Computerized Simulations for Geography Instruction: Sense of Place

by

David T. Hines

Project submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science

APPROVED:

Dr. Clifford A. Shaffer, Chairman

Dr. Lawrence W. Carstensen

Dr. Mary Beth Rosson

May, 1995

Blacksburg, Virginia

Computerized Simulations for Geography Instruction: Sense of Place

by

David T. Hines

Committee Chairman: Dr. Clifford A. Shaffer

Computer Science

(ABSTRACT)

Sense of Place is a **Project GeoSim** educational software module focusing on U.S. states and counties. A comprehensive statistical database is combined with a set of simple visualization techniques to allow geography students to gain an understanding of the characteristics of states and counties. Students can examine data for any one of the statistical variables in the database. In addition, students can create *metrics* consisting of target values for several variables, and rate counties or states by how closely their data match the targets. Symbolic mapping is used to illustrate spatial characteristics of a state or county, such as its size, shape, and location, as well as regional patterns among different states and counties. Graphing is used to illustrate basic concepts of data distributions and to depict the distributions of specific variables. A spreadsheet is used to view the multivariate data sets created by user metrics. These visualization techniques and the graphical user interface tools required to control them are incorporated into interchangeable "views". This approach allows additional views to be added to **Sense of Place** in the future.

ACKNOWLEDGEMENTS

First and foremost, I want to thank my parents for their undying support and belief in my abilities. My desire to excel derives from my admiration of their accomplishments. Thank you with all my heart. I love you.

I extend my thanks to my advisor, Dr. Cliff Shaffer, for his invitation to join the outstanding Project GeoSim team, and for his expert and patient guidance throughout the life of this project.

I also want to thank Dr. Bill Carstensen for his guidance and enthusiasm about this project. I hope that I have made a small contribution to his efforts to enhance the teaching of geography.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

This report describes **Sense of Place**, an interactive simulation program developed for use in geography courses. **Sense of Place** was developed as part of **Project GeoSim**, a joint effort of the Departments of Geography and Computer Science at Virginia Tech. The original concept for **Sense of Place** is due to Dr. Clifford Shaffer in Computer Science and Dr. Lawrence Carstensen in Geography. Dr. Shaffer also developed a prototype for **Sense of Place**, which was the basis for the design and development performed by the author. The remainder of the Introduction consists of four sections. § 1.1 describes **Project GeoSim** and its goals. § 1.2 introduces **Sense of Place** and discusses its general goals. The discussion of goals is continued in more detail in § 1.3, which describes the characteristics of geographical data which we have tried to illuminate. Finally, § 1.4 compares **Sense of Place** to *Places Rated Almanac*, a printed volume that rates metropolitan areas.

1.1 Project GeoSim

Project GeoSim is a multidisciplinary effort involving members of the Departments of Geography and Computer Science at Virginia Tech to develop computer-aided instruction (CAI) modules for teaching introductory geography to first year college students [3]. A **Project GeoSim** instruction module typically consists of two parts: a multimedia tutorial of background information about the geography concepts involved in the lesson, and a simulation program that allows students to apply, experiment with, and enhance the knowledge they have gained.

GeoSim's simulation programs are developed with the **GeoSim Interface Library**

(**GIL**). **GIL** was developed by **Project GeoSim** in order to allow us to use the same application source code on the major computer platforms available to geography classrooms: MS-DOS (versions 3.3 and up), Macintosh (System 7), and UNIX workstations (DECstations and SPARCstations running the X-Window System). In addition to various graphics and interface routines, **GIL** provides an *interface event handler*, which distributes user mouse and keyboard input to the appropriate application *callback functions.* With these tools, **GIL** also supports user interface consistency across all **Project GeoSim** modules[1].

1.2 Sense of Place

The main goal of **Sense of Place** is to allow users to gain a sense of the characteristics of geographical *divisions* of the United States by examining statistical data via a variety of simple data visualization techniques. We use the term geographical division to refer to any area — e.g., a county — for which statistical data are collected, recorded, and archived. Currently, **Sense of Place** makes use of three visualization techniques: choroplethic mapping, plotting statistical distributions, and tabular representation. Each of these techniques is described in Chapter 2. However, **Sense of Place** is also designed so that other techniques may be added in the future, as described in § 6.1.

Each visualization technique is incorporated into a *view.* There are typically two views on screen at a time. In short, a view comprises a particular visualization technique and the user interface tools associated with it. The services these tools provide include easy ways of moving from one view to another. For example, once a user finds the symbol for a division in one view, he or she can click on it to find that division in the accompanying view.

The visualization tools in **Sense of Place** can be used to evaluate divisions in terms of either raw statistical data, or user-created *metrics.* A metric is a statistical standard composed of target values for a set of variables. **Sense of Place** determines a score for each division based on how close its values are to the user's targets. Metrics can be created for a variety of purposes. For example, a user can create a metric that specifies the characteristics of his or her "dream" home or travel destination, or a metric that identifies the important qualities of a successful business location. § 4.5 describes the algorithm used to determine a division's *match score*.

Two levels of geographic division are considered in **Sense of Place**: states and counties (or county equivalents, i.e., divisions which are treated as counties for statistical purposes). We do not consider cities per se, although it may appear that we do, since some counties share their names with familiar cities (Los Angeles County, for example). Cities which are politically independent of any county are treated as county equivalents, since their data would otherwise go uncounted. This is true, for example, of all cities in Virginia. In addition, Washington, D.C. is considered alongside both counties and states, since it does not belong to any state nor any county.

1.3 Characteristics of Geographical Data

Each visualization technique used in **Sense of Place** was chosen to show certain aspects of the data associated with geographical divisions. Some of these aspects are inherently geographical. For instance, geography is unique among disciplines in its concentration on the spatial characteristics of data. Other aspects of geographical data are common to other disciplines. For example, the distributions associated with variables are of interest to all fields of statistical study. It would be difficult to present all of these aspects with one data representation technique. Therefore, we have incorporated several visualization tools into **Sense of Place**: maps, distribution plots and tables. Chapter 2 describes how these techniques were chosen to illustrate different data characteristics. § 2.3 describes how the mapping technique used in **Sense of Place** provides insight into certain spatial characteristics of geographical data. § 2.4 discusses how distribution plots are used to illustrate some general concepts of statistical variables, as well as specific characteristics of the variables used in **Sense of Place**. Finally, § 2.5 describes how tabular representation is used to allow users to examine *multivariate data*, i.e., data sets whose elements are composed of information from multiple variables. A metric created by a user for evaluating and ranking counties or states provides a "template" for a multivariate data set. In other words, the metric defines the different variables measured for each "element" (county or state).

1.4 Relationship of Sense of Place and *Places Rated Almanac*

Anyone familiar with *Places Rated Almanac* might notice that there are some similarities between this regularly published resource and **Sense of Place.** There are also some important differences, however. These differences arise not only because of the different media (computer program versus printed book), but also because of the intended audiences of the respective products. *Places Rated Almanac* is "meant for people who are mulling over a relocation as well as for anyone who enjoys finding out about cities and towns and what they have to offer"[2]. **Sense of Place** hopes to satisfy this audience as well, but its main audience consists of students in geography courses, who will want to characterize areas in many different ways, not just in terms of a "good" place to live or visit. Among the differences between the two products are:

- Geographical Areas Considered: *Places Rated Almanac* focuses on metropolitan areas, which are the most likely relocation destinations. **Sense of Place** considers all states and counties, since all are important in the context of a geography course.

- Representation of Data: *Places Rated Almanac* relies mainly on tabular data representation techniques. In addition to tables, **Sense of Place** offers graphical representation in the form of maps and cartesian plots for every variable and conceivable metric.

- Metrics: The authors of *Places Rated Almanac* use their own tried-and-true metric. **Sense of Place** allows users to create their own metrics.

The remainder of this report is structured as follows. Chapters 2, 3, and 4 discuss visualization techniques, user interface issues, and functional design of **Sense of Place**, respectively. Chapter 5 describes the modifications made to the **GeoSim Interface Library** to support **Sense of Place**. Chapter 6 discusses possible future developments to the program.

Chapter 2

Visualization Techniques

This chapter describes the educational goals for **Sense of Place**, the visualization techniques utilized to meet these goals, and the way these techniques are combined with direct manipulation interface features to create a visualization tool that we refer to as a *view*. § 2.1 focuses on the educational goals and the techniques selected to fulfill them. § 2.2 introduces the concept of a view, and elaborates on each visualization technique in the context of the view into which it is incorporated.

2.1 Selecting Visualization Techniques

The primary goal of **Sense of Place** is to help students learn about various characteristics and relationships of states and counties. A secondary goal is to teach some basic concepts regarding data distributions via hands-on interaction with substantial data sets. We selected visualization techniques which we felt best revealed the variety of characteristics and concepts that we wanted to feature. § 2.3 describes some of the spatial data characteristics we wanted to depict, and the mapping technique used to represent them. § 2.4 discusses the benefits of examining data distributions, and how **Sense of Place** allows users to do so. Finally, § 2.5 focuses on the difficulty of analyzing *multivariate* data — data which reflects multiple qualities of an individual county or state — and the tool in **Sense of Place** used to facilitate such analysis.

2.2 Sense of Place View Structure

Each visualization technique used in **Sense of Place** is incorporated into a *view*. A view encompasses both the visualization technique employed and the interface features used to manipulate it. *View type* might be a more appropriate term, since future development of **Sense of Place** could allow for multiple instances of the same type of view. Since **Sense of Place** currently contains only one instance of each view type, the terms *view* and *view type* are considered equivalent here. See § 6.2 for a discussion of multiple instances of a view.

Each view is contained in a **GIL** window, a rectangular region of the screen within which other interface elements may be grouped[8]. Note that **GIL** windows are a distinct type of window from the *application window* used to display **GIL** applications on some platforms. The application window is controlled by the operating system or a window manager program, not by **GIL**. **GIL** windows appear inside the application window. All view windows are the same size in order to allow any view to fit into one of two *view slots*.

The other interface elements of each view are grouped inside its window. Among these elements are push-buttons, drag areas, and fields. Drag areas are rectangular regions within which a user may perform direct manipulation, such as dragging objects from one place to another or clicking on objects which access some program functionality. Each of the views currently in use in **Sense of Place** contains at least one drag area. Fields are rectangular regions in which output (most frequently text) is displayed.

The interface elements associated with a view are important only insofar as they allow users to manipulate and gain information from the visualization tool used in the view. Each visualization tool is used to represent certain aspects of a set of data. The following sections describe the views currently available in **Sense of Place**. The discussion in these sections focuses mainly on the visualization tools and techniques used in each view. Chapter 3 discusses the interaction tools associated with each view.

2.3 Representing Spatial Data Characteristics

Geography is unique among sciences in its emphasis on spatial relationships such as direction, distance, relative area, and *spatial autocorrelation.* Spatial autocorrelation describes a tendency of the values for adjacent divisions (or divisions that are near to one another) to be strongly related[5]. Average daily temperature is an example of a variable that is normally spatially autocorrelated. Other variables may demonstrate less obvious spatial autocorrelation.

Relative area is another important spatial characteristic of geographical data. The value distributions of some variables do not always tell the whole story. For example, given only population density figures, one might like to get a rough idea of the relationship between the total populations of two counties with similar values. This relationship could be derived if the relative areas of the two counties were known.

Choroplethic mapping is a technique that *can* show both numerical and spatial aspects of data. Mapping is an excellent technique for illustrating the spatial characteristics of geographical divisions. Maps clearly show such qualities as relative area, relative distance, latitude, and longitude. Choroplethic mapping shows the numerical qualities of statistical data by assigning the same color to divisions with similar values. Thus, a choroplethic map that showed population density by county would assign the same color to counties with the same density. Relationships between the total population figures of counties with similar density values could be estimated from the relative sizes of the counties on the map. Spatial autocorrelation is also evident on a choroplethic maps. Spatial autocorrelation appears as large regions (consisting of more than one division) of the same color. Conversely, variables which are not strongly autocorrelated will result in maps with frequent color changes.

Choroplethic mapping in **Sense of Place** is incorporated into the Map View, which is shown in figure 2.1. The Map View contains two maps, referred to as the state map and the county map, respectively. The state map is drawn at a scale small enough to allow presentation of the entire United States on the screen. This scale is large enough for users to

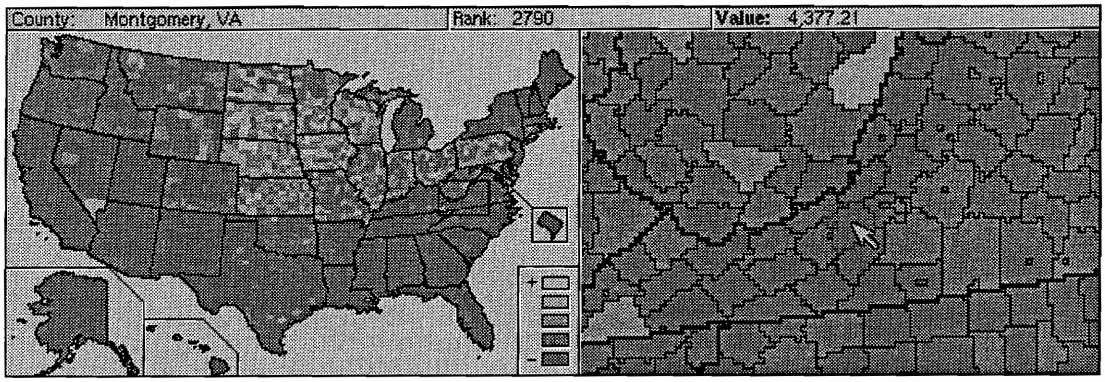


Figure 2.1: **Sense of Place** Map View. The state and county maps are on the left and right respectively.

access each state with the mouse, but too small for them to access most individual counties. Washington, D.C. is shown at a larger scale in an inset on the state map.

