

STACK DATABASE MODEL/VIEW OF MULTIMEDIA DATA

Alfonso F. Cardenas, Panayiotis Michael and Bassam Islam
Computer Science Department, University of California at Los Angeles, California, USA

Abstract: **Growing amounts of multimedia data (alphanumeric, image, sound and video) are being captured with the increasing deployment of sensors. We present the image stack model or view of data as a convenient way to not only visualize such variety of data but also as a basis for querying it, even if the data is not stored as such. We outline briefly the requirements with example queries, and the recommended extensions to the latest object database management developments particularly the ODMG standard.**

Keywords- multimedia data, streams, querying, stack model/view, data warehousing, ODMG extensions

1. Introduction

We present the image stack database model or view of selected multimedia data at some point in time from a variety of sources. An increasing multitude of sensors is being deployed to capture growing amounts of multimedia data (alphanumeric, image, sound and video) [1]. The image stack is a convenient way to not only visualize such variety of data but also as a basis for querying it, even if the data is not stored as such. We outline the requirements and example queries, and the needed extensions to the latest object database management developments particularly in the ODMG standard to support the image stack model/view. This parallels current thrusts into data warehousing and OLAP (On Line Analytic Processing) in conventional alphanumeric-only data management and viewing.

2. Multimedia Data Streams and Databases

A *stream* is an ordered sequence of frames or values, Figure 1 top left. A frame could be an image or photograph, or a frame in a video stream, or a text report, or even an alphanumeric record changing through time. Conventional current DBMS (relational or newer object DBMS) deal

well only with alphanumeric record type structures once they are stored and loaded into a database. Unfortunately, it is impractical to store in a DBMS the voluminous data that is coming in increasingly rapid rates from an increasing number of sensors generating such streams; furthermore, there are no DBMS facilities to search for and select specific elements coming rapidly in a stream of alphanumeric data. We and others are currently researching this challenge and the broader challenge of being able to pick specific elements based on content of the element (e.g., select the images with red objects, select images with tumor sizes larger than 10 cm (a very complex operation involving image segmentation of tumors that is not readily possible today automatically in real nor even near real time basis!) or on content of other elements in other streams (e.g., select photographic image slices corresponding in time to the largest peak value of the seismic stream between time A and B).

We have proposed and implemented in part the M Model [2] which considers the stream as a fundamental construct in addition to all the constructs of today's DBMS, and supports a stored repository of multimedia data streams which is then accessible via the MQuery language [3]. Selected subsets of data streams are feasible to store in a database.

3. Dynamic Image Stack Model/View and Example

We envision and propose complementing the above stream data management facility and conventional DBMS that handle well alphanumeric data and associated operations with the *logical dynamic image stack model/view*. The stack data model is a very attractive way to not only visualize and present multiple types of multimedia data but also to set up as a local or user view database to directly support major types of multimedia queries that we

illustrate subsequently herein. Figure 1 shows an example of the image stack which consists of several planes (a stack can have an arbitrarily large number of planes), each plane with a different type of two dimensionally encoded data, and co-registered to the same coordinate system. This stack would be composed of elements at a point in time from different data streams and other sources. The example in Figure 1 shows several data sources from which data at one point in time could be gathered logically or physically and viewed as planes in the image stack, each plane with related alphanumerically recorded data (not geographically encoded):

1. Static topological maps which are digitized and segmented to identify the geographical location of streets and roads and city boundaries (one plane), of school (including university) districts (one plane), and of contour elevation lines showing peaks and valleys (another plane in the stack). Each of the planes could be complemented by its associated alphanumeric data that is not geographically encoded; for example, for school districts plane, the related school name, student population, ethnic distribution percentages, ...)
2. Relational or object DBMS tables with alphanumeric information that is converted to two-dimensional representation. For example, smog numeric data or biochemical or poison fumes data from a number of sensors could be stored in a DBMS and then its geographic distribution presented as a two-dimensional plane of the stack with the grid cell values filled in as a result of interpolation of data between sensors (if this distribution was needed only for one-shot presentation, then there would be no need to permanently store it physically as a plane in the stack).
3. Digitized aerial/satellite photos, with a plane for each type of data, for example, smog levels, fog levels, and reflectivity/rain levels.

