ON THE USE OF HIERARCHIES AND FEEDBACK FOR INTELLIGENT VIDEO QUERY SYSTEMS

Kristin Eickhorst, Peggy Agouris

Dept. of Spatial Information Engineering and
National Center for Geographic Information and Analysis
University of Maine
348 Boardman Hall
Orono, ME 04469-5711
{snoox, peggy}@spatial.maine.edu

Working Group IV/5

KEY WORDS: Querying, Video, Metadata, Hierarchy, Intelligent Feedback, Data Cubes

ABSTRACT:

Current query systems for video databases rely heavily on structured schemas and often require user annotation of data. These systems need to be made more flexible and accessible to the diverse organizations that utilize them. By building on current work with lifelines, and incorporating new structures for organizing metadata, we can make great strides towards accomplishing this goal. This paper examines current methods for working with video queries, and explores how the use of hierarchical organization can improve on these methods. We look at how data cubes can be combined with image pyramids and scale space theory to facilitate this hierarchy. Finally, metadata structures and intelligent feedback systems are discussed and their uses both before and after the query are explained in the context of our system. This system represents a new way of looking at the information contained within a video sequence and the resultant ways in which the information should be organized.

1. INTRODUCTION

As the medium of video is used with increasing frequency for both business and personal purposes, there has been a search for new and better ways to structure video databases for querying. The system that we are constructing builds on currently existing technologies, and incorporates user feedback as well. In this paper, we present our system in five parts: Section 2 provides an overview of the current video database methods in use today, and also gives a flow chart of our system for comparison. Section 3 focuses on the use of lifelines to gather basic information about objects in the video. Section 4 examines data cubes as a means of organizing the data within a video sequence. Section 5 offers a new hierarchical way of looking at metadata based on existing FGDC standards. Finally, Section 6 discusses intelligent feedback systems, and the ways in which they may be implemented to increase accuracy of video retrieval.

2. CURRENT SYSTEMS

2.1 Schema-Dependent Query Systems

The methods currently being used to perform video queries are often schema-dependent. A schema is a highly structured description of all predicates or relations in a database. Using schemas locks the user into a set pattern of looking at the data. For example, several systems classify a body of video clips as a collection. This collection is further subdivided into clips of varying length and subject matter based on the specific needs of the user and the preferences of the database administrator. One of the most common schema-dependent methods is indexing by stratification, in which each element of interest is associated with a specific time interval, and these elements are then annotated (Hacid et al, 2000). Groupings can be made of elements that exist at discrete temporal intervals, in a manner known as temporal cohesion.

The problem with traditional systems like this is that they all involve a great deal of annotation. This is not yet an automated system, so a great deal of work is required on the part of the database administrator. Another problem that comes into play with annotated systems is that the annotator's interpretation of a scene may differ greatly from that of the user. For example, when dealing with video of a parade, the annotator may be interested in the types of floats passing down the street, while the user may be more interested in a particular tree in the background. Annotated systems cannot possibly extract all the information present in a given scene, and thus are inherently limited in their resultant querying capabilities.

2.2 Object-Oriented Modelling

There is also a movement toward object-oriented modelling, which does not require the use of a schema. In this case, a sequence of video frames can be modelled as an object, with associated attributes and attribute values to describe their contents (Oomoto & Tanaka, 1993). This kind of model is closer to our proposed system, as it includes the possibility of inheritance and thus a hierarchical type of organization. Any objects located within a specific time interval are able to inherit properties unique to that interval.

While it approaches what we would like to accomplish with our system, even this type of organization has some limitations.

There is no way to specify relationships between objects within a time interval, though all may inherit properties from a parent. Many users may be interested in either the connections between two particular objects or in an overall description of the behaviors of all objects in a scene. Our system approaches this problem by examining the lifelines of specific objects within the video and extracting attributes that help with these comparisons.

