coordinates of the upper-left corner of the MBR, and $height$, $width$ are the height and width of the MBR, respectively.

A video object m which appears in a video frame i is represented as a 3-tuple:

$VO^i_m =< VOId_m, MBR^i_m, VFS_m >$

where $VOId_m$ is the identifier of the video object, MBR^i_m is its MBR at frame i, and $VFS_m = \{VC_m, VT_m, VS_m\}$ is the set of visual features whose elements represent color, texture and shape of the VO, respectively. However, usually, the background video objects such as a mountain, a forest, the sea and the sky fill most of the *screen*. Screen here refers to the 2-D space $(x - y)$ of the a video frame. These background video objects impose a visual effect on the audience that they are "behind" or "contain" a foreground video object, such as a person, an animal or a car.

Only the "depth" information is used by background objects to show their relationships to the other video objects. In 2D space, background video objects always "cover" other video objects. Therefore, it is not necessary to represent background video objects using MBRs and modeling their relations with other video objects, which causes the same relation (e.g. "front" or "cover") to be stored redundantly. Thus, we differentiate *foreground video objects* and *background video objects* to simplify their representations.

Definition 1. *Background Video Object* (BVO)

A BVO is a video object that does not have regular shape of its contour and always occupies most of 2D space of a video frame. Examples are sea, sky, forest. Unlike other video objects, A BVO is not described by MBR or its visual features; it is represented as a 2-tuple:

$BVO_n =< VOId_n, Position_n >$

where the domain of $Position_n$ is {*behind*, *full*}. *Behind* implies that all the foreground video objects are in front of the BVO, whereas *full* means that all the foreground video objects are surrounded by the BVO.

Definition 2. *Foreground Video Object* (FVO)

A FVO is a video object that has regular shape of its contour. Examples are persons, buildings and vehicles. A FVO is described by its spatial layout as approximated by its MBR, color, texture, shape, moving trajectory and spatio-temporal relationships with other VOs.

The properties of FVOs are classified into two categories: static attributes and dynamic attributes. *Static attributes* are those features that will not change during the *life span* of the video object. These are color, texture, and shape of the video object. *Dynamic attributes* are the features that will change during the life span of the video object. These are properties such as the spatial position and spatial relationship with other video objects. The life span of a video object refers to the duration of its appearance in a video clip.

It is easy to model static features of video objects. However, the dynamic attributes (trajectories of moving objects and the relative spatio-temporal relations among video objects) change with the evolution of time as measured by the change of video frames. It is impractical to store values of these attributes for every frame; instead, we propose a new way to model these dynamic attributes. Based on the differentiation of dynamic attributes and static attributes, we further classify FVOs into *static video objects* (SVOs) and *moving video objects* (MVOs). "Static" and "moving" are defined relative to the BVOs in the video clip. The SVO is defined as follows:

Definition 3. *Static Video Object* (SVO)

A static video object is defined as a FVO which does not change its position during its appearance in the video and has only static attributes. Examples are buildings, light poles, a stopped car. A SVO is represented by a 3-tuple:

$SVO_p =< VOId_p, MBR^0_p, VFS_p >$

where MBR^0_p is its MBR in the first frame where it appears (since it will not change its position during its life span, only one value is sufficient to describe its spatial layout).

Definition 4. *Moving Video Object* (MVO)

A moving video object is defined as a FVO which changes its position over time[1]. It is represented by a 4-tuple:

$MVO_n =< VOId_n, MBR^0_n, Motion_n, VFS_n >$

where MBR^0_n is the initial position of moving object n, and $Motion_n$ is defined as a sequence of *motion vectors*, $Motion_n = [MV_1, MV_2, \ldots, MV_k]$, as defined below.

Definition 5. *Motion Vector* (MV)

[1] In our data model, we assume that a moving object is rigid or consists of rigid parts connected to each other and all the rigid parts will never disintegrate.

A motion vector is defined as an update function to record trajectories of MVOs:

$MV_m = (S_m, D_m, I_m)$

where S_m is the speed (moving distance per frame), D_m is the movement direction of the moving object, whose domain is *strict directional relations* (north, south, west, east) and *mixed directional relations* (northeast, southeast, northwest,and southwest) [5], and $I_m = [MF_{start}, MF_{end}]$ is a time interval in which the moving object moves in direction D_m with the speed S_m, the initial value of MF_{start} is the starting frame of the clip.