The county map allows access to individual counties. The scale of the county map is too large for it to fit on screen in its entirety, so only a portion is displayed at a time. The portion displayed is determined by the current location of a movable box that appears on the state map. The area inside this box is shown at the larger scale in the county map window. Figure 2.2 shows the movable box on the state map.

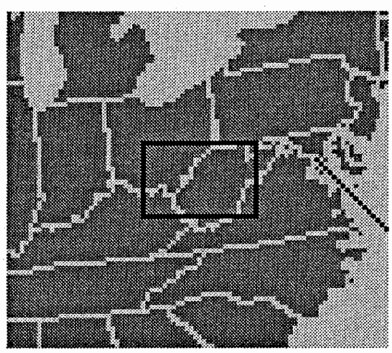


Figure 2.2: Movable Map Box. The contents of the box are displayed on the county map at greater magnification.

The name of each map refers to the smallest division for which the map can display borders. Thus, the county map shows state and county borders, while the state map shows only state borders. Borders define the unit of user access on a map. In other words, a user can access only divisions that have borders on a given map[1] A user accesses a division with the mouse. The names do not refer to the level of data shown on the maps. Both maps show data for the same division level at a time. For example, if a user elects to view county population data, then both maps are colored to match the county data. It is useful to view county data on the state map because regional patterns are more apparent on the state map than they are on the county map.

The particular choroplethic technique used for both maps is *simple choroplethic mapping.* A simple choropleth map symbolizes "magnitudes of statistics as they occur within boundaries of unit areas (such as counties, states, and other kinds of enumeration district)"[11]. The symbolization scheme used in **Sense of Place** is a color scale consisting of five colors, each of which is assigned to a particular group of divisions. Division are separated into groups according to a *range-grading scheme.* **Sense of Place** supports two range-grading schemes: *equal intervals*, in which the total range of observed values is divided into equal-sized portions, and *rank quantiles*, in which divisions are ranked according to their values, and the total range of ranks is divided into contiguous, equal-sized portions.

Although choroplethic mapping is an effective technique for showing spatial relationships, it is not without its objections. The major objection is that choropleth mapping can "wrongly equate the visual importance of each county with its geographical area"[13]. This is particularly true for data which are expressed as rates per population unit, since population and geographical area are not necessarily related. For example, Yukon-Koyukuk County, AK had a birth rate of 2.63% in 1984, while Queens County, NY had a rate of 1.33%. These data, combined with the fact that Yukon-Koyukuk County appears over 1,400 times as large as Queens County on a map, might suggest that there were more births in Yukon-Koyukuk County in 1984. However, birth rates are based on total births and total population, not geographical area. In fact, there were only 218 births in Yukon-Koyukuk

[1]We consider a division to have borders even if those borders are currently hidden from view.

County in 1984, compared to over 25,000 in Queens County. In order to avoid misperceptions, users should be aware of the implications of geographical area for the data they are studying.

Dykes describes objections to the rather coarse classification of data in a typical choropleth map[4]. As he points out, the classification of data into a relatively small number of broad groups (within which there is no symbolic differentiation) can make it difficult for users to get a detailed impression of the distribution of data. However, he goes on to illustrate that attempts to give a more detailed portrayal with finer classification schemes can obscure the patterns that the maps are designed to reveal. Furthermore, colors can be difficult to distinguish on a choropleth map with a great number of color-coded classes.

Lastly, it can be difficult to represent statistical outliers on choropleth maps without unduly distorting the picture of the rest of the distribution. Equal interval range-grading is very sensitive to outliers. For example, the average area for counties is less than 1,000 square miles. However, the largest counties are much larger than the average (the very largest has an area of 159,099 square miles). In a map with equal interval range-grading, any county with an area of less than $159,000/5 = 31,819.8$ square miles (almost every one of them) is placed in the bottom group. For this reason, **Sense of Place** also supports range-grading by rank quantiles. Rank quantiles always contain the same number of individual divisions, since the "values" measured (the ranks) are uniformly distributed. However, range-grading based on rank quantiles also obscures the fact that outliers exist.

These objections serve as caveats that choropleth maps provide only a part of the total picture of geographical data. In particular, choropleth mapping sacrifices fine detail about the distribution of data in order to provide clearer insight into spatial relationships and patterns[4]. The following section discusses why it is equally important to present a detailed view of the data distribution, and describes the visualization technique used to do so.

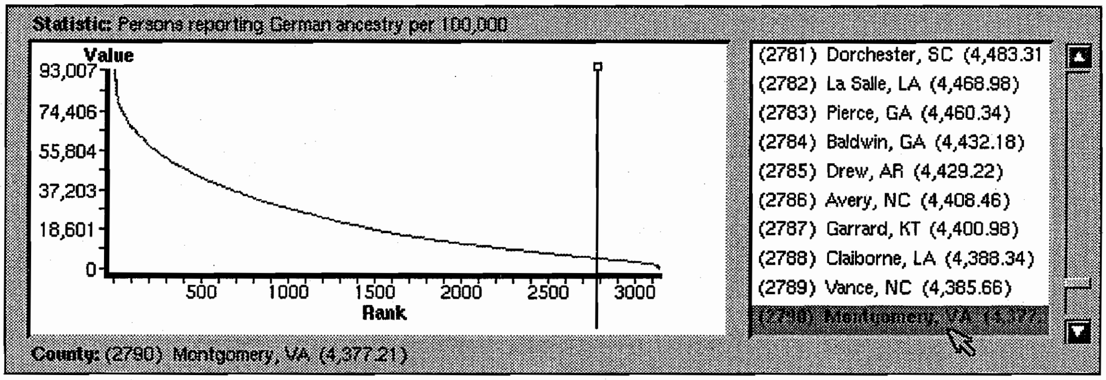


Figure 2.3: **Sense of Place** Graph View. The graph is accompanied by a list of counties or states in rank order. The list can be scrolled with the scroll bar to its right, or with the indicator bar on the graph.

2.4 Representing Statistical Distributions

The distribution for a variable gives a frame of reference for interpreting a single data point. It allows one to determine whether a datum is relatively high or low, or even an outlier. It also describes the characteristics of a data set as a whole, such as the number of data of a certain value, or the set's mean and median values.

In our opinion, a simple cartesian plot — with statistical values along the vertical axis and ranks along the horizontal axis — is an effective way of showing the data distribution. The extreme and median values for a variable are readily apparent on a cartesian plot, and the mean value can be reasonably estimated. The value for any particular state or county can be located on the plot, and the overall rank order of counties or states can be observed. Furthermore, cartesian plots compensate for many of the deficiencies of choropleth maps: they give a much more detailed picture of the distribution than the coarse classification scheme in a choropleth map, are unaffected by relative area, and are not unduly distorted by relative area or the presence of outliers.

Cartesian plotting in **Sense of Place** is incorporated into the Graph View. The Graph View shows the distribution of values associated with divisions in order by rank. These values are either match scores for a ranking metric, or data for a particular variable. The graph is accompanied by a scrollable list of divisions in rank order. This list can be scrolled in two ways: by manipulating a traditional scrollbar, or by dragging an indicator bar across the graph.

As seen in Figure 2.3, the Graph View clearly shows the minimum and maximum values in the data set (-43% and 117%, respectively). The mean and median values can be estimated (both at about 5%) by glancing at the graph. The exact median value can be found by searching through the scrollable list.

The shape of the curve in Figure 2.3 indicates that there are statistical outliers at both the high and low extremes. This can be seen from the steep slope at either end of the curve, which differs sharply from the relatively gentle, regular slope seen in the rest of the graph. Note that these outliers do affect the overall shape of the of the curve, since the y-axis is scaled to fit all data values into a finite space. In other words, the slope of the middle portion of the curve would be steeper in the absence of the outliers, because there would be a smaller range of values along the y-axis. Nonetheless, the graph does differentiate between values at various points along the curve. In general, it does not classify data values into broad groups as choroplethic mapping does.

Some grouping of data values is required when county data are displayed, because there is a maximum of 640 pixel columns available to represent 3,139 data values[2]. The list shows the data points represented by the pixel currently under the indicator bar (the vertical line with the square on top). A ratio of 10 data values per pixel was chosen because it struck a good balance between the width of the graph and the height of the list.

As with choroplethic mapping, there are certain aspects that cartesian graphs simply cannot show. For instance, the Graph View gives virtually no spatial information about the divisions in its list. Location, relative area, and spatial autocorrelation are completely inapparent in the Graph View, because it emphasizes characteristics of data sets that are independent of these physical phenomena. Furthermore, it is difficult to predict the position

[2]All **Project GeoSim** applications use a portion of the screen that is 640 pixels wide and 480 pixels high. This is the resolution of a typical VGA monitor.

of a particular division on the graph. It therefore typically takes more time to find a particular division on the graph than it does to find it on the map (where position is easy to predict).

Lastly, one or the other of the Map View and the Graph View may be more appropriate for viewing a certain variable. Two examples given in § 2.3 illustrate this well. The Map View may be more helpful in gaining insight into population figures for different counties, because it also shows their relative sizes, and hence gives an indication of population density in those counties. The Graph View does not differentiate in any way between two counties with the same value. This is somewhat of a drawback for interpreting population figures, but a boon for understanding birth rates, since differentiating between two counties with the same birth rate based on their relative sizes is generally inappropriate.

2.5 Representing Multivariate Data

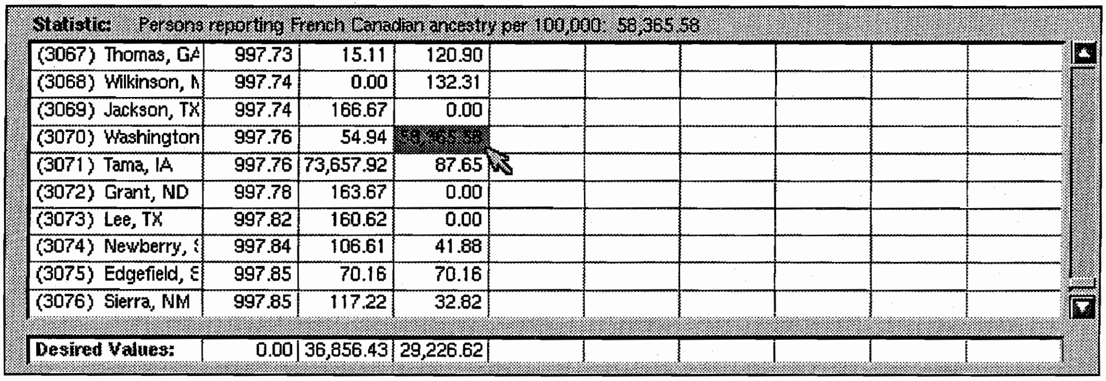
In addition to the spatial characteristics and distribution of a data set, we want to show relationships between sets of data. We also want users to be able to examine several characteristics of a division at once. Although several characteristics are considered when divisions are ranked according to a metric, the resulting scores don't give a clear indication of the precise statistical values for a division. We thus needed a technique for viewing data from several variables simultaneously.

A map can only effectively associate one statistical value at a time — by assigning it a particular color — with each division. Patterns could be overlayed on top of the map's colors to represent a second set of data, but probably would start to degrade the visual quality of the map. A graph can represent each of several variables with its own curve. However, the small amount of space available for plotting a distribution curve makes it difficult to derive precise values from the curve itself. This difficulty inspired us to add a list to the graph which shows precise values, but only for one variable (or set of metric scores). What we really needed was a 'list' with several columns, each containing data from a different variable. In other words, we needed a tabular data representation very much like a spreadsheet. **Sense of Place** offers such a representation in its Spreadsheet View.

The Spreadsheet View is intended to assist users in the analysis of multivariate data. Multivariate data "can be defined as a set of entities E, where the i^{th} element e_i consists of a vector with n observations, $(x_{i1}, x_{i2}, \ldots, x_{in})$"[14]. An "observation" is simply a statistical variable. Thus, each row in the spreadsheet represents one composite data point, rather than a collection of several data points. This is an important distinction, because the variables in a multivariate data set are not necessarily independent. In fact, they are often directly or inversely correlated.

The importance of multivariate data analysis arises in the use of ranking metrics in **Sense of Place**. As mentioned in Chapter 1, a ranking metric defines the variables of a multivariate data set. The score assigned to each division gives an indication of the composite effect of these variables, but it can conceal many statistical differences among divisions. For example, consider a metric consisting of two variables: Dutch and French Canadian ancestry per 100,000 people. If the target values for these variables are set at 36,856.43 and 29,226.62 respectively, then the counties of Tama, IA and Washington, LA obtain identical metric scores (997.76). This might lead one to believe that the populations in these two counties have similar ancestral characteristics. In truth, 73,658 people in 100,000 in Tama county reported Dutch ancestry, while only 55 people in 100,000 in Washington county did. Conversely, 58,366 people in 100,000 reported French Canadian ancestry in Washington county, while only 88 people in 100,000 in Washington county did. The metric scores fail to reveal these differences because the values for the counties are equidistant from the target values. However, the Spreadsheet View *can* reveal the differences between these two counties that are described by their data values. Figure 2.4 shows the Spreadsheet View for the Dutch and French Canadian metric.

