A Spatiotemporal Database for Ozone in the Conterminous U.S.

Lixin Li
Department of Computer Sciences
Georgia Southern University
Statesboro, GA 30460-7997, USA
lli@georgiasouthern.edu

Xingyou Zhang
Department of Geology and Geography
Georgia Southern University
Statesboro, GA 30460-8149, USA
xyzhang@georgiasouthern.edu

Reinhard Piltner
Department of Mathematical Sciences
Georgia Southern University
Statesboro, GA 30460-8093, USA
rpiltner@georgiasouthern.edu

Abstract

This paper considers a set of ozone data in the conterminous U.S., which records the ozone concentration levels at a set of monitoring sites during 1994 and 1999. Existing GIS techniques are insufficient in handling such kind of spatiotemporal data in terms of data interpolation, visualization, representation and querying. We adopt 3-D shape functions from finite element methods for the spatiotemporal interpolation of the ozone dataset and analyze interpolation errors. The 3-D shape function based method estimates ozone concentration levels with less than 10 percent Mean Absolute Percentage Error. We give two approaches for visualizing the data: (i) combining the ArcGIS visualization tool with shape function interpolation results to visualize the ozone data for each year from 1994 and 1999, (ii) using Matlab to visualize the interpolated ozone data in a 3-D vertical profile display. For the spatiotemporal data representation, we use the constraint data model, because it can give an efficient and accurate representation of interpolation results. Finally, we give some practical query examples.

1. Introduction

GIS (Geographic Information System) is a computer system for capturing, storing, querying, analyzing, and displaying geographically referenced data [5]. Traditional geographically referenced data are spatial data, which include spatial information and other characteristics of spatial objects. For example, in order to model a piece of land surface, the Digital Elevation Model (DEM) data are used to collect elevation values for a number of ground point locations.

More and more GIS applications require the use of spatiotemporal data. For instance, in the field of environmental GIS, air pollutant data are often measured by monitoring sites at certain time instances. How to represent the extra *temporal* information becomes the first question to ask when we deal with spatiotemporal data. There exist two approaches.

1. Store the data with different time instances into separate datasets. Each dataset becomes a collection of spatial data. The advantage with this approach is that the existing GIS techniques can be applied to each dataset at a particular time instance. But its disadvantage is that spatial and temporal information is separated, which makes data analysis such as spatiotemporal queries very difficult.

2. Treat *time* as an extra attribute of the data. For example, a set of spatiotemporal data in 2-D space and 1-D time will have such schema as $(x, y, t, value)$.

The second is a better approach to handle temporal information in spatiotemporal data, because it keeps the spatial and time properties of the data together and makes data analysis easier. However, by using the second approach, the existing GIS techniques become insufficient.

There are four challenging spatiotemporal issues which are not solved in traditional GIS, but we can solve using newer techniques. As an illustration, we use the Aerometric Informational Retrieval System (AIRS) data that contains annual measurements of ozone concentration levels.

1. **Spatiotemporal interpolation.** It is very common in GIS that sample measurements are taken only at a set of points. In order to estimate the values at unsampled points, interpolation is needed. There exist a number of spatial interpolation algorithms in traditional GIS. Reference [28] discusses in detail *inverse distance weighting (IDW)*, [9] discusses *kriging*, [13] describes *splines*, [31] introduces *trend surfaces*, and [14] utilizes *Fourier series*. [17] gives a review and comparison of spatial interpolation methods. However, with the additional *time* attribute, these traditional spatial interpolation are insufficient for spatiotemporal data and new spatiotemporal interpolation methods must be developed. For example, suppose that we know the recording of Carbon Monoxide at different monitoring sites at different instances of time. Some spatiotemporal interpolation method is needed to estimate the Carbon Monoxide values at unsampled locations and times.

   Spatiotemporal interpolation is a growing research area. There have been some papers addressing the issue of spatiotemporal interpolation in GIS. For example, [23] utilizes kriging for spatiotemporal interpolation and [22, 20, 21] discuss several shape function based spatiotemporal interpolation methods. Reference [22] also describes IDW and kriging based spatiotemporal interpolation methods and compares them to some shape function based methods by using an actual real estate data set with house prices. References [12, 19, 27] also deal with the use of spatiotemporal interpolations for different applications.

   Solution: based on the nature of the ozone dataset, we use the 3-D method based on shape functions for tetrahedra introduced in [22] as our spatiotemporal interpolation method, which combines both spatial and temporal associations in the process of interpolation.

2. **Spatiotemporal data visualization.** Traditional GIS uses static maps to represent spatial data. Since spatiotemporal data also change with time, a natural way to illustrate spatiotemporal data is animation. Another option is to use a 3-D display with slices, which visualizes *time* as the third dimension.

   Solution: we illustrate our ozone dataset by both static maps at different time instances and 3-D display with slices.

