A Spatio-Temporal Database Model on Transportation Surveillance Videos

Xin Chen and Chengcui Zhang
Dept. of Computer and Information Sciences, University of Alabama at Birmingham, Birmingham AL 35294, USA
{chenxin, zhang}@cis.uab.edu

Abstract

With the rapid growth of multimedia data, there is an increasing need for robust multimedia database model. Such a model should be able to index spatio-temporal data thus efficient access to data whose geometry changes over time can be provided. In this paper, a spatio-temporal multimedia database model for managing transportation surveillance video data is proposed. The objective is to build a spatio-temporal database schema for transportation surveillance videos, in which queries can be answered easily and efficiently. The proposed spatio-temporal model for transportation surveillance videos combines the strength of two general-purpose spatio-temporal multimedia database models - the Multimedia Augmented Transition Network model (MATN) and the Common Appearance Interval (CAI) model. While MATN model is good at modeling the replay of the multimedia presentation and the spatial-temporal relations of semantic objects in the video, it is not efficient in modeling or querying the trajectories of moving objects. In the mean time, while Common Appearance Interval (CAI) model can be used to better answer trajectory-based queries, it explicitly stores the spatial relations of pairs of objects in the model, which is considered redundant in transportation video databases. The proposed model bases its structure on MATNs and adopts the concept of CAI to segment transportation surveillance videos. Since this model is motivated by transportation surveillance applications, it has some domain specific features. It models each traffic light phase in a MATN-like network and models the corresponding video segment using CAIs. In addition, CAIs are further divided into sub-intervals and modeled by the sub-network structure in MATNs. In this paper, the proposed model, together with a brief introduction of the vehicle extraction/tracking/classification, is presented with its formal definition and some sample queries. The advantages of our model in comparison with other models are also demonstrated.

Proceedings of the 3rd Workshop on STDBM
Seoul, Korea, September 11, 2006

1. Introduction

While spatio-temporal applications (Intelligent Transportation Systems, health, climate changes, etc.) have only recently attracted researchers in this field, most of the existing work has concentrated on general-purpose spatio-temporal database models and query languages[2][3][5][6]. The existing models all have their advantages in modeling certain aspects of the data. However, none of them are "jack-of-all-trades" without any redundancy or lost of any efficiency. Since different spatio-temporal applications may have different emphasis on the properties and queries of the domain-specific data, there is a need for designing domain-specific spatio-temporal database model.

In building an intelligent transportation system, a large amount of transportation surveillance videos are collected via surveillance cameras. Various algorithms are proposed to analyze these video data. However, it is still a challenge to store and manage these videos with an efficient indexing and querying schema. In addition, it is necessary to build an integrated system that captures a comprehensive set of requirements which are needed in building a transportation surveillance video database. Such requirements include the extraction of vehicle objects from surveillance videos, vehicle tracking, vehicle classification, and spatio-temporal modeling that provides efficient data indexing and querying for transportation domain. To our best knowledge, there is no such system in this field that is sophisticated enough to manage video data efficiently in transportation surveillance domain. One of the research directions in spatio-temporal database management is to incorporate data streams and evaluate continuous queries over data streams [9] [10] [11]. For example, in [9], Mouza et al. proposes a data model for reporting continuous queries based on mobility pattern matching. Continuous queries are managed as a discrete process relying on events related to the moves of objects. Several general-purpose spatio-temporal models for video databases are proposed in [2][3][5][6]. However, since different spatio-temporal applications may have different emphasis on the properties and queries of the domain-specific data, there is still a need for designing domain-specific spatio-temporal models. In this paper, we proposed a spatio-temporal multimedia database model for storing, indexing, and querying transportation surveillance videos. It is worth mentioning that the proposed work focuses on designing a conceptual model, which serves as a basis for defining entities, attributes, and relationships. As this is our initial work, details such as GUI interface, physical layers, access strategy or operations will be addressed separately in our future work.

In our previous work, we proposed a vehicle extraction, vehicle tracking, and vehicle classification framework [4] [8] for indexing transportation surveillance videos. With this framework, each distinct vehicle can be automatically identified at diffident locations in a video frame. The object-level information such as the bounding box and the centroid can be recorded and stored in the database for future queries. However, it would be unnecessary to record such information in each frame as it will introduce a lot of redundancy in the database. Therefore, an intuitive method is to segment a video into meaningful segments and only record the "key frames". Transportation surveillance video segmentation is not as easy a task as that for regular videos such as movies, since it is hard to detect video shots or events in the continuous transportation surveillance video sequences. In L. Chen's CAI model [2], a video segmentation concept -- Common Appearance Interval (CAI) is proposed, which has some flavor of a video shot in a movie. In this concept, each video segment is endowed with some "semantic" meaning in terms of temporality and spatial relations. This concept is adopted in the proposed model in this paper. In L. Chen's CAI model, the spatial relations of pairs of objects are recorded. This makes it convenient to query the spatial relation of two objects. However, if there are n objects appearing in the frame, there will be C_n^2 pairs of records in the database. Since the spatial relations of vehicles are not the frequent query targets in the transportation video database, this will introduce a lot of redundancy. In this paper, we follow the basic idea of S.-C. Chen's MATN (Multimedia Augmented Transition Network) model [3] in solving this problem.

