PROBE-BASED VISUAL ANALYSIS OF GEOSPATIAL SIMULATIONS

by

Thomas James Butkiewicz

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Information Technology

Charlotte

2010

Approved by:

____
Dr. Zachary Wartell

____
Dr. William Ribarsky

____
Dr. Robert Kosara

____
Dr. Remco Chang

____
Dr. Ross Meentemeyer

ABSTRACT

THOMAS JAMES BUTKIEWICZ. Probe-based visual analysis of geospatial simulations. (Under the direction of DR. ZACHARY WARTELL)

This work documents the design, development, refinement, and evaluation of probes as an interaction technique for expanding both the usefulness and usability of geospatial visualizations, specifically those of simulations. Existing applications that allow the visualization of, and interaction with, geospatial simulations and their results generally present views of the data that restrict the user to a single perspective. When zoomed out, local trends and anomalies become suppressed and lost; when zoomed in, spatial awareness and comparison between regions become limited. The probe-based interaction model integrates coordinated visualizations within individual probe interfaces, which depict the local data in user-defined regions-of-interest. It is especially useful when dealing with complex simulations or analyses where behavior in various localities differs from other localities and from the system as a whole. The technique has been incorporated into a number of geospatial simulations and visualization tools. In each of these applications, and in general, probe-based interaction enhances spatial awareness, improves inspection and comparison capabilities, expands the range of scopes, and facilitates collaboration among multiple users. The great freedom afforded to users in defining regions-of-interest can cause modifiable areal unit problems to affect the reliability of analyses without the user's knowledge, leading to misleading results. However, by automatically alerting the user to these potential issues, and providing them tools to help adjust their selections, these unforeseen problems can be revealed, and even corrected.

ACKNOWLEDGMENTS

This work is dedicated to my parents for always encouraging me to excel academically and enabling me to pursue the highest education possible.

I would like to thank all those who taught and mentored me over the years, including: William Ribarsky, Zachary Wartell, Robert Kosara, Sharon Stansfield, Larry Hodges, Min Shin, Richard Souvenir, Jamie Payton, John Barr, Wanda Dann, David Brown, Michael Rogers, Stephen Clancy, and Roger Davis. For the enriching discussions, inspiration, and great brainstorming sessions, I'd like to thank all of my colleagues and collaborators, including: Remco Chang, Evan Suma, Xiaoyu Wang, Wenwen Dou, Caroline Ziemkiewicz, Dong Jeong, Douglas Shoemaker, Ross Meentemeyer, Jeff Michael, John Chesser, Aaron Houck, Alex Godwin, Ted Carmichael, and Lane Harrison. I could not have gotten where I am today without all of these people.

Funding for my research was provided by: The Renaissance Computing Institute, The U.S. Department of Homeland Security, The Defense Advanced Research Projects Agency (DARPA) of the U.S. Department of Defense (U.S. DoD), and The Army Research Office under contract no. W911NF-05-1-0232.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CAGIS	The Center for Applied Geographic Information Science
ESRI	Environmental Systems Research Institute
DOI	Degree-of-Interest
DSS	Decision Support System
GIS	Geographic Information System
GUI	Graphical User Interface
IR	Infrared
LIDAR	Light Detection and Ranging
MAUP	Modifiable Areal Unit Problem
RENCI	The Renaissance Computing Institute
ROI	Region-of-Interest

CHAPTER 1: INTRODUCTION

Traditional geospatial information visualizations often present views that restrict the user to a single perspective. When zoomed out, local trends and anomalies become suppressed and lost; when zoomed in for local inspection, spatial awareness and comparison between regions become limited. In the interaction model described in this work, coordinated visualizations are integrated within individual probe interfaces, which depict the local data in user-defined regions-of-interest. This probe concept can be incorporated into a variety of geospatial visualizations to empower users with the ability to observe, coordinate, and compare data across multiple local regions. It is especially useful when dealing with complex simulations or analyses where behavior in various localities differs from other localities and from the system as a whole. The effectiveness of this technique over traditional interfaces is illustrated by incorporating it within three existing geospatial visualization systems: an agent-based social simulation, a census data exploration tool, and a 3D Geographic Information System (GIS) environment for analyzing urban change over time. In each case, the probe-based interaction enhances spatial awareness, improves inspection and comparison capabilities, expands the range of scopes, and facilitates collaboration among multiple users.