3. **Spatiotemporal data representation.** In traditional GIS, spatial data are represented in the relational data model, which is the most popular data model. Many database systems are based on the relational model, such as Oracle and MySQL. However, the relational model has disadvantages for some applications, which

may lead to infinite relational databases [25]. An infinite relational database means the database has relations with infinite number of tuples. In reality, only a finite set of the tuples can be stored in a relation. Therefore, a finite set of tuples has to be extracted, which leads to data incompleteness.

The relational data model shows more disadvantages when handling spatiotemporal data. For example, using the relational model, the current contents of a database (database instance) is a snapshot of the data at a given instant in time. When representing spatiotemporal data, frequent updates have to be performed in order to keep the database instance up to date, which erases the previous database instance. Therefore, the information in the past will be lost. This irrecoverable problem makes the relational data model impractical for handling spatiotemporal data.

There exist many spatiotemporal data models, such as the raster snapshot model [1], the temporal map set model [2], the event-based spatiotemporal model [24], space-time composite model [18], data models based on moving points and regions [10, 11], Worboy's data model [29], the parametric rectangle data model [4, 26], the parametric 2-spaghetti data model [7], and the geometric transformation model [8, 6].

Solution: we use the constraint data model [25, 16] because of its efficient and accurate representation of the spatiotemporal data after interpolation.

4. **Spatiotemporal querying.** There are many possible spatiotemporal queries to be asked for a particular set of spatiotemporal data. For example, a very basic query would be, "What is the value of interest at a specific location and time instance?" If we have good interpolation results and efficient representation of the interpolation results, many spatiotemporal queries can be easily answered by existing query languages, such as SQL and Datalog.

   Solution: based on our spatiotemporal interpolation result and its representation in the constraint data model, we give some interesting query examples for our ozone data that are easy to express in Datalog.

The rest of this paper is organized as follows. Section 2 describes our experimental data, which is a set of AIRS data with annual ozone concentration measurements in the conterminous US during 1994 and 1999. Section 3 discusses briefly the 3-D shape function based spatiotemporal interpolation method and evaluates the interpolation errors using leave-one-out cross-validation for the ozone dataset. Section 4 gives the visualization results of the interpolated data by static maps at different time instances and 3-D display with slices. Section 5 shows how the interpolation results
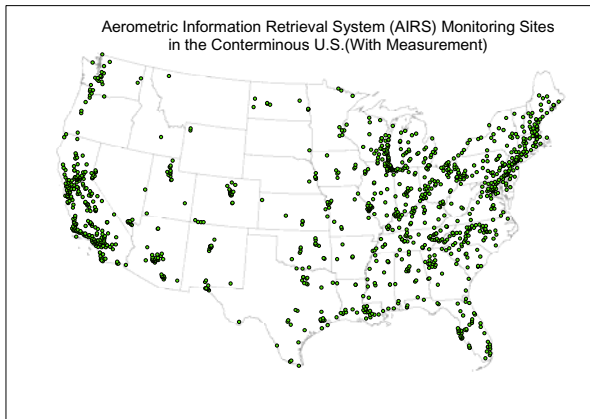
Figure 1: 1209 AIRS monitoring sites with measurements in the conterminous US.
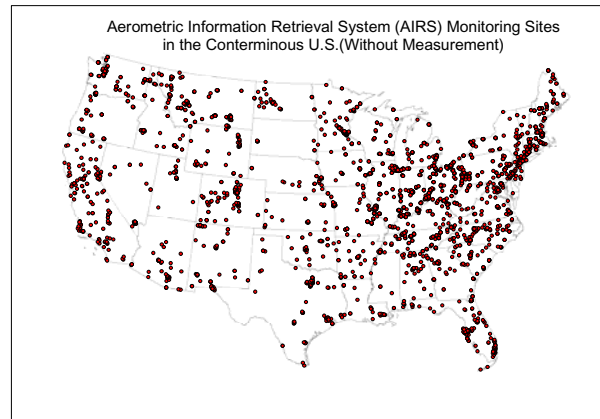

Figure 2: 2904 AIRS monitoring sites without measurements in the conterminous US.

can be stored efficiently in constraint databases. Section 6 describes several practical spatiotemporal queries which are based on our interpolation results, error analysis, and constraint representation of the data.

2. Experimental data

Our experimental data is a set of AIRS (Aerometric Information Retrieval System) data with annual ozone concentration measurements in the conterminous US (website http://www.epa.gov/airmarkets/cmap/data/category1.html). AIRS is a computer-based repository of information about airborne pollution in the United States and various World Health Organization (WHO) member countries. The system is administered by the U.S. Environmental Protection Agency (EPA).

