8th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS
S u c e a v a, R o m a n i a, M a y 25 – 27, 2 0 0 6

SPATIO-TEMPORAL KNOWLEDGE REPRESENTATION AND QUERY PROCESSING IN VIDEO INFORMATION SYSTEMS

Tudor-Ioan CERLINCĂ [1], Ştefan-Gheorghe PENTIUC [2]
"Stefan cel Mare" University of Suceava
str.Universitatii nr.13, RO-720229 Suceav, Romania
[1] *tudor_c@eed.usv.ro*
[2] *pentiuc@eed.usv.ro*

Abstract. *The aim of this paper is to present a multi-level index structure for video spatio-temporal databases modeling and a specific SQL language for query processing. The data structure we propose for video spatio-temporal knowledge representation is intended to minimize the disk space necessary for data storing and to efficiently represent the spatial and temporal relationships among the objects in the video sequences. The data structure and SQL language were designed to support both spatial and temporal queries.*
Keywords: *moving target detection, VHR tree, query processing, spatio-temporal knowledge*

Introduction

Video spatio-temporal databases are highly used in applications where data-types can be characterized by both spatial and temporal semantics. A Video Information System for indoor surveillance can be a good example of such application.
The research and development in this area started decades ago, but there was a separate research both in temporal and spatial databases. Now, the two separate models are integrated into a single one, called the spatio-temporal database model.
A lot of models for spatio-temporal knowledge representation where designed, each of them trying to optimize the spatio-temporal queries. The best models are those who are based on the VHR tree [1] [4]. Another good model is the one that is based on the association map and frame-segment tree [2]. Spatio-temporal queries can be processed very quickly but the model wasn't designed to minimize the storage space, which makes him unusable when the video sequences are big.
The model we developed is based on the VHR tree and was designed for faster access to video data. The storage space is also minimized. The efficient representation of temporal and spatial relationships among objects in the video sequences was another issue we have focused on.
The first step in the development of such model is to efficiently track the moving objects from the video sequence. The Video Information System we propose detects the moving targets on the base of a background model, updated continuously in accordance with light and meteorological conditions [5].

Spatio-Temporal knowledge representation model

Typically, a video spatio-temporal database will store the following information:

- time stamp information
- spatial properties of each object which appears in the video sequence
- spatial and temporal relationships among the objects in the video sequence

The model we developed is based on the VHR indexing tree which was originally designed by Lei Chen & Vincent Oria [1].
The spatio-temporal knowledge representation model is build upon a multi-level indexing structure.
The first level of the indexing structure is a linked list of distinct objects which appears in the video sequence (fig. 1). Every object in the

list is defined as a 3-tuple: <*IDi,SIi,LKFi*> where:

- *IDi* is the object's unique identifier
- *SIi* describes the spatial properties of the object
- *LKFi* is a list of frames which contains the current object. Each node in the list contains a unique identifier of the frame and a pointer to the corresponding root of the VHR tree.

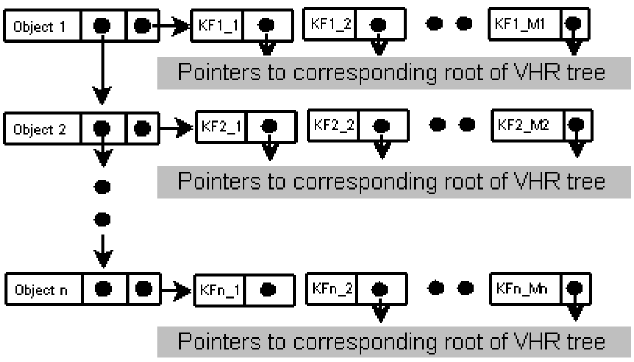

Figure 1. The first level of an indexing structure.

The first level of the indexing structure can be used for faster access to those frames which contains one or more specified objects. Typically, the following queries will be processed:

- *find all frames in which object* ***a*** *appears*
- *find all frames in which both objects* ***a*** *and* ***b*** *appears*
- *find all frames in which object a appears before object* ***b***

To successfully build the first level, all objects in the video sequence must be uniquely identified from one frame to another, which is not always an easy task. Object identification is done using two different methods [5].

The first one is based on the calculation of the minimum Euclidian distance between the weight center of an object from the current frame and the weight center of all objects from the previous frame. This method is quite simple and very fast but there are some ambiguity cases which are not fully covered. This may result in defectous target identification. For such cases, the second method is applied which is based on the calculation of the minimum Euclidian distance between the color histogram of an object from the current frame and the histogram of all objects from the previous frame.