2.3 Current System

Our system consists of several interrelated parts, which can best be explained with the aid of a flow chart:

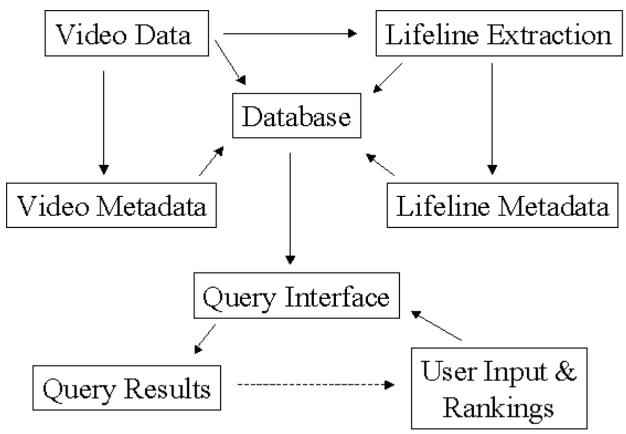

Figure 1. Components of our system

The system's input is video data. Lifelines of objects within the video are extracted in order to obtain additional information about the video contents. These two data sources (video and lifelines) have metadata associated with them, either through annotation by collectors of the video data, or through extraction and computational methods. The video and lifelines along with their respective metadata are all stored in a master database.

This database has a query interface, by which users can express their preferences for metadata values and can also give each category of metadata a subjective ranking. These inputs are used to determine the most likely video clips for a specific user's needs. Once the results have been viewed, the user has the option of updating either metadata input or rankings. This could also be done in an automated manner by the computer.

The remainder of this paper discusses each of the components of our flow chart as follows: Section 3 is concerned with lifeline extraction. Section 4 discusses the ways in which data about these lifelines can best be stored in the database, utilizing data cubes and other hierarchical methods. Section 5 is concerned with metadata and its storage within the database. Section 6 concludes with a look at the query interface and potential feedback mechanisms that can be utilized as well.

3. LIFELINES

3.1 Lifelines

A lifeline can be broadly defined as a sequence of the spatial locations (x,y,z), of an object over a time interval (t_1,t_2), during which an object has moved from one location to another. Lifelines have properties that lend themselves well to defining relationships between objects. During video analysis, extracted lifelines are defined by their nodes, which denote changes in the attributes of cardinality and acceleration of any given object. Groups of lifelines can be compared in terms of either their geometries or their attributes (Stefanidis et al, 2001). When these groups are compared, additional attributes such as topology or average separation can be computed and used as a basis for making decisions about the relationships between objects. Analyzing these lifelines provides an abstraction mechanism for summaries of video content.

One way this can be accomplished is by development of tuples that describe the contents of our database in terms of lifelines and their attributes. Examples of such tuples include:

- Group (grp_id, topology, lifeline_1… lifeline_n)
- Lifeline (ln_id, geometry, acceleration, cardinality)
- Geometry (spatial, temporal)
- Spatial (node_position_1… node_position_n)
- Temporal (node_position_1… node_position_n)

where the user determines n in all instances.

While many tuples could be formulated with our system, those above have been selected as most appropriate for our particular applications. We are certainly interested in the geometry that defines each particular lifeline, and attributes of acceleration and cardinality help describe where the object has changed its movement trends. Including topology when grouping lifelines provides a means of relating each lifeline to others around it. All of these attributes are used in our system. While it is instructive to look at each of these tuples on their own, it is also important to realize that they can be organized in a hierarchical manner. This hierarchy can be expressed in terms of data cubes or their extensions, which will be discussed in Section 4 of this paper.

By breaking the geometry down into spatial and temporal components, we allow the user some freedom in choosing to emphasize one of these dimensions over the other. Of course, it is also possible to look at a combination of spatial and temporal information, which most closely represents the overall content of the video sequence in question.

In looking at spatial coordinates (x and y dimensions), we are primarily interested in the movements of a given object in the video space and the corresponding attribute of cardinality, which changes at each of the nodes. When examining temporal coordinates (z dimension), we are more interested in when the object changes its velocity than in its specific path. In this case, the attribute of acceleration becomes important. An example of the use of lifelines in our system may be instructive.

3.2 Lifeline Example

Topology and other group-oriented attributes can be examined when lifelines are formed into functional groups based on the user's needs. For instance, if a video camera is set up on a tall building to monitor the traffic flow on surrounding roadways, it may be of interest to determine the locations of bottlenecks before the traffic backs up too much.

If each car in the scene is picked out as a separate object and lifelines are constructed in real time, cars on a given road can be grouped together and their topology or average separation can be computed. A group of cars with a low average separation may represent a potential bottleneck, in which case a navigation system could suggest alternate routes for motorists, or a change can be triggered in the timing of traffic lights or lane allocation.