The Spreadsheet View is available after a ranking metric has been processed. It displays a table of values for the variables used in the metric. Each row shows the values for a particular division (county or state) and each column shows the values for a particular



Statistic: Persons reporting French Canadian ancestry per 100,000: 58,365.58

(3067) Thomas, GA	997.73	15.11	120.90
(3068) Wilkinson, M	997.74	0.00	132.31
(3069) Jackson, TX	997.74	166.67	0.00
(3070) Washington	997.76	54.94	58,365.58
(3071) Tama, IA	997.76	73,657.92	87.65
(3072) Grant, ND	997.78	163.67	0.00
(3073) Lee, TX	997.82	160.62	0.00
(3074) Newberry, S	997.84	106.61	41.88
(3075) Edgefield, S	997.85	70.16	70.16
(3076) Sierra, NM	997.85	117.22	32.82
Desired Values:	0.00	36,856.43	29,226.62

Figure 2.4: **Sense of Place** Spreadsheet View

variable.

The purpose of the Spreadsheet View is to allow the user to evaluate the match scores of counties or states for a ranking metric. Two divisions with similar match scores may have vastly different characteristics, particularly if there exists an inverse correlation between two variables used in the metric. Through the evaluation of match scores, it is hoped that the user will gain a better knowledge of the characteristics of counties and states, insight into how certain variables are related, and information necessary to fine tune a ranking metric.

2.6 Setting Target Values for Ranking Metrics

The ranking window (shown in Figure 2.5) is used to set the target values and weights for a metric. It contains a pair of "odometers" for each component variable of the metric. The longer odometer on the left is used to set the target value for the associated variable. The odometer on the right is used to set the weight for the variable. The weight is an expression of the importance of the variable relative to other variables in the metric. The minimum, mean, and maximum values are shown above the odometers.

The ranking window does not fully meet the definition of a view, because it offers little visualization of the database. (A brief listing of the extreme and mean values for each variable is the only "visualization" provided by the ranking window.) However, it operates

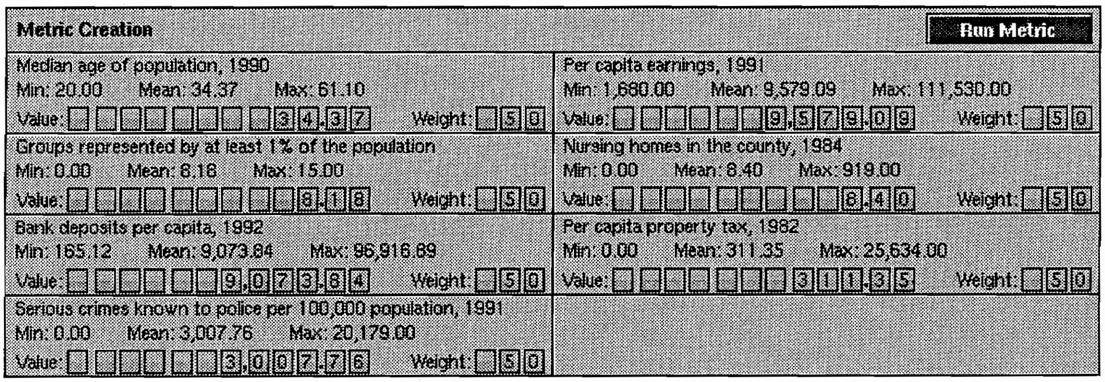


Figure 2.5: **Sense of Place** Ranking Window

functionally as a view, and takes up a view slot. The view slots are the only 'permanent' screen areas large enough to display the ranking window. If the ranking window were not shown in a view slot, it would most likely be displayed in a popup window. **GIL** popup windows are *preemptive*, meaning that other windows (and their buttons and drag areas) are made temporarily inactive while a popup window is on the screen. As such, popup windows are typically meant to appear on screen for relatively short periods of time. We did not want to disable other windows while users set target values in the ranking window, and we wanted to let users keep the ranking window on screen as long as desired, so that they can alter their target values and reprocess the metric. We therefore felt that the ranking window should be treated as a view, rather than a popup window.

Chapter 3

Interface Design

3.1 Implementation

The graphical user interface for **Sense of Place** was implemented with the **GeoSim Interface Library**(GIL). **GIL** is a small, easy to use set of functions for building GUI's that was designed specifically to support development of **Project GeoSim** software applications. In addition to ease of use, the primary goals of **GIL** have been portability and the ability to quickly modify a user interface without recompiling. **GIL** achieves these goals through its layered architecture and its run-time interpretation of interface elements, respectively.

3.2 Sense of Place Target Users

The target audience for **Sense of Place** consists of students in college geography courses. However, we hope that **Sense of Place** will also find users at the secondary and elementary school levels, as well as outside the classroom. For example, **Sense of Place** could be used by adults seeking a place to relocate, to start a business, or to vacation. We therefore expect our audience to be a nonhomogeneous group that includes users of various ages, levels of computer literacy, hardware preferences, and basic skills (e.g., typing). We also expect few people in our audience (aside from geography instructors) to use **Sense of Place** often enough to become experts at it. These expectations led us to strive for a user interface that is easy to learn and use for users at any skill level. (It should be noted that the issue of hardware preference is made mostly moot by the availability of **Sense of Place** for many of the most popular computing platforms.)

3.3 Goals for Sense of Place User Interface

Typing Requirements Because we expected the users of **Sense of Place** to have varying levels of typing experience — including none at all — there is no typing required in the application. Although typing often appeals to so-called "power users" who don't like using a mouse [9], there is actually little in **Sense of Place** that calls for typing. More importantly, most of our audience will probably not use **Sense of Place** enough to become power users. The main task where typing could be applicable is entering the name of a state or county to search for in one of the views. However, this task can be completed easily by selecting the name of the region from a picklist. Furthermore, the main potential advantage of typing — improved access time — would only be appreciable if typical users were expected to search for a large number of states or counties by name, which is not the case. Thus, to accommodate users with poor typing skills, and to simplify understanding the interface, we chose the picklist approach.

Learnability Because **Sense of Place** will be used at most a handful of times by the majority of its audience, it should be as easy as possible to learn to use. In particular, its operation should be as intuitive as possible. In addition, it should behave consistently on all platforms for which it is available, and in a manner that is consistent with other **Project GeoSim** simulation programs. This consistency will prevent users from having to relearn **Sense of Place** when they use it on another machine, and will allow them to build on the interface knowledge they have gained from other **Project GeoSim** modules. This consistency will benefit instructors above all others, since they will be the users most likely to use **Sense of Place** on multiple platforms. Instructors will also be the users most likely to procure **Sense of Place**, so it is important to maintain their satisfaction.

Context Transitions Another goal was to allow users to move easily from one view context to another. For instance, a user looking at a county in the Map View might want to find that county on the graph, or on the spreadsheet. He should be able to do this as quickly and directly as possible, so that he doesn't risk losing his train of thought, or growing impatient. Impatience is particularly problematic, since users will tend to avoid program functionality that requires much time or effort.

Future Development We wanted to allow for the future addition of other views to **Sense of Place.** Because easy context transitions would continue to be important, we wanted to be able to thoroughly integrate any new views into the view structure of **Sense of Place**, instead of "gluing" them on top of the structure. For example, displaying a new view in a window that covered other views would make context transitions cumbersome for the user, and hence should be avoided.

Our solution was to divide the usable portion of the screen into two *view slots* of equal size. This allowed us to display any view in either slot. This is a *tiled* (i.e., non-overlapping) window layout. Any new views should also be made to fit into a view slot. This does limit the user to two on-screen views at any time, but possible alternatives we considered were seen as less satisfactory. The view areas could have been made smaller, thus allowing for more views to be visible simultaneously. This would have been unworkable given the fact that **GIL** currently limits applications to a 640×480 area of the screen. It would have been difficult to see the views, much less manipulate objects in them with the mouse.

The other alternative would have been to allow users to display an arbitrary number of views in overlapping windows. However, tiled windows tend to offer better user performance than overlapping windows[9]. In any case, the effect of overlapping windows would not be that different from that of tiled windows, since each of the current views could realistically occupy no less than half the usable screen space.

3.4 Sense of Place Interface Features

Figure 3.1 shows the series of buttons that appears at the top of the screen. Each of these buttons displays a menu when pressed. The menus are used to control those aspects of **Sense of Place** which are not associated with a particular view. Among the functions

Figure 3.1: **Sense of Place** Buttons

that these menus control are:

1. Quitting the program
2. Dumping the **Sense of Place** screen image to a file
3. Accessing help screens
4. Selecting various ways of accessing county data
5. Selecting various ways of accessing state data
6. Selecting the combination of views to display
7. Setting user preferences

These buttons and the associated menus are discussed at greater length in Appendix A, the *Sense of Place User's Manual* (not provided in this draft).

3.4.1 Map View Features

Finding Map Divisions A highlighting technique is employed to allow users to find divisions on one of the two maps. Map divisions can be found by selecting "Find County (State) On Map" from the "County (State)" Menu, then selecting a county (state) from the picklist window, and finally pressing the button labeled "Find on Map" in the picklist window. Alternatively, users can find a map division by clicking on the division's name in the list in the Graph View or in the table in the Spreadsheet View.

A division is "highlighted" by changing the color of its border from gray to red. In the opinion of the author, this is preferable to changing the color of the division itself, because it doesn't obscure the statistical information conveyed by the map. One disadvantage of this strategy is that a highlighted border may be less noticeable than a highlighted division, because the border generally contains fewer pixels. However, the author feels that this method of highlighting is adequate for getting the attention of most users.

Getting Data for Divisions Text is not used directly on the maps to label division names, ranks or values, because it would add unnecessary clutter and obscure the visual effect of the simple choroplethic technique. Instead, the name, rank, and value of the division currently under the hotspot of the cursor is presented in an information bar at the top of the Map View. Figure 3.2 shows the information bar from the picture of the Map View in Figure 2.1. The fields in the information bar are blank when the cursor is not over any division on the map.

Figure 3.2: Map View Information Bar. This is an excerpt from Figure 2.1

When the mouse cursor is pointing at a state on the state map, the information bar shows data for that state, regardless of whether the map is currently colored to match state or county data. Likewise, when the cursor points at a county on the county map, the information bar shows data for that county. Thus, users can obtain raw data for states and counties at the same time. Note, however, that this flexibility cannot easily be extended to ranking scores, because ranking metrics are usually created with either counties or states in mind. For example, consider a simple metric composed of two variables: land area in square miles and population. Although it would be reasonable to rank counties according to how closely they meet a goal of 500 square miles and 50,000 inhabitants, it wouldn't be very meaningful to use these targets for ranking states. Because we don't want to show ranking scores that are unlikely to be meaningful, we only show scores for the current division level.

3.4.2 Graph View Features

The Graph View should let users to perform the following tasks:

1. Examine the distribution for a variable or metric

2. Discover what divisions are represented by a particular point on the distribution curve.

3. Find the location of a division on the distribution curve.

4. Find the location of a division in the Map View or the Spreadsheet View.

The first goal is achieved by drawing a cartesian plot of the distribution. Because our limited screen space does not allow us to plot a point for each county, a pixel along the horizontal axis is used to represent ten counties. The scale of the vertical axis is determined by the range of values for the plotted variable and the length of the vertical axis, which is constant for all plots. The length of the vertical axis is set so that it fits snuggly into the graph area. The values at the top and bottom of the vertical axis are the maximum and minimum values over all counties or states for the variable, respectively. The tick marks along the vertical axis are evenly spaced, and every second mark is labeled with the value associated with that mark's position.

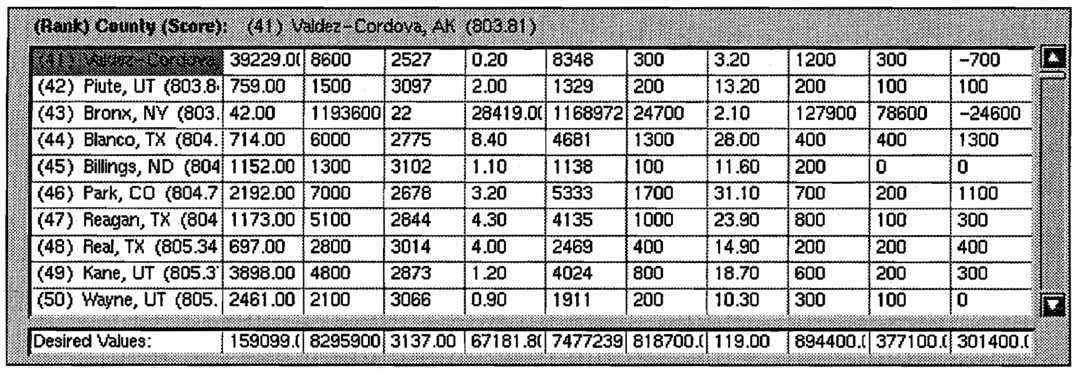
The second goal is achieved by including a picklist to the right of the cartesian plot, and a bar to indicate the point on the curve which represents the divisions currently appearing in the list. The indicator bar is movable, allowing users to examine every point along the graph.

The indicator bar works well for coarse movements to an area on the graph, but not very well for finely controlled movements through the list. Therefore, the list is also scrollable. The indicator bar and the scrollbar each move automatically when the other is moved.

Finding a division on the distribution curve requires some method of user input, such as typing the name of the division, or selecting it from a list. As mentioned earlier, we decided to avoid typed user entry in favor of selection from a picklist. The picklist should be sorted alphabetically, so that users can quickly find the name of any division. Unfortunately, there is not enough space in the graph window for such a list in addition to the graph and the list of names sorted by rank. Therefore, we use a popup window we refer to as the pick window to display the selection list or lists. The pick window is also used to select names of counties and states for other location tasks, such as finding divisions on the maps and in the spreadsheet. The consistency of procedure and technique across these tasks meets user expectations that similar things will be done in similar ways[9]. When the user clicks on a division name in the picklist, the indicator bar moves to the location of the selected division on the distribution curve. The ranked list then automatically scrolls to the location of the selected division, as it does when the indicator bar is manipulated directly.