Finally, a more mature, full-featured implementation is presented in the form of a probe-based interface for the exploration of the results of a geospatial simulation of urban growth. Because this interface allows the user great freedom in how they choose to define regions-of-interest to examine and compare, the classic geospatial analytic issue known as the modifiable areal unit problem (MAUP) quickly arises. The user may delineate regions with unseen differences that can affect the fairness of the comparisons made between them. To alleviate this problem, the interface first alerts the user if it detects any potential unfairness between regions when they are selected for comparison. It then presents the dimensions with potential problematic outliers to the user for evaluation. Finally, it provides a number of semi-automated tools to assist the user in correcting their regions' boundaries to minimize the inequalities they feel could significantly impact their comparisons.

1.1: Theory

A similarity across the majority of GIS applications and geospatial visualizations is the singularity of the viewing perspective. For example, in map-based visualizations, the user is generally restricted to viewing one region of the map at a particular zoom-level. When zoomed out to see the entire extent of the dataset, local trends and anomalies, which are often of interest, become suppressed and ultimately lost in the global picture, especially as the scale of the dataset increases. To discover and inspect these local details, the user must zoom in to a level at which they become visible. However, by doing so, one loses both the global overview and context of the local region. This both limits the user's spatial awareness and prohibits comparison between distant local regions.

In the model presented within this paper, coordinated information visualizations are integrated directly within the main geospatial visualization. User defined regions-of-interest are linked to each coordinated visualization, delineating which data is presented in each visualization. Furthermore, these interfaces, herein referred to as probes, allow the user to interact directly with the geospatial data within the regions-of-interest as well. By using multiple probes, the user can simultaneously observe and interact with many different local regions across the entire range of scales (ranging from global to the smallest individual units) without losing spatial awareness. This is particularly useful when dealing with complex simulations or analyses in which the local values and behaviors differ greatly from each other and/or the system as a whole.

To illustrate the general usefulness of the probe concept for enhancing geospatial visualizations, it has been incorporated within three unique existing applications. First, probes are applied within a 3D GIS environment used to visually explore the changes (new buildings, etc) detected (using aerial laser range-finding) to an urban area between multiple years. The second application augmented is designed for neighborhood-focused visual analysis of census data across large urban areas. Finally, a new, entirely probe-based interface is developed for an agent-based social simulation that models the various factions and behaviors of an entire country. In each case one can see benefits including uninterrupted spatial awareness, improved inspection and comparison capabilities, ability to view data at multiple scales simultaneously, and increased potential for collaboration among multiple users. These common benefits are elaborated on in further discussion in Chapter 4.1.

Informal user evaluations were performed with experts in both GIS and architecture. Each group was presented with the original applications and their probe-enhanced counterparts. It was obvious that these everyday users of geospatial visualization applications had all encountered some of the shortcomings that are addressed in this work. Their comments confirm that, with the addition of probes, the presented applications become increasingly effective and more intuitive to interact with.

1.2: Development

The probe concept was conceived in 2007, when colleagues studying complex adaptive systems enquired as to a solution to increase the effectiveness of their simulation interface. They were developing the Afghanistan simulation described in Chapter 3.3, but had a very simplistic and clunky interface within which to control and view the simulations behavior. In listening to the way they phrased their social theory narratives and questions, as well as how they described the movement and behaviors of the simulation's underlying agents, it was clear that there was a wide range of scales involved. Furthermore, the behaviors at each scale affected those at others, and awareness of what was going on at all scales was important. Understanding what was going on at different scales simultaneously and the interrelationships between these scales was a primary goal. In addition to the issue of visualizing a wide range of scales, there was also a need to control the simulation precisely and locally.

This lead to the idea of being able to explore and examine the simulation at multiple scales, but this complicates navigation. To zoom back and forth between these scales added a huge interaction cost, and comparison between areas came at a heavy cognitive memory requirement in addition to being susceptible to change blindness. This lead to the idea of selecting regions-of-interest at any particular scale, and having the data within it be visible on demand. The benefit of being able to examine local areas while maintaining contextual awareness was immediate and striking. Striving to further reduce cognitive loads and navigation requirements motivated the additional comparison features.

To explore and evaluate the effectiveness of this new technique, two more existing applications, detailed in Chapters 3.1 and 3.2, were extended to include probes in their interfaces. This greatly enhanced their usability and power for complex analysis.