Once all the key entities and their attributes have been extracted from a video sequence of interest, we must turn our attention to how they can be organized to best facilitate queries. A current method that has gotten much attention is the data cube, a tool often used for decision-making processes in businesses. We feel that it can also help us organize our video data.

4. DATA CUBES

4.1 Data Cubes

Data cubes work on relational databases to provide a means of facilitating query performance. These cubes are composed of dimensions, representing categories of data. Within data cubes, dimensions are organized into hierarchies, or levels. The data values to be analyzed are known as the measures of the cubes, and these can be analyzed through queries involving combinations of dimensions and hierarchies (Harinarayan et al, 1996).

To further our query system, we are concerned with the construction of such cubes. The tuples set forth in the description of lifeline abstraction in Section 3.1 will be used as a working example and can be easily adapted for this task. In formulating data cubes, one uses fact tables, which contain data about the topic of interest. Following the traffic example given previously, we will concentrate on group behaviors as our topic of interest. Thus, our fact tables would consist of multiple tuples of the type:

- Group (grp_id, topology, lifeline_1… lifeline_n)

In order to populate our database with these tuples, we would specify a roadway and collect all lifelines with nodes falling within the boundaries of the road. These lifelines would then compose one group, and topology could also be computed.

Once all desired groups of lifelines have been extracted and added to the fact table, we can then look to a data warehouse, where dimension tables are stored. Dimension tables contain information related to the components of the fact table. In our system, the following would be stored in dimension tables:

- Lifeline (ln_id, geometry, acceleration, cardinality)

The query system would refer to a list of lifeline tuples in order to obtain a full picture of all the lifelines within a given group. Data within these dimension tables can easily represent dimension hierarchies, when there are functional dependencies among the attributes of the dimension tables (Mumick, 1997). In our example, ln_id functionally determines geometry, acceleration, and cardinality. Geometry can then be broken down further into its spatial and temporal components. The measures of the cube would be the attributes of acceleration, cardinality, and topology, as well as geometry (node positions).

With the components of the cube in place, a set of likely queries can be defined, such as a list of the nodes at which any object within a group is accelerating. With a small enough set of data, it may be possible to catch all of the most requested queries. If the system is extended to encompass locations of stationary objects within the scene though, the data cube may become too complicated to anticipate all possible user requests.

4.2 Supplements for Cubes

The major problem with the above cubes is that queries must be predefined. We propose a combination of data cubes with other existing mechanisms of data organization, such as pyramids and scale space, in order to allow integration of zooming behaviors and to avoid the need for predefined queries. This also allows us to extend their use to data that is not stored in traditional relational databases.

Pyramids have been used to establish discrete levels of zooming in many digital image processing applications. To improve data cubes we are using the same concepts in our lifelines and the extraction of their attributes. More specifically, the tuples listed in Section 3.1 contain some variables that can be set by the user, namely the number of lifelines in a group and the number of nodes defining a particular lifeline. Choosing just a few nodes or lifelines is equivalent to choosing a low level of detail. The higher the value for n, the more detail will be present in the resultant data set. In this manner, a rudimentary pyramid can be formed and integrated into our query system.

Scale space is a concept by which an image or other object of interest can be examined at any point on a continuum of scale, not just at discrete stops. This allows for zooming to exactly the desired level of detail within an image (Lindeberg, 1994). One of the goals of our research is to incorporate scale space into our query system and even bypass our relational tables. In order to accomplish this, we would need to look not only at the nodes that define lifelines, but also at the coordinates of the lines themselves between these nodes. Future research will focus more intensely on this aspect of our query system.

Extracting lifeline geometry and attributes and organizing the resultant data are the first steps in our video query system. However, it is not wholly effective on its own. The user of any such system will inevitably be interested not only in these descriptors of the dataset, but also in metadata about the quality of the observations being made and the source of the data. The next major component in our system builds on traditional metadata schemes in order to address these concerns and to begin the querying process.

5. HIERARCHICAL METADATA

5.1 Content Summaries

We realize that it is often impractical, and sometimes impossible, to list the entire contents of any data set. Thus content summaries are used to get an idea of which data set will be most appropriate for a given application. These summaries list the most important pieces of information about each data set in question, so that quick decisions can be made about their relative usefulness (Hardy & Schwartz, 1996).

