IMAGE STACK STREAM MODEL OF MULTIMEDIA DATA

Alfonso F. Cárdenas and Panayiotis A. Michael
Computer Science Department, University of California at Los Angeles
3731 Boelter Hall, UCLA, Los Angeles, California 90024, 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 stream model* or *view* of data as a convenient basis for both querying and visualizing the situation and changes through time of phenomena buried in the multitude of such variety of data. We outline briefly the requirements with motivating queries, and the recommended extensions to the latest object database management developments, particularly the ODMG standard to support this.**

Keywords: multimedia streams, querying and visualization, stack stream model/view, evolution over time, ODMG extensions

1. Introduction

An increasing multitude of sensors is being deployed to capture growing amounts of multimedia data (alphanumeric, image, sound and video) [1]. The querying and visualization of relevant information from the multitude streams of different but related types of data being captured is a major challenge that we address in our R&D efforts. Of particular focus is how to access and present the behavior through time of the various types of related information or phenomena in various data streams. Significant work has been done in recent times on video indexing and accessing by content and on visual languages [2-5]. However, it has usually focused on viewing individual video streams and not on the emerging multiple heterogeneous sensor stream world and querying that we are addressing.

We recently introduced the dynamic image stack model/view [6] as a convenient way to not only visualize such variety of data but also as a basis for querying a snapshot of it, even if the data is not stored as such. We now introduce herein the dynamic *image stack stream model* to support further requirements and example queries for accessing and visualizing change or trend patterns of multimedia data streams of interest through time, and the needed extensions to the latest object database management developments particularly in the ODMG standard to support this.

2. Multimedia Data Streams

A *stream* is an ordered sequence of frames or values, for example Figure 1 top left. A frame could be an image, photograph, a frame in a video stream, 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 to select specific elements coming rapidly in a stream of alphanumeric data. We are currently addressing this challenge and the broader challenge of querying and visualizing at selected points in time specific elements based on the content of the element. For example, select the images with red objects, or 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 previously proposed and implemented in part the M Model [7] which considers the stream as a fundamental construct in addition to all the constructs of today's DBMS, and its associated query language MQuery language [8]. It is feasible to store selected substreams in a database, sampled or selected according to various criteria of interest.

3. Image Stack Model/View

We have proposed the *logical dynamic image stack model/view* [2] as 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 arbitrary 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. Digitized aerial/satellite photos, with a plane for each type of data, for example, poison or pollution levels, and reflectivity/rain levels.

2. 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 (another 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 a school districts plane, the related school name, student population, etc.

3. 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 a one-shot presentation, then there would be no need to permanently store it physically as a plane in the stack.

Issues of image co-registration and differing grid sizes (resolution) may arise in materializing the image stack view due to the variety of sensor sources but are not addressed herein. Various image processing may be involved in taking a frame from a stream to place it in an image stack.

In many applications the main interest is in finding out and visualizing what the change and trend of the world order is. We would like to then see image stacks taken at various points of interest and to find out what the change is from one point in time to another. This leads us to introduce the notion of a *stream of image stacks*, illustrated in Figure 2 for a stream with two stacks showing change through time. This is the type of queries demanded in high level decision making. We will present motivating queries in the following sections.

4. Stack Streams and ODMG Database Structure and Extensions

The latest database management system architecture ODMG [9] 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 arrays of arrays or arrays of these collections, or
(2) a collection of data structures or a collection of arrays or a collection of another collection,

and recursively.

The image stack illustrated in Figure 1 and the stream of image stacks in Figure 2 may be supported by the following database architecture 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 a fourth dimension for time, and the corresponding extensions to the Object Query Language (OQL) commands. The I plane in the stack S at the point in time T is denoted by S (*, *, I, T).

In our approach, each array cell at one point in time 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 <X> 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; and the List_of_X_Attributes is composed of instances of the class X_Attribute with attributes Attribute_Name and Attribute_Value, see Figure 4.

We envision the lowest or smallest granularity stream Cell_Stream of one-pixel frames as a one dimensional array of Cells (with t points in time):

```
array (t) <Cell> Cell_Stream
```

The stream of plane I instances in the image stack S is envisioned as

```
array (t) <S(*,*,I,t)> Plane_Name_I_Stream
```

This assigns the name Plane_Name_I_Stream to the stream of plane I instances in the image stack S.