The data coverage contains point locations of the monitoring sites for which AIRS data are collected, the annual concentration level measurements of Ozone (O3), and the years of the measurement. We obtained several datasets from the U.S. EPA (website http://cfpub.epa.gov/gdm) and reorganized them into a dataset with schema (x, y, t, w), where x and y attributes are the longitude and latitude coordinates of monitoring site locations, t is the year of the ozone measurement, and w is the O34MAX (4th Max of 1-hr Values for O3) value of the ozone measurement. The original dataset has many zero entries for ozone values, which means no measurements available at a particular site. After filtering out all the zero entries from the original dataset, there are 1209 sites left with measurements. The other 2904 sites are without measurements. Figure 1 shows the locations of the 1209 monitoring sites with measure-

ments, while Figure 2 illustrates the distribution of the 2904 sites without measurements.

Among the 1209 monitoring sites with measurements, the sites with the most measurements record the ozone values from 1994 to 1999. But not all sites have ozone values during this 6-year period. For example, some sites have measurements from 1995 to 1999, some sites have measurements from 1996 to 1999, some sites have measurements from 1997 to 1999, while some other sites have measurements from 1998 to 1999. In total, there are 6135 ozone value measurements recorded. Each measurement corresponds to the ozone value at a spatiotemporal point (x, y, t), where (x, y) is the location of one of the 1209 monitoring sites, and t is a year between 1994 and 1999.

3. Spatiotemporal interpolation

Reference [21] summaries two basic methods for spatiotemporal interpolation, the reduction method, which treats time as an independent dimension, and the extension method, which treats time as equivalent to a spatial dimension. The reduction approach requires some measurements at a particular location, so that some 1-D interpolation function in the time dimension are needed for a multiplication by some spatial interpolation function. For our AIRS ozone data, there are 2904 monitoring sites without measurement at any year, and other 1209 sites have some measurement between 1994 and 1999. Therefore, if the reduction method is used for our experimental data, there will be no interpolation result for the 2904 sites without measurements.

In order to interpolate for all the monitoring sites with and without measurements during 1994 and 1999, the ex-

tension method could be used. There exist different approaches to the extension method. Reference [22] describes shape function, IDW (Inverse Distance Weighting), and kriging based extension methods. A comparison based on a set of actual real estate data set with house prices has been made in [22]. The comparison criteria include interpolation accuracy, error-proneness to time aggregation, invariance to scaling on the coordinate axes, and the type of constraints used in the representation of the interpolated data. The experimental results show that the extension method based on shape functions is the most accurate and the overall best spatiotemporal interpolation method.

In this paper, we choose the extension method based on shape functions for our ozone dataset. A brief description of this method and error analysis for the interpolation result is given in this section.

3.1. Extension method based on shape functions

Since our dataset is 2-D in space & 1-D in time and the extension method treats *time* as a regular third dimension, this method is very similar to the linear approximation by 3-D shape functions for tetrahedra. Shape functions, which can be viewed as a spatial interpolation method, are popular in engineering applications, for example, in Finite Element algorithms [30, 3].

Interpolation by 3-D shape functions for tetrahedra requires dividing the 3-D domains into a finite number of tetrahedra. For each tetrahedron in the tetrahedral mesh, there are four corner vertices, whose values have been measured. Assume (x_i, y_i, t_i) $(1 \leq i \leq 4)$ are the spatial and temporal information of the four corner vertices of a tetrahedron, and w_i $(1 \leq i \leq 4)$ are the measured values at the vertices. The shape function based spatiotemporal interpolation equation for any point (x, y, t) located inside the tetrahedron with corner points p_1, p_2, p_3, p_4 and corner values w_1, w_2, w_3, w_4 is [22]

$$w(x, y, t) = \frac{\mathcal{V}_1}{\mathcal{V}}w_1 + \frac{\mathcal{V}_2}{\mathcal{V}}w_2 + \frac{\mathcal{V}_3}{\mathcal{V}}w_3 + \frac{\mathcal{V}_4}{\mathcal{V}}w_4 \quad (1)$$

where $\frac{\mathcal{V}_i}{\mathcal{V}}$ $(1 \leq i \leq 4)$ are called shape functions and the value $w(x, y, z)$ is obtained for a point p located inside the tetrahedron.

In Equation 1, $\mathcal{V}_1$, $\mathcal{V}_2$, $\mathcal{V}_3$ and $\mathcal{V}_4$ are the volumes of the four sub-tetrahedra $pp_2p_3p_4$, $p_1pp_3p_4$, $p_1p_2pp_4$, and $p_1p_2p_3p$, respectively; and $\mathcal{V}$ is the volume of the outside tetrahedron $p_1p_2p_3p_4$. These volume values can be computed using the standard volume formula as is done in [22].

We implemented this spatiotemporal interpolation method based on 3-D shape functions into a Matlab program and applied it to the AIRS ozone data. There are 6135 original ozone measurements at 6135 spatiotemporal

points, as described in Section 2. The Matlab function *delaunayn* is used to compute the tetrahedral mesh with the 6135 spatiotemporal points as corner vertices. There are 30897 tetrahedra in the mesh. Using the mesh and the original 6135 original ozone values measured at its corner vertices, we did the following two interpolation tasks, which result in 18543 estimated ozone values:

1. Interpolate the ozone values at all the 2904 sites without measurements during 1994 and 1999.

2. Among the other 1209 sites, for those sites with only partial measurements, interpolate the ozone values for the missing years.