MATN model is good at modeling the replay of multimedia presentations. It also provides an efficient mechanism in modeling the spatial relations of semantic objects in the video. One disadvantage of MATN is that it cannot be efficiently applied to modeling the trajectories of moving objects. However, trajectories of vehicles are often queried in a transportation video database. Therefore, in our model, MATN is adjusted to suit our specific needs.

The MATN model proposed by S.-C. Chen [3] and L. Chen's CAI model [2] are two general-purpose models. Our proposed model combines the strength of these two models. We base our structure on MATN and adopt the concept of CAI to segment traffic videos. Motivated by our specific application, our model has some features neither of the two general-purpose models has. The proposed model models each traffic phase in a MATN-like main network and models the corresponding video segment using CAIs. While the main network models the spatio-temporal relations of vehicle objects at a coarser level, CAIs can capture more details of such relations. In addition, since MATN uses the concept of Multimedia Input Strings (MISs) as the input of an MATN model, we further extend its definition to model CAIs in MATN's main network. CAIs are further divided into sub-intervals and modeled by the sub-network structures in MATNs. The direction information of a moving vehicle rather than the spatial relation between two vehicle objects is recorded in our model. This is due to the fact that transportation video database queries are more often concerned with a moving vehicle's driving direction than its spatial relation with another object. We argue that in this type of applications, there is no need to store a huge amount of redundant information that is not often queried.

Some background information on transportation surveillance video processing is introduced in Section 2. The proposed model is formally defined in Section 3. Section 4 shows query methods and some sample queries. Section 5 analyzes the advantages of the proposed model. Section 6 concludes the paper.

2. Transportation surveillance video processing

In this section, the processing of transportation surveillance videos is introduced to provide some background information on building an intelligent transportation system. In our previous work [4], an unsupervised segmentation method called the Simultaneous Partition and Class Parameter Estimation (SPCPE) algorithm, coupled with a background learning and subtraction method, is used to identify the vehicle objects in a traffic video sequence [8]. The technique of background learning and subtraction is used to enhance the basic SPCPE algorithm in order to better identify vehicle objects in traffic surveillance videos. Figure 1 shows an example of the segmentation result of a vehicle with background extracted. The rectangular area is the Minimal Bounding Rectangle (MBR) of the vehicle that is represented by (x_{low}, y_{low}) and (x_{high}, y_{high}) -- the coordinates of the bottom right point and the upper left point of the MBR. ($x_{centroid}$, $y_{centroid}$) are the coordinates of that vehicle segment's centroid. It is used for tracking the positions of vehicles the across video frames.

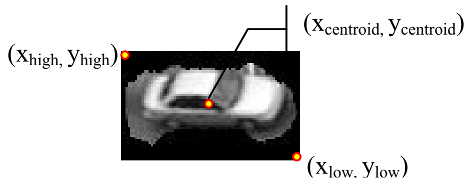

Figure 1 An example vehicle segment.

The framework in [4] has the ability to track moving vehicle objects (segments) within successive video frames. By distinguishing the static objects from mobile objects in the frame, tracking information can be used to determine the trails of vehicle objects. The last phase of the framework is to classify vehicle objects into different classes such as SUVs, pick-up trucks, and cars, etc. The classification algorithm is based on Principal Component Analysis.

With this framework, lots of useful data and information is generated. This provides a basis for an intelligent transportation system. The whole process is automatic. In this paper, a spatio-temporal database model is proposed to further organize, index and query these information.

3. Model definition

While the proposed model is a combination of S.-C. Chen et al.'s MATN model [3] and L. Chen's CAI model [2], its general structure is based on MATN. Therefore, in this section, we will start from introducing the basic idea of MATN model. Following that, the formal definition of our model is presented.

3.1 MATN model

Multimedia Augmented Transition Network (MATN) model originates from the Augmented Transition Network (ATN) [1] which is used for natural language processing. The inputs of an ATN are sentences that are composed of words. Similarly, in MATN, the inputs are Multimedia Input Strings that can be denoted by regular expressions. An MATN model simulates a finite state automaton in that it is constructed by nodes (states), arcs, inputs and transition functions. However, unlike finite state automaton, MATN allows recursions and has the condition/action tables. Recursions allow the user to play the desired video segments repeatedly. Condition/action tables are especially useful in an online environment. With the limitation of bandwidth, the user can choose to play some parts of the video when condition permits or get compressed version of the video. That is, an MATN can control the synchronization and Quality of Service (QoS) of multimedia streams. Another important feature of an MATN is its support for sub-networks. These features of MATN make it more powerful and effective than finite state automaton. It is worth mentioning that MATN not only can be used to model the spatio-temporal relations of multimedia streams in multimedia presentations but also can support multimedia database searching when spatio-temporal relations of multimedia objects are concerned. In this paper, we focus on the spatio-temporal modeling and multimedia database searching capability of MATN models.

3.2 The proposed model

MATN is a general purpose spatio-temporal model. To fit the specific needs of transportation video modeling, the original MATN model needs to be adjusted. This is for the ease of searching and querying the transportation video database. For example, velocity and driving direction are two very important properties of a moving vehicle but they are not explicitly modeled in a MATN model. In L. Chen's CAI model, a concept called Common Appearance Interval is defined to model an interval where a certain set of objects appear in the frame together. We incorporate this concept into our model. CAIs can be automatically generated from the tracking and segmentation phase.In the proposed model, moving vehicles are explicitly modeled, which correspond to the moving objects in CAI model. A Common Appearance Interval is further broken down into sub-intervals in which the relative positions (as defined in MATN models) of vehicles remain unchanged.

