MODELING VIDEO DATA FOR CONTENT BASED QUERIES: EXTENDING THE DISIMA IMAGE DATA MODEL

LEI CHEN AND M. TAMER ÖZSU

School of Computer Science, University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
E-mail:{ l6chen, tozsu}@db.uwaterloo.ca

VINCENT ORIA

Department of Computer Science, New Jersey Institute of Technology
University Heights Newark, NJ 07102, USA
E-mail: oria@ cis.njit.edu

We present an efficient video data model that extends the DISIMA image data model by adding the video components and setting up links between image and video data. Many video data models have been proposed, most of which describe video data independently of image data and therefore fail to consider the relationship between videos and images. Our proposed model expresses the semantics of video data content by means of salient objects and relationships among them. Connections between video data and DISIMA images are made through key frames, which are extracted from each shot. Based on these connections, techniques used to query image data may be used to query video data. In addition, a set of new predicates has been defined to describe the spatio-temporal characteristics of salient objects in the video data. MOQL is used as a query language, with which we present example queries that can be posed on the proposed video data model..

1 Introduction

Considerable research has been conducted on video data modeling and retrieval in last few years. Based on the characteristics of video data, earlier proposals can be classified into the following three categories:

1. Segmentation-based approaches [Zhang 1993, Günsel et al. 1998, Rui et al. 1998, Yeung and Yeo 1996, Hanjalic et al. 1999, Mahdi et al. 2000], where video data is recursively broken down into *scenes*, *shots* and *frames*. Key frames are extracted from shots and scenes to summarize them, and visual features from those key frames are used for indexing.
2. Annotation-based approaches [Smith and Davenport 1992, Weiss et al. 1995, Hjelsvold and Midtstraum 1994, Oomoto and Tanaka 1993, Jiang et al. 1997], in which a content description (annotation) layer is put on top of a video stream. Each descriptor can be associated with logical video sequences or physically segmented shots and scenes. Both effects are independent and additive.

3. Salient object-based approaches [Li et al. 1997, Nabil et al. 1997, Del Bimbo et al. 1995, Day et al. 1995, Khatib 1999, Chen and Kashyap 2001, Chen and Özsu 2002], where *salient objects*, which are physical objects that appear in video data such as persons, buildings or vehicles, are extracted from the videos and the spatio-temporal relationships among them are used to express events and concepts.

Segmentation-based techniques are interesting because individual shots or scenes [Rui et al. 1998] are logically meaningful units. Furthermore, each shot or scene consists of a sequence of frames and each frame can be treated as an image, allowing the use of existing techniques (with some extensions) that have been developed to model and query image data. However, many segmentation-based techniques only use low-level visual features to represent image content. Consequently, posing queries by setting constraints on those visual features may be difficult and unintuitive for the end users.

In this paper, we propose a video data model which is based upon video segmentation and salient objects. A combination of these two features has been selected for the following three reasons:

1. The combined model can capture both salient objects and low level visual features.
2. The combined model can offer users both browsing and querying capabilities.
3. The combined model captures a well-defined structure of video data: video frames, video shots and video scenes.

As is the case in other segmentation-based techniques, our video data model considers a video database consisting of three types of video sequences: *videos*, *scenes* and *shots*. The content of each video sequence is captured through selected *key frames*, which are defined to be subtypes of the image class. Text annotation may also be associated with each video entity. However, in contrast with image data, video data also possess temporal characteristics. To this end, we define additional predicates for describing temporal properties of video sequences and salient objects. Our video data model supports semantic-based queries through salient objects (e.g. "Give me all the videos in which object ***a*** appears to the left of object ***b***") as well as feature-based similarity queries (e.g. "Give me all the videos that are similar to the example video with respect to color histogram matching with the similarity threshold 0.6"). The proposed video data model extends our previous work on the DISIMA [Oria et al. 2002] image database. The DISIMA system captures the semantics of image data through salient objects, their shapes and their spatial relationships with other salient objects. The DISIMA model is composed of two main blocks, as shown in Figure 1: the image block and the salient object block. The

image block consists of two layers: the *image* layer and the *image representation* layer, which maintain the image *representation independence*. There are two types of salient objects: logical and physical, where the former is an abstraction of the latter. Physical salient objects also have two layers in order to separate the content from the representation. The DISIMA system supports a wide range of queries, from semantic-based queries to feature-based queries.