Note that using this spatiotemporal interpolation method, not only the above interpolation tasks can be accomplished, the ozone value at any point inside the tetrahedral mesh can be interpolated.

3.2. Error analysis

We use leave-one-out cross-validation to evaluate the spatiotemporal interpolation errors for the ozone dataset. This cross-validation process removes one of the n observation points and uses the remaining $n - 1$ points to estimate its value; and this process is repeated at each observation point[15]. The observation points are the points with measured original values. For our experimental data, the observation points are the spatiotemporal points (x, y, t), where (x, y) is the location of a monitoring site and t is the year when the ozone measurement was taken. After the leave-one-out cross-validation, each of the observation points will not only have its original value, but also will have an interpolated value. The original and interpolated values at each observation point can be compared for the purpose of an error analysis.

The shape function interpolation method estimates the values inside the convex hull after generating the tetrahedral mesh. It cannot interpolate for the points that are outside the convex hull. Suppose the original number of spatiotemporal observation points is n, which is 6135 for our experimental ozone dataset. The leave-one-out cross-validation method requires generating n different tetrahedral meshes with $n-1$ vertices. It is possible that for a particular tetrahedral mesh, the point left out in the mesh and to be estimated is outside the convex hull. So its value cannot be interpolated. In our experiment, the number of points that cannot be interpolated is 36. Therefore, the rest 6099 points with interpolation results are used for our error analysis.

We used two different approaches for the error analysis in our experiment. The descriptions and results of each approach are given below.

YEAR	N	MIN	MAX	MAE	MBE	RMSE	MAPE
1994	941	-0.081000	0.088568	0.011884	0.000416	0.016892	12.3586
1995	990	-0.099057	0.068092	0.009143	0.001912	0.012720	9.3613
1996	1000	-0.084794	0.063769	0.008389	-0.002869	0.011411	9.1944
1997	1041	-0.065462	0.038500	0.008223	-0.000914	0.011493	9.1170
1998	1063	-0.118325	0.080146	0.009115	0.002416	0.013022	9.4701
1999	1064	-0.078475	0.064124	0.008211	-0.000947	0.011778	9.3932
1994 - 1999	6099	-0.118325	0.088568	0.009118	0.000004	0.013080	9.7792

Table 1: Major statistics of spatiotemporal interpolation errors for US ozone data (1994-1999).

1. We analyze the interpolation error at each data point by calculating the difference between its original and interpolated values as follows:

$$E_i = \frac{\mid I_i - O_i \mid}{O_i} \qquad (2)$$

where E_i is the interpolation error at observation point i, I_i is the interpolated value at point i, and O_i is the original value at point i.

The analysis results are: 40% points have errors below 5%; 66% points have errors below 10%; 82% points have errors below 15%; 90% points have errors below 20%; and 99% points have errors below 50%.

2. We use four types of error statistics, MAE (Mean Absolute Error), MBE (Mean Biased Error), RMSE (Root Mean Squared Error) and MAPE (Mean Absolute Percentage Error). Each of the above error statistics is calculated for each year and the whole period. The definitions of MAE, MBE, RMSE and MAPE are as follows:

$$
\begin{aligned}
MAE &= \frac{\sum_{i=1}^{N} \mid I_i - O_i \mid}{N} \\
MBE &= \frac{\sum_{i=1}^{N} (I_i - O_i)}{N} \\
RMSE &= \sqrt{\frac{\sum_{i=1}^{N} (I_i - O_i)^2}{N}} \\
MAPE &= \frac{\sum_{i=1}^{N} \frac{\mid I_i - O_i \mid}{O_i}}{N}
\end{aligned}
\qquad (3)
$$

where N is the number of observations, I_is are the interpolated values, and O_is are the original values.

Since the ozone measures themselves are quite small, the error statistics such as MAE, MBE and RMSE have small values. We think the MAPE is a better statistic error measurement for the ozone dataset. Table 1 shows that 1994 spatiotemporal interpolation has the largest MAPE as expected, since 1994 is the year with the most missing data. There is no significant difference during 1995-1999. The overall MAPE is 9.8%.

4. Spatiotemporal data visualization

We used ArcGIS and Matlab software packages to visualize the spatiotemporal interpolation result of the ozone dataset.

4.1. Using ArcGIS

ArcGIS is the most popular and leading commercial GIS software package with an integrated collection of GIS software products. We combine our interpolation result from the 3-D shape function based spatiotemporal interpolation method with the ArcGIS visualization tool. Since ArcGIS does not provide any spatiotemporal interpolation function, such as animation and 3-D display which treat time as the third dimension, we have to display the interpolation result at each time instance.