Finally, collaborators in The Center for Applied Geographic Information Science (CAGIS) requested a probe-based interface be created to bring interactivity and analytics to their urban growth model. This simulation, described in Chapter 2.2, produced high resolution land use maps for historical and forecasted time steps. However, presentation of these results was limited to static maps and simple animations. Analysis of the results was also limited to expensive hard-to-master GIS tools that required one to know what one was looking for before analysis. The Urban Growth Decision Support System, detailed in Chapter 3.4, was developed to provide a highly interactive, intuitive interface for exploratory analysis of the results of their simulation. It was also intended to serve as a presentation tool to communicate the results and analyses to target users and the general public.

1.3: The Modifiable Areal Unit Problem

The Urban Growth Decision Support application seeks to present the results of an urban growth simulation to policy analysts, urban planners, etc. such that they can analyze historical growth patterns, examine predicted trends, and compare the characteristics of development between different regions. It provides the user with the ability to probe the map-based data via selecting regions of any size and shape, resulting in coordinated visualizations reflecting those regions-of-interest, and to directly compare these regions-of-interest with each other.

However, by giving the user this freedom to select regions at such a wide range of shapes and sizes, it can inadvertently make their analyses particularly vulnerable to unforeseen inequalities between regions being compared. For example, household level data, such as income or population, is aggregated into blocks to protect privacy. Depending on how one defines new regions cutting through these blocks, one can find different average values for the same locations. This is part of the long standing problem in the field of geography and spatial analysis, known as the modifiable areal unit problem (MAUP). Probe-based interaction is particularly prone to being effected by MAUP due to the inherent variability in areal units.

The prevalence of the MAUP in the Urban Growth DSS is compounded by the fact that the target audience does not necessarily have expert knowledge regarding all the “behind the scenes” data layers that have gone into guiding and dictating the underlying simulation’s behavior. For example, a policy analyst may understand the zoning limitations that constrain growth in a particular area, but is unlikely to understand the geologic barriers to construction in the same region, i.e. soil suitability and parcel slope.

To help alleviate the effects of the MAUP within the application, a number of enhancements were provided along with the previously available probe-based interface elements. First, when the user selects multiple regions to directly compare against each other, the statistical distributions within the various dimensions are evaluated in search of outliers with deviations that have the potential to be particularly problematic in the final analyses. When these are detected, the user is alerted to their presence and provided with an overview of the possible inequalities in each dimension that may affect their intended analysis. If the user decides that any of these inequalities might have a significant negative impact on their desired analysis, they can then choose to adjust them using a number of provided tools. These tools provide methods to manipulate the boundaries of regions to assimilate and discard land coverage types, grow and shrink in advantageous directions, and trade area amongst selected regions to attempt to bring their disparities within the user's selected bounds.

The usefulness of these enhancements is illustrated with an example scenario in which the analysis of urban sprawl growth patterns for a number of suburbs around a major metropolitan area is complicated by predefined city boundaries containing disproportionate amounts of water and protected land, which the underlying simulation specifically ignore.

CHAPTER 2: RELATED WORK

This chapter provides a review of interaction and visualization work related to probes as an interaction technique, an explanation of the urban growth model and systems related to the Urban Growth Decision Support System, as well as review of the way the modifiable areal unit problem has been evaluated, understood, and dealt with.

2.1: Interaction in Geospatial Analysis

Donelson's [Donelson 1978] Spatial Data Management System presents a large projected display of a 2D graphical information space. The interface is two-handed and supports both panning and zooming. Two joysticks, a tablet, and two secondary monitors that are touch-sensitive are provided as an interface. One monitor displays a "world view" of the entire information space along with a 'you-are-here' rectangle which provides visual context for the user as he views a particular 2D region on the large display. The other monitor, "the key maps monitor," shows auxiliary information such as a chapter outline when the main screen displays text files, or a time-line when the main screen displays video.