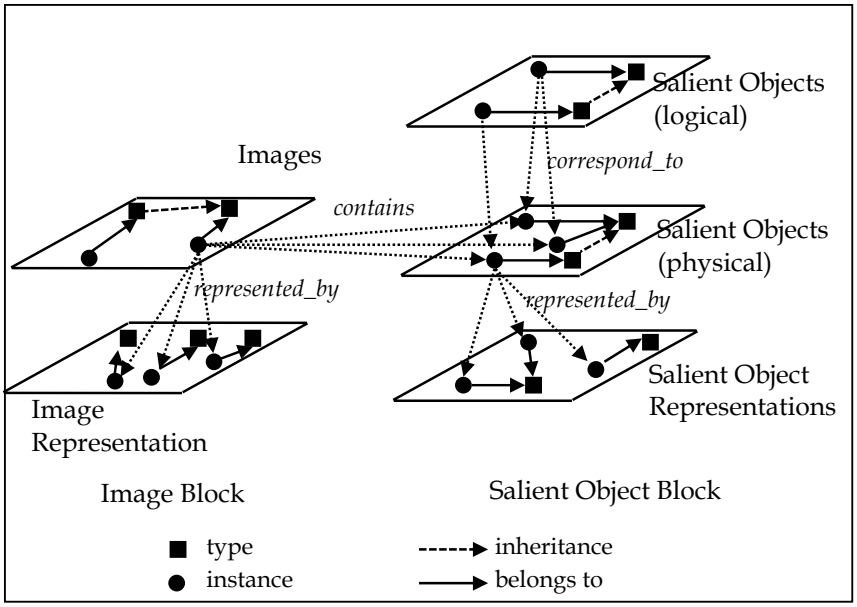

Figure 1. DISIMA Model Overview

The remainder of the paper is arranged as follows. Section 2 presents our video data model. A set of predicates is defined for describing characteristics of video data in Section 3. With the newly defined predicates, Section 4 shows how MOQL [Li et al. 1997a] is used to query video data, and finally, in Section 5 we conclude and indicate some future work.

2 The Video Data Model

A data model is defined as a collection of mathematically well-defined concepts to express characteristics of data. In this section, we present our video data model, its components and some details on representations of video data.

2.1 The Model

The proposed video data model captures the structural characteristics of video data and the spatio-temporal relationships among salient objects that appear in the video. The types of queries which are supported by this model can be classified into the following two categories.

- Semantic-based queries through salient objects. These semantic queries are posed on video data by setting constraints on the properties of salient objects and the spatio-temporal relationships among them. Queries of this category may be further classified into five types as follows.

  - Salient object existence. In this type of query, users are only interested in the appearance of an object, for example, "Give me all the videos in which object ***a*** appears."

  - Temporal relationships. These queries involve temporal relationships among objects in a video. For example, "Give me all the videos in which object ***a*** appears before object ***b***."

  - Spatial relationships. In these queries, users express simple directional or topological relationships among salient objects. For example, "Give me all the videos in which object ***a*** appears to the left of object ***b***."

  - Spatio-temporal relationships. Users are concerned with the spatio-temporal relationships among salient objects in these queries. For example, "Give me all the videos in which object ***a*** appears on the left of object ***c*** before object ***b*** appears to the left of object ***c***."

  - Properties of moving objects. These queries are used for retrieving videos which contain specific properties of a given moving object. For example, "Give me all the videos in which salient objects have trajectories similar to the trajectory of object ***a*** in the example video ***e***." or "Give me all the videos in which object ***a*** and ***b*** move toward each other."

- Feature-based similarity queries. The video data model supports feature-based similarity queries on both salient objects and videos.

  - Feature-based similarity queries on salient objects. These queries retrieve those videos which contain the salient objects with specific values of color, texture and shape. For example, "Give me all the videos that contain a salient object with a color similar to the example color ***x***"(***x*** is specified as the example color value).

- Feature-based similarity queries on videos. In this type of query, users retrieve videos which are similar to an example video in terms of color and texture. For example, "Give me all the videos which are similar to the example video with respect to color histogram matching with the similarity threshold ***y***" (***y*** is a predefined value by users or systems).

