Hierarchical Structures for Video Query Systems

Kristin Eickhorst
Department of Spatial Information Engineering
University of Maine
348 Boardman Hall
Orono, ME 04469-5711, USA
snoox@spatial.maine.edu
http://www.spatial.maine.edu/~snoox

Abstract

Video query systems have been used increasingly for both business and personal applications. Many applications for video data involve stationary cameras, resulting in a stable background and moving objects in the foreground. The movements of these objects can be extracted to form lifelines using techniques such as those developed in our lab. Our current task is to organize these lifelines and their attributes in a way that will make them easy to query, even by inexperienced users. In order to accomplish this, we have employed data cubes and other hierarchical measures, as well as new metadata structures. After a brief review of our ongoing work with lifelines, we will discuss these additional components of our query system in more depth. Our comprehensive system has the potential to change the way in which video databases are organized and queried.

1. Extracting Video Lifeline Data

Work done by members of our research group has dealt with the methods for extracting lifelines from video data sets (Stefanidis et al, 2001a,b). A lifeline can be 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. They are useful because they represent several attributes of an object's spatiotemporal progression, such as acceleration and cardinality, that can then be incorporated into a master database for use in future queries. Lifelines can also be aggregated into groups, which have their own properties, such as topology. Users that require a high level of detail would desire many nodes and lifelines when defining lifelines and groups respectively, while those more concerned with a general overview of the video contents might ask for fewer nodes and lifelines in order to save space and processing time.

Self-Organizing Maps (SOMs), a type of neural network, can handle the computations that must take place in order for an object's movements to be generalized, as is shown in work done in our lab (Stefanidis et al., 2001c). SOMs are automated, so that the first nodes that are placed give a rough outline of the lifeline of interest, and additional nodes are clustered near these vertices in order to help to bring out the details. Our current work builds on this base, and takes it a step further by allowing the user some input in exactly how many nodes will be used to carry out the delineations of lifelines. This introduces the possibility of specifying a level of detail that will work best for any given application.

Once the nodes have been extracted from the objects in question through techniques like Self-Organizing Maps, the next challenge is in how to organize them for storage in our database. We have addressed this issue by looking into data cubes as well as other hierarchical structures, such as pyramids and scale space. The following section elaborates on these options, and shows how the user's ability to choose the number of nodes defining a lifeline comes into play. Metadata about both the video contents and specific lifelines are also stored in the database, and we have developed some new metadata structures in order to best make this aspect of the database available for querying. Metadata issues are presented in Section 3.

2. Data Cubes and Other Hierarchical Organizers

Once all the geometries of the lifelines and their corresponding attributes have been gathered, we shift our focus to how we can best organize the input into the database. Current video database systems often use restrictive schemas in order to accomplish this. We prefer to use more flexible systems that allow some user input as well. Currently, we have developed a system where the components of our lifelines can be easily extracted for storage in data cubes and other hierarchical structures. The data cubes are constructed such that querying for a group and its behavior from a fact table would also pull up information about its component lifelines, their respective nodes, and the attributes associated with them. These could be accessed as needed to answer key questions about the contents of the video sequence.

While data cubes can be very useful if the user already has a set of pre-defined questions to answer and does not anticipate wanting to grow beyond these questions, they do present some problems. The main problem with using a typical data cube setup is that all queries must be predefined. If we wanted to make any advanced queries for which we hadn't set up structures already, we would be unsuccessful. This could happen if we wanted to increase our scale to something beyond groups, or to examine the contents of more than one video clip at a time.

An example application for our system is traffic monitoring by municipalities. This application can be used to illustrate many of the different problems encountered in existing video query systems, and the ways in which we propose to remedy them. For example, if we are dealing with an application that monitors traffic flow through the streets of a busy downtown area, we will be primarily concerned with the level of detail that will allow us to best capture these streets and their associated events. If the scope of the project then expands to include the entire metropolitan area, data cubes would not be able to expand along with them in order to consider more generalized views. These limitations have led us to consider pyramids and scale space as alternatives that are more flexible but still have a hierarchical structure.