Finding a division in the Map View or the Spreadsheet View can be accomplished by using the picklist window, as described above for the Graph View. However, we also offer a more direct route. If a user sees the name of a division in the graph list, he or she should not be required to go to the picklist window to select the name. We thus made it possible for users to find divisions in other views by clicking on the name of the division in the graph list. When a user clicks on a name in the graph list, **Sense of Place** finds and marks it in the other current view, unless the other current view is the ranking window.

3.4.3 Spreadsheet View Features


(Rank) County (Score): (41) Valdez-Cordova, AK (803.81)

(41) Valdez-Cordova,	39229.0(	8600	2527	0.20	8348	300	3.20	1200	300	-700
(42) Piute, UT (803.8	759.00	1500	3097	2.00	1329	200	13.20	200	100	100
(43) Bronx, NY (803.	42.00	1193600	22	28419.0(	1168972	24700	2.10	127900	78600	-24600
(44) Blanco, TX (804.	714.00	6000	2775	8.40	4681	1300	28.00	400	400	1300
(45) Billings, ND (804	1152.00	1300	3102	1.10	1138	100	11.60	200	0	0
(46) Park, CO (804.7	2192.00	7000	2678	3.20	5333	1700	31.10	700	200	1100
(47) Reagan, TX (804	1173.00	5100	2844	4.30	4135	1000	23.90	800	100	300
(48) Real, TX (805.34	697.00	2800	3014	4.00	2469	400	14.90	200	200	400
(49) Kane, UT (805.3	3898.00	4800	2873	1.20	4024	800	18.70	600	200	300
(50) Wayne, UT (805.	2461.00	2100	3066	0.90	1911	200	10.30	300	100	0
Desired Values:	159099.(	8295900	3137.00	67181.8(	7477239	818700.(	119.00	894400.(	377100.(	301400.(

Figure 3.3: Showing Labels in the Spreadsheet View. The cell underneath the tip of the mouse pointer is highlighted, and its contents are echoed above the spreadsheet.

The main issue that influenced the design of the interface features for the Spreadsheet View was the fact that it needed to display a lot of information in a small space. The spreadsheet itself had to be eleven columns wide, to accommodate division names and up to eight variables. The resulting columns are not wide enough for even abbreviated versions of the variable names to be displayed above the table. They are in fact not as wide as the widest numbers that they have to display.

The solution was to clip the right ends of any labels that do not fit in their cells. To allow users to view every label in its entirety, the available space above the table is used to echo the label in the cell currently under the tip of the mouse pointer. If the cell pointed at by the mouse is in a variable column, the name of the variable is also displayed. Figure 3.3 shows an example of this technique.

3.4.4 Ranking Window Features

To select variables for a ranking metric, the user selects either "Rank Counties" from the county menu or "Rank States" from the state menu. The pick window appears with a list of variables in response to the user's selection. A variable is selected by clicking on its label in the list. A selected label is highlighted by redrawing it in red with a bold font. The user can remove a selected variable by clicking on it a second time.

The user presses "Done" after making his or her selections, and the ranking window appears. The ranking window contains a pair of "odometers" for each selected variable. The value odometer (the one one the left) is used to enter a target value, while the weight odometer is used to set the relative importance of this target value to a user. Any digit can be changed by clicking on it. A digit is incremented by clicking in the top half of the surrounding box, and decremented by clicking in the bottom half. The target value must be within the range of observed values. The weight must be between 0 and 100.

Once the input values are set, the user presses "Run Metric", and **Sense of Place** proceeds to score each division. The "Run Metric" button is then disabled until the metric is modified by either altering the set of variables or changing a target value or weight. The current state of the data in other views is always what it was the last time the user pressed "Run Metric".

Chapter 4

Functional Design

4.1 Implementation of the Map View

The maps used in **Sense of Place** are stored as *run-length-encoded* images. Run-length encoding is a simple compression scheme that is effective for most images, except those characterized by frequent color changes[7]. Run-length compression is usually effective for simple choropleth maps, since they generally consist of relatively large contiguous areas of the same color. For example, the image file for the county map in **Sense of Place** is compressed to 10

The essence of run-length encoding is to replace repeated adjacent values in a file by a pair of values, called a *run*. One of these two values is the value found in the file; the other is the *length* of the run, that is, the number of times that value is repeated in sequence. Note that there is not a unique way to split a file into runs. For example, 16 adjacent repetitions of a particular value could be represented as one run of length 16, or two runs of length 8, and so on.

The values in a **Sense of Place** map file represent counties rather than colors. Since the stored representation of a map is independent of any color scheme used when drawing the map, we can use one image file for all variables and all conceivable ranking metrics. If color values were stored in an image file — as they often are for static images — there would have to be a file for each variable and each possible user metric.

In **Sense of Place**, the run-length-encoded map is read from file each time it is drawn. (**GIL** also allows run-length-encoded images to be stored in primary memory, which has potential execution speed advantages, but there was not enough available memory in some programming environments for both maps.) Each row is expanded into a buffer in turn. The county values in this buffer are used as indices into an array of color values, which has been initialized to match the appropriate statistical values. The resulting array of color values is then written to the screen.

It is also possible to add state or county border lines to a map through use of an *edge detection* algorithm. The basic idea of the edge-detection algorithm in **Sense of Place** is to insert a border color value wherever two adjacent dissimilar county values appear. This algorithm has also been fined-tuned to allow for thicker border lines at state boundaries, and highlighting the border of a selected state or county.

4.1.1 County Map Magnification

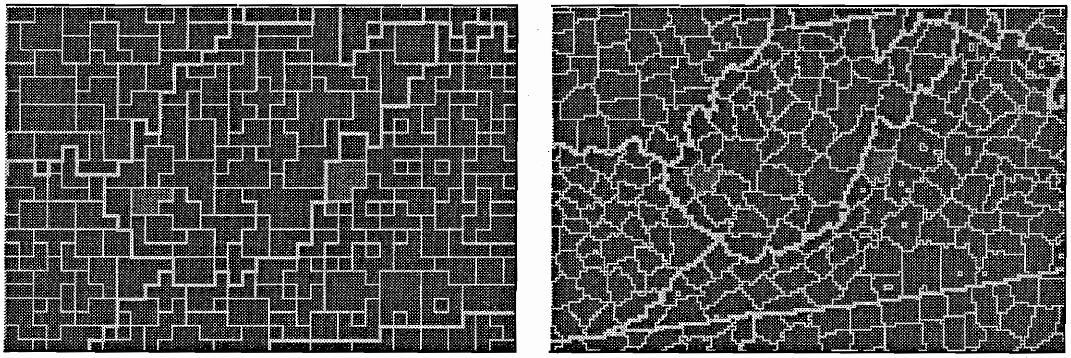


Figure 4.1: At left is the screen image produced by the smaller image file at 800% magnification. At right is the larger file at 200% magnification.

Our first implementation for the county map was to draw a "magnified" view of the image used for the state map. An integral magnification factor specified how many times each point was to be repeated. The resulting map had disappointing visual quality, since the relatively small image file did not contain enough information to represent counties accurately at high levels of magnification. The solution was to use a separate, larger image file for the county map. However, the county map can still be magnified to suit the user. Figure 4.1 shows the improvement in visual quality provided by the larger map.

4.1.2 Drawing Map Borders

Since adjacent divisions might be assigned the same color for a given variable or metric, it is desirable to draw borders to clearly distinguish them. We considered three ways of drawing map borders:

1. Use map image files which include implicit border pixels. The main drawback with this approach is that it would make it difficult to draw the maps without borders (regional patterns can be more apparent on a borderless map). It also would also make the image files slightly larger, since the border pixels would add more runs. Furthermore, border pixels would complicate the implementation of the function that displays the label of the county or state currently under the tip of the mouse pointer.

2. Use one image file to represent county pixels, and another to store border pixels. This would allow the borders to be drawn as an overlay on top of the borderless map image. This approach would make it easier to remove the borders from the screen. However, it would also require extra disk space to store the additional files, extra primary memory to use them, and more effort to maintain them, since most changes to a map file would likely modify the county borders as well.

3. Use an algorithm to dynamically generate the borders from the map image files. This approach allows us to easily draw the map with or without borders, and requires only one image file of minimal size for each map. The main cost associated with this method is the additional time required to "detect" the borders on the fly each time a map is drawn. However, drawing a map generally requires considerably less time than accessing and processing the data used to draw it, so this does not seem to be a problem.

The third method was selected. The algorithm used is a simple edge detection scheme. The essence of this scheme is to compare each pixel to the pixels above it and to its left.

An edge is "detected" when a pixel is discovered to be different in value[1] from either its left or upper neighbor. When an edge is detected, the pixel is drawn in the border color, rather than the color of the county it belongs to. This scheme has been modified for **Sense of Place** to differentiate state borders from county borders in the county map, and to allow highlighting of a county by drawing its border in a unique color. State borders are drawn two pixels wide in the county map to make them distinct from borders between two counties in the same state. This is accomplished by comparing each pixel with the two directly to its left, the two directly to its north, and the four to its northwest. Figure 4.2 summarizes the edge detection scheme.

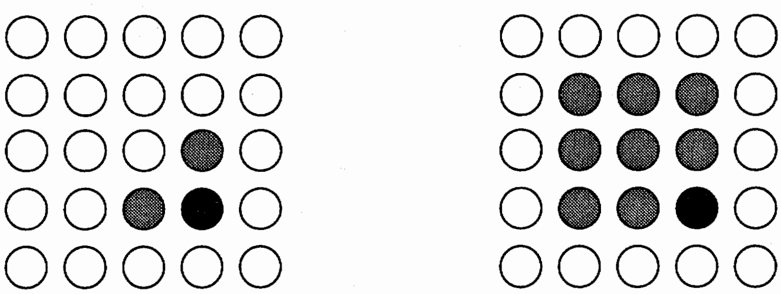


Figure 4.2: Edge Detection. The diagrams at left and right depict the schemes used for drawing single-thickness and double-thickness borders, respectively. In either diagram, the pixel shown in black is drawn in the border color if its value is different from the value of any of the pixels shown in gray.

4.2 Implementation of the Graph View

Implementation of the Graph View was fairly straight forward. The majority of the decisions to be made regarding the Graph View had to do with the visualization technique and the user interface. A **GIL** picklist implements the list of divisions displayed on the right side of the view. A **GIL** drag area is used to allow users to drag a movable indicator bar horizontally along the distribution curve. The indicator bar itself is drawn using XOR

[1]The value described here is an integer that identifies the division to which the pixel belongs. The values for pixels from different divisions are guaranteed to be different. This value is distinct from the pixel's color value, which may or may not be different from the color value of a pixel in a different division.

mode, which allows for easy removal of the bar from its previous position when it is moved.

4.3 Implementation of the Spreadsheet View

The Spreadsheet view presents data in tabular form. Because **GIL** picklists were designed to allow for multiple columns, they could be used for simple tables. Before **Sense of Place** was developed, however, the **GIL** implementation for picklists had several limitations which prevented the effective use of a single picklist in the Spreadsheet view:

1. All columns in a picklist had to be of identical width.
2. A picklist did not know how to draw gridlines between cells, as is customary in a spreadsheet.
3. Picklists would overdraw any gridlines that the application code drew itself.

These limitations are addressed in § 4.3.1 and § 4.3.2, respectively. One problem was the fact that memory constraints made it impossible in the DOS version to store all data for the spreadsheet in primary memory simultaneously. On the other hand, performance expectations made it impractical to rely entirely on secondary memory. The solution to this storage problem is described in § 4.3.3.

4.3.1 Column Width

Uniform column width was an obstacle for the Spreadsheet View because more space is required in general for division names than for data values. Thus, the first column in the spreadsheet, which contains division names, needs to be wider than the others. This obstacle is surmounted by using one picklist for the name column, and another for the data columns. This actually makes for cleaner application coding, since it puts the decision as to what type of label to draw (name versus number) in the **GIL** event handler, instead of the application code. Each picklist has a *label function*, which is called by **GIL** whenever it has to draw a label (i.e., an item) in the list. The label function sends **GIL** a pointer to the text for the item. If one picklist were used for the Spreadsheet View, its label function would have to be intelligent enough to determine whether to send a name or a number, depending on what column the slot were in. The use of two picklists allows for the use of two label functions, each of which only has to send back one kind of label (names for the name picklist, numbers for the data picklist).

4.3.2 Gridlines

Because of the limited screen space available for the Spreadsheet View, labels in the spreadsheet had to "extend" virtually past the bounds of their cells. To prevent overlap, the labels were physically clipped at cell boundaries (see § 5.2). This clipping made gridlines necessary to clearly demarcate cell boundaries.

The only way to create these gridlines with the original **GIL** implementation for picklists was to draw them in the application code. This approach was attempted, but was abandoned due to the visual distraction of redrawing the gridlines every time the list was redrawn. The author decided that the best tack was to modify **GIL** so that this redrawing could be avoided. This modification is described in § 5.1.2.

4.3.3 Storing Spreadsheet Data

Data for the spreadsheet cannot all be stored simultaneously in primary memory, as they would require 201,024 bytes[2] for a eight-variable metric. This would cause problems for the MS-DOS version, which is nominally[3] limited to 640,000 bytes of available primary memory. On the other hand, the data are stored in multiple files — according to type of variable — which would have to be repeatedly opened and closed as the data were accessed, since ten open files would require considerable buffer space. This was seen as a possible hindrance to performance.