Furnas [Furnas 1986] describes generalized fisheye views. In the spatial domain, the metaphor is a fisheye camera lens that shows higher detailed, less distorted imagery toward the center of the field-of-view, and less detailed, compressed imagery toward the outer field-of-view. In additional to the geospatial example of Steinberg's famous poster "New Yorker's View of the United States", Furnas presents experimental studies showing that people's concepts of complex non-spatial structures also exhibit fisheye character. Furnas presents degree-of-interest (DOI) functions to describe fisheye display of information for both spatial and non-spatial data. He also acknowledges the significance in geospatial contexts of supporting multiple foci in fisheye views. He gives an example of a person who has lived in multiple states, whose mental map of the geography is fisheye in character, but with foci at each location in which he has lived. In the context of non-interactive cartography, Kadmon and Shlomi present a mathematical approach for such multi-focal map rendering [Kadmon 1978]. The modifications to the UrbanVis [Chang 2007] tool, described in Chapter 3.2, applies the multiple foci concept, but in 3D, driven by an 'urban legibility' level-of-detail algorithm. Leung and Apperley [Leung 1994] give an overview of distortion based techniques circa 1994.

More recently, Furnas [Furnas 2006] focuses not on the variations of geometric distortion, but on the different degree-of-interest functions and how these determine what information is and is not included in the display. He discusses how this concept can be carried to non-visual domains as well.

Bier et al. [Bier 1993] [Bier 1994] present the Toolglass and Magic Lenses, a see-though 2D, two-handed GUI interface. The Toolglass and Magic Lens are see-though windows whose positions are controlled by the user's secondary hand with a trackball & wheel device. The user's primary hand controls a regular mouse and pointer. Graphical filters in a Toolglass can be overlaid on other objects to reveal alternate visual representations, while the mouse cursor continues to allow direct manipulation of the objects through the Toolglass. Bier et al. cite earlier works with similar concepts of filters for changing information in visualized systems but these earlier works lacked the metaphor of a movable lens. Viega et al. [Viega 1996] extend the concept of Magic Lens to 3D, including both flat, planar lenses and volumetric lenses.

Perlin and Fox [Perlin 1993] introduce the zoomable 2D Pad interface. This interface includes portal filters, which show “non-literal views of cooperating objects.” For instance, when a portal filter is positioned over tabular data, within the portal a bar chart could be displayed.

The concept of probes relates to this prior work as follows: Probes begin with a View+Close-up [Furnas 2006] implementation of the Focus+Context metaphor. However, the user can define, place, and scale multiple regions in the view for which the Close-up windows, or insets, are generated. The interactive manipulation of the regions-of-interest (ROI) boundaries, and the fact that the view geometries within the ROI are drawn in a specialized manner borrows from Magic Lens and portal filters. However, while Magic Lens or portal filters just present an alternate rendering of the selected geometry in the main view, with probes an additional inset window displays secondary representations of the selected data. Unlike a standard View+Close-up inset in cartography, this inset is typically an alternate 2D infovis representation of the data in the ROI. Further, the inset window can contain interactive controls that affect the ROI and the inset’s infovis graphic supports linked brushing.

Compared to a Toolglass, the probe inset pane with interactive controls decouples the ROI from the location of the controls. Significantly, probes are more than just labeled push-pins found in physical and digital 3D maps such as Google Earth [Google 2010]. Push-pins are not areal, and labels are not dynamically varying infovis displays with optional 2D GUI controls. Commercial GIS tools such as ArcGIS [ESRI 2010] provide map views, tabular views, and various basic graphing capabilities, but it is not possible to interactively tie a multiplicity of these latter two view types to a multiplicity of ROIs on the map view.

Linking an information visualization view to a separate map view is an established and effective method of providing spatial context by connecting abstract depictions (e.g. parallel coordinates) with the geospatial origin (e.g. a region on a map) of the data being depicted. Andrienko and Andrienko [Andrienko 1999] present an application in which a choropleth map is coupled with a separate window displaying box-and-whiskers plots of group statistics. Each plot is linked to the choropleth map by common shading color. While similar to probes, in that the plots are linked back to their spatial origins, the basic pair being linked is fundamentally different. As later defined in Chapter 3, the basic unit of this work, a probe, consists of a user-defined region-of-interest and a linked visualization, whereas the basic unit in Andrienko's work is a class. In the map view, each entity or region belongs to one of a specified number of classes. In the linked visualization, there is one plot for each of the same number of classes. Classes do not overlap, are not necessarily connected, and are thus not equivalent to the type of freely user-defined regions-of-interest utilize in probe-based interfaces.