There are two steps of our visualization procedure using ArcGIS.

1. Importing into ArcGIS the shape function based interpolation result at the total of 4113 monitoring sites during 1994 and 1999, which is a point dataset.

2. Based on the point data imported from the previous step, using the kriging interpolation method provided by ArcGIS to produce six continuous maps from 1994 to 1999.

We have to use some other interpolation method such as kriging, instead of shape functions, in Step 2. This is because shape function based interpolation method is newly introduced into the area of GIS, and it is not implemented in ArcGIS. Figure 3 illustrates the visualization results with the combination of shape functions and ArcGIS during 1994 and 1999.
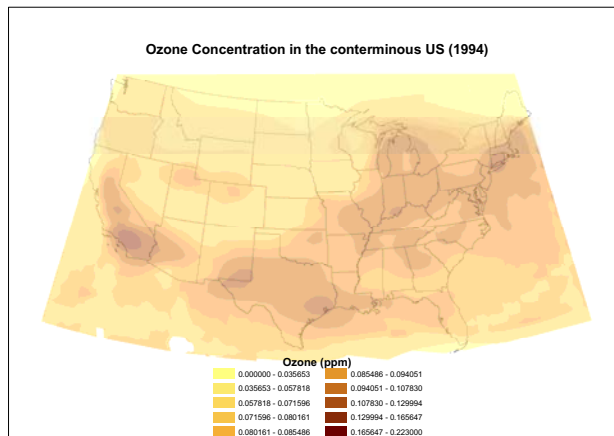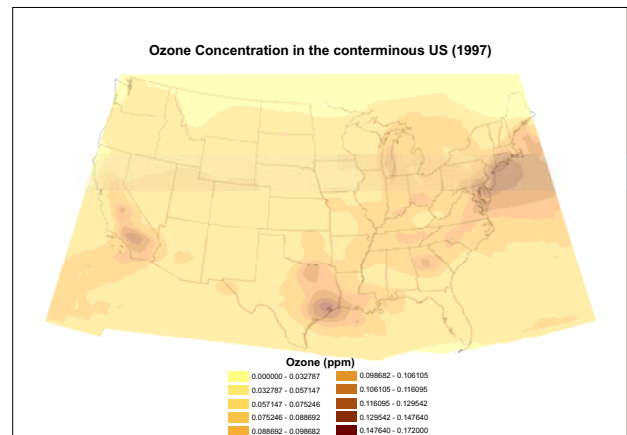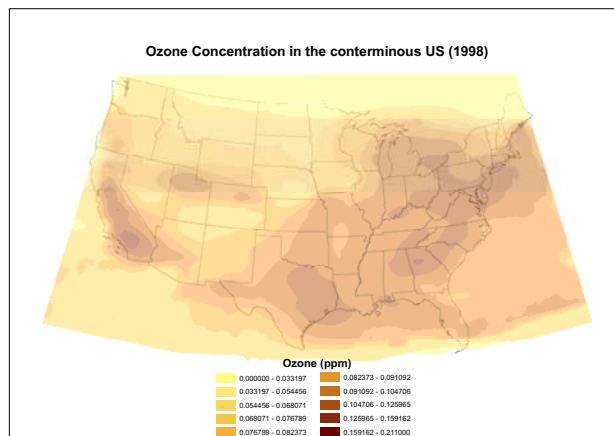
(a) Visualization results for 1994.
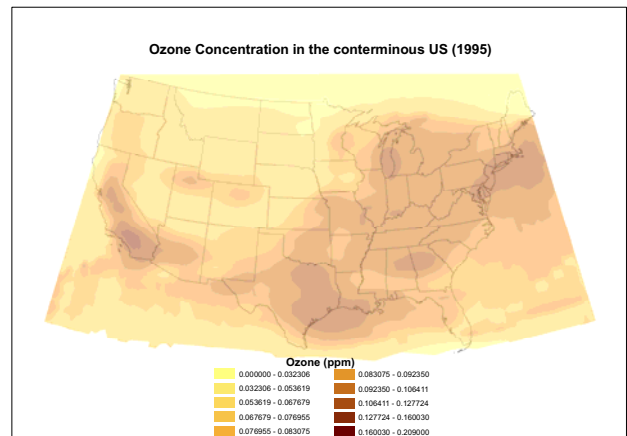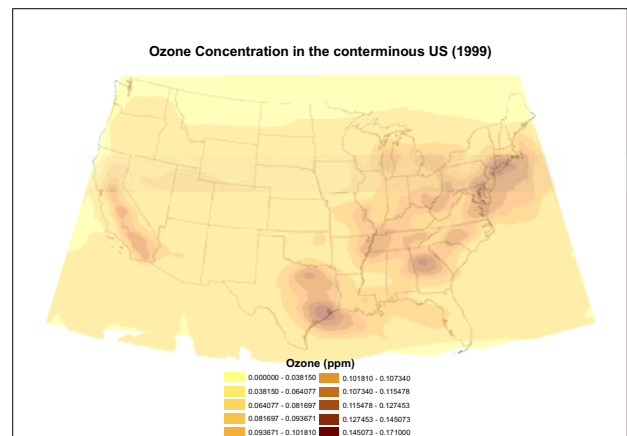
(b) Visualization results for 1995.
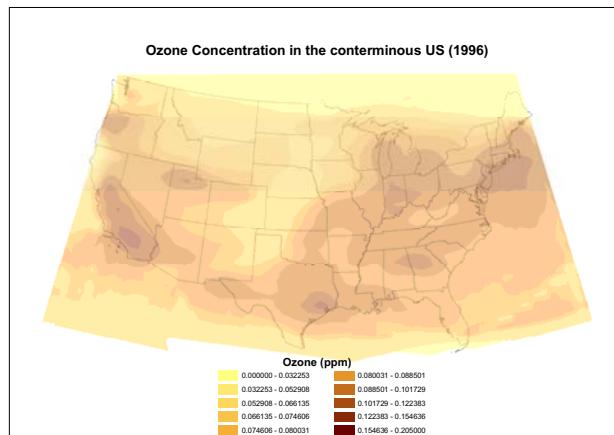
(c) Visualization results for 1996.