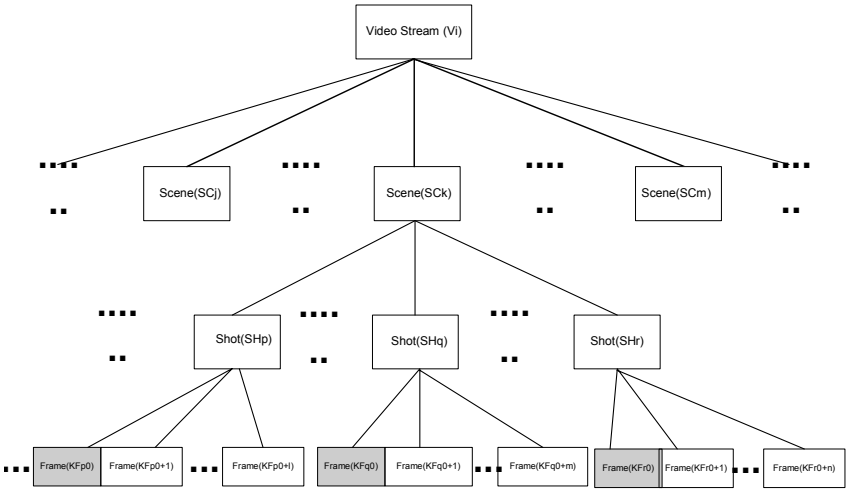

Figure 2. Hierarchical Structure of Video Data

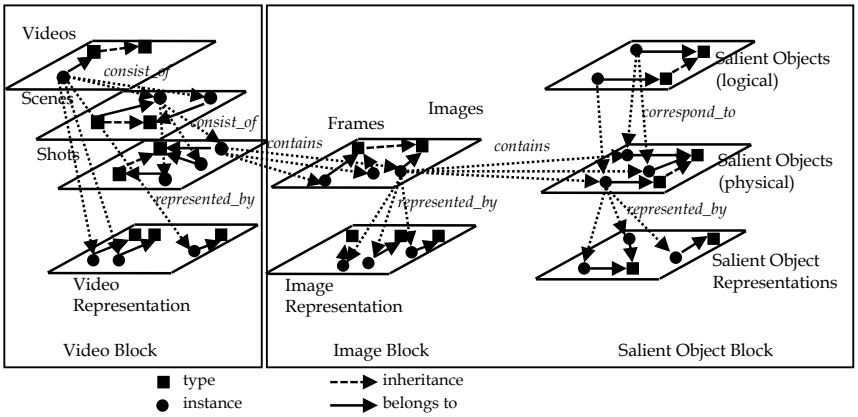

Figure 3. Overview of the Video Data Model

Modeling video data based on segmentation may be divided into three steps. Firstly, video is segmented into *shots*; secondly, *key frames* are selected to represent the shots; finally, *scenes* or *story units* are constructed on the basis of the key frames. A video data model based on video segmentation possesses a hierarchical structure, as seen in Figure 2. A video stream contains several scenes or story units, each scene contains a sequence of shots and each shot contains a sequence of video frames. The shaded frames in Figure 2 are examples of key frames.

Figure 3 shows an overview of the proposed video data model and its links to the DISIMA image data model. A video block is introduced to model video data. As defined in [Oria et al. 1997], a block represents a group of semantically related entities. In Figure 3, the video block has four layers: *video*, *scene*, *shot* and *video representation*. The relationships among the four layers are also shown. The basic composition unit of a video (scene, shot) is a video *frame*, which is treated as a special type of image. It inherits all the attributes from image entities and adds a new time attribute to model its temporal characteristics. In our data model, only the *key frames* are used to represent the contents of a shot. The relationship between key frames and shots sets up the connection between a video block and a DISIMA image block. The definitions of the components of the model are given below.

Definition 1 A *key frame* is a video frame that is selected from a shot to represent the salient contents of the shot. A key frame $KF_i = \langle i, R_i, C_i, D_i, SH_i, I_i \rangle$ is defined as a six-tuple, where:

- i is the unique frame identifier;
- R_i is a set of representations of the raw frame (e.g. JPEG, GIF);
- C_i is the content of a key frame KF_i (see Definition 3);
- D_i is a set of descriptive alpha-numeric data associated with $KF_{i;}$
- SH_i is the shot (see Definition 4) to which KF_i belongs;
- I_i is a close time interval [T_s, T_e], which specifies the portion of the shot that KF_i represents. Since I_i is within the shot, it must satisfy $I_i \prec SH_iI_i$, where $\prec$ is a "sub-interval" operation, defined as follows: Given two time intervals I_A and I_B, $I_A \prec I_B$ if and only if $I_BT_s \leq I_AT_s$ and $I_AT_e \leq I_BT_e$, where T_s and T_e are the starting and end times of an interval.

We identify, as in DISIMA, two kinds of salient objects: physical and logical.

Definition 2 A *physical salient object* is a part of a key frame and is characterized by a position (i.e. a set of coordinates) in the key frame space. A *logical salient object* is used to give semantics to a physical salient object.

For example, a logical salient object actor *Cage* may be created in the video database to store generic information about the actor "Nicolas Cage" such as his name, career, hobbies, etc. A physical salient object $POCage_1$ may be created for an instance of this logical salient object that appears in a key frame. $POCage_1$ would then be linked to the logical salient object *Cage* with reference to the key frame. Based on the definitions of physical and logical salient objects, we define the content of a key frame.

Definition 3 The content of key frame KF_i, $Ci = \langle P^i, s \rangle$,is defined by a pair, where:

- P^i is the set of physical salient objects which appear in KF_i and $\mathcal{P}$ is set of all physical salient objects $\mathcal{P} = \cup_i P^i$;
- $s : P^i \rightarrow \mathcal{L}$ maps each physical salient object to a logical salient object, where $\mathcal{L}$ is the set of all logical salient objects.

Similarly to images in DISIMA, two main representation models are used to represent key frames: the *raster* and the *vector*. Raster models are employed for image application and vector representations are used to reason about the spatial relationships among salient objects in a frame.

Definition 4 A *shot* is an unbroken sequence of frames recorded from a single camera operation. A Shot $SH_j = \langle j, I_j, KFS_j, SC_j, D_j \rangle$ is defined as a five-tuple, where:

- j is the unique shot identifier;
- I_j is a time interval which shows the starting and end time of SH_j;
- SC_j is the scene (see Definition 5) to which SH_j belongs. Since SH_j is within SC_j, it satisfies: $I_j \prec SC_j.I_k$;
- KFS_j is a sequence of key frames [$KF_{j,1}$,…, $KF_{j,m}$], where m is the number of key frames in SH_j. KFS_j is used to represent the content of a shot. The selection of key frames for a shot is discussed in Section 2.3;
- D_j is as given in Definition 1.

Definition 5 A *scene* is a sequence of shots which are grouped together to convey the concept or story. A scene is $SC_k = \langle k, I_k, SHS_k, V_k, D_k \rangle$ defined by a five-tuple, where:

- k is the unique scene identifier;
- I_k is a time interval which shows the starting and end time of SC_k;
- V_k is the video (see Definition 6) to which SC_k belongs. SC_k is a part of V_k, therefore, SC_k satisfies: $I_k \prec V_k.I_1$;
- SHS_k is a sequence of key frames $[SH_{k,1},\ldots, SH_{k,m}]$, where m is the number of shots in SC_k. SHS_k is used to construct SC_k;
- D_k is as given in Definition 1.