2.2. Modeling Spatio-temporal Relationships Among VOs

We extend the mechanism proposed in [5] to describe spatio-temporal relationships among pairs of moving video objects to model spatio-temporal relationships among FVOs. In our model, the relations between a FVO and a BVO are already captured by the "Position" property of the BVO. Therefore, only the spatio-temporal relationships among FVOs are modelled. Spatial relations between two FVOs are classified into 12 directional relations (*south, north, west, east, northwest,northeast, southwest, southeast, left, right, below and above*) and 8 topological relations (*equal, inside, contain, cover, covered by, overlap, touch, disjoint*). Detailed definitions of these relations can be found in [5].

The spatio-temporal relationships between two FVOs, A_i and A_j , in a given ordered list of time intervals $[I_1, I_2, ..., I_n]$ in a clip,is represented as a list (called a $st - list$):

$[(\alpha_1, \beta_1, I_1), (\alpha_2, \beta_2, I_2), \ldots, (\alpha_n, \beta_n, I_n)]$

where α_i is one of 8 topological relations, β_i is one of 12 directional relations, and (α_i, β_i, I_i) means that α_i and $\beta_i$relations hold between these two FVOs during the interval I_i.

2.3. Modeling of Video Data

As indicated earlier, we view a video as a sequence of clips. A video is represented as a 2-tuple:

$Video_i =< VideoId_i, CList_i >$

where $VideoId_i$ is the identifier of video, and $CList_i$ is a sequence of clips, $CList_i = [Clip_1, Clip_2, ..., Clip_p]$.

A clip is an abstraction of a shot or a scene and a video frame is the basic "building block" of the video, which can be treated as an image, allowing the regular image processing techniques to be applied to frames.

A clip is represented by a 2-tuple:

$Clip_j =< ClipId_j, CAIL_j >$

where $ClipId_j$ is the unique identifier of the clip (each clip in the database has a unique identifier), and $CAIL_j = [CAI_1, CAI_2, \ldots, CAI_n]$ is a sequence of *common appearance intervals* (CAI) as defined below.

Definition 6. *Common Appearance Interval* (CAI)

A common appearance interval $CAI(VO_1, ..., VO_m)$ is the interval, measured in frames, in which video objects VO_1, $...VO_m$ appear all together. It is represented by a 5-tuple:

$CAI_i =< I_i, BVOS_i, MVOS_i, SVOS_i, STS_i >$

where $I_i = \{F_{start}, F_{end}\}$, where F_{start} and F_{end} are the starting and ending frames of the interval; $BVOS_i$, $MVOS_i$, $SVOS_i$ are sets of BVOs, MVOs, SVOs, respectively, that appear in I_i; $STS_i = \{st - list_{i1}, st - list_{i2}, \ldots, st - list_{ik}\}$ is a set of $st - list$s, each element of which is a $st - list$ between two FVOs. The spatio-temporal relations between any two FVOs in a CAI can be expressed in a $p \times p$ matrix, where p is the number of FVOs in that CAI. Since the two inverse spatio-temporal relationships between two FVOs can be derived from each other, we need only to save upper triangle of the matrix. From the above definition, we can see that a CAI not only captures the trajectories of moving objects, but also the spatio-temporal relationships among FVOs and relations between FVOs and BVOs.

A $Clip_j$ is segmented into a sequence of CAIs according to the appearance/or disappearance of the video objects. Therefore, the transition among these CAIs record the appearance/disappearance of video objects.

3. COMPARISON WITH RELATED WORK

We compare our model (for convenience, we call it CAI model) with VSDG [6], HPNs [7], and ATN [8]. Li's [5] work is not included in this comparison since it only focuses on moving video objects. In VSDG, the spatial relationships between any two video objects for any sampled frame are derived from their positions in that frame. The sampling rate directly affects accuracy in capturing spatial relationships among video objects. The same sampling technique has been used in HPN's. In ATN, any relative position (they define 27 relative positions) change among video objects is recorded, causing storage of redundant information. For example, if there are 100 video objects in a video clip, but only one of them is moving, ATN will record all the spatial locations of all the video objects when the relative position between only two video objects changes. ATN also tries to capture the trajectories of video objects by recording the relative positions between two video objects. However, two parallel moving video objects in the scene will not cause the relative position changes between them. CAI model represents the spatio-temporal relationships among video objects using a sequence of intervals, therefore, there is no possibility to miss some spatial relationship. BVOs, SVOs, and MVOs are represented using the different structures in CAI, which eliminates the possibility of storing redundant information. The moving trajectories of MVOs are captured with a sequence of motion vectors. Table 1 shows the comparison among the four models with respect to PMS[2], SRI[3], and CTV[4].