In MATN models, the spatial relations of moving vehicles are recorded based on 27 three dimensional relative positions. That is, the 3-D space is evenly divided into 27 positions with one of them being the reference position. The coordinates of moving vehicles are then compared with this reference position to decide their relative positions. The details of these 27 positions are shown in Figure 2, where position #1 is the reference position. The first and the third columns indicate the relative position numbers, while the second and the fourth columns are the relative coordinates. (x_t, y_t, z_t) and (x_s, y_s, z_s) represent the X-, Y-, and Z-coordinates of the reference position and the position of a vehicle object, respectively. The '≈' symbol means the difference between two coordinates is within a threshold value. For example, the relative position number 25 indicates an object's X- and Y-coordinates (x_s and y_s) are greater than that (x_t, y_t) of the reference position, while their Z-coordinates are approximately the same. We adopt this approach in modeling the spatial relations of moving vehicle objects. In the proposed model, the center of a video frame is chosen to be the reference position and each vehicle object is mapped to a point object represented by its centroid as illustrated in Figure 1. The centroid point of each vehicle object is then used to derive the relative position of that object to the reference position. However, it is worth mentioning that we only use the 2-D relative positions in MATN models as the *z* values (or the depth information) of all vehicle objects in a 2-D video sequence are zeros. Therefore, there are only 9 relative positions in our model, which are used to record the relative positions of vehicles in a video frame at a coarse granularity. More or fewer numbers may be used to divide an image or a video frame into sub-regions to allow more fuzzy or more precise queries as necessary.

For this specific application, the driving direction of a vehicle is also recorded. However, there is no need to record this information for all video frames as this will introduce a large amount of inter-frame redundancy. Only the changes of directions between intervals are computed and recorded in the proposed model.

Before the formal definition of the proposed model is presented, we first define the terms and concepts that will be used in the rest of the paper.

Definition 1. A *Vechile Object* (*VO*) is a 3-tuple *(OID, MBR, VSFs)*. *ID* is the unique identifier of a distinct vehicle in the database. *MBR* is the *Minimal Bounding Rectangle* of the Vehicle Object. (x_{low}, y_{lo}, z_{low}) and (x_{high}, y_{high}, z_{high}) are the 3-D coordinates of the bottom right point and the upper left point of the rectangle. The coordinates of the centroid of the *MBR* is used to determine the object's relative position in 27 positions. However, since only 2-D information is available, we will use the 9 positions as shown in Figure 2 only. *VSFs* stands for *Vehicle Segment Features*, which can be used for vehicle classification and tracking purpose. In our previous work [4], a set of vehicle features based on Principle Component Analysis was proposed actually used in our vehicle classification framework to decide the vehicle type (cars, pick-up trucks, SUVs, etc.)

Definition 2. *Traffic Light Phase (TLP)* is a segment of the traffic video during which there is no traffic light change. It is denoted by a 4-tuple *(PID, PSF, PEF, META)*. *PID* is the ID of the video segment. *PSF* is the starting frame of the traffic light phase; *PEF* is the ending frame of the phase; *META* is the meta-data of this phase. It includes the allowed driving directions, the duration of the phase, etc.

Definition 3. *Traffic Video Clip* (*TVC*) is a contiguous trunk of traffic video. It consists of *TLPs* and can be denoted as a 2-tuple (*CID*, *META*). *CID* is its clip ID. *META* is the meta-data of this clip which can include such information as the time the clip is shot, the road/intersection location, camera settings, etc.

TVCs, *TLPs* and *VOs* are the constituting units of a transportation surveillance video database with *VO* being the smallest unit. Next we will give the formal definition of the proposed model and explain in detail how it can be used for transportation video database modeling.

Number	Relative Coordinates	Number	Relative Coordinates	Number	Relative Coordinates
1	$x_s \approx x_t$, $y_s \approx y_t$, $z_s \approx z_t$	10	$x_s < x_t$, $y_s \approx y_t$, $z_s \approx z_t$	19	$x_s > x_t$, $y_s \approx y_t$, $z_s \approx z_t$
2	$x_s \approx x_t$, $y_s \approx y_t$, $z_s < z_t$	11	$x_s < x_t$, $y_s \approx y_t$, $z_s < z_t$	20	$x_s > x_t$, $y_s \approx y_t$, $z_s < z_t$
3	$x_s \approx x_t$, $y_s \approx y_t$, $z_s > z_t$	2	$x_s < x_t$, $y_s \approx y_t$, $z_s > z_t$	21	$x_s > x_t$, $y_s \approx y_t$, $z_s > z_t$
4	$x_s \approx x_t$, $y_s < y_t$, $z_s \approx z_t$	13	$x_s < x_t$, $y_s < y_t$, $z_s \approx z_t$	22	$x_s > x_t$, $y_s < y_t$, $z_s \approx z_t$
5	$x_s \approx x_t$, $y_s < y_t$, $z_s < z_t$	14	$x_s < x_t$, $y_s < y_t$, $z_s < z_t$	23	$x_s > x_t$, $y_s < y_t$, $z_s < z_t$
6	$x_s \approx x_t$, $y_s < y_t$, $z_s > z_t$	15	$x_s < x_t$, $y_s < y_t$, $z_s > z_t$	24	$x_s > x_t$, $y_s < y_t$, $z_s > z_t$
7	$x_s \approx x_t$, $y_s > y_t$, $z_s \approx z_t$	16	$x_s < x_t$, $y_s > y_t$, $z_s \approx z_t$	25	$x_s > x_t$, $y_s > y_t$, $z_s \approx z_t$
8	$x_s \approx x_t$, $y_s > y_t$, $z_s < z_t$	17	$x_s < x_t$, $y_s > y_t$, $z_s < z_t$	26	$x_s > x_t$, $y_s > y_t$, $z_s < z_t$
9	$x_s \approx x_t$, $y_s > y_t$, $z_s > z_t$	18	$x_s < x_t$, $y_s > y_t$, $z_s > z_t$	27	$x_s > x_t$, $y_s > y_t$, $z_s > z_t$