Definition 6 A video consists of a sequence of scenes. A *video* $V_n = \langle n, I_n, R_n, SCS_n, D_n \rangle$ is defined by a five-tuple, where:

- n is the unique video identifier;
- I_n is a time interval which describes the starting and end times of the video V_n. $I_nT_s=0$, since all the video start at time 0;
- SCS_n is a sequence of scenes $[SC_{n,1},\ldots, SC_{n,m}]$, that contained by V_n, where m is the number of scenes in V_n;
- R_n is a set of representations of V_n. Two main representation models are developed for videos: *raster* and *CAI*. Raster representations are used for video presentation, browsing and navigation, while CAI (Common Appearance Interval) representations are used to express spatio-temporal relationships among salient objects and moving trajectories of moving objects. The detailed definition of CAI is presented in the following section. The raster presentation may be one of MPEG-1, MPEG-2, AVI, NTSC, etc. Shots and scenes are not directly represented in the representation layer. Through time intervals which record durations of shots or scenes and video identifiers which indicate the video to which shots or scenes belong, portions of video representations can be quickly located and used as the representation for shots or scenes
- D_n is as given in Definition 1.

In order to model the movement of salient objects that appear in the video, we defin motion vectors and moving trajectories.

Definition 7 A motion vector $MV_p = \langle R_p, D_p, I_p \rangle$ is defined as a three-tuple, where:

- R_p 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) [Li et al. 1997];
- D_p is the movement distance in the direction of R_p, which is normalized by removing the visual effect, such as zoom in or zoom out;
- I_p is a closed time interval in which the moving object moves in direction R_p a distance of D_p.

Definition 8 A trajectory $TR_q = [MV_1^q, MV_2^q, ..., MV_k^q]$ is a sequence of motion vectors, where k is the length of the trajectory.

2.2 CAI Presentation

Several earlier proposals on video data modelling [Bimbo et al. 1995, Day et al. 1995, Tush et al. 2000] only deal with the video representation at the shot level, which presents difficulties in answering queries related to spatio-temporal relationships among salient objects within a shot. In order to answer video object existence and temporal relationship queries quickly, an efficient representation scheme must be defined. In fact, only when salient objects appear all together can the spatial relationships among them be computed. For instance, given a shot in which objects ***a*** and ***b*** appear such that ***a*** appears before ***b***, the spatial relationships between the objects can be computed only during the time interval when both of them appear. Therefore, we propose a time interval-based representation to capture the appearance/disappearance of objects within a video.

Definition 9 *Common Appearance Interval* (CAI)
A common appearance interval of a set of objects, $O_1, O_2, ..., O_n$ denoted as $CAI(O_1, O_2, ..., O_n)$ is a time interval within a video in which $O_1, O_2, ..., O_n$ appear together.

Based on the CAI definition, we define the common appearance interval list (CAIL) of a set of objects $O_1, O_2, ..., O_n$ as an ordered list of time intervals (see Definition 10) within a video in which $O_1, O_2, ..., O_n$ appear together.

Definition 10 A list of intervals $[I_1, I_2, ..., I_n]$ is *ordered* if and only if $I_k T_e < I_{k+1} T_s (\forall 1 \le k \le n-1)$.

Figure 4. CAILs of an Example Video

Assume there are *n* salient objects in a video, 2^n-1 CAILs are needed to record all the possible combinations of object appearances. Figure 4 shows an example video extracted from the movie "Gone in 60 seconds". In this video, object O_1 is Randall and object O_2 is Sara. Three CAILs are needed to record the appearance of objects: $CAIL(O_1)=[I_1, I_3]$, $CAIL(O_2)=[I_2, I_3]$, and $CAIL(O_1, O_2)=[I_3]$, according to the appearance/ disappearance of the objects, where I_1, I_2, and I_3 are time intervals in which O_1, O_2, and $O_1\&O_2$ appear, respectively. With a CAIL representation, the queries that test the existence of a salient object and simple temporal relationship between two salient objects (e.g. O_1 before O_2) can be quickly answered.

2.3 Key Frame Selection

The content of a shot is captured by key frames selected from the shot. Therefore, the selection of key frames will greatly affect the amount of information that can be captured and the types of queries that are possible. Several key frame selection algorithms have been proposed in literature [Zhang et al. 1993, Yeung and Yeo 1996, Günsel and Tekapl 1998, Defaux 2000]. In those approaches, low-level visual features (color, shape, texture, luminance and motion) are employed to choose key frames. However, the results of these algorithms cannot reflect semantic changes within the shot, such as the appearance of a salient object or a change of a spatial relationship, because there exists a gap between semantic concepts and low-level features. Nevertheless, semantic information is very important for supporting the five types of semantic queries listed in Section 1. In our model, we identify appearance/disappearance of salient objects and changes of spatial relationships within video shots by combining manual and automatic interpretation techniques. Key frames are first selected through the automatic processes (using the off-the-shelf key frame selection algorithms) and manual interpretation processes are used to mark out the changes of salient objects. With these two steps, key frames are selected whenever the spatial relationships among salient objects change. In other words, a key frame is selected to represent duration within a shot in which the spatial relationships among salient objects contained in that video frame hold.

