GIS MODELING OF URBAN COMPONENTS TO MONITOR AND MODEL URBAN RESPIRATION

July, 2003

Mizuki Kawabata
Center for Spatial Information Science
University of Tokyo
Bunkyo-ku, Tokyo, JAPAN
Email: *mizuki@csis.u-tokyo.ac.jp*

Joseph Ferreira, Jr.
Department of Urban Studies and Planning
Massachusetts Institute of Technology
Cambridge, MA 02139-4307, USA
Email: *jf@mit.edu*

Abstract

A Geographic Information System (GIS) is increasingly recognized as an effective tool for urban modeling. As a part of NASA's multi-disciplinary project, this article develops GIS modeling methods of constructing measurements of urban components to monitor and model urban respiration. The first step is to create an Oracle database that stores spatially and temporally referenced measures of trace gases. Next, the measurements of land use, population density, roads, and wind are modeled and combined into a base grid cell matrix. Finally, spatial regression analysis is performed to estimate the effect of urban components on air quality. The usability of GIS in the context of environmental modeling is discussed.

1. Introduction

The increasing urbanization and the emergence of megacities are prompting the need for the study of *urban respiration*, the interaction between urban activities and atmospheric chemistry. Urban activities consume atmospheric oxygen and expire carbon dioxide as well as various other trace gases and small particles. Although extensive research has been done to develop strategies to reduce air pollutants, the magnitudes and spatial-temporal dimensions of gaseous and aerosol pollutants and their relation to urban activities are not well known. This limited knowledge is partly due to the difficulty in examining the large and varied contents of the urban emissions, and partly due to the difficulty in measuring urban components.

A Geographic Information System (GIS), a technology that can handle information on both location and its characteristics within a single system, is increasingly recognized as a powerful modeling tool in a variety of fields (Batty and Xie 1994a, 1994b, Sui 1998; Yeh 1999). While GIS's visualization capability is widely used, the modeling capability of GIS is relatively underutilized (Klosterman 1998; Nedovic-Budic 1998; Wegener 1998). With the increasing supply and usability of spatial data, GIS has a great potential to easily and accurately model the urban components and to systematize the modeling process.

This study demonstrates how GIS can be effectively used for monitoring and modeling urban respiration. Specifically, we develop GIS modeling methods of constructing measurements of urban components and investigate their influence on urban air quality. Our focus is on building prototype modeling methods that can be readily replicated and applied to other urban areas.

This research is a part of Urban Metabolism and Trace Gas Respiration Project, a large and multi-disciplinary project aiming to monitor and model the distribution and emission patterns of pollutants in urban areas. Funded by National Aeronautics and Space Administration (NASA), the project was taken place between February, 1997 and October, 2001. The project was conducted through a highly interdisciplinary collaboration between five research teams from the Center for Atmospheric and Environmental Chemistry at Aerodyne Research, Inc., the Department of Chemical Engineering and the Department of Urban Studies and Planning (DUSP) at the Massachusetts Institute of Technology (MIT),
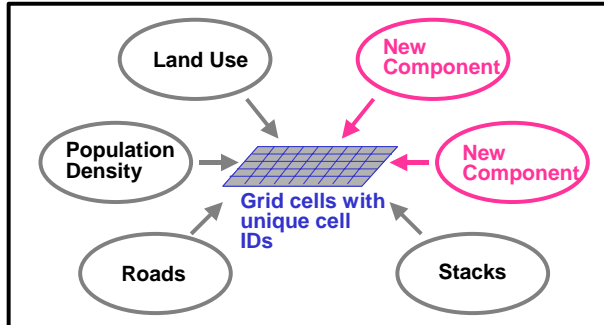
the Institute of Earth, Oceans, and Space at the University of New Hampshire, and the Laboratory for Atmospheric Research at Washington State University. This paper reports a part of the work conducted by the DUSP team with a focus on the development of GIS and spatial data infrastructures for environmental modeling.

The reminder of this paper is organized into five parts. Section 2 presents the methodological framework, and Section 3 describes how trace gas measures are collected and stored in a database. Section 4 shows how the measurements of urban components are developed and combined, and Section 5 demonstrates the modeling process for spatial regression analysis. Finally, Section 6 concludes with a discussion of the usability of GIS in the context of urban environmental modeling.

2. Methodology Overview

We develop GIS-based modeling methods that can readily replace an old component by an improved component, or to add more components to the modeling pipeline. Using GIS, we seek to construct modeling methods that can be easily replicated and applied for different urban areas.

Boston in Massachusetts, a northeastern area in the U.S., is selected for the study area. The urban components examined are land use, population density, roads, and wind. We employ a grid-based approach to the modeling process. Across the study area, a base grid cell matrix with the cell size of 200m×200m is created. The cell matrix contains 100 rows and 101 columns (10,100 cells in total), and each cell has a unique cell ID. All measurements of urban components are converted to grid cell matrices and then combined into a grid data layer, which

Figure 1 Grid-based modeling approach

serves as a base map (Figure 1). The grid-based approach makes it flexible in adding or subtracting urban components. The combined measurements are used to estimate their effect on air quality.

This study employs three applications: ArcView, Oracle, and S-PLUS. The first application, ArcView, is a widely-used GIS package, and the second application, Oracle, is an advanced package of relational database management systems (RDBMS). The third application, S-PLUS, is a statistical package that can be dynamically linked with ArcView. The use of RDBMS in the GIS modeling process improves the flexibility, repeatability, and efficiency of the data processing pipeline. The statistical package, which is coupled with GIS, helps model development.

The methodological framework is as follows. First, we load the spatially-temporally

referenced measures of multiple trace gases and particulates into an Oracle database. In order to examine the spatial and temporal distributions of trace gas measures and their relation to urban components, we create base maps that show trace gas measures on top of the layers of urban backgrounds such as orthophotos and roads. Second, measurements of land use, population density, and roads are developed and combined into a grid cell matrix. Third, we create Oracle tables that can adjust the cell values according to the directions and speed of wind. Finally, spatial regression analysis is performed to examine the effect of urban components on trace gas concentrations.