[2]Eight values for each of 3,141 counties would mean a total of 25,128 values to be stored. Eight bytes of storage are dedicated to each value, in order to assure accuracy during computations. $8 \times 25,128 = 201,024$.

[3]On our MS-DOS system, other software reduces this limit to approximately 533,000 bytes

It was expected that users' scrolling would exhibit *locality of reference*[12] similar to that of page references in a virtual memory system. In other words, users would probably do mostly line-at-a-time scrolling as opposed to scrolling in large leaps. It was also thought that greater distance implies greater time, and thus most users wouldn't mind the additional time required to scroll a large distance.

The approach taken is a simple virtual memory scheme. A contiguous portion of the table is loaded into primary memory. To allow efficient line-at-a-time scrolling, this portion is larger than that displayed on screen (meaning that most of the time, scrolling up or down a line does not require new data from secondary memory).

The data type used is referred to as a "virtual array". A programmer using a virtual array accesses data in much the same way as he or she would in a normal array. The main difference is that subscripting operators are replaced by a single function call. For example, where one might use to $A[i][j]$ to refer to the j^{th} element in the i^{th} row of a normal array A, one would make the function call `element`(A, i, j) to access the corresponding element in a virtual array.

4.4 Design and Implementation of the Sense of Place Database

The original **Sense of Place** database was implemented as a collection of ASCII text files. This implementation offered the advantage of human-readable database files, but also had the following disadvantages:

- A typical text file was larger than a binary file that contained the same data, due to the relatively uneconomical storage of numbers in text format[4].
- The code required to parse the text files was relatively large compared to the code required for binary data files. This was of particular concern under MS-DOS, where primary memory is relatively scarce.

[4]Four bytes can store no number larger than 9,999 in a text file. By contrast, four bytes are enough to store numbers as large as 2,147,483,647 in a binary file.

- The expected access time for data in the text files was relatively slow, because of the time required to parse the text. Files that contained the offsets of important positions in the data files were used to accelerate the parsing, but these files required additional secondary storage. A uniform field-width for each datum would allow the computation of offsets, but would also increase the size of the data files, since the field-width would have to be as large as the widest datum. A binary format allows a relatively small uniform field-width. Thus, the binary files are relatively small, and do not have to be parsed to access data.

The most crucial[5] disadvantage to the original implementation was the size of the text files. We felt that the advantages of a smaller database to our customers (including less time spent downloading **Sense of Place**) warranted a change to binary data files.

To allow humans to read the binary data files, we implemented a utility called **spb2spa**, which translates a binary data file into a cleanly-formatted ASCII text file. The resulting output file is useful as model to a instructor who wants to create a database file to add to **Sense of Place.** Once the new file is created, the instructor can use our utility **spa2spb** to translate it into binary format. The last step for the instructor is to declare the name of the new database file in the configuration file. Appendix B, the *Sense of Place Instructor's Manual*, describes the process of adding files to the database.

The naming conventions for **Sense of Place** data files are the same as those of MS-DOS: up to eight characters for the name, plus a three-character extension. The extension for data files is ".spb" (Sense of **P**lace **B**inary Datafile). For each data file, there is a dictionary file of the same name, but with the extension ".spd" (Sense of **P**lace **D**ictionary File). A dictionary file contains information about the variables whose data is stored in the corresponding data file, such as their order, names, and extreme and mean values.

[5]The ascii data files for our first database (which contained of 203 variables, slightly more than the 169 variables in the current database) required 5,519,138 bytes of storage. Binary data files for the same database would have only required 2,590,280 bytes ($3,190$ divisions $\times 203$ values per division $\times 4$ bytes per value).

4.5 Metric Matching Algorithm

The visualization tools in **Sense of Place** can be used to evaluate divisions in terms of user-created *metrics*. A metric is a statistical standard composed of target values for a set of variables. When a metric is created by a user, all counties or all states can be compared to that metric and scored according to how closely they match the metric. The algorithm used to determine a division's match score is as follows:

Algorithm SCORE_DIVISION_BY_METRIC

INPUT: User metric M and a division D

OUTPUT: A match score for D

```
Sum = 0
NumVars = 0
FOR each variable S in M
        IF the database contains a value for D in S THEN
                NumVars = NumVars + 1
                Let V be D's value for S
                Let T be the user's target value for S
                Let Max be the maximum value over all divisions for S
                Let Min be the minimum value over all divisions for S
                Sum = Sum + |T-V| / MAX(T-Min, Max-T)
        ENDIF
ENDFOR
Score = Sum / NumVars
```

The score is thus the average of the relative distances between the target values in the metric and the values for the division. The maximum relative distance between a variable's target value and it's value for a division is 1, while the minimum relative distance is 0. Thus, a perfect match is indicated by a score of 0, while the worst possible score is 1.

Note that only variables for which there are data are considered. This allows scoring of divisions which have missing data. There is an inherent danger in this policy. If enough data are missing, then a division's score can be quite misleading, because it does not reflect the same multiplicity of characteristics that scores of divisions with no (or even few) missing data do. Users should be aware of this potential danger, but should not be unduly concerned about it, because great pains have been taken to collect complete data sets. Furthermore, the Spreadsheet View reveals any missing data for division. We therefore decided to award scores based on whatever data is present, and allow the user to determine the validity of these scores by examining the Spreadsheet View.

Chapter 5

Enhancements to GIL for Sense of Place

The philosophy behind the ongoing development of the **GeoSim Interface Library** has been "smaller is better". Thus, additions to the library are not made casually. The first consideration when evaluating a potential addition to the library is, "Can this be done effectively and efficiently in the application code?" If the answer is "no", then the addition should probably be made. An example is the use of gridlines in the Spreadsheet View, which could not be done satisfactorily without some modifications to **GIL** picklists, as described in § 5.1.2. If the answer is "yes", then a second question should be asked: "Will **Project GeoSim** need the capabilities this addition will provide in the future?" For example, drop-shadowing is fairly easy to do at the application level. However, it's done enough in **Project GeoSim** applications that it seemed appropriate to include functions for doing this in the library. Described below are the major changes to **GIL** which were made to better support the development of **Sense of Place.**

5.1 Picklists

A picklist is a list of selectable items that is in many ways similar to a menu. The chief distinction between a picklist and a menu in **GIL** is size. **GIL** menus are currently limited to no more than 25 items, whereas picklists may contain an arbitrary number of items.

A picklist also offers more flexibility of look and feel than a menu. Picklists can have multiple "pages", multiple columns, and allow the application programmer to specify the color and font of each item independently. By contrast, a **GIL** menu must present all of its items on screen at one time, cannot have more than one column, and all of its items are drawn in the same color and font. The flexibility of a picklist allows for such useful interface features as scrollable lists, lists with too many items to simultaneously fit on screen, and highlighting of multiple selected items.

5.1.1 Multiple Picklists

Previous versions of **GIL** allowed for only one picklist on screen at a time. This limitation was not a handicap for any **Project GeoSim** application before **Sense of Place**. As **Sense of Place** developed, however, it became clear that it would require multiple picklists. For instance, the graph view required a list of divisions in rank order, while the spreadsheet view was implemented with a pair of picklists. It would have been impractical to present these two views simultaneously without multiple picklists.

The new implementation was fairly straight forward. The collection of picklist state variables were consolidated into a data structure called `GSlistType`. An array of type `GSListType` was then declared. The return type of the picklist initialization function `GSlistsetup` was changed from `void` to `GSlistHandle`. The returned value is used by the application code to identify the list in all subsequent **GIL** picklist functions.

5.1.2 Spreadsheets

The picklist was a natural choice for implementation of the spreadsheet view in **Sense of Place**. The only problem was that **GIL** didn't offer a means of drawing and maintaining gridlines between items. The application itself could have taken responsibility for gridlines, but this meant redrawing the lines around each slot any time an item was drawn there. This approach was attempted, but the results appeared clumsy.

The solution was to let **GIL** set gaps of arbitrary size between rows and columns in the list, respectively. The application is responsible for drawing the lines when the picklist is initialized, but not each time it changes.

5.2 Clipped Messages

Sense of Place uses every bit of screen real estate at its disposal. The thickness of its window borders was even trimmed from one pixel to two to squeeze everything in. There are in fact some situations which require more screen space than is available. For instance, long county names can extend past the end of the available space at the top of the county map window.

Since screen real estate could not be increased under MS-DOS, a policy was required for shortening these names to fit in the available space. Rather than abbreviate the names themselves, the author decided that the names should be "clipped" to fit in the allotted space. The function `GSwwriteclippedmsg` was added as companion to `GSwwritemsg`. `GSwwritemsg` requires only an origin relative to the current window, and draws the portion of the string that lies within the current window's borders. In addition to an origin, `GSwwriteclippedmsg` requires the dimensions of a "clip area". Only the portion of the string that fits inside the clip area is drawn. Figure 5.1 shows how the parameters to `GSwwriteclippedmsg` are used.

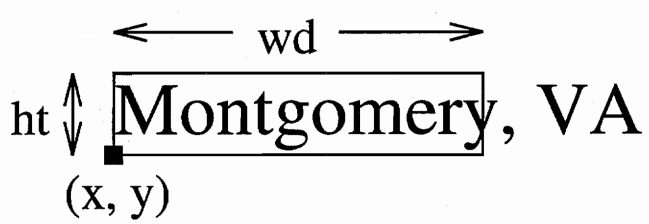


Figure 5.1: Example of `GSwwriteclippedmsg`. The portion of the string that lies outside of the box described by the arguments `x, y, wd,` and `ht` is not drawn.

Chapter 6

Future Developments

Although **Sense of Place** has graduated from a prototype to bona fide educational software application, there are several areas in which its capabilities could be expanded. § 6.1 examines a visualization technique that could be added to **Sense of Place** in the future. § 6.2 describes how the same view could be instantiated multiple times to compare several sets of data. The requirements for adding other countries and world regions is addressed in § 6.3.

The main impediment to further development of **Sense of Place** is the size of conventional memory under MS-DOS. **Project GeoSim** has made it a policy to restrict our software to unreserved conventional memory under MS-DOS, so that the software will run under even the simplest PC configurations. There is not more than 640Kb of conventional memory under MS-DOS. Our development platform makes at most 533Kb of conventional memory available for applications, scarcely more than the 518Kb **Sense of Place** requires to run under MS-DOS. In short, there is not much space for added features. There are currently plans underway for porting the **GeoSim Interface Library** to Microsoft Windows. This port could solve the storage limitation, if the decision were made to discontinue support of the MS-DOS version of the library.

6.1 Additional Views

The **Sense of Place** view display structure was designed to simplify adding new views. Views must be made to fit into a rectangular space 630 pixel wide and 215 pixels high. If a view cannot be displayed in its entirety in this limited space, then a scheme should be developed for displaying only a portion of the view at once. Some ideas for views are discussed in the following two sections.

6.1.1 Representing Statistical Correlations

We feel that it is important to teach students about basic concepts of statistical correlation, so that they understand how certain phenomena interact, and how uncollected data can be predicted based on the data one has on hand. We therefore see the need in the future to enhance **Sense of Place** with a visualization technique that is specifically designed to convey correlations. Although the Spreadsheet View gives users the information necessary to infer correlations between data sets, it does little to make these correlations readily apparent.

Ward describes one possible approach for uncovering correlations between variables called the method of parallel coordinates[14]. In this technique, each variable is repre-

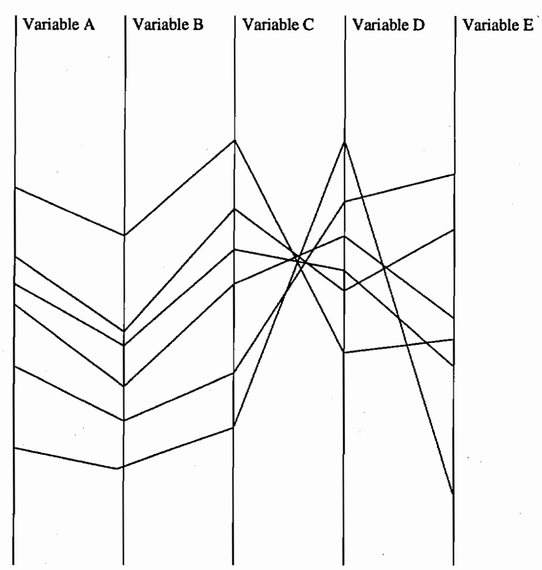


Figure 6.1: Method of Parallel Coordinates. The almost-parallel lines between A and B indicate direct correlation (likewise for B and C). The lines that form an 'X' between C and D indicate inverse correlation. The haphazard pattern of the lines between D and E indicate that there is no correlation between these variables.

sented by a graduated vertical axis. A division would be represented by plotting its value for each variable as a point on the corresponding axis, and then connecting all the points for the division in left to right order.

A direct correlation between two variables would manifest itself as a set of parallel lines between the corresponding axes. Inverse correlations, on the other hand, would appear as a set of intersecting lines between the axes. An example which shows both direct and inverse correlations can be seen in Figure 6.1.

The effectiveness of the method of parallel lines as a visualization tool tends to degrade as the size of the data set increases. As more lines are added, the clarity of observable patterns begins to blur. In particular, presentation of data for over 3000 counties in a limited area of a computer monitor would be unworkable. In order for this technique to be effectively used in **Sense of Place**, a limitation would have to be imposed on the number of divisions plotted. There could be numerous selection criteria for a set of divisions to display: counties in a particular state, or a particular range in the ranking order, for example.

6.2 Multiple Instances of Views

The user may wish to view more than one instance of the same view type. For example, a user might want to modify a desirability metric with poor match values, and compare either the distribution curve or the maps for the modified metric against the original. The distribution curves for the two metrics could be displayed on the same graph, or they could be displayed on two separate graphs. Additional maps would obviously require additional view windows.