3 Video and Salient Object Functions and Predicates

In DISIMA [Oria et al. 1997], a set of predicates have been defined for describing the characteristics of images and salient objects. All of these predicates may be used to characterize static properties of video data. However, video queries may also involve temporal constraints on videos (scenes, shots) and on salient objects. In this section, we first give definitions of some basic functions and then define a set of predicates on spatio-temporal characteristics of salient objects and similarity comparison of videos (scenes, shots). Finally, several possible metrics for similarity comparison are proposed.

3.1 Salient Object Related Functions

In this paper, we use the notation $\rightarrow$ as a syntactic representation for a function that carries multiple function specifications. A general function specification is of the form $\mathcal{A} \rightarrow \mathcal{R}$, where $\mathcal{A}$ represents the argument type expression of the function and $\mathcal{R}$ represents the result type.

- *FrameInShot*: $\mathcal{SH} \rightarrow \mathcal{KF}$. Given a shot $SH_i \in \mathcal{SH}$, *FrameInShot*(SH_i) returns a sequence of key frames [$KF_{j,1}$,…, $KF_{j,m}$] ($KF_{j,k} \in \mathcal{KF}$) contained in SH_i, where $\mathcal{SH}$ is the set of all shots, $\mathcal{KF}$ is the set of all key frames, $1 \le k \le m$, and m is the number of key frames in SH_i.

- *ShotInScene*: $\mathcal{SC} \rightarrow \mathcal{SH}$. Given a scene $SC_i \in \mathcal{SC}$, *ShotInScene*(SC_i) returns a sequence of shots [$SH_{j,1}$,…, $SH_{j,m}$] ($SH_{j,k} \in \mathcal{SH}$) that SC_i contains, where $\mathcal{SC}$ is the set of all scenes, $1 \le k \le m$, and m is the number of shots in SC_i.

- *SceneInVideo*: $\mathcal{V} \rightarrow \mathcal{SC}$. Given a video $V_i \in \mathcal{V}$, *SceneInVideo*(V_i) returns a sequence of scenes [$SC_{j,1}$,…, $SC_{j,m}$] ($SC_{j,k} \in \mathcal{SC}$) that V_i contains, where $\mathcal{V}$ is the set of all videos, $1 \le k \le m$ and m is the number of scenes in V_i.

- *ObjectsInFrame*: $\mathcal{KF} \rightarrow \mathcal{L}$. Given a key frame KF_i, *ObjectsInFrame* (KF_i) returns a set of objects $\{O_1, ..., O_n\}$ that appear in key frame KF_i, where $\mathcal{L}$ is the set of all logical salient objects.

- *TrajectoryInShot*: $\mathcal{SH} \times \mathcal{L} \rightarrow \mathcal{MVS}$. Given a shot SH_i and a logical salient object O_j, *TrajectoryInShot*(SH_i, O_j) returns the moving trajectory of salient object O_j in shot SH_i, where $\mathcal{MVS}$ is the set of all sequences of motion vectors.

Similar to the definition of functions *ObjectsInFrame* and *TrajectoryInShot*, we can define functions such as *ObjectsInShot*, *ObjectsInScene*, *ObjectsInVideo, Trajecto-*

ryInScene, and *TrajectoryInVideo*. Because of lack of space, their detailed definitions are not given in this paper.

3.2 Salient Object Related Functions

As we mentioned earlier, the video data are connected with image data through the key frames. Therefore, a set of predicates are defined to describe the spatio-temporal relationships among the salient objects that appear in key frames. By definition, the content of a key frame is represented by salient objects. Therefore, we define ***keyframe_contains*** to check whether a salient object appears in a particular key frame.

Definition 11 ***keyframe_contains*** predicate
keyframe_contains(KF_i, O_j) will return true if O_j appears in KF_i, otherwise, it will return false.

In order to describe the "contain" relations between salient objects and shots, scenes, or videos, we also define three other contain predicates: ***shot_contains***, ***scene_contains***, and ***video_contains***. Based on the composition relationships in Figure 2, we can deduce the definitions of these three predicates from *keyframe_contains* predicate**.** The definition of ***shot_contains*** is presented here as an example.

Definition 12 ***shot_contains*** predicate
shot_contains returns true if O_j appears in shot SH_i.

$$shot\_contains(SH_k, O_j) \Leftrightarrow \exists KF_i \in FramesInShot(SH_k), \; keyframe\_contains(KF_i, O_j)$$

The spatial layout of salient objects contained in key frames are captured through the regions. Therefore, a set of spatial predicates are defined on *regions* in key frames, which are: directional relation predicates: ***keyframe_south, keyframe_north, keyframe_west, keyframe_east, keyframe_northwest, keyframe_northeast, keyframe_southwest, and keyframe_southeast*** and topological relation predicates: ***keyframe_inside, keyframe_covers, keyframe_touch, keyframe_overlap, keyframe_disjoint, keyframe_equal***. The definition of ***keyframe_west*** is given as an example.

Definition 13 ***keyframe_west*** predicate
keyframe_west(O_i, O_j, KF_k) will return true if O_i.*region* appears to the west of O_j.*region* in KF_k, otherwise, it will return false.

With the defined predicates on key frames, we can use them to check the spatial relationship between two salient objects within a shot. ***shot_west*** is presented here as an example, the rest of the predicate definitions, such as ***scene_west***, ***video_west***, etc., can be defined similarly.