(d) Visualization results for 1997.

(e) Visualization results for 1998.

(f) Visualization results for 1999.

Figure 3: Visualization results with the combination of shape functions and ArcGIS during 1994 and 1999.
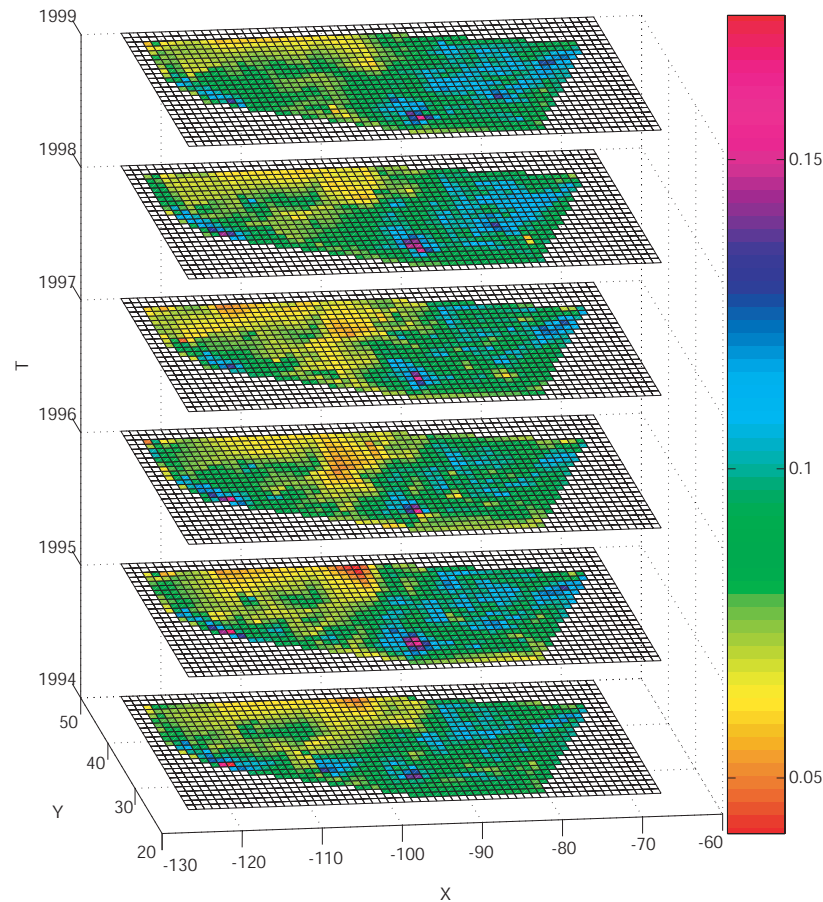
Figure 4: Visualization results using Matlab.

4.2. Using Matlab

We also programmed in Matlab to visualize our spatiotemporal interpolation result in a vertical profile display, as shown in Figure 4. The six slices in the figure corresponds to ozone concentration visualizations in six years during 1994 and 1999. They are obtained by intersecting six horizontal time planes with the tetrahedral mesh of the 6135 measured spatiotemporal points.

5. Spatiotemporal data representation in constraint databases

Theoretically, there are infinite number of points inside the (x,y,t) domain that have interpolated values. Representing the interpolated values at an infinite number of points is impossible in relational databases. However, in constraint databases, we can efficiently represent 3-D shape function based spatiotemporal interpolation results by a finite number of tuples [21].

Since the 3-D shape function based spatiotemporal interpolation Equation 1 is linear, we can store the interpolation results in the MLPQ constraint database system [25]. Suppose the constraint relation $Ozone\_interp$ is used to store the interpolation results. The MLPQ input file to create $Ozone\_interp$ is given in Figure 5.