The second level of the indexing structure is the VHR tree, which will store the following information:

- the unique identifier of each frame in the video sequence
- scene objects defined by their spatial properties
- spatial and temporal relationships among objects in the video sequence

The VRH tree is an extension to HR tree and was designed to minimize the disk space needed for data storing and optimize the spatio-temporal queries.

For every frame in the video sequence, the HR model computes the difference between R trees of current and previous frames. The major advantage of VHR model is that he computes the difference between R trees of the current frame and R trees of the two last frames. The main idea of both HR and VHR trees is to reuse the R trees at different moments of time.

Figure 2 presents an example of a VHR tree. As it is shown, the R tree corresponding to Frame 3 is using an important part of the R tree for Frame 1, because there are no big differences between these frames. The only difference is that the object 3 has moved to a new location. The same situation can be observed for frame 4 and 2. The HR tree will not reuse the R trees but will add new R trees which are pretty similarly one to each other. It is obvious that the VHR model will save more disk space than HR model.

Few examples of spatio-temporal queries that can be processed using the VHR tree are listed below.

- *find all frames in which object* ***a*** *appears in the left of object* ***b***
- *find all frames in which object* ***a*** *appears in the right of coffee maker after object* ***b*** *appears in the left of coffee maker*
- *find trajectory of object* ***a***

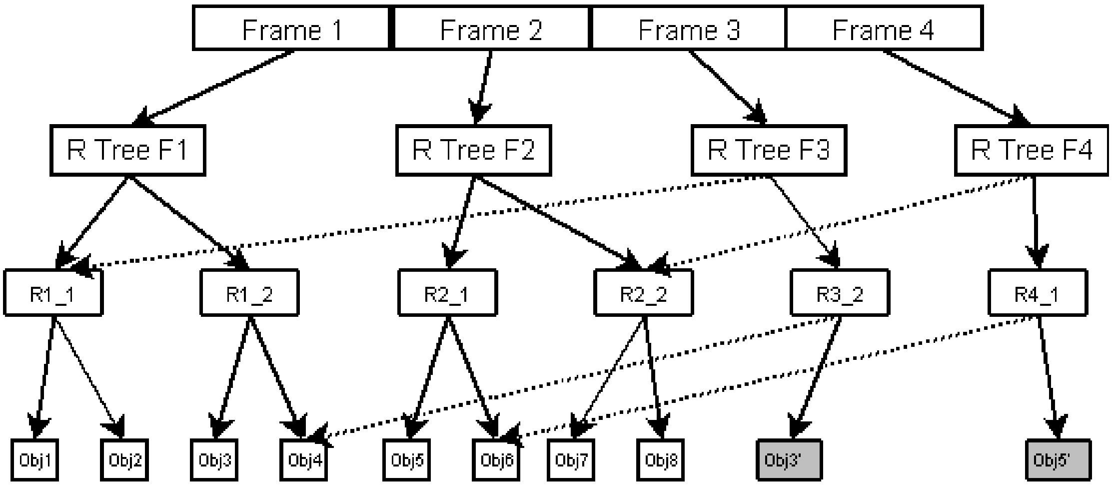

Figure 2. An example of a VHR tree.

Spatial properties modeling

The spatial property of an object can be easily defined using the 2D coordinate system. There are many ways to define the spatial property of an object. An interesting definition is proposed in [2]: the spatial property of an object is defined as a pair *SP(a)=(R,FI)* where *R* is the rectangle that covers all areas in which object ***a*** appears in the frame interval $FI = [FK_i, FK_f]$.

When the object ***a*** is not moving in the frame interval *FI, R* will be the smallest rectangle which covers all parts of the object.
When object ***a*** is moving, *R* should be choose in a such manner that object ***a*** to have a uniform movement in *R*.

Spatio-temporal relationships modeling

We have focused on the representation of the following classes of spatio-temporal relationships:

- temporal relationships: *together, before, after*
- spatial relationships which are divided in two different categories: directional (*left, right, up, bottom*) and topological (*partial covered, total covered, touched*)

According to [2], the spatio-temporal relationship between two objects *Ai* and *Aj* in the frame interval $FI = [FK_i, FK_f]$, may be expressed as $A_i(r, fz, Ik)$, where r is a directional or topolocigal relationship and fz is a fuzzy member who has a value between 0 and 1. Taking into consideration the spatial distance between two objects, the fuzzy member fz will specify how much this distance satisfy a specific spatial relationship. Figure 3 presents different cases for a ***Object1***(*left,fz,1*)***Object2*** relationship between two objects in a video sequence.