The current view structure of **Sense of Place** could support two instances of the same view type on screen simultaneously. Three or more instances would require modification of this structure. Alternatively, **GIL** could be expanded to allow applications programmers to create more than one application window. This alternative assumes the use of a windowing environment such as Microsoft Windows on the PC.

6.3 Other Regions of the World

The capabilities of **Sense of Place** are currently only applied to the United States. It would, however, be possible to include other countries and regions of the world at some point in the future. Each country would require its own database and map image files. Some changes in nomenclature might be required, such as replacing "county" and "state" with the equivalent terms in the new country. Careful planning would also be required to incorporate new countries into the view structure, so that users could compare views from different countries without confusing them. The use of multiple application windows as described in § 6.2 might be useful in this respect.

REFERENCES

[1] James M. A. Begole, Clifford A. Shaffer, and Mark Lattanzi. The Project GeoSim graphical user interface. In *Proceedings of the 23rd Annual Virginia Computer Users Conference*, pages 17–28. Department of Computer Science, Virginia Tech, October 1993.

[2] Boyer and Savageau. *Your Guide to Finding the Best Places to Live in America.* Rand McNally & Company, Chicago, 1989.

[3] Lawrence W. Carstensen, Clifford A. Shaffer, R.W. Morrill, and Edward A. Fox. GeoSim: A GIS-based simulation laboratory for introductory geography. *Journal of Geography*, 92(2):217–222, 1993.

[4] J. Dykes. *Visualization in Geographical Information Systems*, chapter 12. John Wiley & Sons, Inc., Chichester, 1994.

[5] David Ebdon. *Statistics in Geography.* Basil Blackwell, Inc., 108 Cowley Road, Oxford OX4 1JF, UK, second edition, 1985.

[6] Edward B. Espenshade, Jr., editor. *Goode's World Atlas.* Rand McNally & Company, Chicago, 17th edition, 1988.

[7] Andrew S. Glassner. *Graphics Gems II*, chapter II.8. Harcourt Brace Jovanovich, Boston, 1991.

[8] David T. Hines, James M. A. Begole, Colin A. Klipsch, and Clifford A. Shaffer. *The GeoSim Interface Library (GIL) Programmer's Manual, Version 1.0.1*, 1994.

[9] Deborah Hix and H. Rex Hartson. *Developing User Interfaces: Ensuring Usability Through Product & Process.* John Wiley & Sons, Inc., New York, 1993.

[10] Vincent F. Miranda. Computerized simulations for introductory geography instruction: Mental Mapping. Master's thesis, Virginia Polytechnic Institute and State University, 1991.

[11] Arthur H. Robinson, Jeol L. Morrison, Phillip C. Muehrcke, A. Jon Kimerling, and Stephen C. Guptill. *Elements of Cartography.* John Wiley & Sons, Inc., New York, sixth edition, 1995.

[12] Andrew S. Tanenbaum. *Operating Systems: Design and Implementation.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1987.

[13] Edward R. Tufte. *The Visual Display of Quantitative Information.* Graphics Press, Cheshire, Connecticut, 1983.

[14] Matthew O. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proceedings Visualization '94*, pages 326–333. IEEE Computer Society Technical Committee on Computer Graphics and ACM/SIGGRAPH, October 1994.

Appendix A
Sense of Place User's Manual

Project GeoSim: **Sense of Place** User's Manual, Version 1.0

A.1 Introduction

This manual describes **Sense of Place**, an interactive simulation program developed for use in geography courses. **Sense of Place** was developed as part of **Project GeoSim**, a multidisciplinary effort involving members of the Departments of Geography and Computer Science at Virginia Tech to develop computer-aided instruction (CAI) modules for teaching introductory geography to first year college students A **Project GeoSim** instruction module typically consists of two parts: a multimedia tutorial of background information about the geography concepts involved in the lesson, and a simulation program that allows students to apply, experiment with, and enhance the knowledge they have gained.

The main goal of **Sense of Place** is to allow users to gain a sense of the characteristics of U. S. counties by examining statistical data via a variety of simple data visualization techniques. Currently, **Sense of Place** makes use of three visualization techniques: choroplethic mapping, plotting statistical distributions, and tabular representation. However, **Sense of Place** is also designed so that other techniques may be added in the future.

Each visualization technique is incorporated into a *view.* There are typically two views on screen at a time. In short, a view comprises a particular visualization technique and the user interface tools associated with it. The services these tools provide include easy ways of moving from one view to another. For example, once a user finds the symbol for a division in one view, he or she can click on it to find that division in the accompanying view.

The visualization tools in **Sense of Place** can be used to evaluate counties in terms of either raw statistical data, or user-created *metrics.* A metric is a statistical standard composed of target values for a set of statistics. **Sense of Place** determines a score for each county based on how close its values are to the user's targets. Metrics can be created for a variety of purposes. For example, a user can create a metric that specifies the characteristics of his or her "dream" home or travel destination, or a metric that identifies the important qualities of a successful business location.

We use the term "county" rather broadly to refer to counties, parishes, and borough, as well as cities which are are politically independent of any county, since their data would otherwise go uncounted. This is true, for example, of all cities in Virginia, as well as Washington, D.C.

A.1.1 Characteristics of Geographical Data

Each visualization technique used in **Sense of Place** was chosen to show certain aspects of the data associated with geographical divisions. Some of these aspects are inherently geographical. For instance, geography is unique among disciplines in its concentration on the spatial characteristics of data. Other aspects of geographical data are common to other disciplines. For example, the distributions associated with variables are of interest to all fields of statistical study. It would be difficult to present all of these aspects with one data representation technique. Therefore, we have incorporated several visualization tools into **Sense of Place**: maps, distribution plots and tables.

Section A.2 presents a tutorial introduction to the operation of **Sense of Place**. You are encouraged to run through the program as you read it.

A.2 Sense of Place Tutorial

A.2.1 Starting Sense of Place

To start **Sense of Place** on a DECstation, SPARCstation, or MS-DOS PC, type `snsplace` at the command prompt, and hit the return key. To start **Sense of Place** on a Macintosh, open the folder labeled **exe** in the **snsplace** folder, and double-click on the icon labeled **snsplace**. The screen should appear as shown in Figure A.1 when you start **Sense of Place.**

Figure A.1 draws attention to several important screen areas. The various **Sense of Place** data visualization techniques are presented in the top and bottom view slots. Context-sensitive hints about how to use **Sense of Place** are shown in the message window. The menu buttons are used to access popup menus which are used to control the program.

When you start **Sense of Place**, the top view slot shows the Graph View, and the bottom view slot shows the Map View. Note that there is no data shown in either view when the program begins. Thus, there is only a pair of axes in the Graph View, and all states and counties are colored gray in the Map View. Section A.2.2 describes how you can select data for viewing.

A.2.2 Selecting Data

To select county data for viewing, press [Data] in the row of menu buttons at the top of the screen. This button causes a popup menu to appear when pressed. This menu is shown in Figure A.2. Press [Data] now to access the menu. The items in the menu are listed alphabetically. Each item tells **Sense of Place** to perform some service for you. The service that corresponds to each item is described in the following list.

- **Examine variable by county** allows you to select a single variable. Once you select a select a variable, **Sense of Place** shows you its values for every county.
- **Find county on graph** allows you to locate the datum for a county in the Graph View. This item is not active when the program first starts, because there is no data shown in the Graph View. The item is printed in gray to show that it is inactive.
- **Find county on map** allows you to locate a county on the county map in the Map View. This item *is* active when the program starts, because counties can be shown on a map even when it is not showing data.
- **Rank counties** allows you to select a set of variables for a *ranking metric.* A ranking metric consists of a set of target values (one for each selected variable) used to evaluate

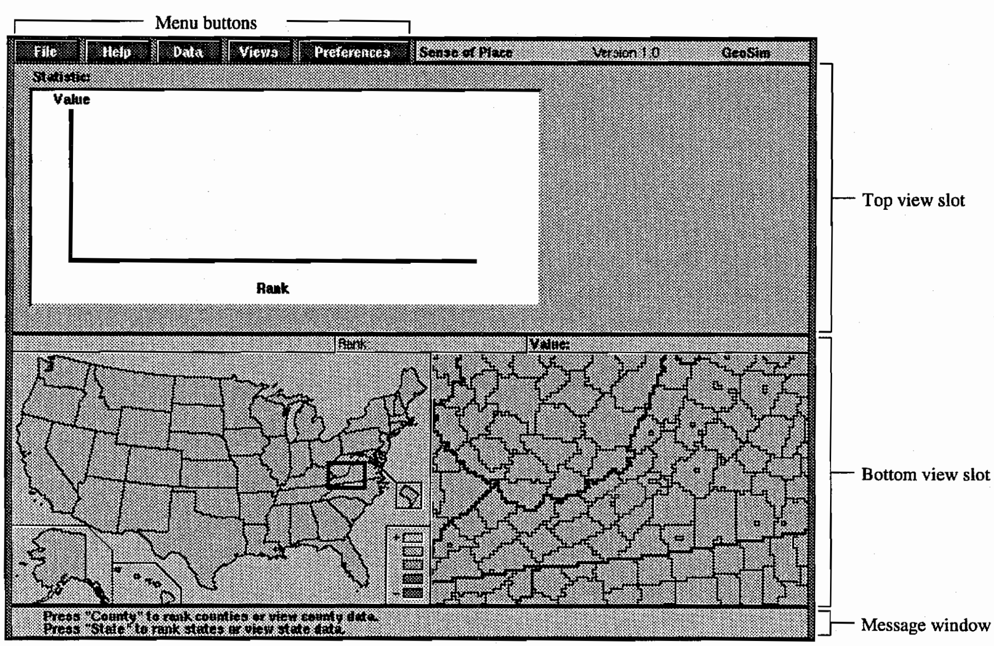


Figure A.1: Initial screen layout

counties. Each county is given a *metric score* which reflects how close its statistical values are to the target values.

- **Show all statistics for a county** allows you to select a single county. Once you select a county, **Sense of Place** shows you a list of all statistical values in our database for that county.

To get a better idea of what the items in the county menu do, try using each of them. Each of the following sections will walk you through the use of one of the items on the county menu.

A.2.2.1 Examine variable by county

When you select this item, the list window shown in Figure A.3 will appear. The window contains two list areas. On the left is the subject list area. This area will contain a list of statistical subjects from which to choose. On the right is the variable list area. It is blank when the window first appears. When you select a subject, the variable for that subject are listed in the variable list area.

Note that the subjects and variables you see on your screen may not be the same as the ones in Figure A.3, since the list of variables is dynamic. Since there are more items than can be shown at one time in the limited screen space available, the lists are scrollable. The arrow buttons move the list up or down one item at a time. Try pressing the down arrow button, and then the up arrow button now to see how they work.

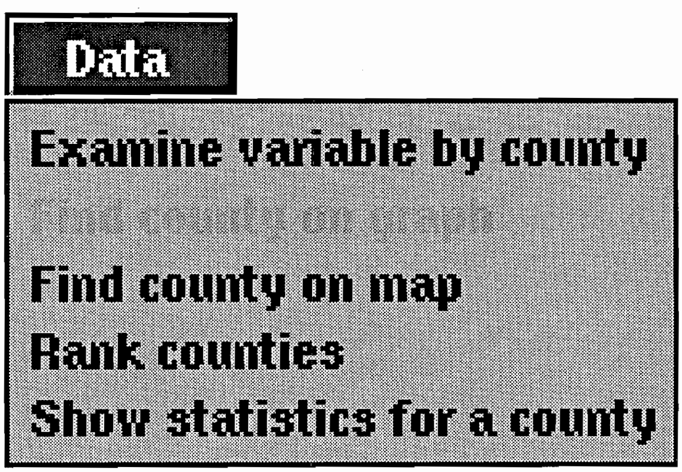


Figure A.2: Data menu

The arrow buttons will allow you to go to any item on the list, albeit rather slowly. If you want to go faster, use the scroll bar. To use the scroll bar, place the mouse cursor over the gray scroll button, hold the mouse button down, and drag the scroll button up or down as desired. Try this now to see how it works.

Once you have found the item you want, place the mouse cursor over it and press the mouse button once to select it. When you select an item, it is highlighted in the list and echoed in the selection field above the list. Try selecting a subject and then a variable now. You can change your selection easily by clicking on another item. Select another variable. Note that the highlight moves from the old selection to the new one, and the selection field is updated to show the new selection. You can also change subjects if you don't see a variable you like. Once you are satisfied with your selection, press [Done]. You can press [Cancel] at any time to cancel the selection process and go back to where you were.

When you press [Done], the list window disappears, and **Sense of Place** retrieves the data for the selected variable from its database. A splash window appears to show the progress of the data retrieval. Once all the data have been retrieved, the splash window disappears, and the Graph View and the Map View are filled in. Figure A.4 shows the Graph View and the Map View for the variable **Persons reporting United States ancestry per 100,000.**

Note that the Graph View now shows a plotted curve, a list of counties, and an indicator bar. The indicator bar shows the graph position of the counties in the list. The list can be scrolled just like the variable list. Alternatively, you can drag the indicator bar to any position on the graph, and the list will automatically scroll to show the counties at that position. Try moving the indicator bar, and note how the list changes. Note that the list also contains the precise value for each county.