In order to facilitate content summaries and their querying, we propose the construction of new metadata structures. Each level of the video hierarchy discussed in the previous sections needs its own metadata components, because we are able to view successively more detail as we move from groups to lifelines to individual nodes. It would not make sense to examine metadata about specific nodes when the user is only interested in an overall description of the data set.

The Federal Geographic Data Committee's (FGDC) metadata standards are already structured in a somewhat hierarchical manner, with the following categories on their topmost level:

- Identification
- Data Quality
- Spatial Data Organization
- Spatial Reference
- Entity and Attribute
- Distribution
- Metadata Reference

Each of these categories of information has been subdivided further in order to fully elucidate the metadata for a specific data set (FGDC, 1998). The data that we extract from our lifeline abstraction process about geometry, acceleration, cardinality, and topology is most closely linked to the "Entity and Attribute" category, and can be added as an additional piece of metadata for querying purposes.

5.2 Additions to FGDC Metadata

If someone wanted a broad overview of the contents of the dataset, FGDC metadata on topics such as distribution or metadata reference would be appropriate. Once the user decides that the data set may be of some interest and starts zooming in to look at specific entities though, information about these entities and their attributes and about data quality may become more important.

For example, suppose a bank robbery had taken place, and the user wanted to track all cars that accelerated and turned toward the highway within a five-minute window after the crime is committed. In this case, the most important metadata would deal with entities and attributes on the FGDC level. It would also be helpful to know how accurate we could expect the results to be. Both temporal and geometric accuracy would need to be scrutinized at the level of specific nodes. This accuracy may differ from that of the groups or lifelines, due to error propagation (Clarke, 1998). Thus it is important to specify a level of interest, either through data cubes or one of the alternative zooming methods presented in Section 4.

Our system is designed to ask the user which level of the hierarchy is most important. Then the appropriate metadata categories could be presented for input and analysis through feedback systems, such as those presented in Section 6.

6. FEEDBACK SYSTEMS

6.1 Basic Operation

The last major component of our current work is the use of intelligent feedback systems, whereby a user could express preferences for such qualities as coarseness of spatial resolution, frequency of event occurrence, and overall accuracy of data. These preferences would be gathered before a query is processed and would be used in mapping a path through the hierarchical metadata for use in processing.

In our system, the user is presented with a list of descriptions for each of the seven FGDC metadata categories, including a sample of the subdivisions within each category. The main categories can then each be given weights between zero and one, with one meaning that only exact matches between the requested metadata and that found in the metadata file would be deemed acceptable. A weight of zero would mean that any response in the metadata file would be acceptable.

Any of the seven categories given a ranking above zero is then subdivided into the next level of the hierarchy, and these could be weighted as well. For instance, if the user were interested in the results of the Data Quality category, these levels would be presented:

- Attribute Accuracy
- Logical Consistency Report
- Completeness Report
- Positional Accuracy
- Lineage

The user can then assign weights to these subdivisions. This continues until either all elements on the lowest level are given weights of zero, or no further subdivisions can be made.

One of the biggest challenges of the system would be in determining whether a match had actually been made or not. In this case, algorithms would need to be developed to compare similarity of responses (Sharma 1997). It is relatively simple to compare two numerical responses, and geometric responses can be mapped out and compared by computing distances. It is much more complicated if the category being addressed is answerable only with text however. This area of our system is still under development.

Once similarity measures had been determined, they would then be multiplied by the weights given by the user such that:

$$S = w_1 s_1 + w_2 s_2 + \ldots + w_m s_n \qquad (1)$$

where S = total similarity
w_i = weight assigned by user
s_i = similarity measure for given metadata

These weights are based on the importance that each user gives to the different categories of metadata, and are used to select a group of video clips that most closely match the desired criteria (Jain, 1994). While an expert user may want to utilize the system fully by choosing all possible weights, the non-expert may desire a more automated system.

Consider a database administrator at a police station, who is in charge of collecting surveillance tapes at local cultural and sporting events, to be used in searching for suspicious activities. This individual would be interested in nearly every aspect of the available metadata, in order to be sure that the source, accuracy, and even the spatial reference data was such that the videos could be utilized by the members of the police department. In this case, all possible metadata categories would be available for ranking.