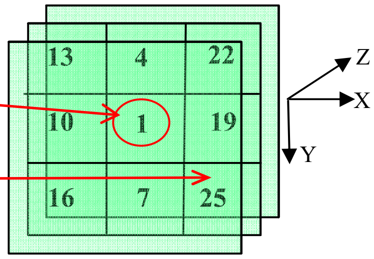

Figure 2 Three dimensional relative positions for vehicle objects.

3.3 Formal Definition of the proposed model

For convenience, we call the proposed model TVDM (Transportation Video Database Model). TVDM can be formally defined as follows.

Definition 4. A TVDM is an 8-tuple (Σ_m, Σ_s, Ω, δ, Q_m, Q_s, F, S). Σ_m,Σ_s are the alphabets of the traffic video streams that can be expressed by regular expressions. The meaning of the regular expression symbols used in this model is illustrated in Table 1.

Table 1. Meaning of Regular Expression Symbols

Symbol	Meaning
Unquoted Characters	Non-terminal symbols
'…'	Terminal symbols
=	Is defined as
[…]	Optional symbols
{…}+	One or more repetitions
{…}	Zero or more repetitions
&	Concurrent
…|…	Or
;	Rule terminator
,	Concatenation
[$c_1 - c_2$]	ASCII Characters

1. Σ_m is the Transportation Video Stream alphabet in the main network. A Transportation Video Stream in the main network is a *Common Appearance Interval (CAI)* as proposed in L. Chen's CAI model [2]. A *CAI* is an interval in which vehicle objects VO_1, VO_2, …, VO_m appear all together. A new *CAI* starts when there is a new vehicle appears in the video or an old one disappears in the video or both. Therefore, a *CAI* is a media stream that can be expressed in regular expressions,

CAI = CID, PID, CAID, {OID, ['&']};
CID = 'C', {'0'-'9'};
PID = 'P', {'0'-'9'};
CAID = 'CA', {'0'-'9'};
OID = 'O', {'0'-'9'};

where '&' means concurrent appearance of multiple vehicle objects, CID, PID, CAID, and OID stand for the ID's of *Clip*, *Phase*, *CAI* and *Object(s)*.

2. Σ_s is the Transportation Video Stream alphabet in the sub-network. A Transportation Video Stream in the sub-network corresponds to a sub-interval of a *CAI*. In the proposed model, sub-network models a *CAI* in a way that a *CAI* is further divided into sub-intervals. In each sub-interval, the relative positions of all objects in the *CAI* remain unchanged. We call such an interval CAI_{sub}. Therefore, we have

CAI_{sub} = CID, PID, CAID, CAISUBID, {OID, ('1' | '2' | …| '27'), ('N' | 'NW' | 'NE' | 'S' | 'SW' | 'SE' | 'E' | 'W'), ['&']};
CAISUBID = 'CAS', {'0'-'9'};

From the above regular expression, we can see that in each CAI_{sub}, a vehicle object is denoted by an object ID followed by a number and a symbol. The number is one of the 27 3-D relative positions as illustrated in Figure 1. (‘N’ | ‘NW’ | ‘NE’ | ‘S’ | ‘SW’ | ‘SE’ | ‘E’ | ‘W’) denotes the moving direction of that vehicle object, where N stands for north, NW stands for northwest and so forth. The driving direction can be induced from the change of relative positions between sub-intervals. This direction information is kept here for easily capturing the trajectory of a vehicle which is frequently queried in transportation video database.

3. Ω is the special input symbol alphabet. $\Omega = \{\&\&, ||, \sim, *, \alpha, \beta\}$. The meaning of these special symbols is shown in Table 2. These symbols are used in querying the transportation video database.

Table 2. Special Input Symbols

Symbol	Meaning
&&	Logical And
||	Logical Or
~	Logical Not
*	Wildcard
α	Arithmetic operators such as ‘+’, ‘-’…
β	Condition operators such as ‘<’, ‘>’, ‘= =’, ‘!=’…

4. Q_m is the set of nodes (states) in the main network. Each node in Q_m is defined as a 4-tuple *(NID, FID, OID_{in}, OID_{out})*. *NID* is the node ID. *FID* is the frame ID. This frame is the starting frame of the next *CAI* that is on the outgoing arc of this node (state). OID_{in} is the list of IDs of the vehicle objects that newly appear in the next *CAI*. OID_{out} is the list of IDs of the vehicle objects that disappear in the next *CAI*. However, these two lists cannot be both empty. If both of them are empty, there is no new *CAI* generated.