3. Creating Database for Trace Gas Measures

The first step is to create a database that stores trace gas measures. The Aerodyne team prepare a mobile laboratory that is equipped with high-end sensors, a GPS device, and a central data logging computer. Using this mobile laboratory, the Aerodyne team collected real-time (~1 second response) mobile measures of multiple trace gases and particulates in Boston metro. Figure 2 plots the GPS readings on May 23, 1999.
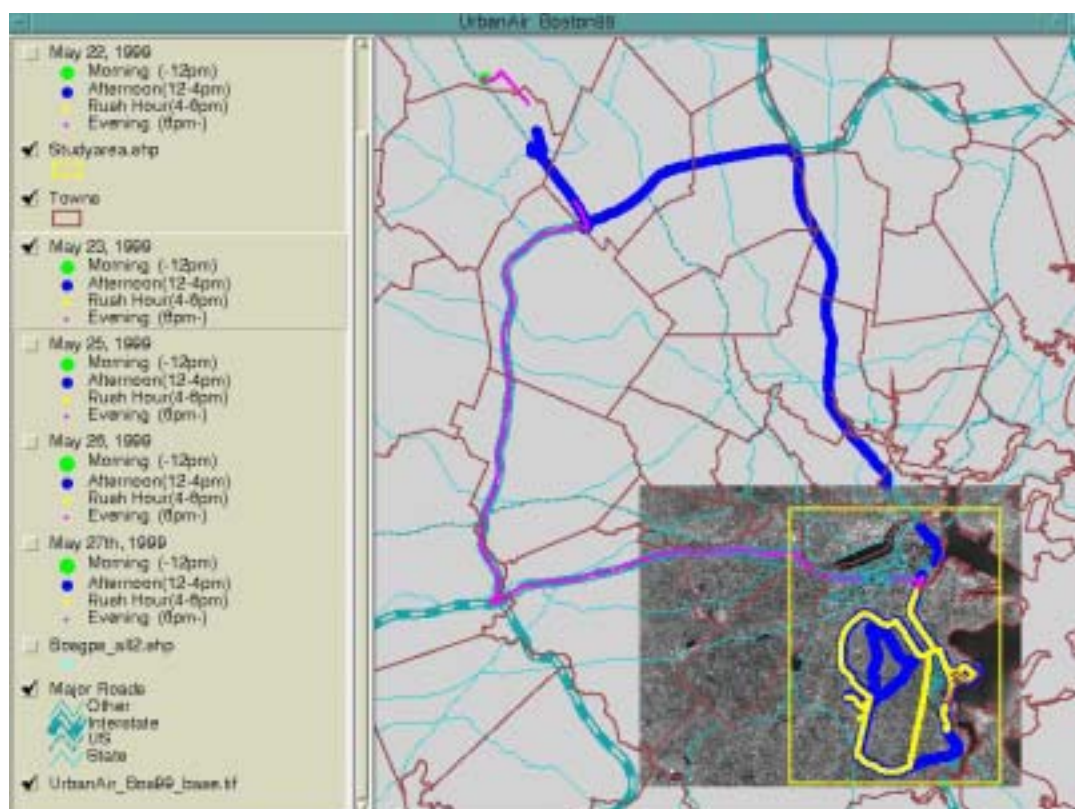

Figure 2 GPS readings in Boston metro on May 23, 1999

The Aerodyne team provided our DUSP team with their GPS readings of six types of gas measures (NO_2, NO, O_3, CO_2, uv, and sf_6) and two types of particulates (heated and unheated particles). We load these spatially-temporally referenced gas measures into

Oracle and create an database, URBANRESP.

The datasets are quite large, having tens of thousands of data records for each gas type for each observation day. Storing these vast amounts of data in the Oracle database helps us investigate, transform, and manage the datasets. The use of Oracle also facilitates streamlining the data processing. The procedures are written in Structured Query Language (SQL) to automatically check, transform, and query the data. For example, adjusting GMT time to EDT time, filtering out outliers, and detecting system anomalies are performed with SQL in Oracle.

To explore spatial and temporal distributions of trace gas measures in relation to the urban components, we import trace gas measures into ArcView and map them by gas type and time period. Figure 3 shows GPS readings of NO_2 overlaid on top of the layers of a digital orthophoto and major roads.
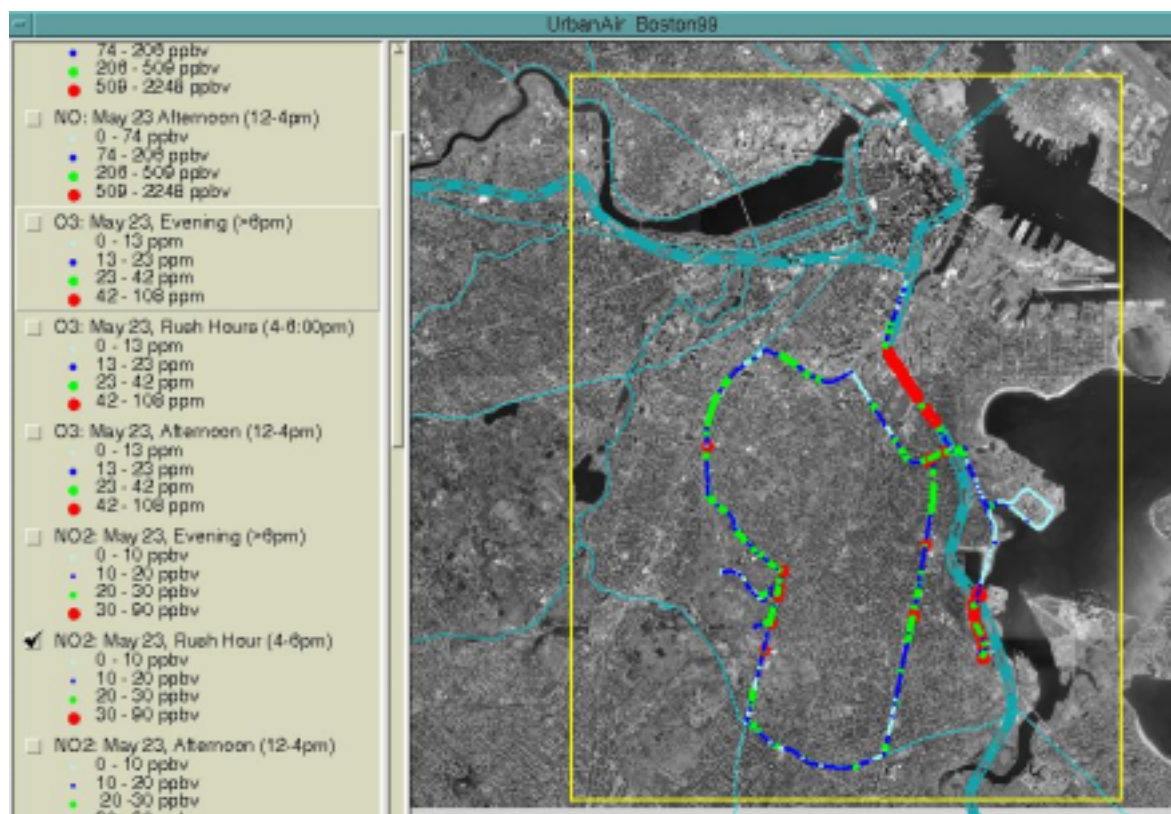

Figure 3 Spatial and temporal distributions of NO_2 measures in study area

The orthophoto is downloaded from MIT Digital Orthophoto Project (http://ortho.mit.edu/), and the major roads are obtained from the Massachusetts Geographic Information System (MassGIS), the leading GIS agency in Massachusetts (http://www.state.ma.us/mgis/). To be able to show data layers in one view, all georeferenced data including trace gas measures, digital orthophotos, and major roads are

converted to Mass State Plane coordinates (NAD 1983). Making the projection consistent across the data layers facilitates the modeling process in the subsequent sections.