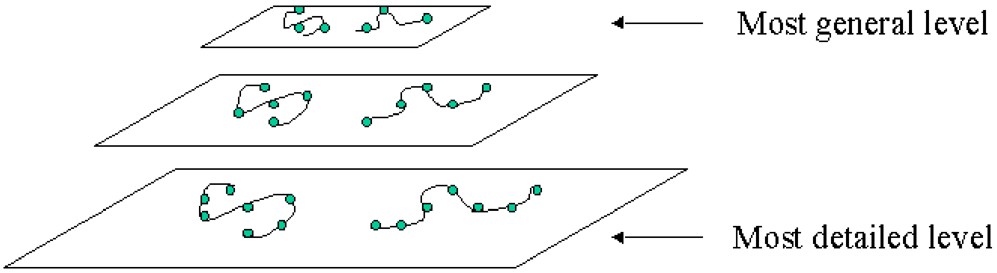

Figure 1: Pyramid representations of the same lifeline

Pyramids are basically discretized versions of scale space mechanisms. In a pyramid set up, there are pre-defined levels of the hierarchy that can be accessed (Chen et al, 2000). These would correspond to choosing different values for n, where n is the number of nodes that define a lifeline, or the number of lifelines that define a group. Pyramids would have some already-defined values for n, so that the broadest level of the hierarchy might only use three nodes to define a lifeline (perhaps the beginning and end of the lifeline as well as one key node between these two). A more detailed level might have ten nodes, or a proportional number for the temporal length of the video sequence (Figure 1).

In contrast to the pyramid structure, a scale space method would be more continuous in the number of nodes that could be used to define a lifeline (Ogniewicz, 1994). While the pyramid scheme may allow the user to choose only multiples of 10 for n (or some other number suggested by SOM principles) when looking beyond the coarsest level of detail, the scale space method would allow the use of any integer value for n. This would give the user much more flexibility in the choice of exactly how much detail to extract from any given video sequence, and in the case of our traffic monitoring applications, it would allow us to zoom to whatever area of coverage is desired for a given situation. For that reason, we have chosen to implement scale space methods in our query system whenever possible.

3. A New Metadata Structure

Now that we have the data from our videos and the lifeline extraction process stored in hierarchical structures within our database, we turn to their associated metadata. This metadata can also be stored in the same database in a hierarchical fashion. We can easily take advantage of the current structuring of the Federal Geographic Data Committee (FGDC)'s suggested metadata standards (FGDC, 1998). In this structuring, there are seven main categories of metadata, each of which branches into several subcategories, which in turn have their own subcategories (Figure 2).

1. Identification

2. Data Quality Information
    - 2.1 Attribute Accuracy
    - 2.2 Logical Consistency Report
    - 2.3 Completeness Report
    - 2.4 Positional Accuracy
        - 2.4.1 Horizontal Positional Accuracy
        - 2.4.2 Vertical Positional Accuracy

    - 2.5 Lineage
    - 2.6 Cloud Cover

3. Spatial Data Organization
4. Spatial Reference Information
5. Entity and Attribute
6. Distribution
7. Metadata Reference Information

Figure 2: Sample FGDC Metadata Standards

The metadata that we have extracted from our lifelines and their nodes fits most closely category five in this scheme: Entity and Attribute. As demonstrated by category 2.6, the FGDC Standards do leave room for application-specific metadata to be added into their hierarchy (Galhardas et al, 1998). For example, in an oceanographic application, cloud cover might be a piece of metadata that would be useful to know when assessing data quality of a satellite image. We could very easily add such subcategories to Entity and Attribute, assessing the positional accuracy of the lifelines we extracted, as well as any metadata we would like to include on the algorithms used to perform the lifeline extractions.

The benefits of hierarchical metadata used in conjunction with hierarchical data structures is that more specific levels of metadata can be presented as the user zooms into the video sequence of interest to glean more data as well. When the user is just browsing around to find a clip that might be appropriate for use in a given application, broad-level metadata to be found in the first level of our hierarchy would suffice. For instance, if looking to see whether a specific entity (e.g. represented by a lifeline) is present in the clip, the first level of the metadata hierarchy would be adequate. This might occur when a user is interested in the path followed by a given car in the video database.

If this entity is present, and the user zooms in for a closer look, the first subcategories of metadata would be activated too. When the user sees the entity of interest, questions may arise about how it was extracted. This information would be present in the current level of metadata. In the case of the traffic monitoring application, this information might include the type or position of the camera that obtained the video footage or the person operating this camera. If satisfied by the veracity of the source, the user may zoom in once more. In this case, accuracy of that specific entity and any associated attributes would be available as metadata as well. For our traffic case, this would mean positional accuracy of the car at any given moment or the overall accuracy of the extracted lifeline (Figure 3).