5. Q_s corresponds to a sub-network. It is defined as a 2-tuple *(NID_{sub}, FID)*. NID_{sub} is the ID of a node in the subnetwork. Each node is associated with a *FID* which is a frame ID. This frame is the starting frame of the next CAI_{sub} that is on the outgoing arc of this node (state).

6. δ is a set of the transition functions from one node (state) to another. In the proposed model, two nodes are connected by an arc that is denoted by a transportation video stream. $\delta : Q_m \times CAI \rightarrow Q_m$ or $Q_s \times CAI_{sub} \rightarrow Q_s$.

7. F is the set of final states, where $F \subset Q_m \cup Q_s$.

8. S is the set of starting states, where $S \subset Q_m \cup Q_s$.

3.4 Modeling spatio-temporal relations in transportation video database with TVDM

In this section, the above-defined TVDM is applied to model a transportation video database. Details will be explained in an example shown in a TVDM diagram (Figure 3).

In the diagram, circles are nodes/states of the network. For simplicity, only the node ID is shown in the figure. However, the identification of a node is also dependent on the traffic light phase and the video segment it is in. The network flow can be easily traced by following the arcs (arrows) in the diagram. The solid arcs are the flow in the main network while the dotted ones are in the sub-network. Each arc can be distinguished by the unique traffic media stream on it. Again for simplicity, the full name of each vehicle object in the video streams (*CAI*/CAI_{sub}) is not given in the figure. Instead, characters such as ‘A’ and ‘B’ are used as the symbols to represent the vehicle objects.

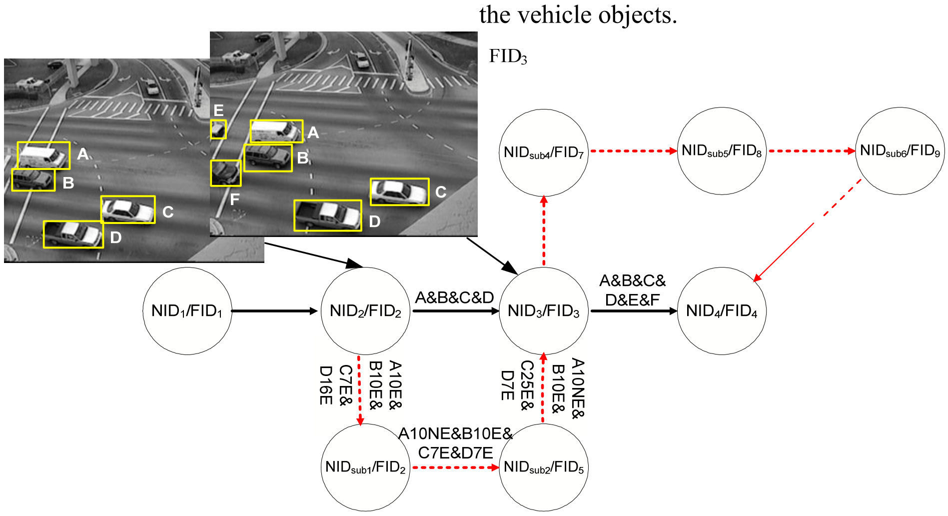

Figure 3. TVDM diagram

In this example, a traffic video clip shot at a major intersection is modeled. A video frame is divided into nine 2-D sub-regions. The entire traffic video clip is divided into phases according to the change of traffic lights. Each phase can be modeled with a network as shown in Figure 3. By connecting each such network with an arc, the network of the entire traffic video clip can be set up. There are two key frames shown in this figure. FID_2, FID_3 are the IDs of the two frames which are the turning points of two consecutive *CAIs*. Starting from the node NID_3, two new vehicle objects (E and F) move into the scene, signifying the end of the previous *CAI* and the beginning of the next *CAI*. The video stream between

NID_2 and NID_3 is denoted as 'A&B&C&D', meaning the concurrent appearance of vehicle objects A, B, C and D.

In the sub-network originating from NID_2 and ending at NID_3, the corresponding traffic video stream (A&B&C&D) is further modeled by the vehicle objects' relative positions and driving directions. Each such sub-stream is called a CAI_{sub} in which the relative positions of all vehicle objects remain unchanged. For example, the first arc of this subnetwork carries the sub-stream (A10E&B10E&C7E&D16E) in which vehicle A is in position 10 heading east; vehicle B is in position 10 heading east; vehicle C is in position 7 heading east; vehicle D is in position 16 also heading east. The arc originating from NID_{sub2} is the sub-stream in which vehicle A is still in position 10 but heading northeast; vehicle B is still in position 10 heading east; vehicle C has moved to position 25 and is still heading east; vehicle D has moved to position 7 heading east. In this traffic light phase, we can tell that vehicles are either driving west/east or northeast i.e. left turn. By storing this information in the meta-data of each traffic light phase, the vehicles driving toward illegal directions can be easily detected.

In the following sections, we will discuss the transportation video database queries with the proposed model and explore some typical scenarios that may be of users' interest in querying the transportation video database.