Figure 5 shows one rule in the MLPQ input file. The other omitted rules are of similar format. Since there are 30897 tetrahedra generated in the tetrahedral mesh, there should be 30897 rules in the MLPQ input file.

The rule shown in Figure 5 corresponds to the interpolation results of all the points located in the tetrahedron with corner vertices (-68.709, 45.217, 1996), (-68.672, 44.736, 1999), (-67.594, 44.534, 1995), and (-69.214, 45.164, 1999). The ozone values measured at these four points are

```
begin %ozoneInterpolationResults%

  Ozone_interp(x,y,t,w) :- 0.002532x + 0.003385y + 0.000511t >= 1,
                           0.002709x + 0.003430y + 0.000517t >= 1,
                           0.002659x + 0.003593y + 0.000511t <= 1,
                           0.002507x + 0.003175y + 0.000515t <= 1,
                           v = 0.0127,
                           v1 = 1/6 |1.71x+2.17y+0.35t-682.87|,
                           v2 = 1/6 |2.10x+2.84y+0.40t-790.39|,
                           v3 = 1/6 |1.28x+1.63y+0.24t-474.05|,
                           v4 = 1/6 |2.53x+3.38y+0.51t-999.13|,
                           wv = 0.063v1 + 0.087v2 + 0.096v3 + 0.074v4.


                                    ⋮

end %ozoneInterpolationResults%
```

Figure 5: The input file to create $Ozone\_interp(x, y, t, w)$ in the constraint database system MLPQ.

0.063, 0.087, 0.096, and 0.074, respectively. In the body of the rule, there are 10 constraints. The relationship among these constraints is AND. The first four constraints define the four facets of the tetrahedron, the next five constraints give the volume values, and the last constraint is the interpolation function described by Equation 1.

6. Spatiotemporal queries

In this section we give some sample spatiotemporal queries. We assume that the input constraint relations are

- $Ozone\_orig(x, y, t, w)$, which records the original measured ozone value w at monitoring site location (x, y) and time t;

- $Ozone\_interp(x, y, t, w)$, which stores the 3-D shape function interpolation results of the ozone data, as described in Section 5;

- $Ozone\_loocv(x, y, t, w)$, which stores the interpolated ozone concentration level at each monitoring site (x, y) and time t after applying the leave-one-out cross-validation, as discussed in Section 3.2.

We write queries in the Datalog query language, which is a rule based language used in several constraint database systems.

Query 6.1 *For a given location with longitude x and latitude y, find the ozone concentration level in year t.*

This can be expressed in Datalog as follows:

$$Ozone\_value(w) \quad :- \quad Ozone\_interp(x, y, t, w).$$

Query 6.2 *Suppose that in the future years, there will be budget increase so that new ozone monitoring sites can be added. Find the best areas where new monitoring sites should be installed.*

In order to decide the best locations to add new monitoring sites, we can first find those monitoring sites that have average large interpolation errors according to Equation 2, for example, over 20%. Then, we do a *buffer* operation on the set of monitoring sites with big errors to find out the areas within certain distance to each site, for example, 50 miles. Since the buffered areas are the areas with poor interpolation result, these areas can be considered the possible areas where new monitoring sites should be built.

To find the monitoring sites with more than 20% interpolation errors, we can perform the following Datalog queries:

$$
\begin{aligned}
Error(x, y, t, r) \quad :- \quad & Ozone\_orig(x, y, t, w1), \\
& Ozone\_loocv(x, y, t, w2), \\
& r = |w1 - w2|/w1.
\end{aligned}
$$

$$Avg\_error(x, y, avg(r)) \quad :- \quad Error(x, y, t, r).$$

$$
\begin{aligned}
Sites\_Chosen(x, y) \quad :- \quad & Avg\_error(x, y, ae), \\
& ae >= 0.2.
\end{aligned}
$$

To find the areas within 50 miles to the sites with more than 20% interpolation errors, we can perform a GIS **Buffer** operation on the relation *Sites_Chosen*. The buffer operation is provided by many GIS software packages and the MLPQ constraint database system. After performing the

buffer operation, an output relation will be created which contains a 50-mile buffer around the locations stored in the *Sites_Chosen* relation.

Similarly, if there will be budget cut, we could use similar queries to find out and shut down the monitoring sites with least interpolation errors.

References

[1] M. P. Armstrong. Temporality in spatial databases. In *URISA Conference on Geographic and Land Information Systems*, volume 2, pages 880–889, 1988.

[2] A. Beller, T. Giblin, K. V. Le, S. Litz, T. Kittel, and D. Schimel. A temporal GIS prototype for global change research. In *URISA Conference on Geographic and Land Information Systems*, volume 2, pages 752–765, 1991.

[3] G. R. Buchanan. *Finite Element Analysis*. McGraw-Hill, New York, 1995.