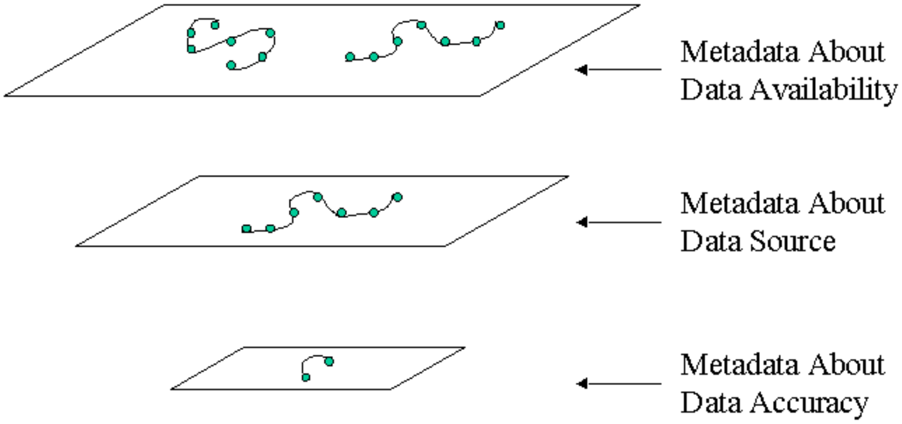
Metadata About Data Availability
Metadata About Data Source
Metadata About Data Accuracy

Figure 3: Zooming Behaviors and Associated Metadata

In the figure above, a user may first be looking at the overall video clip in order to determine whether any cars rounded a particular bend in the road during a given time interval. In this case, not only would all lifelines be displayed, but metadata about data availability would also be accessible. This would allow the user to utilize both visual and textual cues in order to determine whether to proceed to more detailed levels of the hierarchy. If such a car is found in the video, then its lifeline can be isolated as in the second level of Figure 3. Metadata about the data source would also be associated with this level, so that the user can decide whether the data is reliable enough to zoom in further. In the last level of the above figure, only a segment of the lifeline is presented, and metadata about data accuracy is also accessible. This might be useful if the user was interested in whether or not the car drifted into the wrong lane as it rounded the bend. In this case, accurate measurements would be of great importance.

This setup is useful as a default system for the inexperienced user or browser, who becomes more concerned about detailed metadata as increasingly detailed data sets are viewed. Of course, an experienced user who knows exactly what is desired to solve an application problem would be able to query even the most deeply embedded metadata without needing to zoom all the way down to the level of an individual node. Thus, the hierarchical metadata of our system can work either in tandem with the hierarchies of data in our data cubes, or as an independent component for querying purposes.

4. Conclusions

This paper is just a short sampling of the work being done in our lab on lifelines as a means for extracting information from videos, as well as the methods by which we can organize this information and its corresponding metadata within a database. Other work that is currently ongoing in our lab takes this a step further, and deals with the process by which users may query the database and give feedback on the results (Eickhorst & Agouris, 2002). Our goal and overall vision is to create a comprehensive environment whereby users can indicate preference for a level of detail to be extracted, and then be able to query the resulting database of videos, lifelines, and metadata for the most likely video sequences.

Acknowledgements

This work is supported by NSF through Digital Government Award #9983445.

References

Chen, J., C. Bouman, J. Dalton, 2000. Hierarchical browsing and search of large image databases. *IEEE Transactions on Image Processing,* 9(3), pp. 442-455.

Eickhorst, K., P. Agouris, 2002. On the use of hierarchies and feedback for intelligent video query systems. *Proceedings of ISPRS 2002 Symposium of Commission IV* (in press).

Federal Geographic Data Committee (FGDC), 1998. FGDC-STD-001-1998 "Content Standard for Digital Geospatial Metadata (Revised June 1998)", Washington DC.

Galhardas, H., E. Simon, A. Tomasic, 1998. A framework for classifying scientific metadata. *AAAI'98 Workshop on AI and Information Integration,* Madison, Wisconsin.

Ogniewicz, R., 1994. Skeleton space: A multiscale shape description combining region and boundary information. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 746-751.

Stefanidis, A., P. Partsinevelos, P. Agouris, 2001a. Using lifelines for spatiotemporal summaries. *DG.O 2001 Proceedings, National Conference on Digital Government Research,* Los Angeles, CA.

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

Stefanidis, A., P. Partsinevelos, P. Agouris, 2001c. Automated spatiotemporal scaling for video generalization. *IEEE International Conference on Image Processing (ICIP) 2001*, Vol. 1, pp. 177-180.