4. Modeling Urban Components

The second step is to construct measurements of urban components. We develop measurements of four urban components: land use, population density, roads, and wind.[1] The objective of this section is to create a base Oracle table that contains all urban components. Employing the grid-based approach described in Section 2, we first prepare an empty grid-shaped polygon layer in which the base grid cell matrix is stored as attribute data. The grid cell size is 200m×200m, and the matrix for the study area consists of 100 rows and 101 columns, having a total of 10,100 cells. We assign a unique cell ID to each cell.
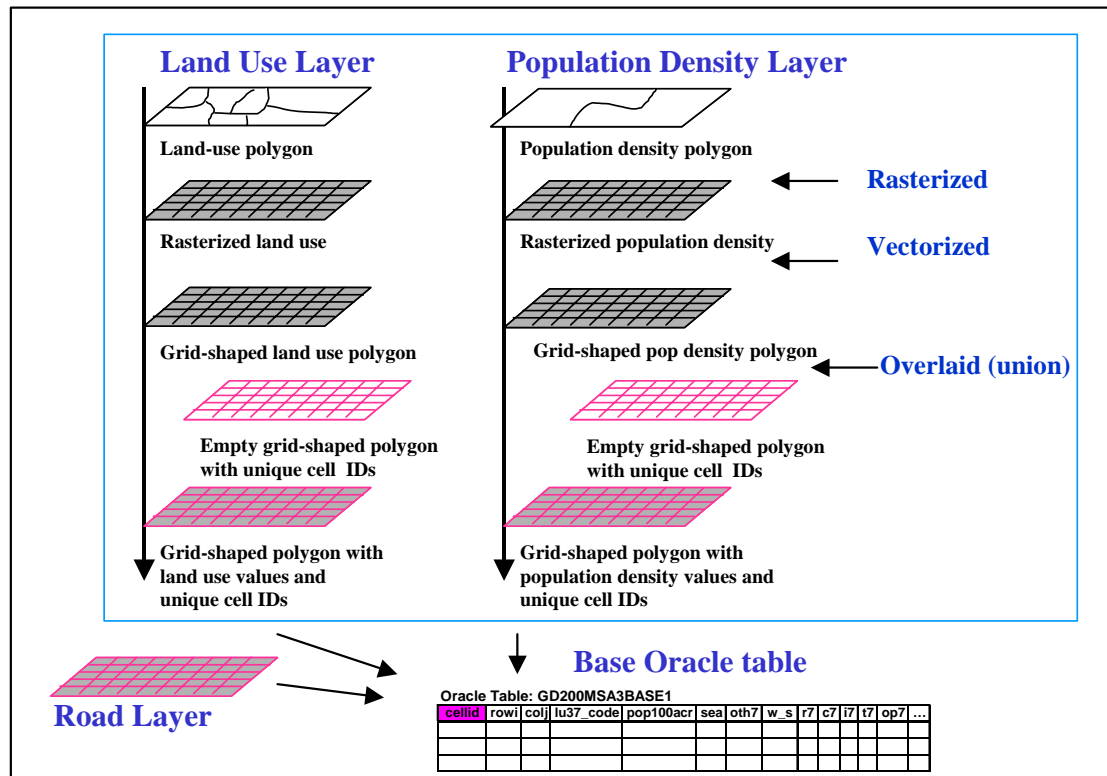
Figure 4 Diagram for developing measurements of urban components

Next, each data layer is rasterized to a grid cell layer. The rasterized layer is then vectorized and overlaid with the empty grid-shaped polygon layer to add unique cell IDs to the cell values. Finally, the cell values are imported into the URBANRESP database and combined within Oracle. The diagram of the modeling process is depicted in Figure 4.

---

1 Later we added two urban components: traffic congestion and stack emission. The development methods of the additional two components are described in Cao and Ferreira (2002).

The data processing involves the extensive use of ArcView's spatial analysis functions such as neighborhood statistics and map calculator, as well as Oracle's capabilities as a powerful RDBMS. The following subsections describe the development process in more detail.

4.1 Land Use and Population Density Models

Land use data are obtained from MassGIS, and population data are extracted from 1990 Census. Land use is classified into seven categories: residential, commercial, industrial, transportation, open space, water/water base, and others categories. Land use codes are available from MassGIS (http://www.state.ma.us/mgis/lus.htm). The geographic unit of population is the census block group, which generally contains between 250 and 550 housing units with an optimal size of 400 housing units (U.S. Census Bureau 1993). Population is divided by block group area to obtain population density (population per acre).

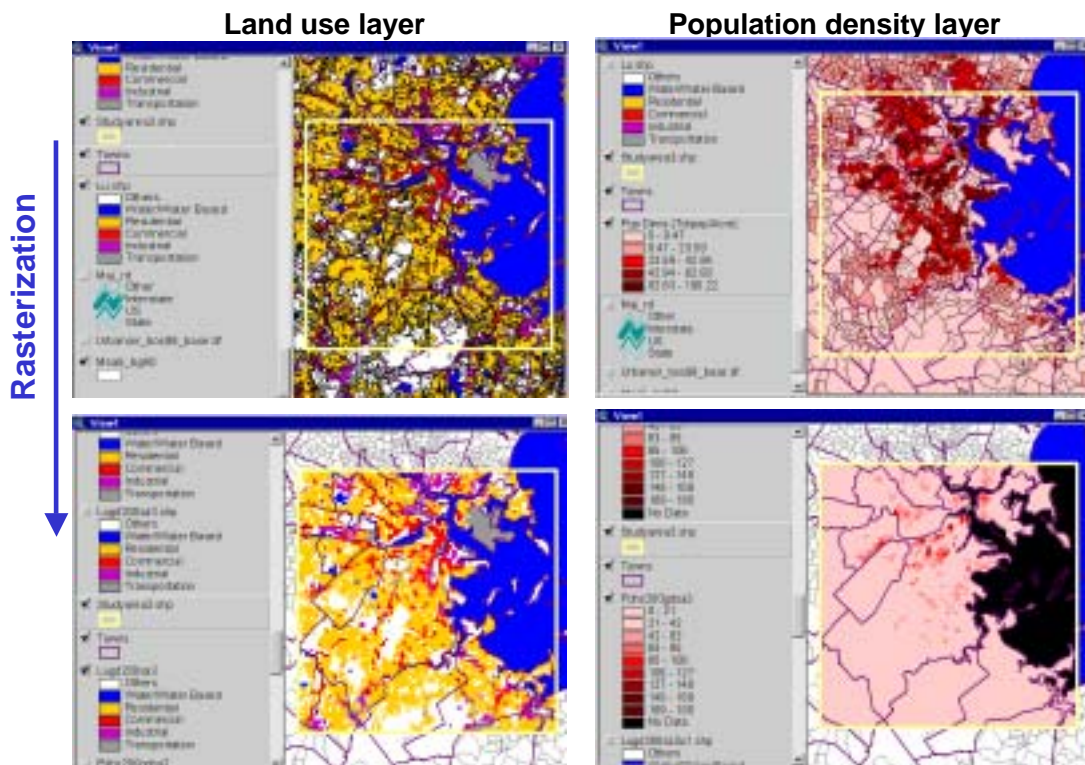**Figure 5 Rasterization of land use and population density layers**