A stream of stacks at four selected points in time a, b, c and d illustrated in Figure 3 is denoted in our model by

array (t) <S (*,*,*,T = a, b, c, d)> Stack_Stream

Note that this database approach and 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 at the pixel level, and at any point in time (!), 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).

In addition to the above class and array database structures, we would have the class definitions for 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.

5. Sample Queries

Below are several motivating sample queries in English leading to our innovations. These queries are typical of the type of major decision making queries that are not automated today by any generalized DBMS, GIS or visualization system, and would take much painful and custom data retrieval, processing and visualization to implement. The requirement for easily accessing and visualizing the situation and changes through time of phenomena buried in the multitude of multimedia data streams is illustrated.

.

We show 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 an imaging or visualization system for display. Our focus in the article is on the dynamic image stack streams 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 2-3D visualization presentation interfaces that are obviously used as the final step to each of the queries below for final visualization of query results; these are 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, and compare the fume level to 12 and 24 hours earlier showing clearly the differences in red for higher poison levels and in blue for lower poison fume levels.

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. Then the query is run for the stream world of 12 and 24 hours earlier, and compares the resulting poison levels with the now levels at the corresponding location points to calculate and show the changes that occurred.

Extended OQL query for retrieving the data at the "Now" point in time:

```
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
          AND Time = Now ;
 /* 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 at time point Now */
```

Next, the above query modified to get the PL.Cell_Value at the locations of interest is run for 12 and 24 hours earlier. The resulting PL.Cell_Values are then compared to obtain the desired output showing the increases or decreases in poison levels. Note that we thus create a stream of three stacks (one image per stack in this example) specific for this user view and query.

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 purple. Compare the smog level to a year and two years earlier showing clearly the differences in red for higher smog levels and in blue for lower smog levels.

This involves thresholding or segmentation, a contouring operator, and a pictorial superimpose operation, followed by a difference operation

Extended OQL query:

```
DISPLAY S.Cell_Value IN RED,
          (E.Cell_Value MOD 25) AS CONTOURS
FROM  Smog AS S, Elevation AS E,
      School_Districts AS SD
WHERE S.Cell_Value > X AND
        SD.Cell_Value EXISTS AND
        Time = Now;
/* CONTOURS operator produces lines every 25 feet
(MOD 25) */
```

Next the above query but modified to get the S.Cell_Value at the locations of interest is run for one and two years earlier. The resulting S.Cell_Values are then compared to obtain the desired output showing the increases or decreases in smog levels.

Query 3 - For the East Los Angeles area where smog level is greater than the average smog value X within last year, show an aerial view with smog levels shown in purple, superimposing major freeways, and provide a display (an image in the resulting stack view) of detailed information on population density and ethnic mix for where this information is available in the database. Then compare the smog level to two, four, six, eight and ten years earlier and show a stream of smog images/pictures clearly indicating for each smog picture the differences in red for higher smog levels and in blue for lower smog levels compared to the prior period; provide also the percentage change in population density and ethnic mix vs. the prior period.

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 in Figure 4 (this is a matter of access time and storage efficiency which should be transparent to the user).

This is indeed a complex query that we would expect to be highly automated by an intelligent multimedia stream databases management system embodying the innovations that we propose herein. The system would reissue the first query modified appropriately to retrieve the smog information for the previous periods, and then calculate the differences requested for each point in time.

6. Conclusion

We are currently researching at UCLA the development and implementation of the image stack model view and of streams of stacks over a multitude of multimedia streams. 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. The medical environment has been a major focus area in our work and will continue in this domain along with the bio-geographical domain (such as bio-terrorism).

Acknowledgement

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

References

1. A. F. 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), Spinger-Verlag, London, Great Britain, 2000.

2. M. Flickner, et. al., “Query by Image and Video Content: The QBIC System,” *Computer*, Volume 28, No. 9, September 1995.

3. S. Chang, W. Chen, et. al, “VideoQ: An Automated Content Based Video Search System Using Visual Cues,” *ACM Multimedia*, 1997.

4. S. Srinivasan, D. Ponceleon, et. al., “CueVideo: Automated Video/audio Indexing and Browsing.” *Proceedings ACM SIGIR 99*, Berkeley, CA, 1999.