Definition 14 ***shot_west*** predicate
shot_west(O_i, O_j, SH_k) is true if O_i is to the west of O_j in shot SH_k.

$$shot\_west(O_i, O_j, SH_k) \Leftrightarrow \exists KF_l \in FramesInShot(SH_k),$$
$$keyframe\_contains(KF_l, O_i) \wedge$$
$$keyframe\_contains(KF_l, O_j) \wedge$$
$$keyframe\_west(O_i, O_j)$$

Besides spatial relationships, temporal relationships also play important roles in describing the characteristics of salient objects that appear in video. Allen [Allen 1983] defines interval algebra for describing and reasoning about the temporal relations between intervals. Seven basic temporal predicates (***before, meet, overlap, during, starts, finishes, equal***) are defined between two intervals to describe the temporal relationships. In the proposed video data model, each key frame is associated with an interval which is the portion of the shot that this key frame represents, therefore, Allen's seven temporal predicates can be used to describe the temporal relationship between two key frames, which has the following basic form: *temporal_predicate(KF_iI_i, KF_jI_j);temporal_predicate*$\in$*{**before, meet, overlap, during, starts, finishes, equal**}*. With these temporal predicates, we can define a new set of temporal predicates to describe the temporal relationship between two salient objects that appear in the same shot, scene, or video. ***shot_before*** is defined here as an example.

Definition 15 ***shot_before*** predicate
shot_before(O_i, O_j, SH_k) returns true if O_i appears before O_j in shot SH_k.

$$shot\_before(O_i, O_j, SH_k) \Leftrightarrow \exists KF_l \in FramesInShot(SH_k),$$
$$\forall KF_m \in FramesInShot(SH_k)$$
$$keyframe\_contains(KF_l, O_i) \wedge$$
$$keyframe\_contains(KF_m, O_j) \wedge$$
$$before(KF_l.I_l, KF_m.I_m)$$

In addition to simple spatial and temporal relations, there exist spatio-temporal relationships between two salient objects, such as *enter, cross, leave, bypass* [Erwig and Franzosa 1999]. We also define the predicates to describe these relationships.

For example, the predicate ***shot_enter*** is used to check whether the sequence order of spatial relations between two salient objects in a shot follows [*disjoint, touch, inside*].

Definition 16 ***shot_enter*** predicate
shot_enter(O_i, O_j, SH_k) is true if O_i enters O_j in shot SH_k.

$$
\begin{aligned}
shot\_enter(O_i, O_j, SH_k) \Leftrightarrow\ & \exists KF_l \in FramesInShot(SH_k), \\
& \exists KF_p \in FramesInShot(SH_k), \\
& \exists KF_q \in FramesInShot(SH_k), \\
& keyframe\_contains(KF_l, O_i) \wedge \\
& keyframe\_contains(KF_l, O_j) \wedge \\
& keyframe\_disjoint(O_i, O_j, KF_l) \wedge \\
& keyframe\_contains(KF_p, O_i) \wedge \\
& keyframe\_contains(KF_p, O_j) \wedge \\
& keyframe\_touch(O_i, O_j, KF_p) \wedge \\
& keyframe\_contains(KF_q, O_i) \wedge \\
& keyframe\_contains(KF_q, O_j) \wedge \\
& keyframe\_inside(O_i, O_j, KF_q) \wedge \\
& before(KF_l.I_l, KF_p.I_p) \wedge \\
& before(KF_p.I_p, KF_q.I_q)
\end{aligned}
$$

3.3 Similarity Predicates

In the context of image or video retrieval, similarity-based queries are more meaningful than exact queries, which are widely used in traditional databases. In DISIMA image database system, a ***similar*** predicate is defined to comparing the similarity between two images with respect to some metric such as salient objects, spatial relationships, colors, textures, or combinations of these. We extend this predicate to compare the similarity between shots (scenes, videos). Furthermore, in video data, the properties of a salient object can be classified into two categories: *static* attributes and *dynamic* attributes. Static attributes are those features that will not change during the *life-span* of the salient object in a shot. These are color, texture and shape. Dynamic attributes are the features that will change during the life-span of a salient object. These are properties such as spatial positions and spatial relationships with other salient objects. Because static properties of salient objects

that appear in a video shot do not change during their life-span, the similarity computation based upon color, texture and shape of salient objects in DISIMA [Oria et al. 1997] can be easily extended to examine the properties of salient objects in a video shot. For dynamic properties of salient objects, we define a predicate (***trajectory_similarity***) on the similarity between trajectories of two moving salient objects. We also propose some possible similarity metrics for the proposed predicates.

Definition 17 ***shot_similarity*** predicate
shot_similarity(SH_i, SH_j) is true if two shots are similar with respect to some metric, such as color, texture, etc.
We have previously proposed some metrics for measure the similarity between two images in [Oria et al. 2002] that can be extended to measure similarity between shots. As each shot is represented by a sequence of key frames, the metric for sequence matching can be applied here, such as longest common subsequences, dynamic time warp, weighted distances, etc.