Note that not all data in a stack has to be stored as such, but only appear as such, e.g., there could be a pointer from a plane to point to where the actual image is physically stored to avoid replication. Issues of image co-registration and differing grid

sizes (resolution) may arise due to the variety of sensor sources but are not addressed herein.

4. ODMG Database Structure and Extensions Needed for Image Stack Viewing and Processing

The latest database management system architecture ODMG [4] proposed by the Object Database Management Group provides data modeling and querying facilities not in standard SQL (predominating in today's commercial DBMS world) that are needed towards supporting the dynamic image stack model. These include the definition of lists, sets and bags (collectively referred to as collections) to complement record structures or classes and arrays, and the great ability to intermix these and define: (1) arrays of data structures or of arrays or of these collections, or (2) a collection of data structures or of arrays or of another collection, and recursively.

The image stack illustrated in Figure 1 may be supported by the following database structure whose merits will be more evident through the wide range of example queries below supported by it. This requires extending the ODMG model to handle two and three-dimensional arrays and the corresponding Object Query Language (OQL) commands. The *ith* plane in the stack *S* is denoted by $S(*, *, ith)$.

Figure 2a shows the database diagram representing roads, city boundaries and school boundaries appearing in a topological image so as to support answering point, line and polygon queries (like queries 1 and 2 below); this would be stored in addition to the original 2-dimensional image.

Figure 2b shows the skeleton of the ODMG schema definition for a dynamic image stack with the extensions to the ODMG standard needed to support it. Each array cell is a class (or record structure) composed of a pixel value and a List_X of classes:

```

Class Cell
  Attribute Cell_value /*pixel value or
  gray level of an image*/
  Attribute List_of_X
  
```

Where the list List_of_X is composed of instances of the class X with attributes X_Name and List_of_X_attributes; where the List_of_X_attributes is composed of instances of the class X_attribute with attributes Attribute_name and Attribute_value.

Note that this structure has the ability to optionally store with the pixel value of any image in the stack a list of class or record structures and their attributes along with the actual value for each attribute. Thus, it is possible to store any kind of alphanumeric data particular even to the pixel level, although in most cases such data applies to larger objects composed of many pixels (e.g., a tumor in an X-ray or MRI image).

5. Sample Queries

Below are several sample queries in English for the image stack above and sample queries that we are addressing in our research. We include the hypothetical extended OQL query incarnation for two queries. The extensions are significant, but are possible to implement in an extended ODMG DBMS. The results of the queries would be sent to a drawing or image system to convert to display form. Our focus in the article is in the dynamic image stack model and some highlights of its database structure model support and query language support needed for the types of queries illustrated below; the focus is not on internal query processing, traditional image processing, or pictorial/image presentation interfaces – beyond space limits herein.

Query 1 - Retrieve the location of the intersections of the UCLA boundaries with Westwood Blvd and Sunset Blvd. where the poison fume level exceeds value Y now.

This uses a polygon and line intersection function on x, y coordinate data retrieved from the alphanumeric database section, and then checks that the involved poison fume cell values exceed Y for the latest available poison fume profile.

Extended OQL query:
 SELECT *
 FROM Road AS R, School_Boundaries AS SB,
 Poison_Levels AS PL

WHERE SB.Name = "UCLA" AND R.Name = "Westwood Blvd" OR R.Name = "Sunset Blvd"
 AND POLYGON_LINE_INTERSECT (SB, R) AND
 PL.Cell_Value > Y
 /* Poison_Levels plane PL is an image in the stack. PL.Cell_Value checks for the poison level at each grid point of interest in image PL */

Query 2 – For locations with smog levels over X, show terrain elevation with contour lines every 25 feet for places in which there are school districts, showing such smog level in red.

This involves thresholding or segmentation, a contouring operator, and a pictorial superimpose operation.