Another basis for comparison is the four types of queries specified in Section 1. CAI model can answer all types of

[2]Possible Missing Spatial relation

[3]Saving Redundant Information

[4]Capturing Trajectories of Video objects

Model	PMS	SRI	CTV
VSDG	Yes	Yes	No
HPNs	Yes	Yes	Yes
ATN	No	Yes	No
CAI	No	No	Yes

Table 1: Comparison among video data models

queries and has the advantage to answer the simple temporal and spatio-temporal queries. Compared to VSDG, which uses transitions among the segments to record only the appearance of new semantic video objects, CAI captures both the appearance and the disappearance of VOs. Therefore, with our CAI model, queries such as "Give me all the clips in which actors a and b appear together and a leaves the scene first" (query type 3) and "Give me all clips in which actor a to the left of actor b and a leaves the scene first" (query type 4) can be easily answered. HPNs also do not capture the transitions of disappearance of the video objects. ATN can record appearance and disappearance of video objects, however, as we mentioned above, like VSDG, ATN does not record the moving trajectories of video objects, so VSDG and ATN could not answer the type 1 queries. Table 2 shows the comparison of four models on handling fours types queries.

Model	Query Type 1	Query Type 2	Query Type 3	Query Type 4
VSDG	No	Yes	Yes*	Yes*
HPNs	Yes	Yes	Yes*	Yes*
ATN	No	Yes	Yes	Yes
CAI	Yes	Yes	Yes	Yes

Table 2: Comparison of query handling (Yes* means the case can be handled with some difficulty)

Finally, our model captures the BVOs as an attribute of the CAIs and uses "Position" of BVOs to describe relations between FVOs and BVOs. Therefore, our CAI model can easily answer range queries such as "give me all the objects that are in front of the sea in video3" or "give me all the objects that are surrounded by a mountain in video3". The other three models do not directly support such rang queries.

4. CONCLUSION AND FUTURE WORK

Properties of video objects and spatio-temporal relationships among them are very important for content-based video retrieval. Most of the video events can be expressed in terms of these relationships. For example, a person (video object) having a spatial relationship (e.g., on top of) with a stone (another video object) may convey the meaning of "stand" or "sit". We define a video model that captures these relationships. Specifically, we differentiate between foreground and background video objects. Among foreground video objects, static video objects will not move during their appearance in the video; however, moving video objects change their position from time to time. This separation allows us to more efficiently model object movement since we do not redundantly store properties of static video objects nor do we capture all the spatio-temporal relations among all video objects. A *common appearance interval* is introduced to capture appearance/disappearance of video objects. Currently, we are defining an accurate similarity measure to compare the trajectories of moving objects.

5. ACKNOWLEDGMENTS

This research is funded by Intelligent Robotics and Information Systems (IRIS), a Network of Center of Excellence of the Government of Canada.

6. REFERENCES

[1] H.J. Zhang, C.Y. Low, S.W. Smoliar, and J.H. Wu, "Video parsing, retrieval and browsing: An integrated and content based solution," in *Proceedings of ACM Multimedia*, San Francisco, CA, 1995, pp. 15–24.

[2] M.L. Cascia and E. Ardizzone, "Jacob: Just a content-based query syste for video databasesthe characteristics of digital video and consideration of designing video databases," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1996, vol. 2, pp. 1216–1219.

[3] T.G.A. Smith and G.Davenport, "The stratification system: A design environment for random access video," in *Workshop on Networking and Operating Syste Support for Digitial Audio and Video*, 1992.

[4] H.T. Jiang, D. Montesi, and A.K. Elmagarmid, "Video-text database systems," in *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, June 1997, pp. 344–351.

[5] J.Z. Li, M.T. Özsu, and D. Szafron, "Modeling of moving objects in a video databas," *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pp. 336–343, 1997.

[6] Y.F. Day, S. Dagtas, M. Iino, A. Khokhar, and A. Ghafoor, "Object-oriented conceptual modeling of video data," in *Proceedings of the Eleventh International Conference on Proceedings on Data Engineering*, March 1995, pp. 401–408.

[7] A.K. Wasfi and G. Arif, "An approach for video metadata modeling and query processing," in *Proceedings of ACM Multimedia*, 1999, pp. 215–224.

[8] S-C. Chen and R. L. Kashyap, "A spatio temporal semantic model for multimedia presentations and multimedia database systems," *IEEE Transaction on Knowledge and Data Engineering*, vol. 13, no. 4, pp. 607–622, 2001.

[9] Y. Rui, T. S. Huang, and S. Mehrotra, "Exploring video structure beyond the shots," in *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, 1992, pp. 237–240.