Definition 18 ***trajectory_similarity*** predicate
trajectory_similarity(TR_i,TR_j) is true if two trajectories are similar with respect to some metric, such as moving direction or moving patterns.
Based on the representation of the trajectories, Li et.al [Li et al. 1997] have developed a function to measure the similarity between two trajectories with respect to moving directions. We extend it by taking the moving distance into consideration. Since L_1 norm performs better than the L_2 in terms of robustness to outliers [Rousseeuw and Healey 1994], L_1 norms is used to measure the differences between moving directions and moving distances. Given two trajectories of two salient objects O_i *and* O_j, $TRi = \left\lfloor MV_1^i, MV_2^i, ..., MV_m^i \right\rfloor$ and $TRj = \left\lfloor MV_1^j, MV_2^j, ..., MV_n^j \right\rfloor$, we compute the similarity as following (without loss of generality, we assume $m \leq n$):
The difference between the moving direction of MV_k^i and that of MV_k^j is

$$rdiff\left(Mk_k^i, MV_k^j\right) = rdistance\left(R_k^i, R_k^j\right) \Big/ Maxdistance$$

where *rdistnace* is a function which is defined in [Li et al. 1997] to measure the difference between two moving directions as show in Table 1. *Maxrdistance* is a constant (here is 4) which is defined as the maximum difference between two moving directions. In Table 4, NT, NW, NE, WT, SW, ET, SE, ST are abbreviations of north, northwest, northeast, west, southwest, east, southeast, south, respectively.

	NT	NW	NT	WT	ST	ET	SE	ST
NT	0	1	1	2	3	2	3	4
NW	1	0	2	1	2	3	4	3
NE	1	2	0	3	4	1	2	3
WT	2	1	3	0	1	4	3	2
SW	3	2	4	1	0	3	2	1
ET	2	3	1	4	3	0	1	2
SE	3	4	2	3	2	1	0	1
ST	4	3	3	2	1	2	1	0

Table 1. Distances of Moving Directions

The difference between moving distance of MV_k^i and that of MV_k^j is

$$ddiff\left(Mk_k^i, MV_k^j\right) = ddistance\left(D_k^i, D_k^j\right) \Big/ Maxddistance$$

where $ddistance\left(D_k^i, D_k^j\right) = \left|D_k^i - D_k^j\right|$ and *Maxddistance* is the maximum value of *ddistance* between D_k^i and D_{k+l}^j where *k=1, ...,m and l=1, ..., n-m.* When *Maxddistance*=0, $ddiff\left(Mk_k^i, MV_k^j\right) = 0$.

The similarity between two trajectories is

$$traj\_sim\left(TR_i, TR_j\right) = 1 - MIN\left\{\frac{\sum_{p=1}^{m}\left(rdiff\left(MV_p^i, MV_{p+q}^j\right) + ddiff\left(MV_p^i, MV_{p+q}^j\right)\right)}{2m}\right\}$$

$$\left(\forall\, 0 \le q \le n-m\right)$$

where *MIN* is a function to get the minimum value of moving direction and moving distance difference.

4 Querying Video Databases

With the newly defined predicates, we can extend the functionality of MOQL [Li et al. 1997a] to query a video database. MOQL is an extension of the standard object query language, OQL [Cattel 1994], which is designed for posing queries over image. In this section, we show how MOQL can be used to pose queries on the video data based on the proposed video data model. In the proposed video databases, different implementations of the predicates have been defined depending on

type of the medium and the query processor is in charge of calling the right predicates. In the following example, queries are posed against shots.

- **Query 1** Find all the shots contains actor ***a***

```
SELECT c
FROM Shots c, Actors a
WHERE c contains a
```

- **Query 2** Find all the shots in which actor ***a*** appears before actor ***b***.

```
SELECT c
FROM Shots c, Actors a, Actors b
WHERE c contains a
AND c contains b
AND a before b
```

- **Query 3** Find all the shots in which actor ***a*** appears to the left of actor ***b***.

```
SELECT c
FROM Shots c, Actors a, Actors b
WHERE c contains a
AND c contains b
AND a left b
```

- **Query 4** Find all the shots in which actor ***a*** enters building ***b***.

```
SELECT c
FROM Shots c, Actors a, Building b
WHERE c contains a
AND c contains b
AND a enters b
```

- **Query 5** Find all the shots in which actor ***a*** has a moving trajectory similar higher then 80% as that of actor ***b*** in shot ***e***.

```
SELECT c
FROM Shots c, Actors a, Actors b
WHERE c contains a
AND e contains b
AND a.trajectory similar b.trajectory
similarity >0.8
```

- **Query 6** Find all the shots that contain a silent object with a color similar at 80% to RGB value (255,0,255).

```
SELECT c
FROM Shots c, LSO o
WHERE c contains o
AND o.color similar colorgroup (255,0, 255)
similarity 0.8
```

5 Conclusion

Previous video data modelling approaches ignore connections between video data and images and they lack facilities to represent semantics of video data. In this paper, we propose a video data model that is an extension of the DISIMA image database system. Each video frame can be considered as a special type of image. Based on this principle, we add a video block to the existing DISIMA data model and set up links between videos and images. Therefore, operators defined for querying image data can be used to answer queries related to salient object and image features of video data. We also define a set of spatial and temporal predicates on salient objects in video data by extending the predefined predicates in DISIMA. Based on the composition relationships among videos, scenes, shots and frames, the semantics of video data are represented in terms of salient objects. In order to support both semantic queries through salient objects and feature-based similarity queries, similarity computations on static attributes and dynamic attributes of salient objects and video are defined. MOQL has been extended to facilitate queries on video data with the extended predicates. In our future work, we will implement possible similarity metrics for comparing shots and trajectories and compare their efficiency and accuracy. In addition to that, an efficient indexing structure will be designed to improve query efficiency. The design of the indexing structure should consider query efficiency as well as space requirements.