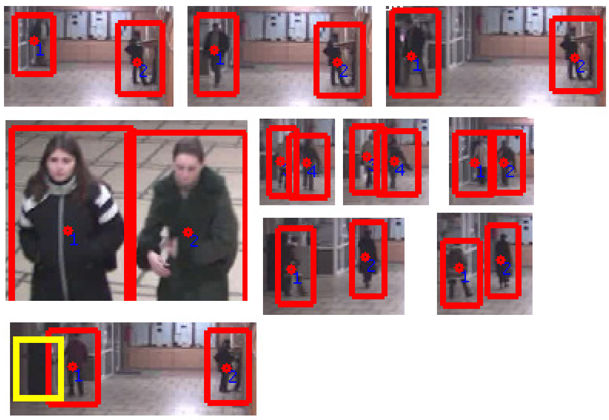

Figure 3. Different cases of objects relationship in video sequence.

As it is shown we have focused on the representation of two different spatial relationships: between two moving objects and between one moving object and one stationary object. In our case, the stationary object (the coffee maker) is marked with yellow rectangle.

For every kind of spatial relationship, the fuzzy member *fz* will be computed as follows [2]:

RELATIONSIP	ANGLE	VALUE *fz*
TOP(A,B)	actan(x/y)	1-(ANGLE/90)
LEFT(A,B)	actan(y/x)	ANGLE /90
TOP-LEFT(A,B)	actan(x/y)	1(abs(ANGLE-45)/45)
TOP-RIGHT	actan(y/x)	1-((ANGLE-45)/45)
BOTTOM(A,B)		TOP(B,A)
RIGHT(A,B)		LEFT(B,A)
BOTTOM-RIGHT(A,B)		TOP-LEFT(B,A)
BOTTOM-LEFT(A,B)		TOP-RIGHT(B,A)
x and y are the horizontal and vertical distance between the weight center of two objects		

The temporal relationships can be defined as follows:

together: Two objects *Obj1* and *Obj2* shows up *together* only if there is a frames interval $FI = [FK_i, FK_f]$ in which both objects appears.

before: An object *Obj1* appears *before Obj2* only if two frames interval $FI1$, $FI2$ exists for which $FK1_f \leq FK2_i$ and $Obj1 \in FI1$, $Obj2 \in FI2$.

SQL Language for Video Spatio-Temporal Databases

The SQL language we have developed supports both spatial and temporal queries. At this moment, the SQL engine which is responsible with query translation and processing is not integrated into a DBMS but in our Video Information System.

The SELECT statement syntax which is listed below is pretty similar with the one from the most of the DBMS.

SELECT FRAMES|COUNT [ALL|FROM start_frame TO [end_frame]]
WHERE condition

Some of the conditions that are supported by our SQL language are listed below:

1. NUMBER OF OBJECTS IS EQUAL| BIGGER|SMALLER|BETWEEN(mi*nNr Obj*,*maxNrObj*) [WITH|THAN *RefValue*]
2. OBJECT *Obj1* APPERARS [IN AREA (X1,X2,Y1,Y2)]
3. RELATIONSHIP *Obj1 fuzzy*/*spatial /temporal relationship Obj2*

Logical operations are also supported by using *AND, OR, NOT* keywords.

Now, let's take two examples of queries written in natural language and show how we can translate them into SELECT statements.

Query 1:

Natural language: *find all frames in which object **a** appears*

SQL Language**:** *SELECT FRAMES ALL WHERE OBJECT a APPEARS*

Query 2:

Natural language: *find all frames in which object **a** appears in the left of object b and object c appears in the right of coffee_maker*

SQL Language**:** *SELECT FRAMES ALL WHERE RELATIONSHIP a LEFT b AND RELATIONSHIP c RIGHT cofee_maker*

The Video Information System

The VIS we developed was tested with video sequences recorded in different locations of the "Stefan cel Mare" University of Suceava. The main window of the application is presented in figure 4. The application consists of four modules:

1. moving target detection module
2. video spatio-temporal database builder based on the VHR tree
3. SQL engine for query processing
4. query builder which helps users to build their own SELECT statements using graphical controls

To increase the speed of VIS, users can define one or more exclusion areas, which are rectangular areas where moving object will never be present.