Both of the land use and population density layers are in a polygon format, but the polygon shape (geographic boundary) differs between them. The incorporation of land use and population density data into the base grid cell matrix is conducted as follows. First, we rasterize each of the land use and population density layers within the study area. The

rasterized layer has a common grid cell matrix with a grid cell size of 200m×200m. Figure 5 displays the rasterization process of the layers, where the rasterized layer shows land use or population density at the center of each 200m×200m cell.

Second, for each cell, we generate smoothed values by calculating the percentage of land in the neighboring 25 grid cells that are of each land use type and by computing the average population density in the surrounding 25 grid cells. This process is conducted using ArcView's neighborhood statistics and map calculator functions. The geographically smoothed values are generated because when estimating the effect of land use and population density on air quality, the smoothed values, rather than the original values created from the rasterization process, would represent better variables.

Third, each rasterized layer is vectorized and overlaid with the empty grid-shaped polygon layer with unique cell IDs to create a grid-shaped polygon layer that has unique cell IDs as well as the values of land use or population density. Finally, the cell values are stored in the URBANRESP database.
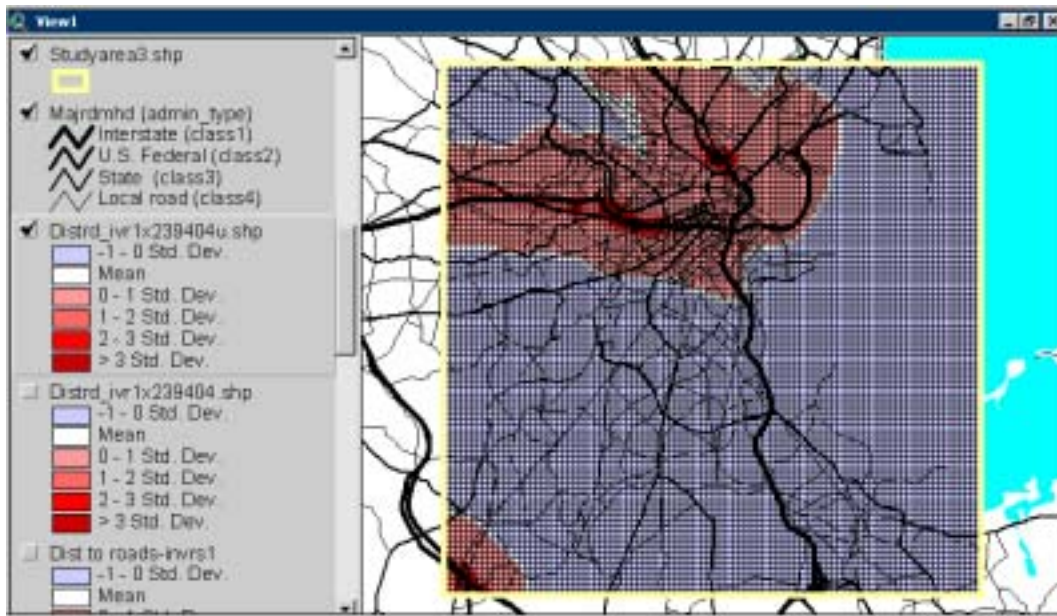

4.2 Road Model
For the third urban component, roads, we develop measurements indicating nearness to roads. Data on road networks are provided by MassGIS. Assuming that larger roads have a greater influence on air quality than do smaller roads, we model the nearness-to-road measurements as follows.

First, the following four separate line layers are created based on the road type: the interstate road layer, U.S. Federal road layer, state road layer, and local road layer. Second, we rasterize each of the four line layers by assigning the value of distance to roads to each cell. Third, we combine the four rasterized layers in a way that each cell has the value of weighted total distance that puts more weights on larger roads and less weights on smaller roads. Alternative specifications using different weights are developed to examine sensitivity. Fourth, we take the inverse of the weighted total distance, assuming that the pollution level is inversely proportional to distance to roads. Fifth, we normalize the inversed values, in order to obtain comparable estimates from spatial regression models. Figure 6 shows a sample nearness-to-road measurement by the standard deviation classification. As expected, areas with dense networks of larger roads have high levels of nearness-to-road measures.

Finally, as in the case for land use and population density, the grid layers of the nearness-to-road measurements are vectorized, combined, and overlaid with the empty grid-shaped polygon layer to create a data layer that contains all the road measurements as well as unique cell IDs. The cell values are then imported into the URBANRESP database.

Figure 6 A sample nearness-to-road measurement

4.3 Wind Model

The fourth urban component is wind, which is likely to affect atmospheric chemical measures. Using SQL, we develop an Oracle table, WINDSHIFT, which shifts the grid cell values according to a direction and speed of wind. The SQL scripts to develop the WINDSHIFT table are shown in Appendix A. Corresponding to the base grid cell matrix, the WINDSHIFT table has unique cell IDs, row numbers, column numbers, and new cell IDs. The new cell IDs are assigned according to eight wind directions and three wind speeds. The table can be used to generate wind-adjusted values. For example, the land use values can be shifted 1, 2, or 3 grid cells in the direction of wind.