4. Transportatino video database queries

4.1 Multimedia Input String

As introduced in Section 3.1, the inputs of MATN models are Multimedia Input Strings (MIS) together with condition/action tables. An MIS is to simulate the input of an ATN, which is a natural language sentence. In MATN, there are two types of basic inputs. One is "Control_Command". It represents a control message that may occur during the multimedia presentation. A "Control_Command" is composed of conditions and actions. For example, a "Control_Command" can be "if the bandwidth < θ, display the compressed version of the specified media stream". "Bandwidth < θ" is the condition and "display the compressed version of the specified media stream" is the action. However, since multimedia presentation is not the focus of this paper, we concentrate on the other type of multimedia input strings which is "Media_Stream". In this paper, we designed our own input strings by following the basic idea of MATN's "Media_Stream". It has been used in Section 3 to define the traffic video stream in a CAI or CAI_{sub}.

Query strings are constructed based on Media_Streams. Query strings are used as the query input of the proposed model for querying the transportation video database. They include some special input symbols which have special meanings. The list of such symbols is in Table 2. For example, the symbol '||' means one of the conditions on the two sides has to be satisfied. "CAI.(A&B)||CAI.(A&C)" is the query string used to find CAIs containing "A&B" or "A&C". "CAI_{sub}.(A*N&B*S)" means moving vehicles A and B are in the same CAI_{sub} with A driving northward and B driving southward. "*" is the wild card symbol meaning that A and B can be in any relative positions. It is obvious that the first query string can easily find its matches (or no match) in the main network. The answer to the second query string can only be found in sub-networks of the model. However, the search shall start from the main network to find all video streams containing "A&B" and delve into the sub-network thereafter.

4.2 Object oriented transportation video database management and sample queries

In [7], a spatio-temporal model is proposed and integrated into ODBMS. Since our model also targets on modeling traffic video objects, we argue that ODBMS is suitable for transportation video databases.

In the proposed model, there are 6 classes of objects. The class names and properties are illustrated in Table 3, which is a generalization of the definitions introduced in Section 3. The first three classes are defined in Definitions 1, 2, and 3. The NODE class is defined in the formal definition of TVDM. For CAI/CAI_{sub} class, there are four important properties i.e. the starting node (NID_s) and the ending node (NID_e) in the main network, the starting nodes (NID/NID_{sub}) in the main/sub network, and the traffic video streams in main/sub network as defined in Section 3.3. The FRAME class contains the actual video frames in the definition of nodes of the network. Its properties include FID (frame ID), the Video Clip it is in i.e. CID and the actual frame in the form of an image file. A VO object corresponds to a distinct vehicle object. For example, if there is a vehicle object A, A.OID is its ID. Note that the MBR property in VO class is actually a list $<mbr_{f1}, mbr_{f2}, ..., mbr_{fn}>$ denoting the MBR property of a VO across the frames $f_1, f_2, \ldots, f_n$.

Table 3. Objects and Properties

Class	Properties
VO	OID， MBR， VSF
TLP	PID, PSF, PEF, META
TVC	CID, META
NODE	NID, OID_{in}, OID_{out}, FID
CAI/CAI_{sub}	NID_s, NID_e, NID/NID_{sub}, Media_Stream
FRAME	FID, CID, Frame

4.3 Sample queries

In order to test the effectiveness of the proposed spatio-temporal model, some typical transportation video database queries are studied in this section. In transportation video database, the user is often more interested in retrieving video data through queries on vehicle objects' properties and spatio-temporal relations among them.

Query Example 1. Given one vehicle, find all the vehicles north to it.

This is a simple query in which only the information of spatial relations between vehicles is needed. Suppose this vehicle A is in position 1 and the target vehicle is B. The user can issue this query using a high-level language which will then be translated into a Media_Stream together with some arithmetic and logic operations:

$$\mathbf{CAI_{sub}.(A1^{*}\&B4^{*})||}$$
$$\mathbf{(CAI_{sub}.(A1^{*}\&B1^{*})\&\&}$$
$$\mathbf{(B.mbr_{(CAIsub.NIDsub.FID)}.y < A.mbr_{(CAIsub.NIDsub.FID)}.y))}$$

The meaning of the above expression is that in some CAI_{sub}, vehicle object A is in position 1 and B is in position 4 OR both A and B are in position 1 with the y-coordinate of B's centroid less than that of A's. It does not matter in what directions the two vehicles are moving. In either of these two situations, B is considered north to A. In the CAI model [2], the spatial positions between each pair of vehicle objects are recorded explicitly in the database. Therefore, for Query Example 1, there would be no need to compute the relation between B.mbr.y and A.mbr.y. Our model did not adopt this approach. In both the CAI model and our model, each vehicle object's coordinates (bounding box and centroid) are already stored in MBR. Therefore, it is easy to compute the relative spatial relation between two vehicle objects. However, our model chooses to store the spatial relation between pairs of vehicle objects at a coarser granularity. That is, for each vehicle object, only its relative position to the reference position is stored. If two vehicle objects happen to fall into the same position relative to the reference position, their spatial relation is examined on the run to avoid storing too much information in the database. This can reduce the redundancy in transportation surveillance video database, since the queries on spatial relations of vehicles are not as often as on vehicles' moving trajectories.