For personalized queries, users can give a name to each object present in the video sequence. Even the static objects can have names: coffee maker, access door 1, access door 2, etc. These names can be further used in the SELECT statements.

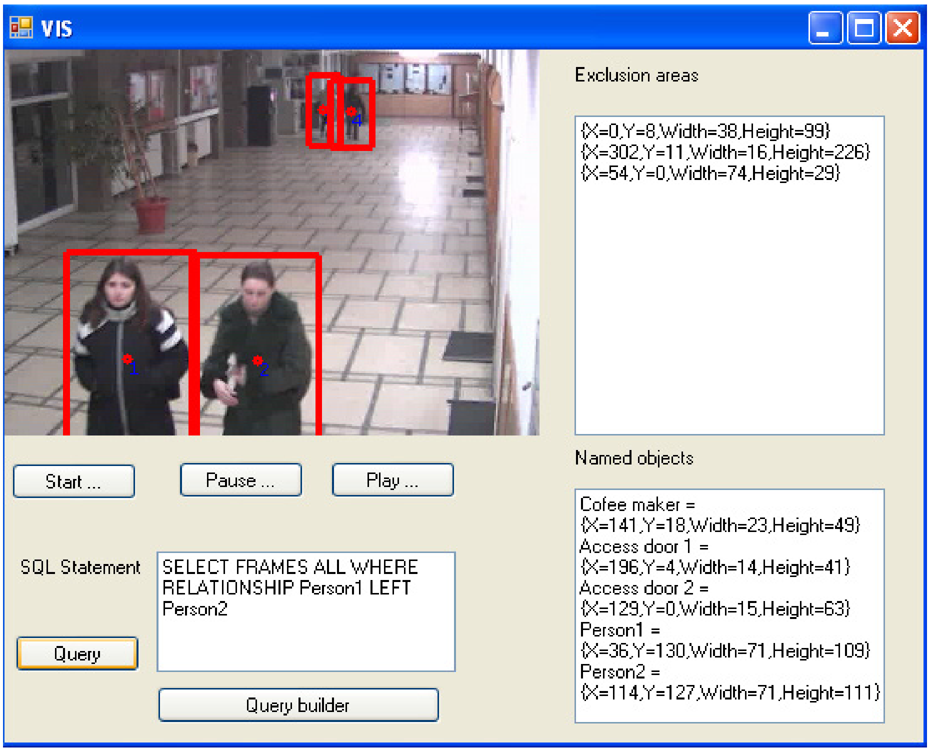

Figure 4. The main window of the Video Information System.

Conclusions

In this paper is presented the background and the overall structure of a VIS elaborated in the University of Suceava. The system was experimented in the University video surveillance system.
The main contributions that may be revealed are the SQL language we defined, SQL engine, and data structure we implemented and improved for spatio-temporal relationship and knowledge representation.
Future work will be focused on the integration of the SQL engine into PostgreSQL [3] or to develop a framework in ActiveX similar to [6]. The engine will also be improved, in order to be able to translate the natural language into SELECT statements. All modules will be integrated in a collaborative environment [7].

References

[1] Lei Chen, Vincent Oria (2002), *A multi-level index structure for video databases*, Multimedia Information Systems , pag, 28-37

[2] Mesru Koprulu, Nihan Kesim (2004), *Spatio - Temporal Querying in Video Databases*, Inf. Sci., pag. 131-152

[3] Mohamed Minout, Esteban Zimanyi (2004), *Algebra-to-SQL Query Translation for Spatio-Temporal Databases*, DEXA pag. 904-913

[4] Lei Chen, Vincent Oria (2004), *MINDEX: An Efficient Index Structure for Salient Object-based Queries in Video Databases*, Multimedia Systems, 2004, pag. 56-71

[5] Tudor-Ioan Cerlinca (2004) – *A distributed system for real time traffic monitoring and analysis*, Advances in Electrical and Computer Enginering, Suceava, Romania, ISSN-1582-7445 pag. 82-86

[6] Cornel Turcu, Cristina Turcu (2003), *Facilities in Designerax – An All-Purpose Automation Activex Framework*, Advances in Electrical and Computer Engineering, ISSN 1582-7445 - No 1 / 2003, pp. 17-20

[7] Cristina Turcu (2003), *A Proposal to Develop A Collaborative Learning Environment*, Advances in Electrical and Computer Engineering , ISSN-1582-7445 no.2, pp.105-109