6 Acknowledgements

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

References

W. Al-Khatib and A. Ghafoor (1999). An approach for video meta-data modeling and query processing. In Proceedings of ACM International Conference on Multimedia, pp 215-224.

A. Del Bimbo, E. Vicario, and D. Zingoni (1995). Symbolic description and visual querying of image sequences using spatio-temporal logic. IEEE Transactions on Knowledge and Data Engineering, 7(4): pp 609-622.

L. Chen and M. T. Özsu (2002). Modeling video objects in video databases. In Proceedings of IEEE International Conference on Multimedia and Expo, pp 171-175.

S.-C. Chen and R. L. Kashyap (2001). A spatio-temporal semantic model for multimedia presentations and multimedia database systems. IEEE Transactions on Knowledge and Data Engineering, 13(4): pp 607-622.

Y.F. Day, S. Dagtas, M. Iino, A. Khokhar, and A. Ghafoor (1995). Object-oriented conceptual modeling of video data. In Proceedings of the 11th International Conference on Data Engineering, pp 401-408.

F. Dufaux (2000). Key frame selection to represent a video. In Proceedings of IEEE International Conference on Image Processing, pp 275-278.

M. Erwig and R. Franzosa (1999). Developments in spatio-temporal query languages. In Proceeding of DEXA Workshop on Spatio-Temporal Data Models and Languages, pp 441-449.

B. Günsel, A. M. Ferman, and A. M. Tekapl (1998). Temporal video segmentation using unsupervised clustering and semantic object tracking. Journal of Electronic Imaging, 7(3): pp592-604.

B. Günsel and A. M. Tekapl (1998). Content-based video abstraction. In Proceedings of IEEE International Conference on Image Processing, pp 128-131.

A. Hanjalic, R.L. Lagendijk, and J. Biemond (1999). Automatically segmenting movies into logical story units. In Proceedings of International Conference on Visual Information Systems, pp 229-236.

R. Hjelsvold and R. Midtstraum (1994). Modelling and querying video data. In Proceedings of 20th Very Large Data Base Conference (VLDB), pp 686-694.

J.F.Allen (1983). Maintaining knowledge about temporal intervals. ACM Communications, 26(11): pp 832-843.

H.T. Jiang, D. Montesi, and A. K. Elmagarmid (1997). VideoText database systems. In Proceedings of IEEE International Conference on Multimedia Computing and Systems, pp 344-351.

J.Z. Li, M.T. Özsu, and D. Szafron (1997). Modeling of moving objects in a video databas. Proceedings of IEEE International Conference on Multimedia Computing and Systems, pp 336-343.

J.Z. Li, M.T. Özsu, D. Szafron, and V. Oria (1997a). MOQL: A multimedia object query language. Proceedings of the Third International Workshop on Multimedia Information Systems, pp 19-28.

W. Mahdi, M. Ardebilian, and L.M.Chen (2000). Automatic video scene segmentation based on spatial-temporal clues and rhythm. Networking and Information Systems Journal, 2(5): pp 1-25.

M. Nabil, A.H. H. Ngu, and J. Shepherd (1997). Modeling moving objects in multimedia database. In Proceedings of the 5th Conf. on Database Systems for Advanced Applications, pp 67-76.

E. Oomoto and K. Tanaka (1993). OVID: Design and implementation of a video-object database system. IEEE Transactions on Knowledge and Data Engineering, 4(5): pp 629-643.

V. Oria, M.T. Özsu, and P. Iglinski (2002). Querying images in the DISIMA DBMS. Multimedia Application and Tools, 2002. in press.

V. Oria, M.T. Özsu, L. Liu, X. Li, J.Z. Li, Y. Niu, and P. Iglinski (1997). Modeling images for content-based queries: the DISIMA approach. In Proceedings of Second International Conference on Visual Information Systems, pp 339-346.

R.Cattel (1994). The Object Database Standard: ODMG-93. Mogan Kaufmann.

P. J. Rousseeuw and G. Healey (1987). Robust Regression and Outlier Detection. John Wiley & Sons.

Y. Rui, T. S. Huang, and S. Mehrotra (1992). Exploring video structure beyond the shots. In Proceedings of IEEE International Conference on Multimedia Computing and Systems, pp 237-240.

T.G.A. Smith and G. Davenport (1992). The stratification system: A design environment for random access video. In Proceedings of Workshop on Networking and Operating System Support for Digital Audio and Video pp 250-261.

R. Tusch, H. Kosch, and L. Böszömenyl (2000). VIDEX: an integrated generic video indexing approach. In Proceedings of ACM international Conference on Multimedia, pp 448-451.

R.Weiss, A. Duda, and D.K Gifford (1994). Composition and search with a video algebra. IEEE Multimedia, pp 12-25.

M.M. Yeung and B.-L. Yeo (1996). Time-constrained clustering for segmentation of video into story units. In Proceedings of 13th International Conference on Pattern Recognition, pp 375-380.

H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H. Wu (1995). Video parsing, retrieval and browsing: An integrated and content based solution. In Proceedings of ACM International Conference on Multimedia, pp 15-24.

H.J. Zhang, A. Kankanhalli, and S.W. Smoliar (1993). Automatic partitioning of full-motion video. Multimedia Systems, 1(1): pp 10-28.