Query Example 2. Two vehicles are meeting each other with one from northwest and another from south east.

$$\mathbf{CAI_{sub}.(A^{*}NW\&B^{*}SE)\ \&\&}$$
$$\mathbf{(Dist(A.mbr_{(CAIsub.NIDsub.FID)}.high,\ B.mbr_{(CAIsub.NIDsub.FID)}.low)<}$$
$$\mathbf{Dist(A.mbr_{(CAIsub.NIDsub.FID)}.low,\ B.mbr_{(CAIsub.NIDsub.FID)}.high))}$$

$Dist(C_1, C_2)$ is the subroutine to calculate the distance between two points whose coordinates are C_1 and C_2. A.mbr.high is the coordinates of the upper left corner of A's MBR and A.mbr.low is the coordinates of the bottom right corner. The meanings of B.mbr.low, A.mbr.low and B.mbr.high are similar.

Query Example 3. Find vehicles that are speeding.

$$\mathbf{(A.OID \in Node1.OID_{in})\&\&(Node2.OID_{out} \in A.OID)\ \&\&}$$
$$\mathbf{(dist(A.mbr_{(Node1.FID)}.centroid,\ A.mbr_{(Node2.FID)}.centroid)\ /}$$
$$\mathbf{((Node2.FID - Node1.FID)\ /\ TVC.META.FR) > \theta\ )}$$

This query can be conducted on the main network. Node1 and Node2 are two nodes in the main network where one of the incoming vehicles in Node1 is an outgoing vehicle in Node2. From the frame IDs of these two nodes, the number of frames in between can be calculated, which is divided by the frame rate (TVC.META.FR). In this way, the average velocity of the vehicle when passing the intersection can be calculated. If a vehicle's velocity is larger than θ, the allowed maximum speed, this vehicle is considered speeding. "dist(A.mbr$_{(Node1.FID)}$.centroid, A.mbr$_{(Node2.FID)}$.centroid)" is the subroutine that computes the travel distance of the vehicle in passing this intersection. "FR" in TVC.META.FR" means the frame rate (frs/sec) which is stored in the meta-data of that video clip.

Query Example 4. Find vehicles that take a U-turn.

$$\mathbf{((CAI_{sub1}.(A^{*}S)\&\&CAI_{sub2}.(A^{*}N))||}$$
$$\mathbf{(CAI_{sub1}.(A^{*}N)\&\&CAI_{sub2}.(A^{*}S))||}$$
$$\mathbf{(CAI_{sub1}.(A^{*}E)\&\&CAI_{sub2}.(A^{*}W))||}$$
$$\mathbf{(CAI_{sub1}.(A^{*}W)\&\&\ CAI_{sub2}.(A^{*}E))||}$$
$$\mathbf{(CAI_{sub1}.(A^{*}NW)\&\&\ CAI_{sub2}.(A^{*}SE))||}$$
$$\mathbf{(CAI_{sub1}.(A^{*}SE)\&\&\ CAI_{sub2}.(A^{*}NW))||}$$
$$\mathbf{(CAI_{sub1}.(A^{*}NE)\&\&\ CAI_{sub2}.(A^{*}SW))||}$$
$$\mathbf{(CAI_{sub1}.(A^{*}NE)\&\&\ CAI_{sub2}.(A^{*}SW)))\ \&\&}$$
$$\mathbf{((CAI_{sub2.NIDsub.FID} - CAI_{sub1.NIDsub.FID})/\ TVC.META.FR < \delta)}$$

In this query, the sub-networks are searched for any vehicle that drives toward opposite directions within reasonable time duration δ when passing the intersection.

Query Example 5. Find vehicles that drive toward illegal direction in the traffic light phase when only north-bound and south-bound vehicles are allowed.

$$\mathbf{\sim(CAI_{sub}.(A^{*}S)||CAI_{sub}.(A^{*}N))}$$

The south-bound and the north-bound allowed directions are information extracted from the meta-data of that traffic light phase. According to this information, we can detect vehicles driving at wrong directions by using similar queries as the above.

Query Example 6. Find vehicles that stop at some time.

$$\mathbf{CAI.(A^{*})\&\&}$$
$$\mathbf{(dist(A.mbr\ _{(CAI.NIDs.FID)}.centroid,\ A.mbr\ _{(CAI.NIDe.FID)}.centroid)}$$
$$\mathbf{/}$$
$$\mathbf{((CAI.NID_{s}.FID - CAI.NID_{e}.FID)\ /\ TVC.META.FR) == 0)}$$

A vehicle's velocity when passing the intersection cannot be zero. Otherwise, it is considered to have stopped. This can be used in identifying accidents or traffic jams. If one or more vehicles stay still for a long consecutive sequence of CAIs, there might be an accident or jam occurring in the intersection.

Query Example 7. Given a vehicle A that is driving eastward, find a vehicle B that overtakes A.

$$\mathbf{(CAI_{sub1}.(A^{*}E\&B^{*}E)\&\&}$$
$$\mathbf{(B.mbr_{(CAIsub1.NIDsub.FID)}.x<A.mbr_{(CAIsub1.NIDsub.FID)}.x)\&\&}$$
$$\mathbf{(B.mbr_{(CAIsub1.NIDsub.FID)}.y \approx A.mbr\ _{(CAIsub1.NIDsub.FID)}.y))\&\&}$$
$$\mathbf{(CAI_{sub2}.(A^{*}E\&B^{*}E)\&\&}$$
$$\mathbf{(B.mbr_{(CAIsub2.NIDsub.FID)}.x>A.mbr_{(CAIsub2.NIDsub.FID)}.x)\&\&}$$
$$\mathbf{(B.mbr_{(CAIsub2.NIDsub.FID)}.y \approx A.mbr\ _{(CAIsub2.NIDsub.FID)}.y))}$$

The condition of an overtaking is that both vehicles are driving in the same direction with one behind another. The one behind overtakes the other by using an empty lane next to the lane both vehicles are on. Therefore, initially the x-coordinate of A (the leading vehicle) is larger than that of B's. After overtaking, A's x-coordinate is smaller than that of B's. As the whole process may not finish within one CAI_{sub}, CAI_{sub1}, and CAI_{sub2} do not have to be two consecutive intervals (the same applies to all the previous queries.) However, we do need to find out the closest pair of CAI_{sub2} and CAI_{sub1}, since any CAI_{sub} after/before CAI_{sub2}/CAI_{sub1} with A and B in it may also satisfy the above conditions.