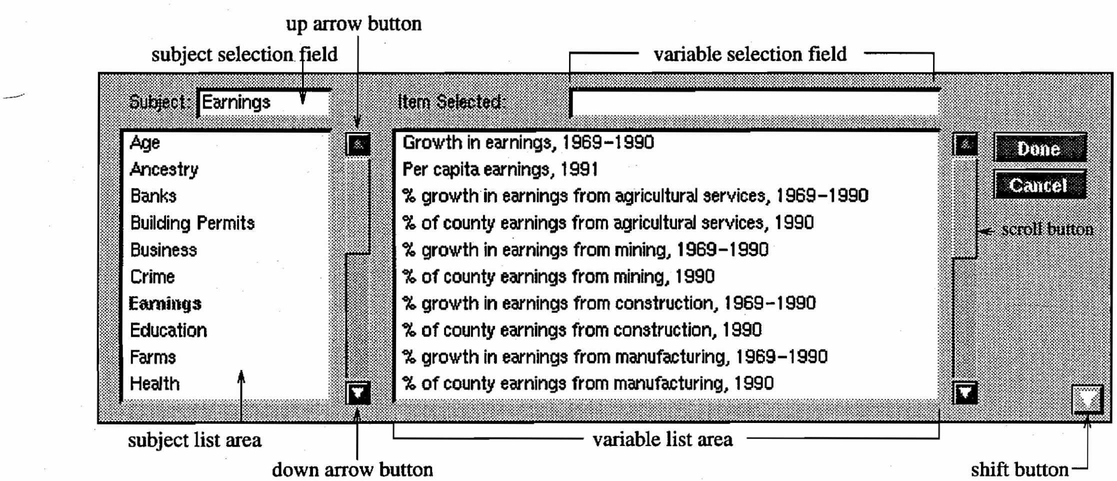


Figure A.3: Statistic list

Also note that the maps in the Map View are colored according to the data. The color scheme is shown at the bottom right of the state map (the one on the left). The brighter the color, the higher the value for the county. The county map (on the right) shows the portion of the state map that lies inside the box on the state map. This box can be moved with the mouse to any position. Try moving it now — by holding the mouse button down and moving the mouse — and watch how the county map changes. Note also that when the mouse cursor is over a state, that state's name is shown in the information bar above the maps.

The information bar also contains value and rank fields, which are used to show data values and ranks for states and counties. "Value" is currently highlighted because the coloring of the maps has been calculated according to the statistical values for all the counties. (The coloring can also be calculated according to rank, as we will see in Section A.2.4.1.) Try placing the mouse cursor over a county in the county map. Notice that, in addition to the county's name, its value for the variable and its rank among all counties is shown in the information bar.

We will examine other aspects of the Graph View and the Map View later. Let's go on and look at the other items in the county menu.

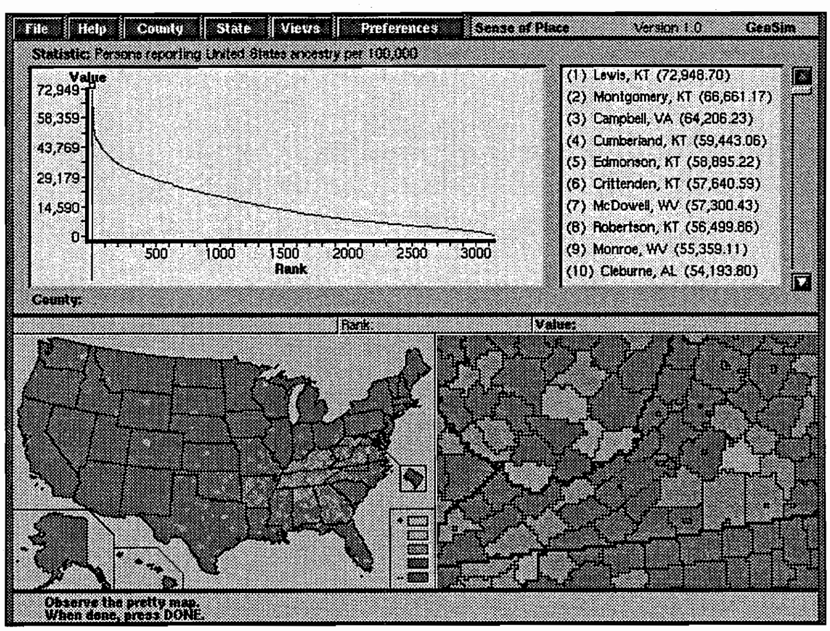


Figure A.4: Graph View and Map View for **Persons reporting United States ancestry per 100,000**

A.2.2.2 Find county on graph

If you look back at the county menu, you will notice that this item now appears in black. This means that you can select this item to find the position of a particular county on the graph. Go ahead and select this item now.

You should see window with a list of state names on the left, and a blank list on the right. The blank list will be used to show counties after you select a state. The state list works just like the variable list: you can scroll it up and down, make a selection, and change your selection once you've made it. Try selecting a state now. (The state you live in might be a good one to pick.)

When you select a state, the name of the state is highlighted, and echoed in the selection field above the list. The counties in the selected state are then shown in the list on the right. The selection process for a county is the same as a state. Try selecting a county now. (Again, we suggest picking the county you live in.)

Note that you can change states or counties at any time during the selection process. (Of course, if you change states, you will have to reselect a county as well!) **Sense of Place** doesn't actually look for the selected county until you press [Done]. You can also press [Cancel] at any time to cancel the search. Go ahead and press [Done] when have are satisfied with your selection. (Make sure that you have selected both a state *and* a county.)

When you press [Done], the list window disappears, and **Sense of Place** looks for the position of the selected county on the graph. Once the position is found, the indicator bar is moved there, and the list is scrolled to show the highlighted name of the county. You can try finding several more counties now if you like.

A.2.2.3 Find county on map

This item works much the way **Find county on graph** does. You select a county in exactly the same way. The movable box on the state map is then moved to the position of the selected county, and the county's border is highlighted (shown in red instead of black) on the county map. Try finding the county in which you live.

A.2.2.4 Rank counties

The variable list appears when you select this item as it does when you select **Examine variable by county**. The difference here is that you can (and should) select more than one variable. You can pick as many as eight variables for a ranking metric. (The number of variables currently selected is shown at the top of the list window.) Again, press [Done] once you are satisfied with your selections.

After you press [Done], you should see the the Ranking Window, which is shown in Figure A.5. The Ranking Window is used to set target values for the selected variables. It is split into eight sections, each of which may[1] contain a pair of "odometers" for a selected variable. The leftmost of the two odometers is used to set the target value for the associated variable. The odometer on the right is used to set the importance of the variable relative to the other variables in the metric.

The target value for each variable is shown in the boxes of the odometer on the left. The mean value is initially selected as the target value, but you can change any digit of the value by clicking on it. To increase a digit, click on the top of its box. Click on the bottom of the box to decrease the digit. You can also click on the min, mean, and max boxes to quickly go to these values. Although you are limited to target values that are within the range of observed values for a variable, this limitation is usually not constricting. The range of observed values usually includes *outliers* (unusually small or large values) at both ends.

Once you have set the target value and importance for each variable, press [Run Metric]. **Sense of Place** will then compute a score for each county based on how closely its values match the target values. Scores are between 0 and 1000 (the lower the score the better). This takes a little time, so a splash window appears to show the progress of the computation. When the computation is done, the current views are redrawn to show the metric scores. The views we have seen so far are the Graph View and the Map View. There is also a third view called the Spreadsheet View, which will be described in Section A.2.2.6.

[1]The first n sections are used (starting from the top left and working first down and then to the right) for n variables, where n is a positive integer between one and eight.

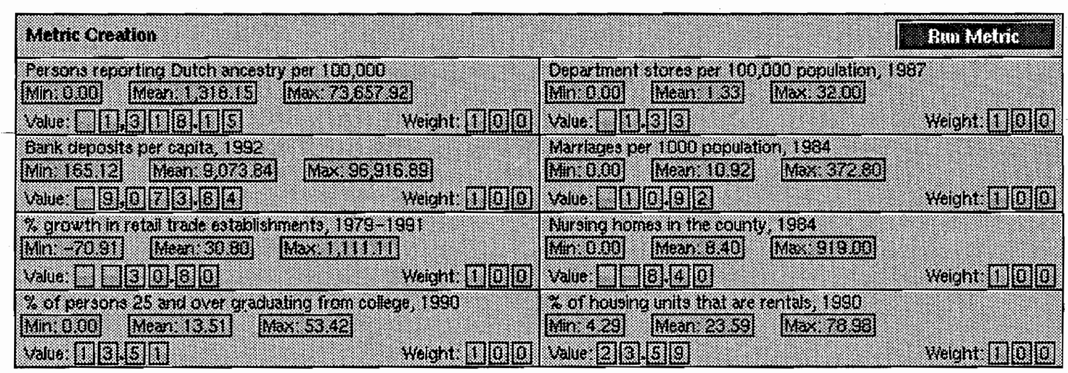


Figure A.5: Ranking Window

Note that a metric score by itself does not mean much. It is mainly useful for comparative purposes.

A.2.2.5 Show all statistics for a county

This item allows you to look at all data for a selected county. The selection process is the same as that used for **Find county on graph** and **Find county on map**. When you press [Done] after selecting a county, the variable list is shown with the values for that county. Section A.2.2.1 describes how to select variables from this list.

A.2.2.6 The Spreadsheet View

So far, we have looked at two views: the Map View and the Graph View. We have also looked at the Ranking Window, which can also be considered a view, since it shows the ranges and mean values for the metric variables. We will now look at the fourth and final **Sense of Place** view: the Spreadsheet View. We will also look at how you can switch back and forth between views.

We are introducing the Spreadsheet View last because it is the last one to become available when you run the program. It is used to show the details of ranking metric scores, so you cannot access it until you run a metric.

The Spreadsheet View contains a scrollable spread sheet which lists all counties one to a line. Each line contains the name of a county, its metric score, and the county's values for all the variables in the metric. The Spreadsheet View is shown in Figure A.6. County names and ranks (in parentheses) are shown in the leftmost column. Metrics scores are

Statistic: Persons reporting French Canadian ancestry per 100,000: 58,365.58

(3067) Thomas, GA	997.73	15.11	120.90
(3068) Wilkinson, N	997.74	0.00	132.31
(3069) Jackson, TX	997.74	166.67	0.00
(3070) Washington	997.76	54.94	58,365.58
(3071) Tama, IA	997.76	73,657.92	87.65
(3072) Grant, ND	997.78	163.67	0.00
(3073) Lee, TX	997.82	160.62	0.00
(3074) Newberry, S	997.84	106.61	41.88
(3075) Edgefield, S	997.85	70.16	70.16
(3076) Sierra, NM	997.85	117.22	32.82
Desired Values:	0.00	36,856.43	29,226.62

Figure A.6: Spreadsheet View

displayed in the second column from the left. Values for the metric variables are shown in the remaining columns. Note that the columns do not have headers. Since variable names are generally fairly lengthy, there is only room enough to show one name at a time. When you aim the mouse pointer at a variable's column, the name of the variable name is shown above the spread sheet. The words "Metric Score" appear at the top of the window when the pointer is in the metric score column. When the pointer is in the county name column, the contents of the cell that it's in are echoed above the spreadsheet.

The counties are initially listed in ascending order by metric score, but you can also sort them by any one of the variables in the metric. Variable values are sorted by how closely they meet the target value. To sort by a variable, click on its target value at the bottom of the Spreadsheet View (the target values are aligned with the spread sheet columns). There is also a target value (always 0.00) listed for metric scores which you can click on to go back to the original sorting.

A.2.3 Changing Views

Since there are more views than slots in which to place them, there is a menu for selecting the views that appear on the screen. This menu — called the "Views" menu — is shown in Figure A.7. To access the menu, press [Views] at the top of the screen.

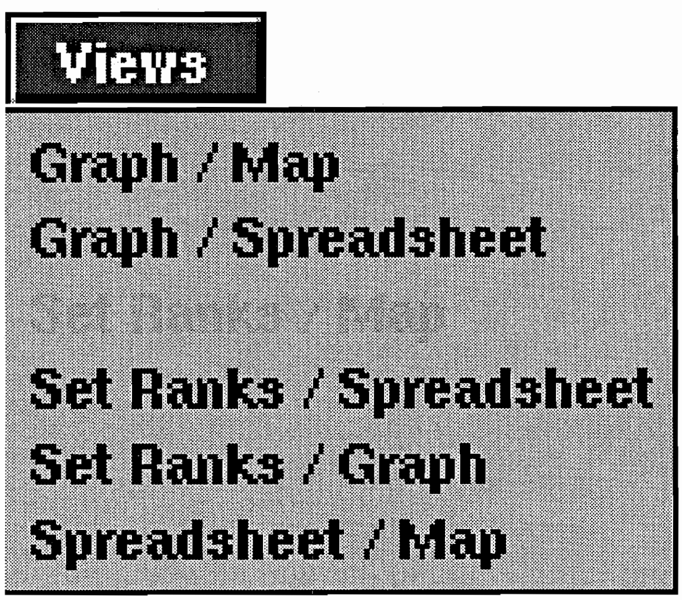


Figure A.7: Views menu

The Views menu is accessible at any time during the program, but it's items are inactive until you run a ranking metric. If you haven't run a metric yet, you might do so now, and observe how the menu changes after you do. Each item lists a pair of views, separated by a slash (/). When you select an item, the view to the left of the slash is placed in the top view slot, and the view to the right is placed in the bottom.

You may notice that one of the items is still inactive in Figure A.7. This item lists the two views that are currently active. The item that corresponds to the currently active views will always inactive, since it makes little sense to re-select those views.

A.2.3.1 Coordination of Current Views

We have tried to coordinate the interaction tools in the respective views. For example, if the Graph View and the Map View are currently active (and show data), you can click in a county on the county map, or on the name of a county in the list in the Graph View, and the county will be highlighted in both views. In fact, you can do the same with any pairing of the Map View, the Graph View, and the Spreadsheet View.