4.4 Combining Trace Gases and Urban Components

The final step is to combine trace gas measures with the measurements of urban components. Before the combining process is carried out, trace gas measures are averaged for each cell as shown in Figure 7. First, we perform spatial join using the point layer of trace gas measures and the grid-shaped polygon layer with unique cell IDs to create a point layer that contains both trace gas measures and unique cell IDs. Next, the attribute table is loaded into the Oracle database. We then calculate the average trace gas measures per cell using SQL and store the results in a new table.
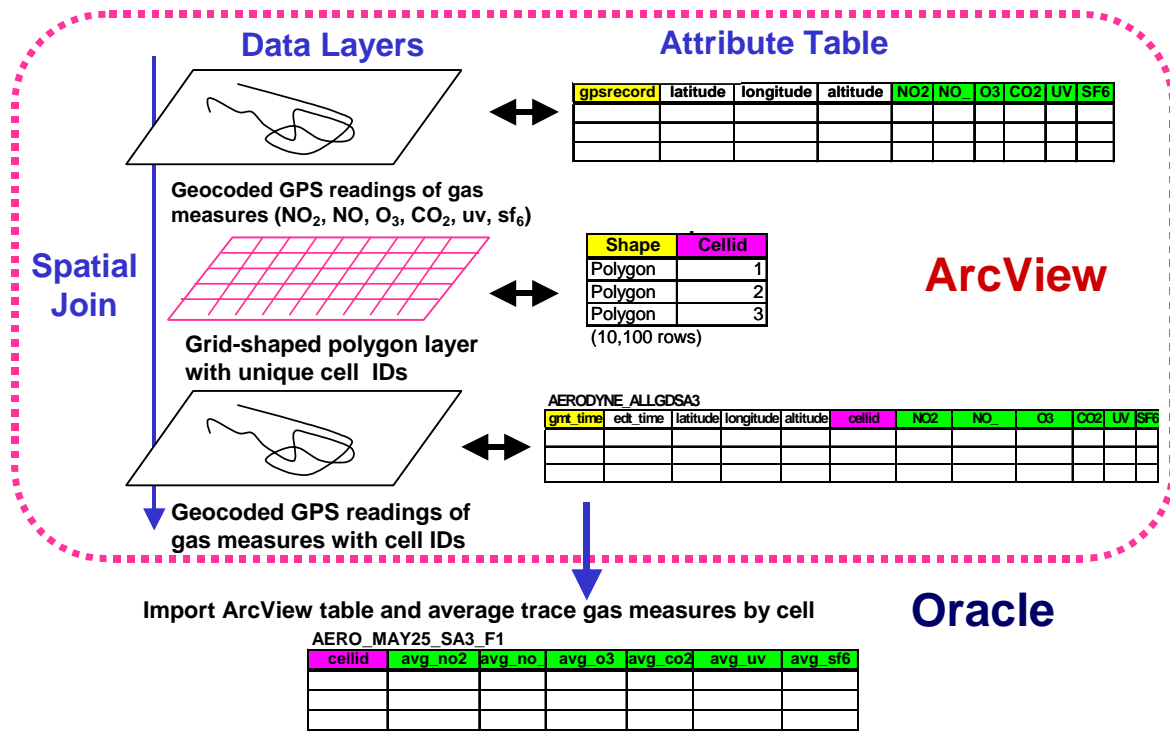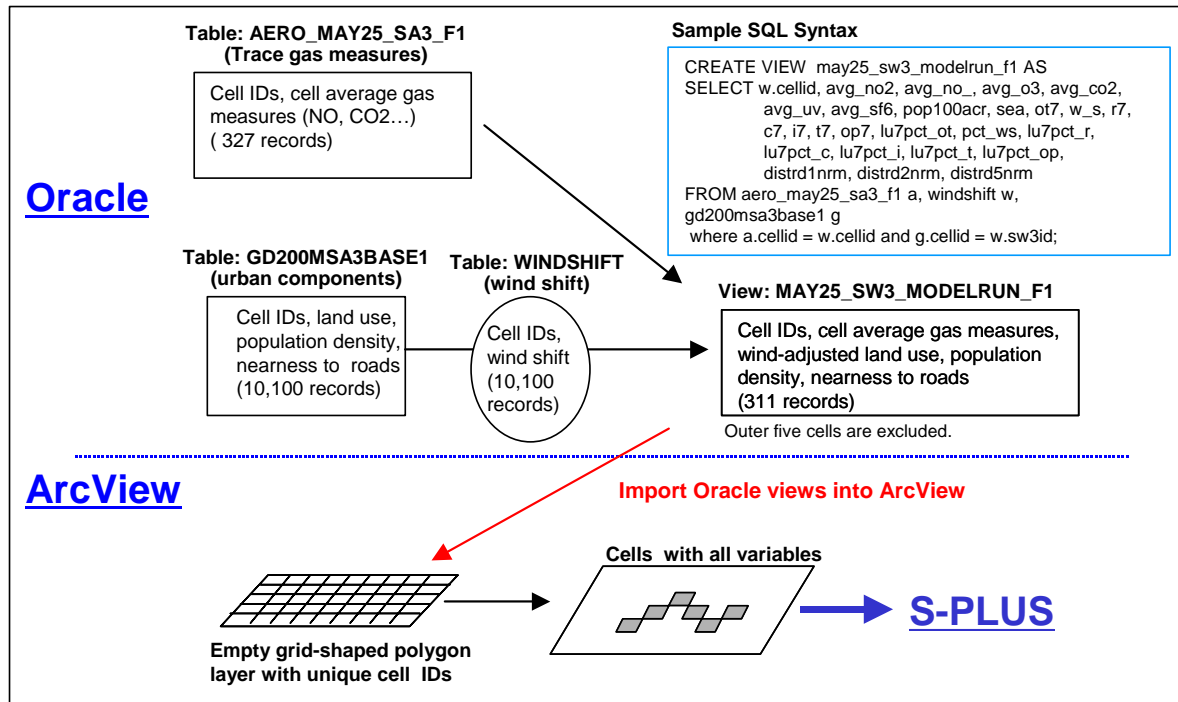
Figure 7 Averaging trace gas measures by cell

Finally, the average trace gas measures and urban component are joined. Joining the variables in the Oracle side, rather than in the ArcView side, offers a great deal of flexibility and repeatability in the data processing. For instance, we can easily redo the process when parameters are changed or when new components are added.

To prepare the data for spatial regression, we create a polygon layer that includes all the variables. Specifically, for each wind direction and speed, an Oracle table that has trace gas measures and wind-adjusted measures of urban components is created. A table of our interest, then, is imported into ArcView and joined with the grid-shaped polygon layer to create a layer that contains both cell-averaged trace gas measures and wind-adjusted values of all urban components. Finally, the cell values are exported to S-PLUS for spatial regression analysis. Figure 8 shows the diagram for the data preparation process.