5. Advantages

The goal of this paper is not to design a master-of-all spatiotemporal database model, which is often not feasible. Instead, the proposed model in this paper is domain-specific. That is, it focuses on modeling the transportation surveillance video database. Targeting at the specific characteristics of transportation video, the proposed model can extract, index, and store the key information in the video. With these information stored in the database, transportation video data can be efficiently accessed and queried. This is one of the major advantages of our model.

The proposed model combines the strength of two general purpose spatio-temporal database models – MATN and CAI. We follow MATN's basic structure as well as its way of modeling spatial relations among objects. The key concept in L. Chen's model – CAI is also adopted in our model. This provides a way in partitioning transportation videos into "meaningful" segments and extracting the spatio-temporal information out of that. By combining the advantages of the two general-purpose models, our proposed model can better meet the needs of a transportation surveillance video database.

More specifically, motivated by this specific application, the proposed model only stores information that is frequently queried. Therefore, unlike CAI model, the proposed model does not record spatial relations between each pair of moving objects as this is not the frequent query type in the transportation video database. Furthermore, this will introduce redundancy into the database. The proposed model only records the relative spatial-relation of moving objects at a coarse granularity based on MATN model. The direction information of a moving vehicle is also recorded since this is a big concern of the user's queries. For example, the illegal driving direction and U-turn can be easily detected in our model, while it is not that easy in either the MATN or the CAI model. In order to further extract useful information from the video, CAIs are further divided into sub-intervals in which all moving vehicles' relative positions remain unchanged. We call this sub-interval CAI_{sub}. This sub-division enables us to model the video streams at a finer granularity instead of simply using CAIs.

6. Conclusion

This paper proposed a spatio-temporal multimedia database model for transportation surveillance video. The formal definition of the model provides the basis for constructing a transportation surveillance video database, which is a key in building an Intelligent Transportation System. The proposed model combines the strength of two general-purpose spatio-temporal models –MATN and CAI. The proposed model is also adjusted to suit the specific needs of this specific application. To avoid redundancy, only the frequently queried information is stored in the database. Thus the extraction of key information from the transportation video is performed in a way to facilitate queries in this specific domain. With this model design, data indexing and database queries can be performed more efficiently.

8. References

[1] Woods, W. Transition network grammars for natural language analysis. *Comm. ACM,* vol. 13, pp.591-602, Oct. 1970.

[2] Chen, L., and Özsu, M. T. Modeling of video objects in a video database. In *Proc. of IEEE International Conference on Multimedia,* Lausanne, Switzerland, August 2002, pp.217-221.

[3] Chen, S.-C., and Kashyap, R. L. A spatio-temporal semantic model for multimedia database systems and multimedia information systems. *IEEE Transactions on Knowledge and Data Engineering,* vol. 13, no. 4, pp. 607-622, July/August 2001.

[4] Zhang, C., Chen, X., and Chen, W.-B. A PCA-based vehicle classification framework. *IEEE International Workshop on Multimedia Databases and Data Management,* in conjunction with *IEEE International Conference on Data Engineering (ICDE 2006),* April 8, 2006, Atlanta, Georgia, USA.

[5] Day, Y.F., Dagtas, S., Iino, M., Khokhar, A., and Ghafoor, A. Object-oriented conceptual modeling of video data. In *Proc. of Eleventh International Conference on Data Engineering (ICDE),* March 1995, pp. 401-408.

[6] Wasfi, A.K., and Arif, G. An approach for video meta-data modeling and querying processing. In *Proc. of ACM Multimedia,* 1999, pp. 215-224.

[7] Li, J.Z., Özsu, M. T., and Szafron, D. Modeling of moving objects in a video database. In *Proc. of IEEE International Conference on Multimedia Computing and Systems,* pp. 336-343, 1997.

[8] Chen, S.-C., Shyu, M.-L. , Peeta, S., and Zhang, C. Learning-Based spatio-temporal vehicle tracking and indexing for transportation multimedia database systems", *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, no. 3, pp. 154-167, September 2003.

[9] Mouza, C., and Rigaux P. Mobility patterns. In *Proc. of 2nd Workshop on Spatio-temporal Database Management (STDBM'04),* Toronto, Canada, August 30th, 2004.

[10] Babu, S., and Widom, J. Continuous queries over data streams. In *Proc. of SIGMOD Record,* September 2001.

[11] Mokbel, M. F., Xiong, X., Hammad, M. A., and Aref, W. G. Continuous query processing of spatio-temporal data streams in PLACE. In *Proc. of Second Workshop on Spatio-temporal Database Management (STDBM'04),* Toronto, Canada, August 30th, 2004.