The one view that is not necessarily coordinated with the others is the Ranking Window. The other three views always depict the same data. However, while you are creating a new metric with the Ranking Window, it may be out of synch with the other views. For instance, the Spreadsheet View may show variables and target values from a previous metric; it will not be updated to match the Ranking Window until you run the new metric.

A.2.4 User Preferences

Pressing [Preferences] accesses a menu of attributes that you can set to suit your preference. Most of the items on the menu belong to a pair of opposites. For example, the item **State map borders: Remove** is the counterpart of the item **State map borders: Add.** Only one item of such a pair is visible at any given time. For example, if the state map currently has borders on it, only **State map borders: Remove** will appear on the menu. The preferences menu is shown in Figure A.8. The following list describes the items on the menu.

- **County map magnification: Decrease.** This item decreases the magnification factor[2] of the county map by one. This item is deactivated when the magnification factor reaches one.

[2]This factor expresses the relationship between the scale of the county map as it appears, and the actual scale of the county map as stored in its image file. doesn't express any relationship between the scales of the state and county maps.

- **County map magnification: Increase.** This item increases the magnification factor of the county map by one. This item is deactivated when the magnification factor reaches eight.

- **Graph curves: Discrete.** This item causes the graph to be drawn as a series of discrete points. The value for every tenth county is plotted as a point, because there is too little room to plot values for all 3,141 counties. This item is the counterpart to **Graph curves: Continuous.**

- **Graph curves: Continuous.** This item causes the graph to be drawn as a continuous line that connects the values for every tenth county. This item is the counterpart to **Graph curves: Discrete.**

- **Map drag box: Thin lines.** This item causes the movable box on the state map to be drawn one pixel thick. This item is the counterpart to **Map drag box: Thick lines.**

- **Map drag box: Thick lines.** This item causes the movable box on the state map to be drawn three pixels thick. This item is the counterpart to **Map drag box: Thin lines.**

- **Map range-grading: Quantiles.** This item selects rank quantile range-grading for the Map View. This is discussed further in Section A.2.4.1. This item is the counterpart of **Map range-grading: Equal intervals.**

- **Map range-grading: Equal intervals.** This item selects equal interval range-grading for the Map View. This is discussed further in Section A.2.4.1. This item is the counterpart of **Map range-grading: Quantiles.**

- **State map borders: Remove.** This item removes borders from the state map. This item is the counterpart of **State map borders: Add.**

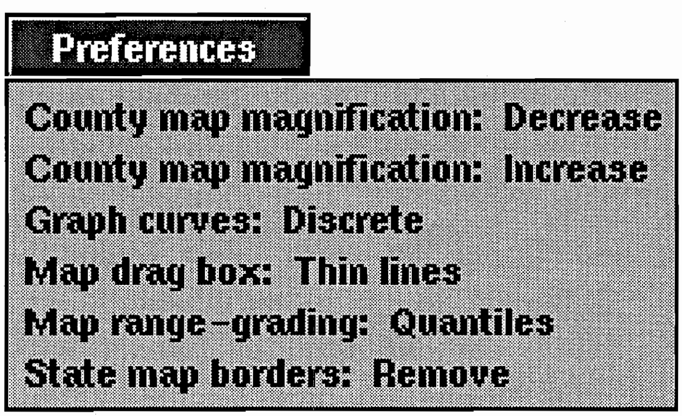


Figure A.8: Views menu

- **State map borders: Add.** This item adds borders to the state map. This item is the counterpart of **State map borders: Remove.**

A.2.4.1 Map Range-Grading

Most of the items on the preferences menu should be fairly straight-forward. However, the range-grading items warrant some explanation. The maps in the Map View are colored according to one of two classification schemes. Each scheme has its pros and cons, depending on what information you're seeking. The first scheme is called "equal intervals". The total range of a set values is divided into several equal-sized intervals. Each county is colored according to the interval into which its value falls. The important thing to note about this scheme is that it separate counties into groups which are not generally equal in size, because the values are not generally uniformly distributed. Because outliers increase the size of the intervals, they can greatly affect the results of this scheme. This is good if you are looking for outliers, but not so good if you want to see subtle differences at other parts of the range.

The other classification scheme is called "rank quantiles". In this scheme, values are ranked in either descending or ascending order, and the total range of ranks is divided into several equal-sized intervals. This scheme guarantees equal-sized groups (± 1), because the ranks *are* uniformly distributed. This means that outliers have no affect on rank quantiles, so you can see subtle differences at all points in the range of values. However, it also means that outliers are inapparent under this scheme.

A.2.5 Final Comments

There are two more important menus: the Help menu and the File menu. The Help menu offers information about the program, and assistance in operating it. The File menu allows you to print the contents of the screen to a file, or quit the program.

A.2.5.1 Help Menu

You can access the Help menu by pressing [Help]. There are two items on this menu. **Sense of Place Info** accesses information on Sense of Place, including the current version of the program. **Tutorial Help** accesses a text version of this manual.

A.2.5.2 File Menu

You can access the File menu by pressing [File]. There are two items on this menu. **Quit** causes execution of the program to terminate. **Save Screen** allows you to save the image on the screen in several different formats. After you select **Save Screen**, a dialog box will appear with buttons for changing the file format, saving the screen, or canceling the operation. When you press [Save], you should see a white line run down each side of the screen. When the save operation is finished, a window will appear with the name of the output file on it. Press [Done] in this window to continue with the program.

Appendix B

Sense of Place Instructor's Manual

Project GeoSim: **Sense of Place**
Technical Manual, Version 1.0

Sense of Place uses several text files for configuration and data. This manual describes those files and their formats. A **Sense of Place** user may change the configuration and/or data files.

B.1 Sense of Place Configuration File: `config.gcf`

The file `config.gcf` must be in the same directory (or folder on Macintosh) as the **Sense of Place** executable file (`snsplace.exe` on MS-DOS, Sense of Place on Macintosh and `snsplace` on Unix). This configuration file is a plain ASCII text file that may be edited with any text editor software.

`config.gcf` is a **Project GeoSim** database file. The format of an entry in a **Project GeoSim** database file is:

parameter= value

where *parameter* is a string that is the name of the parameter, such as *database*, and *value* is a string that represents the value that the parameter is assigned. Values include file names, path names, numbers and Boolean values (true or false). This is an example of an entry in the file:

```
database= SP_config
```

Any line in the file that begins with a pound sign (#) is a comment that will be ignored when **Sense of Place** reads the file. If you want to temporarily "remove" a line, simply place a pound sign in front of it. When you want to restore the line, remove the pound sign. The following are the valid parameter names that **Sense of Place** uses. Each description is followed by the entry that `config.gcf` originally contained when first installed.

DataBase is the name of this **Project GeoSim** database. It must be the first entry in the file and its value must always be `SP_config`.

`DataBase= SP_config`

DataPath is the path to the **Sense of Place** data files. This path is relative to the directory in which the executable program (e.g. `snsplace.exe`) resides. Directory or folder names in the path must be separated by a *forward slash*: `/`. Do not use a back slash (\) or colon (:) to separate names in the path.

`DataPath= dbase/`

InfoPath is the path to the help files. This path is relative to the directory in which the executable program (e.g. `snsplace.exe`) resides. Directory or folder names in the path must be separated by a *forward slash*: `/`. Do not use a back slash (\) or colon (:) to separate names in the path.

`InfoPath= info/`

InterfaceFile is the name of the file that contains the user interface information, such as window and button names and positions.

`InterfaceFile= snsplace.inf`

StMapFile is the name of the state map image file. The state map appears in the lower left hand corner of the screen when the program begins. The value for this parameter should not be changed.

`StMapFile= stmap.rlc`

CntyMapFile is the name of the county map image file. The county map appears in the lower right hand corner of the screen when the program begins. The value for this parameter should not be changed.

`CntyMapFile= cntymap.rl0`

BStore is a Boolean parameter. If BStore = true, then **Sense of Place** will use backing store to save and restore the screen when windows pop up and disappear. If BStore = false, then **Sense of Place** will redraw the screen when a pop-up window is closed. Using backing store (i.e., BStore = true) is faster, but requires more RAM. If you are short on RAM, set BStore to false.

`BStore= true`

ConfirmQuit is a Boolean parameter. If ConfirmQuit = true, then the user will always be asked to confirm a request to quit the program. If ConfirmQuit = false, then the program will terminate immediately whenever the user selects "Quit" from the "File" menu.

```
ConfirmQuit= true
```

YourState and YourCounty are the names of your home state and county, respectively. The maps feature the county and state named with these parameters when the program begins. We have set these parameters to the home county and state of Virginia Tech. You should reset them to your own state and county.

```
YourState= VA
YourCounty= Montgomery
```

IntroHelpDesired is a boolean parameter. If IntroHelpDesired = true, then a brief help screen is shown whenever the program begins. If IntroHelpDesired = false, the help screen is bypassed.

```
IntroHelpDesired= false
```

MapBoxLineWidth is the width frame of the movable map box on the state map. This portion of the state map inside this box is shown (at greater magnification) in the county map. The legal values for this parameter are `thick` and `thin`.

```
MapBoxLineWidth= thin
```

CntyMapMagnify is the default magnification of the county map when the program begins. For example, if CntyMapMagnify = 3, then the county map is drawn at 3 times its actual size. The legal values for this parameter are integers in the range 3-8.

```
CntyMapMagnify= 3
```

NDataSections is the number of database sections currently in use. This number must represent the number of DataSection (see below) entries in the configuration file.

```
NDataSections= 23
```

DataSection is the name of a section of the database. There is a DataSection declaration for each section of the database. The name of a section indicates what type of variable it contains. You can omit a section by removing its declaration from the configuration file. (Remember that you can temporarily remove a declaration by putting a pound sign in front of it.) Remember to change the NDataSections value if you remove a data section. You may notice that some DataSection declarations have some additional text enclosed in "pipes" (|). Section B.2.2 explains how this extra text is used.

```
DataSection= age
DataSection= ancestry
DataSection= banks
DataSection= bpermits |Building Permits|
DataSection= business
```

```
DataSection= crime
DataSection= earnings
DataSection= educ |Education|
DataSection= farms
DataSection= health
DataSection= household |Households|
DataSection= housing
DataSection= income
DataSection= laborfc |Labor Force|
DataSection= localgov |Local Government|
DataSection= physical
DataSection= pop |Population|
DataSection= poverty
DataSection= retail
DataSection= service
DataSection= unemploy |Unemployment|
DataSection= vitals
DataSection= votes
```

B.2 Data Files

Each database section specified in **config.gcf** has a pair of files associated with it: a dictionary file, which contains information about the section's variables, and a data file, which contains the actual data. Both files of a pair must share the same prefix, which is usually a name that describes the general subject for the section, such as "ancestry", or "banks". The suffix for dictionary files is **.spd**, which stands for "**S**ense of **P**lace **d**ictionary file". The suffix for data files is **.spb**, which stands for "**S**ense of **P**lace **b**inary data file".

These files must be in the directory path specified by the **DataPath** entry in **config.gcf**, which is relative to the directory or folder that contains the executable program (usually /snsplace). Upon installation, the directory path is **dbase/**. Typically then, the full path and file name for the banks data file, for instance, is **/snsplace/dbase/banks.spb**.

B.2.1 Excluding a Variable from a Database Section

A variable can be "removed" from a database section by editing its dictionary file. "Removing" a variable means excluding it from the selection list in the program, so that students will never see it. Each dictionary file has two columns — labeled "CntyUse" and "StUse", respectively — which tell **Sense of Place** whether or not to include variables. A "1" in the

"CntyUse" column signals **Sense of Place** to include a variable, while a "0" indicates that the variable should be excluded. The same is true for values in the "StUse" column. For example, here are the steps to take to remove "Bank deposits per capita" from the county variable list:

1. Find the file `banks.spd`.
2. Find the "LongLabel" column.
3. Find the row with "Bank deposits per capita" in the "LongLabel" column.
4. Change the "1" in this row and the"CntyUse" column to "0".

B.2.2 Creating Your Own Database Sections

You can create your own sections for the **Sense of Place** database. To do so, you need to create a dictionary file and data file for the section. **Sense of Place** data files are stored in binary — rather than text — format. Binary format has several advantages over text format: it makes the data files smaller, it allows the program to read them faster, and it makes the program itself smaller.

The main disadvantage of binary files is that you cannot use a text editor to create or read them. To remedy this problem, we have created utility programs that translate binary files into text format, and text files back into binary format. Thus, you can translate one of our data files into text format, and use it as a model for creating your own (with a text editor). Then you can translate your text file into binary format, so that **Sense of Place** can read it. The name of the program that translates binary files into text format is `spb2spa`. The program that translates text files into binary format is called `spa2spb`. The names of these programs are derived from the suffixes for binary files (`spb`) and text files (`spa`, which stands for "Sense of Place ascii data file".) The dictionary files are text files, so you can create and read them with a text editor.

Here is a summary of the procedure for creating your own database sections:

1. Collect the data.
2. Create a dictionary file for the section. Use one of our dictionary files as a model.
3. Create an ascii text version of your data file. Use `spb2spa` to obtain an ascii text version of one of our files to use as a model.
4. Translate your text file into binary format with `spa2spb`. (Note that the dictionary file must be present in the current directory for `spa2spb` to work.)

5. Place the dictionary and data files in the database directory with our files.

6. Create a DataSection entry for your section in `config.gcf`. Use the prefix for the two files as the value for the entry. If the prefix is too short for **Sense of Place** to use in its selection list, you can put a longer name after the entry. For example:

   ```
   DataSection= bpermits |Building Permits|
   ```

   The name must be enclosed in "pipes" (|).