**Table: AERO_MAY25_SA3_F1
(Trace gas measures)**

Cell IDs, cell average gas
measures (NO, CO2…)
(327 records)

Oracle

Sample SQL Syntax

```
CREATE VIEW  may25_sw3_modelrun_f1 AS
SELECT w.cellid, avg_no2, avg_no_, avg_o3, avg_co2,
       avg_uv, avg_sf6, pop100acr, sea, ot7, w_s, r7,
       c7, i7, t7, op7, lu7pct_ot, pct_ws, lu7pct_r,
       lu7pct_c, lu7pct_i, lu7pct_t, lu7pct_op,
       distrd1nrm, distrd2nrm, distrd5nrm
FROM aero_may25_sa3_f1 a, windshift w,
gd200msa3base1 g
 where a.cellid = w.cellid and g.cellid = w.sw3id;
```

**Table: GD200MSA3BASE1
(urban components)**

Cell IDs, land use,
population density,
nearness to roads
(10,100 records)

**Table: WINDSHIFT
(wind shift)**

Cell IDs,
wind shift
(10,100
records)

View: MAY25_SW3_MODELRUN_F1

Cell IDs, cell average gas measures,
wind-adjusted land use, population
density, nearness to roads
(311 records)

Outer five cells are excluded.

ArcView

Import Oracle views into ArcView

Cells with all variables

S-PLUS

**Empty grid-shaped polygon
layer with unique cell IDs**

Figure 8 Combining all variables for spatial regression analysis

5. Performing Spatial Regression Analysis

Spatial regression is performed to estimate the effect of urban components on trace gas concentrations. We employ spatial regression, instead of ordinary least squares (OLS) regression, because spatial regression can take into account the likely existence of spatial autocorrelation, which would yield biased estimates for OLS regression. Since the focus of this paper is not on regression results but on how GIS can be used for environmental modeling, this section presents an example of the modeling process.

A sample case is to model NO values as a function of urban components. We use ArcView as a tool for data exploration and mapping, and use S-PLUS as a tool for spatial regression. Figure 9 displays the cell-averaged observed measures of NO on May 25, 1999. The NO measures are shaved data that exclude outliers.

The visualization capability of GIS helps us explore the data, which contributes to the advancement of the modeling process including the development of hypotheses about the spatial regression estimates. Such data driven analysis is often called *exploratory spatial data analysis*. By viewing the spatial patterns of NO measures over the urban components, we can elaborate a priori consideration for the regression results; for example, industrial land use would be likely to increase NO values.
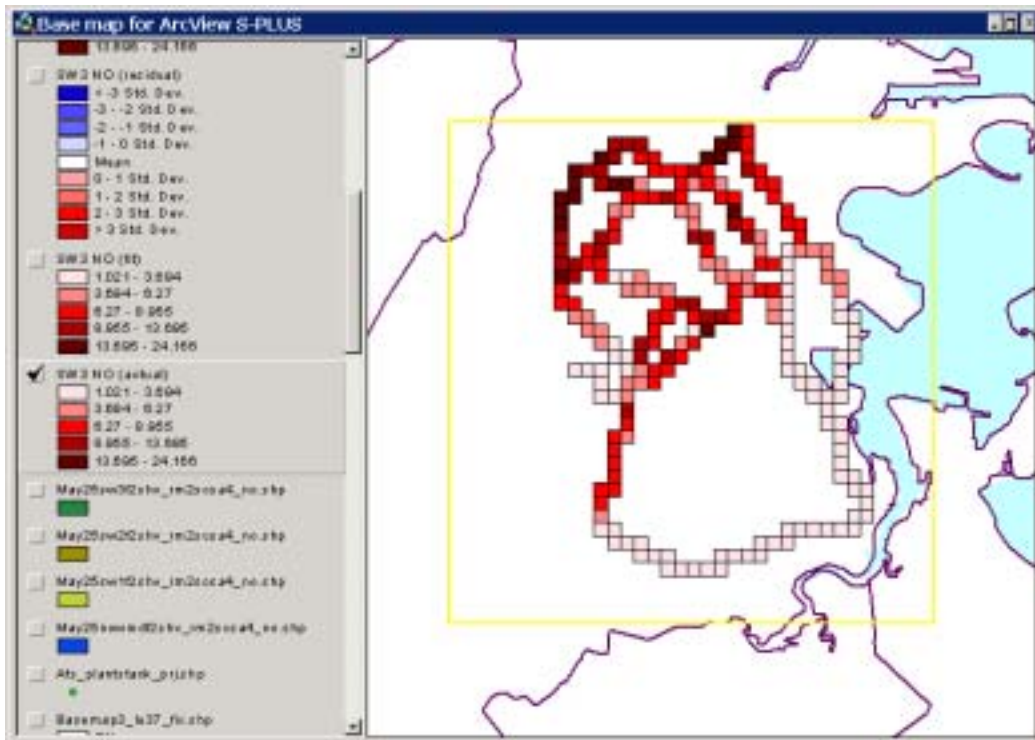
Figure 9 Average NO measures by 200m×200m grid cell

The dynamic link between ArcView and S-PLUS greatly facilitates the statistical modeling process. The ArcView extension for Spatial Statistics sets up the dynamic link with S-PLUS, enabling us to directly export the grid cell values to S-PLUS for statistical analysis and import the results back into ArcView for visualization.

In the sample case presented here, the cell values are exported to S-PLUS, where numerous model specifications that regress NO values against a function of urban components are tested. Estimated results from some of the best model specifications are then imported back into ArcView and mapped for examining the estimated results.