5. T. Cataraci, M. F. Costabile, et al., “Visual Query Systems for Databases: A Survey,” *Journal of Visual Languages and Computing* , 1997, pp.215-260

6. A. F. Cárdenas**,** Michael, P. A. and Islam, B. S., "Stack Database Model/View of Multimedia Data," *Proceedings of the 2002 International Conference on Information and Knowledge Engineering*," Las Vegas, Nevada, June 24-27, 2002

7. 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.

8. 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.

9. R. G. Cattell (editor), *The Object Database Standard: ODMG*, *Release 3.0*, Morgan Kaufmann, San Francisco, CA 2000.

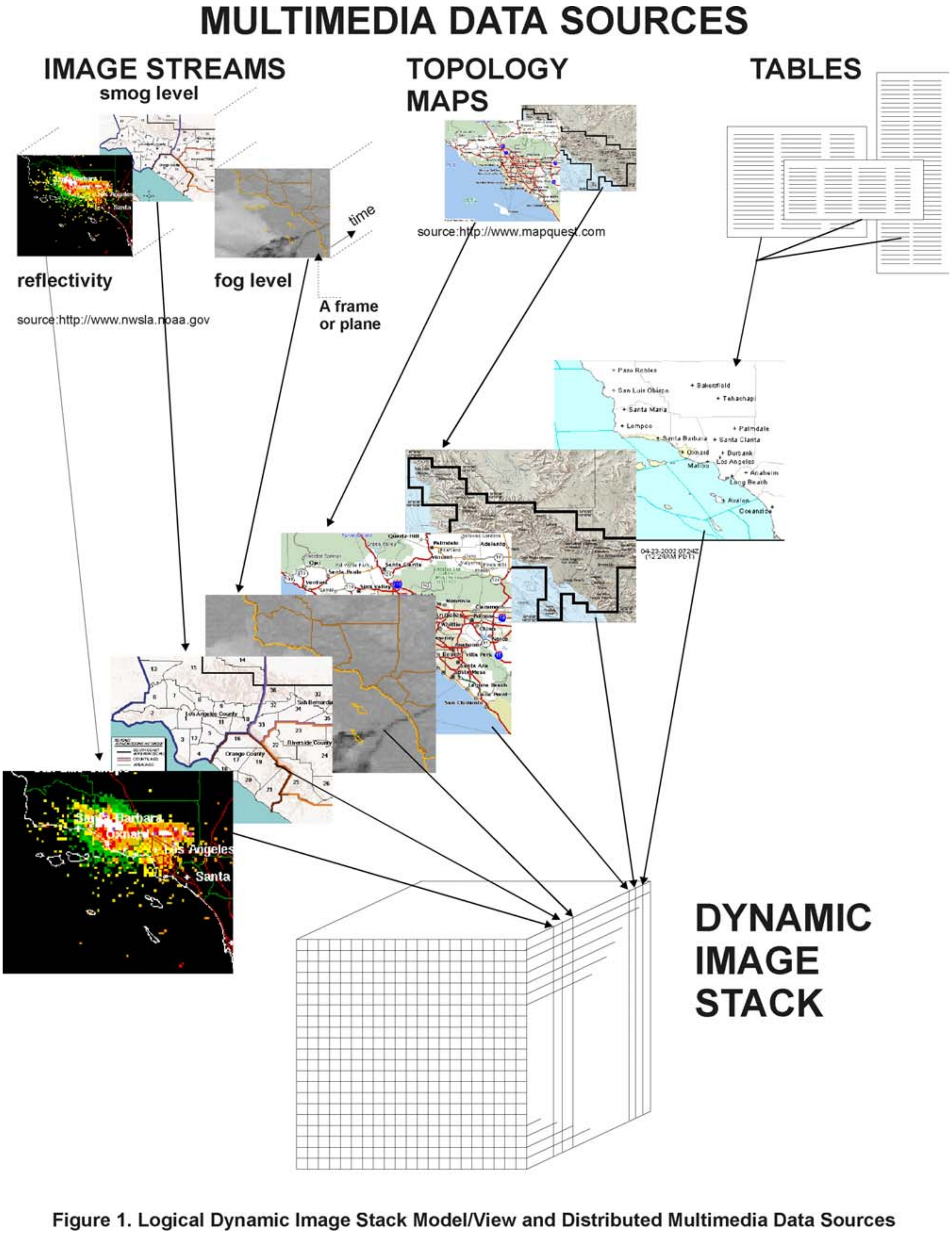

Figure 1. Logical Dynamic Image Stack Model/View and Distributed Multimedia Data Sources