[4] M. Cai, D. Keshwani, and P. Revesz. Parametric rectangles: A model for querying and animating spatiotemporal databases. In *7th International Conference on Extending Database Technology*, volume 1777 of *Lecture Notes in Computer Science*, pages 430–444. Springer-Verlag, 2000.

[5] K. Chang. *Introduction to Geographic Information Systems*. McGraw-Hill, New York, 2nd edition, 2004.

[6] J. Chomicki, S. Haesevoets, B. Kuijpers, and P. Revesz. Classes of spatiotemporal objects and their closure properties. *Annals of Mathematics and Artificial Intelligence*, 39(4):431–461, 2003.

[7] J. Chomicki and P. Revesz. Constraint-based interoperability of spatiotemporal databases. *Geoinformatica*, 3(3):211–243, 1999.

[8] J. Chomicki and P. Revesz. A geometric framework for specifying spatiotemporal objects. In *International Workshop on Time Representation and Reasoning*, pages 41–46, 1999.

[9] C. V. Deutsch and A. G. Journel. *GSLIB: Geostatistical Software Library and User's Guide*. Oxford University Press, New York, 2nd edition, 1998.

[10] M. Erwig, R. H. Güting, M. M. Schneider, and M. Vazirgiannis. Abstract and discrete modeling of spatio-temporal data types. *Geoinformatica*, 3(3):269–296, 1999.

[11] L. Forlizzi, R. H. Güting, E. Nardelli, and M. Schneider. A data model and data structure for moving object databases. In *ACM SIGMOD International Conference on Management of Data*, pages 319–330, 2000.

[12] J. Gao and P. Revesz. Voting prediction using new spatiotemporal interpolation methods. In *Seventh Annual International Conference on Digital Government Research*, San Diego, 2006.

[13] J. E. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry*. CRC Press, Boca Raton, New York, 1997.

[14] J. W. Harbaugh and F. W. Preston. *Fourier Analysis in Geology*, pages 218–238. Prentice-Hall, Englewood Cliffs, 1968.

[15] U. Hjorth. *Computer Intensive Statistical Methods, Validation, Model Selection, and Bootstrap*. Chapman and Hall, 1994.

[16] P. C. Kanellakis, G. M. Kuper, and P. Revesz. Constraint query languages. *Journal of Computer and System Sciences*, 51(1):26–52, 1995.

[17] N. S. Lam. Spatial interpolation methods: A review. *The American Cartographer*, 10(2):129–149, 1983.

[18] G. Langran. *Time in Geographic Information Systems*. Taylor and Francis, London, 1992.

[19] J. Li, R. Narayanan, and P. Revesz. A shape-based approach to change detection and information mining in remote sensing. In C. H. Chen, editor, *Frontiers of Remote Sensing Information Processing*, pages 63–86. WSP, 2003.

[20] L. Li, Y. Li, and R. Piltner. A new shape function based spatiotemporal interpolation method. In *First International Symposium on Constraint Databases 2004*, volume 3074 of *Lecture Notes in Computer Science*, pages 25–39. Springer, 2004.

[21] L. Li and P. Revesz. A comparison of spatio-temporal interpolation methods. In *Second International Conference on GIScience 2002*, volume 2478 of *Lecture Notes in Computer Science*, pages 145–160. Springer, 2002.

[22] L. Li and P. Revesz. Interpolation methods for spatiotemporal geographic data. *Journal of Computers, Environment and Urban Systems*, 28(3):201–227, 2004.

[23] E. J. Miller. Towards a 4D GIS: Four-dimensional interpolation utilizing kriging. In *Innovations in GIS 4: Selected Papers from the Fourth National Conference on GIS Research U.K, Ch. 13*, pages 181–197, London, 1997. Taylor & Francis.

[24] D. J. Peuquet and N. Duan. An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *International Journal of Geographical Information Systems*, 9(1):7–24, 1995.

[25] P. Revesz. *Introduction to Constraint Databases*. Springer, New York, 2002.

[26] P. Revesz and M. Cai. Efficient querying of periodic spatiotemporal objects. *Annals of Mathematics and Artificial Intelligence*, 36(4):437–457, 2002.

[27] P. Revesz and S. Wu. Spatiotemporal reasoning about epidemiological data. *Artificial Intelligence in Medicine*, 2006.

[28] D. Shepard. A two-dimensional interpolation function for irregularly spaced data. In *23nd National Conference ACM*, pages 517–524. ACM, 1968.

[29] M. F. Worboys. A unified model for spatial and temporal information. *Computer Journal*, 37(1):26–34, 1994.

[30] O. C. Zienkiewics and R. L. Taylor. *Finite Element Method, Vol. 1, The Basis*. Butterworth Heinemann, London, 2000.

[31] E. G. Zurflueh. Applications of two-dimensional linear wavelength filtering. *Geophysics*, 32:1015–1035, 1967.