An example is shown in Figure 10, which displays NO residual values (fitted values minus observed values) by the standard deviation classification. It is easy to understand the locations and patterns of overestimated or underestimated cells, which help us develop the model specification.
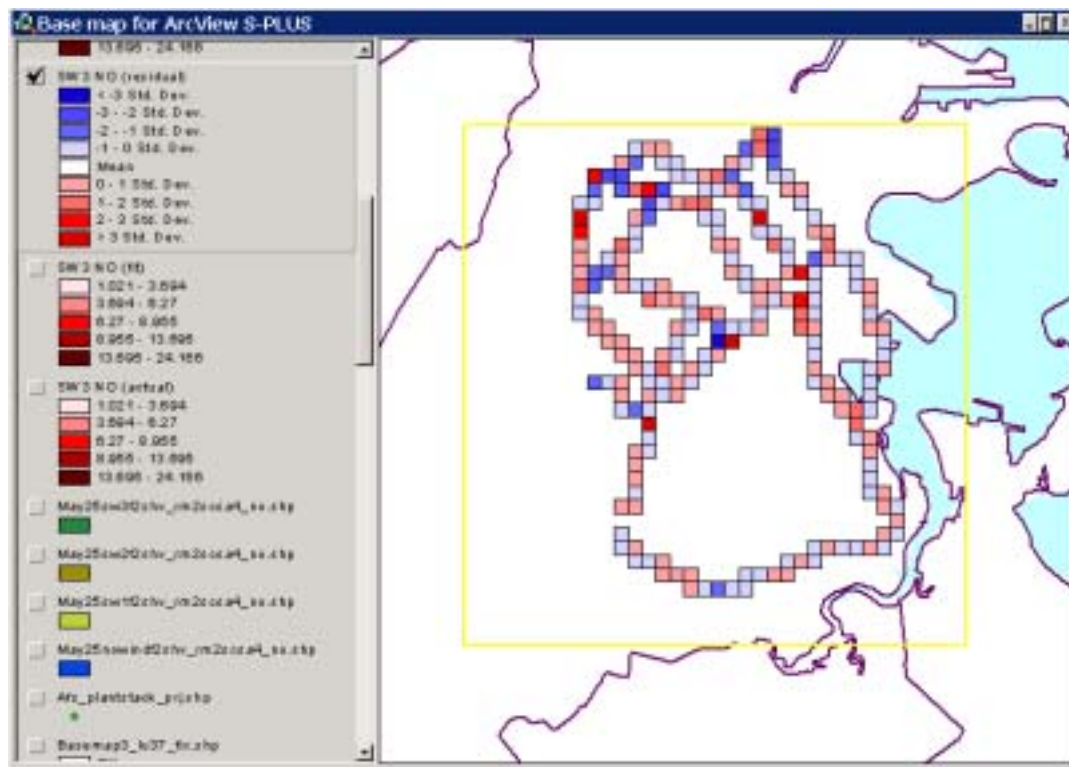
Figure 10 NO residual values by standard deviation classification

6. Conclusions

Selecting Boston as the study area, this paper presented how GIS could be an effective tool to develop measurements of urban components in the context of urban environmental modeling. The work was our venture to build prototype GIS modeling methods that could be repeatable and applicable to other urban areas. The modeling process exhibited some characteristic features.

The first feature is the use of the grid-based approach. All layers of the urban components were converted into a common base grid cell matrix, making the combining process straightforward and computationally manipulative. The modeling process was greatly facilitated by GIS's spatial analysis functions such as rasterization and map calculator.

The second feature is the coupling of GIS and RDBMS, which brought much flexibility and repeatability into the modeling process. Loading the copious amounts of trace gas observations into the Oracle database helped and fastened the data processing. Such procedures as storing, transforming, and checking the data were written in SQL, which could automate the procedures. Complex filtering and querying of the data were performed in Oracle, and the results were imported into ArcView for visualization and analysis. The measurements of urban components created with ArcView were joined in Oracle, and then selected datasets were imported back into ArcView to create a base layer

that included both trace gas measures and urban components.

The third feature is the use of the dynamic link between GIS and statistics packages, which highlighted the GIS's usability for exploratory spatial data analysis. Viewing a layer of trace gas measures on top of the layers of urban components helped us develop hypotheses about regression results. The cell values stored in ArcView were directly exported to S-PLUS to run spatial regression models that estimate the influence of urban components on trace gas concentrations. Some of the best specifications were imported back into ArcView for visualization, which helped us improve the model specification. For example, we could compare the spatial patterns of observed and fitted gas measures by investigating the spatial distributions of residual values.

The GIS modeling methods we proposed could serve as templates for more realistic and complex models. The urban components developed in this study were land use, population density, roads, and wind. Each of these components can be refined. The road component, for example, can be improved to take into account traffic congestion levels, in addition to the levels of nearness to roads. A number of other components, such as surface roughness and landfills, are also likely to affect air quality. The proposed modeling framework can readily replace an old component by an improved component or to add new components, and to rerun the statistical models with changed components. The statistical models would be improved by using better-specified variables. Incorporating interactions among the variables would also improve the models. Once a realistic model is developed, we can use the estimated parameters to interpolate gas concentrations throughout the surface of the urban area.

While this study focused on Boston, the modeling framework can be applied to other metropolitan areas. The measurements of urban components were constructed using readily available data, and much of them were available online. Similar data are obtainable for other urban areas, and new datasets are coming online. With these datasets, the GIS-based modeling methods could be applied for other metropolitan areas in a reasonably standardized way.

The advancement of new technology has a great potential to streamline the modeling process. The current GIS functions were too limited to completely automate the modeling process. We experimented with ModelBuilder, ArcView's new tool for building spatial models, but it was too inflexible in handling and automating the complex modeling process. To build an automated data process pipeline, a great deal of custom programming would be needed. Given the rapid technological development, however, powerful and flexible modeling tools will soon be available, which would make the modeling process much faster and more repeatable.

References

Batty, M. and Xie, Y. 1994a. "Modeling Inside GIS: Part 1. Model Structures, Exploratory Spatial Data Analysis and Aggregation." *International Journal of Geographic Information*

Systems 8(3): 291-307.

Batty, M. and Xie, Y. 1994b. "Modeling Inside GIS: Part 2. Selecting and Calibrating Urban Models Using ARC/INFO." *International Journal of Geographic Information Systems* 8(5): 451-470.

Cao, L. and Ferreira, J. Jr. 2002. "Integrating GIS and RDBMS to Model Traffic Congestion and Urban Air Pollutants." *Proceedings of Urban and Regional Information Systems Association 39th Annual Conference*. Long Beach, CA, USA.

Environmental Systems Research Institute (ESRI). 2001. *Dictionary of GIS Terminology*. Redlands, California: ESRI Press.

Klosterman, R. 1998. "Computer Applications in Planning." *Environment and Planning B* Anniversary Issue: 32-36.

Nedovic-Budic, Z. 1998. "The Impact of GIS Technology." *Environment and Planning B* 25(5): 681-692.

Sui, D. Z. 1998. "GIS-Based Urban Modeling: Practices, Problems, and Prospects." *International Journal of Geographical Information Science* 12(7): 651-671.

U.S. Census Bureau. 1993a. *1990 Census of Population and Housing: Public Use Microdata Samples United States Technical Documentation*. Washington, DC: U.S. Census Bureau.

Wegener, M. 1998. "GIS and Spatial Planning." *Environment and Planning B* Anniversary Issue: 48-52.

Yeh, A. G-O. 1999. "Urban Planning and GIS." in Longley, P. A., et. al. eds. *Geographical Information Systems: Principles and Technical Issues*. 2nd Edition. Vol. 2. John Wiley & Sons. 877-887.