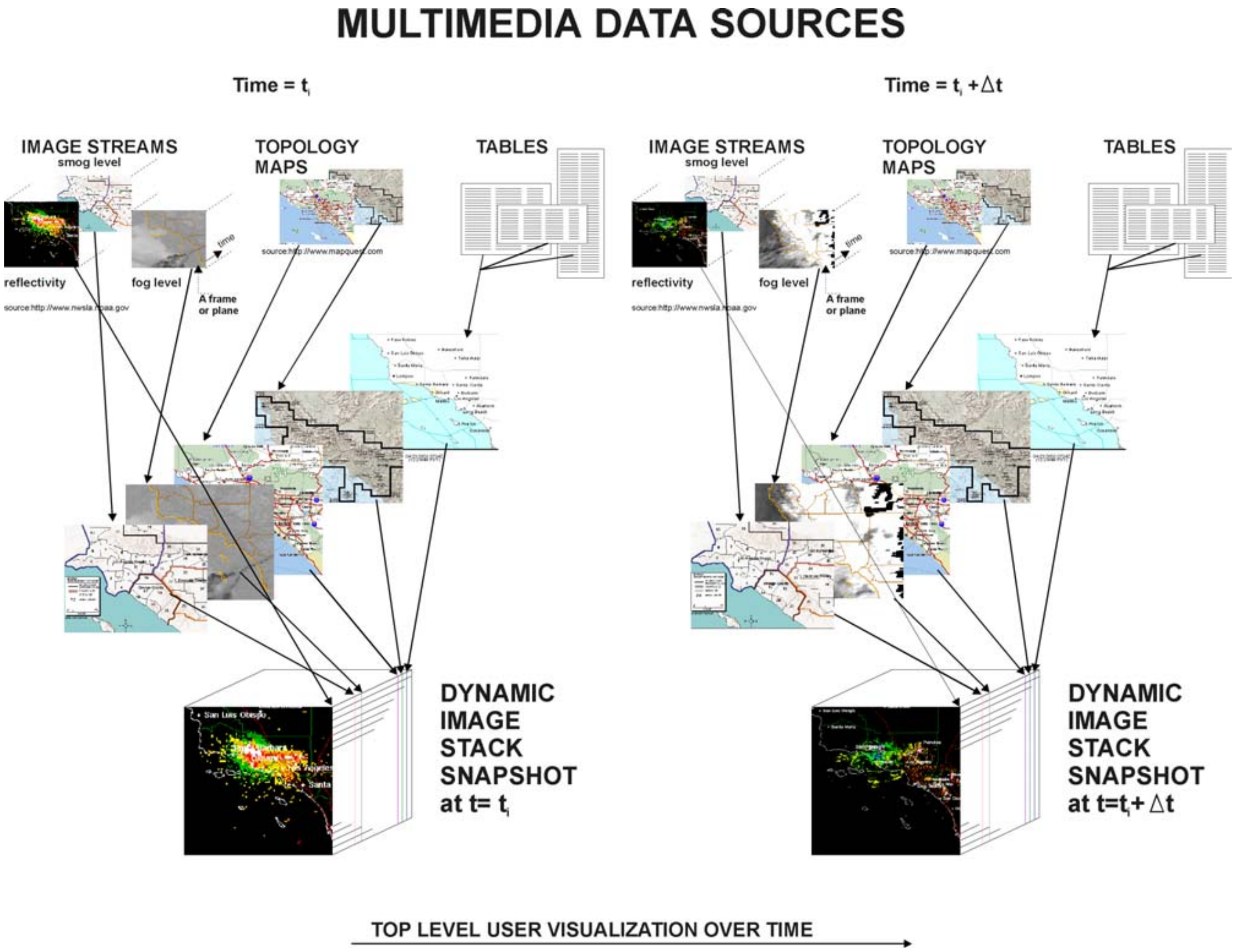

Figure 2. Stream of two image stacks showing change through time