Extended OQL query:
 DISPLAY S.Cell_Value IN RED, (E.Cell_Value MOD 25) AS Countours
 FROM Smog AS S, Elevation AS E,
 School_Districts AS SD
 WHERE S.Cell_Value > X AND
 SD.Cell_Value EXISTS /*
 Contour lines every 25 feet */

Query 3 - For the East Los Angeles area where smog level is greater than value X within last year, show an aerial view with smog levels shown in red, superimposing major freeways, and provide detailed information on population density and ethnic mix for where this information is available in the database.

Alphanumeric population density could be either stored outside the stack with a pointer to it from each pertinent cell in the image stack, or else it could be recorded with each cell as defined above in Figure 2b (this is a matter of access time and storage efficiency which should be transparent to the user).

6. Conclusion

We are currently researching at UCLA the development of the image stack model view along with the stream model M2 as one encompassing model. The intent is to provide such an image stack view whether or not data is actually physically stored as such. We highlight herein proposed extensions to object database management developments particularly the ODMG standard that

we consider feasible. We are also developing further functionality in the area of stack relationships, including related information available in the Internet. The medical environment has been a major focus area in our work and will continue in this domain along with the biogeographical domain (like bio-terrorism).

Acknowledgement

This research work is partially supported by National Science Foundation Grant # IIS 0140384, "Multimedia Stream Modeling, Relationships and Querying."

References

1. Cárdenas, "A 2025 Scenario and Vision on Stream Data Modeling," in *Information Systems Engineering -- State of the Art and*

- Research Themes*, Brinkkemper, S., Lindencrona, E. and A. Solvberg (Editors), Springer-Verlag, London, Great Britain, 2000.
2. J. D. N. Dionisio and A. F. Cárdenas, "A Unified Data Model for Representing Multimedia, Timeline and Simulation Data," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 10, No. 5, pp. 746-767, September/October 1998.
3. J. D. N. Dionisio and A. F. Cárdenas, "MQuery: A Visual Query Language for Multimedia, Timeline and Simulation Data", *Journal of Visual Languages and Computing*, Vol. 7, Academic Press Ltd., pp. 377-401, 1996.
4. R. G. Cattell (editor), *The Object Database Standard: ODMG, Release 3.0*, Morgan Kaufmann, San Francisco, Ca. 2000.