Individual police officers, on the other hand, would be much more interested in the entity and attribute information that could be extracted from the tapes. They would probably not care much about where the tapes originated or even about the accuracy of the data to be extracted. If the video is available for them to view, it is because the database administrator has deemed it acceptable in these categories. This leaves the individual officers free to concentrate on matching very specific attribute information instead. Defaults could be accepted for the other categories, and only those metadata categories that are of interest at the time would be ranked.

Once all the rankings had been completed and the similarity indices had been computed, the user would be presented with a list of the video clips that most closely fit the given criteria.

6.2 Advanced Systems

After the query results had been presented to the user, additional adjustments could be made to the preferences via user feedback. The users would be allowed to rank the clips that were presented as possible solutions. These rankings would then be used to gauge the criteria being used to make decisions about what metadata are important.

For example, if an individual has given data quality a very high ranking, but then chooses clips near the bottom of the ranked list as being more desirable than those near the top, perhaps lower values for data quality should have been chosen. Internal adjustments could be made to the weights for each category and new sets of possible solutions could be constructed.

This feedback system would allow the user to obtain the best possible results from the query process. Additional features that could prove useful to users include a default set of weights for given applications, as well as storage of preferences for future queries by the same user (Keogh & Pazzani, 1999). In either of these cases, the weights would be handled behind the scenes, unless the user specifically wanted to modify them.

Going back to our original example of a camera mounted on a tall building, information about the data quality, distribution, or metadata reference would probably not be as important as information about entities and attributes within the video sequences. We could assume that the user of this system would already know who had collected the data, and would be more interested in the attributes, so as to determine where a potential bottleneck might be found.

Anyone who uses this system often enough for the same purposes could store profiles with associated weights, so that they would only need to log in, choose their task, and give their preferences for metadata results. The system would do the rest.

7. CONCLUSIONS

The combination of feedback systems with modified data organization tools and new hierarchical structures for metadata represents a powerful new environment for querying video databases. While there is still work remaining to be done both on the underlying theories and their implementation, we believe that the proposed system has the potential to make video querying faster and more effective.

REFERENCES

Clarke, J., 1998. Modelling uncertainty: a primer. Technical Report 2161/98, University of Oxford, Dept. Engineering Science.

Federal Geographic Data Committee (FGDC), 1998. FGDC-STD-001-1998 “Content Standard for Digital Geospatial Metadata (Revised June 1998)”, Washington DC. http://www.fgdc.gov/standards/documents/standards/metadata/v2_0698.pdf.

Hacid, M., C. Decleir, J. Kouloumdjian, 2000. A database approach for modelling and querying video data. *IEEE Transactions on Knowledge and Data Engineering*, 12(5), pp. 729-750.

Hardy, D., M. Schwartz, 1996. Customized information extraction as a basis for resource discovery. *ACM Transactions on Computer Systems*, 14(2), pp. 205-216.

Harinarayan, V., A. Rajaraman, J. Ullman, 1996. Implementing data cubes efficiently. *Proceedings of the ACM SIGMOD Int. Conference on Mgt. of Data 1996,* pp. 205-216.

Jain, R., A. Hampapur, 1994. Metadata in video databases. *ACM SIGMOD Record.* 23(4), pp. 27-33.

Keough, E. M. Pazzani., 1999. Relevance feedback retrieval of time series data. *Proceedings of the 22nd Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval.*

Lindeberg, T., 1994. Scale-space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2), pp. 225-270.

Mumick, I., D. Quass, B. Mumick, 1997. Maintenance of data cubes and summary tables in a warehouse. *Proceedings of the ACM SIGMOD Int. Conference on Mgt. of Data 1997,* pp. 100-111.

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

Sharma, V., 1997. Organization and visualization of metadata for spatial information in a digital library. Master’s Thesis, University of Maine, Orono, ME.

Stefanidis, A., P. Partsinevelos, K. Eickhorst, P. Agouris, 2001. Spatiotemporal lifelines in support of video queries. *Proceedings Twelfth International Workshop on DEXA,* pp. 865-869.

ACKNOWLEDGEMENTS

This work is supported by the National Science Foundation through grants DGI-9983445 and ITR-0121269.