Appendix A: SQL Scripts to create WINDSHIFT table

/* WINDMODEL.SQL - Script to build WINDSHIFT table that computes new grid cell average landuse values for use with each of 8 wind directions and 3 wind speeds. Note that new cell IDs taking account of wind = 0 for rowi <6 or rowi>95 or colj <6 or colj>96. */

```
CREATE TABLE windhold (
  nn1id      integer,
  ne1id      integer,
  ee1id      integer,
  se1id      integer,
  ss1id      integer,
  sw1id      integer,
  ww1id      integer,
  nw1id      integer,
  nn2id      integer,
  ne2id      integer,
  ee2id      integer,
  se2id      integer,
  ss2id      integer,
  sw2id      integer,
  ww2id      integer,
  nw2id      integer,
  nn3id      integer,
  ne3id      integer,
  ee3id      integer,
  se3id      integer,
  ss3id      integer,
  sw3id      integer,
  ww3id      integer,
  nw3id      integer);

INSERT INTO windhold values(
  0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0);


CREATE TABLE windshift AS
SELECT cellid, rowi, colj, cellid nowind,
       nn1id, ne1id, ee1id, se1id, ss1id, sw1id, ww1id, nw1id,
       nn2id, ne2id, ee2id, se2id, ss2id, sw2id, ww2id, nw2id,
       nn3id, ne3id, ee3id, se3id, ss3id, sw3id, ww3id, nw3id
 from windhold w, GD200MSA3 g;

drop table windhold;

create index windcell on windshift(rowi,colj);


/* Update according to the wind direction & strength */

UPDATE windshift
SET nowind = 0
```

```
where (windshift.rowi   between 1 and 5 or
windshift.rowi between 96 and 100)
    or (windshift.colj   between 1 and 5 or
windshift.colj between 97 and 101);


UPDATE windshift
SET nn1id =
     (SELECT cellid
        FROM windshift w
        WHERE w.rowi = windshift.rowi - 1
          AND w.colj = windshift.colj )
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET nn2id =
     (SELECT cellid
        FROM windshift w
        WHERE w.rowi = windshift.rowi - 2
          AND w.colj = windshift.colj )
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET nn3id =
     (SELECT cellid
        FROM windshift w
        WHERE w.rowi = windshift.rowi - 3
          AND w.colj = windshift.colj )
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;


UPDATE windshift
SET ne1id =
     (SELECT cellid
        FROM windshift w
        WHERE w.rowi = windshift.rowi - 1
          AND w.colj = windshift.colj + 1)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET ne2id =
     (SELECT cellid
        FROM windshift w
        WHERE w.rowi = windshift.rowi - 2
          AND w.colj = windshift.colj + 2)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;
```

```
UPDATE windshift
SET ne3id =
    (SELECT cellid
       FROM windshift w
       WHERE w.rowi = windshift.rowi - 3
       AND w.colj = windshift.colj + 3)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

select cellid, rowi, colj, nowind, ne1id, ne2id, ne3id
  from windshift
  order by cellid;

UPDATE windshift
SET ee1id =
    (SELECT cellid
       FROM windshift w
       WHERE w.rowi = windshift.rowi
       AND w.colj = windshift.colj + 1)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET ee2id =
    (SELECT cellid
       FROM windshift w
       WHERE w.rowi = windshift.rowi
       AND w.colj = windshift.colj + 2)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET ee3id =
    (SELECT cellid
       FROM windshift w
       WHERE w.rowi = windshift.rowi
       AND w.colj = windshift.colj + 3)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

select cellid, rowi, colj, nowind, ee1id, ee2id, ee3id
  from windshift
  order by cellid;

UPDATE windshift
SET se1id =
    (SELECT cellid
       FROM windshift w
       WHERE w.rowi = windshift.rowi + 1
       AND w.colj = windshift.colj + 1)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET se2id =
    (SELECT cellid
       FROM windshift w
       WHERE w.rowi = windshift.rowi + 2
       AND w.colj = windshift.colj + 2)
```

```
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET se3id =
    (SELECT cellid
       FROM windshift w
       WHERE w.rowi = windshift.rowi + 3
       AND w.colj = windshift.colj + 3)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

select cellid, rowi, colj, nowind, se1id, se2id, se3id
  from windshift
  order by cellid;

UPDATE windshift
SET ss1id =
    (SELECT cellid
       FROM windshift w
       WHERE w.rowi = windshift.rowi + 1
       AND w.colj = windshift.colj )
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET ss2id =
    (SELECT cellid
       FROM windshift w
       WHERE w.rowi = windshift.rowi + 2
       AND w.colj = windshift.colj )
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET ss3id =
    (SELECT cellid
       FROM windshift w
       WHERE w.rowi = windshift.rowi + 3
       AND w.colj = windshift.colj )
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET sw1id =
    (SELECT cellid
       FROM windshift w
       WHERE w.rowi = windshift.rowi + 1
       AND w.colj = windshift.colj - 1)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET sw2id =
    (SELECT cellid
       FROM windshift w
       WHERE w.rowi = windshift.rowi + 2
```

```
        AND w.colj = windshift.colj - 2)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET sw3id =
    (SELECT cellid
        FROM windshift w
      WHERE w.rowi = windshift.rowi + 3
        AND w.colj = windshift.colj - 3)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;


select cellid, rowi, colj, nowind, sw1id, sw2id, sw3id
  from windshift
  order by cellid;


UPDATE windshift
SET ww1id =
    (SELECT cellid
        FROM windshift w
      WHERE w.rowi = windshift.rowi
        AND w.colj = windshift.colj -1)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET ww2id =
    (SELECT cellid
        FROM windshift w
      WHERE w.rowi = windshift.rowi
        AND w.colj = windshift.colj -2)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;
```

```
UPDATE windshift
SET ww3id =
    (SELECT cellid
        FROM windshift w
      WHERE w.rowi = windshift.rowi
        AND w.colj = windshift.colj -3)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;


UPDATE windshift
SET nw1id =
    (SELECT cellid
        FROM windshift w
      WHERE w.rowi = windshift.rowi -1
        AND w.colj = windshift.colj -1)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET nw2id =
    (SELECT cellid
        FROM windshift w
      WHERE w.rowi = windshift.rowi -2
        AND w.colj = windshift.colj -2)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;

UPDATE windshift
SET nw3id =
    (SELECT cellid
        FROM windshift w
      WHERE w.rowi = windshift.rowi -3
        AND w.colj = windshift.colj -3)
where windshift.rowi   between 6 and 95
  and windshift.colj   between 6 and 96;
```