MULTIMEDIA DATA SOURCES

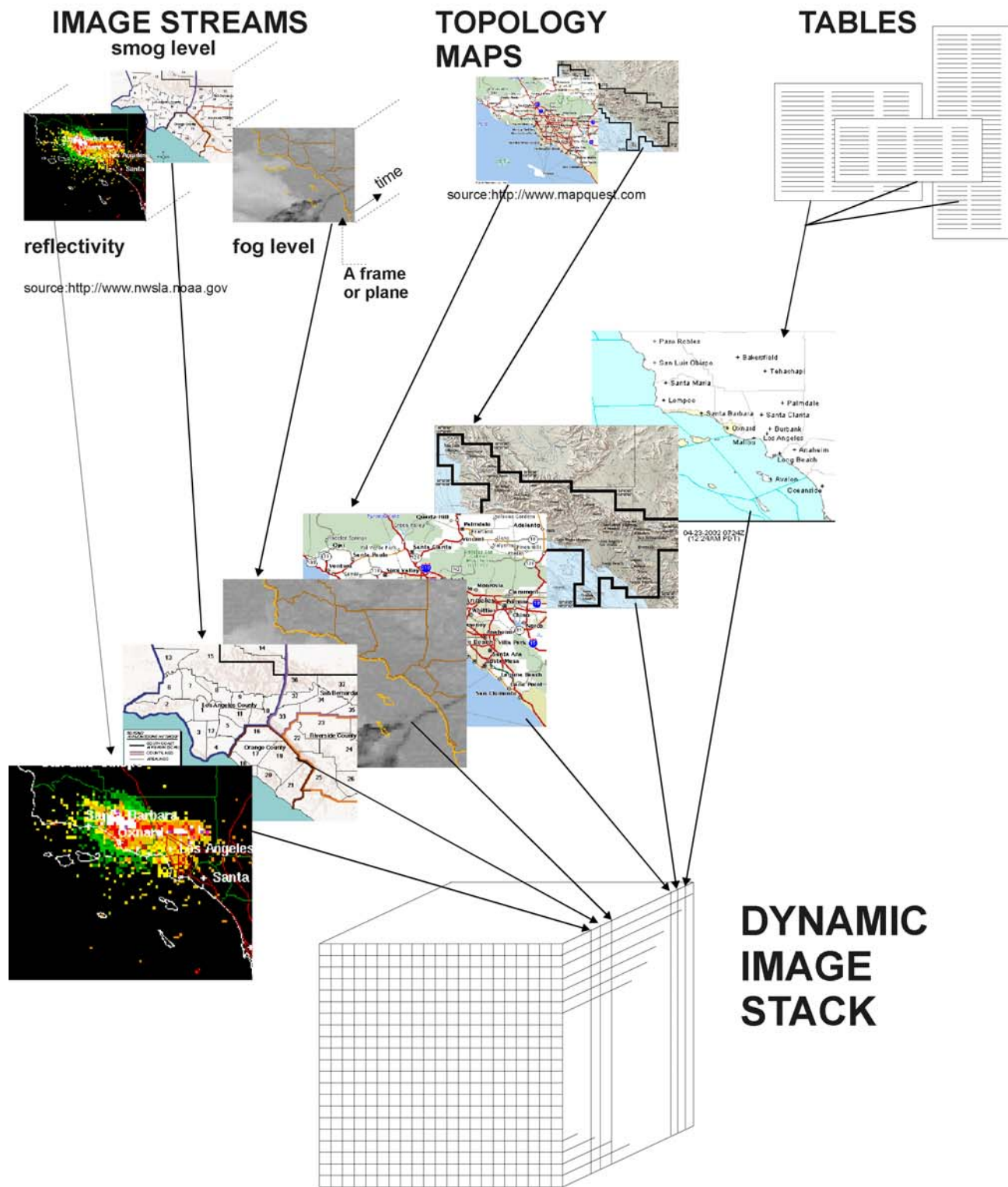


Figure 1. Logical Dynamic Image Stack Model/View and Distributed Multimedia Data Sources

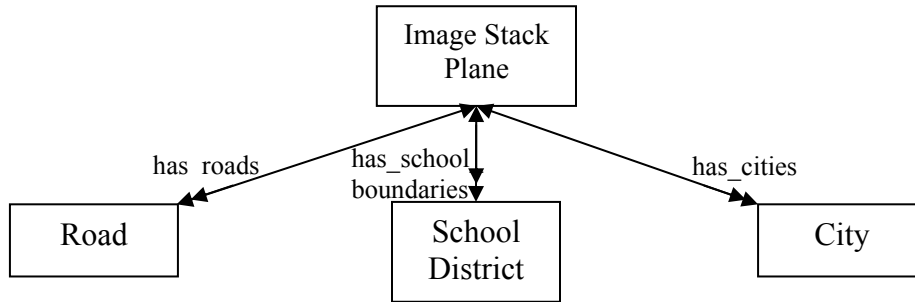


Figure 2a – Partial data model representing point, line and polygon objects for one image stack plane

*/*Class and relationship definitions for the block diagram in Fig. 2a above go here*/*

attribute array(x,y,p)<Cell> Dynamic_Image_Stack

/ ODMG Standard to be expanded to handle from 1 to 3-dimensional arrays */*

class Cell

```
{
  attribute Cell_Value; /* data type to be added: bit_value or other */
  attribute List<X> List_of_X; };
```

class X

```
{
  attribute string X_name;
  attribute List<X_Attribute> List_of_X_Attributes; };
```

class X_Attribute

```
{
  attribute string Attribute_Name;
  attribute Attribute_Value; /* data type to be defined */ };
```

typedef Plane_Name_I Dynamic_Image_Stack(*,*,i)

/ This assigns the name Plane_Name_i to a plane in the image stack, part of 3-dimensional array extension to ODMG */*

Figure 2b – Partial database schema definition of stack model/view in extended ODMG