Edsall [Edsall 2003] presents a similar, but more advanced multiple linked views environment with HealthVisPCP, in which he links a parallel coordinate plot with a scatter plot and a choropleth map. The map is colored to indicate each region's classification (into a user-specified number of classes) according to a statistic. Lines in the parallel coordinate plot, and dots in the scatter plot are then likewise colored through the same classification. Brushing the records in either visualization highlights those records in the map view. Again the user does not define regions-of-interest (e.g. by circling areas), but instead chooses a variable-of-interest and a number of classifications across that variable's range. The intended usage here could be considered the reverse of the probe-based system: Here the user picks an interesting region of the information visualization to find the spatial areas contributing to it, whereas in probe based interfaces, that user picks an interesting region in the geospatial view to find information about the data associated with or contained within that region.

MANET [Unwin 1996] supports moving, rescaling, tiling, stacking, and overlaying multiple plots of various types (scatter plots, box plots, etc.). Sets of arbitrary windows can be designated as siblings, which can be manipulated (e.g. opened and closed, scaled, resized) as a group. An index window provides a virtual display showing all the windows on the screen, where each window is represented by a grey box. Re-arranging these mini-windows re-arranges the actual windows on the desktop. Up to four virtual screens can be configured, and the index window allows rapid switching between these virtual screens. Similar concepts are now ubiquitous in multi-desktop extensions for common operating systems' GUI shells. Plots can be interactively interrogated by mousing over plotted points or groups of plotted points. The particular type of interrogation varies with the plot type. MANET supports 'cues' which are special locations on a plot where the mouse cursor changes shape to indicate some type of manipulation of the plot is possible with further mouse input. Generalized brushing of data is also supported. MANET also supports interactive choropleth maps [Unwin 1998], which can be linked via brushing to other plots. For instance, the user could bring up a choropleth map and a set of box plots, brush one or more regions of the map, and the associated data points on in the box plots would be highlighted. However, it does not appear to be possible to select two separate regions-of-interest of the map and then display two sets of box plots, each of which summarizes only one of the two separate map selections. Therefore, there is no way to generate a third set of box plots to highlight the differences in the statistics between the two regions.

Dykes [Dykes 1997] presents *cdv*, which provides interactive, dynamic 2D cartographic information visualization through a web browser. Dykes demonstrates a number of applications using *cdv*. One application is for exploring the spatial-temporal distribution of tourists in a German park. Selecting a given map location pops-up a 2D line plot showing the number of people visiting that location versus time. A line is drawn connecting the 2D plot to the map location whose data it reflects. Multiple 2D plots can be brought up simultaneously for the selected locations. Dynamic linking and brushing between these plots is supported. This is similar to the probes concept presented here. However, probes are more general: First, in a probe-based system, the user can select arbitrary regions-of-interest on the map. Second, probe interfaces' plot windows are capable of displaying one of many types of 2D visualizations as selected by the user. Third, the probe-based interface provides a mechanism that allows multiple probe windows to be selected, then generates an additional window that directly compares the plots in the original windows.

Dykes strongly advocates cursor driven map interrogation for examining local phenomena. With his "continuity probe" the user selects a zone (country, state, etc.) on a map and the selected zone and the neighboring zones are colored to reflect local tendencies of various statistics. For example, the mean income of the selected zone may be calculated and then the deviation from the mean can be used to color the zones in the selected neighborhood with red hues indicating above local average income and blue indicating below local average incomes. Multiple color-coded co-variance matrices show the co-variance among all the selected zones. This demonstrates the ability to display multiple 2D plots of the same type for a given selected region. Again, this is not quite as flexible as the probe interface, as it does not appear to support displaying multiple 2D plots of different types for a given selection region. Furthermore, it is unclear if multiple neighborhoods can be selected and have their individual 2D plots displayed and compared.

Dykes also discusses interactive adjustment of the distance weights applied to a selected zone's neighbors when computing various neighborhood statistics. The probe-based applications described here do not currently support any explicit weighting within a probe's region-of-interest. In many of the applications described in Chapter 3, the user can paint the selected region as an arbitrary shape (e.g. circle, a general blob, or a curve following a road) so there is not always a meaningfully defined center of the selected region from which to measure the distance for computing a cell weight (although often some centroid could be computed). However, the UrbanVis environment (Chapter 3.2) does allow the user to "weight" how much of the visualization to devote to depicting the data from the center of a region versus the outlying portion of the region.

2.2: Urban Growth Decision Support System

While the actual urban growth model and simulation used within the Urban Growth Decision Support System is not within the scope of this work, some brief discussion of similar systems and their approaches is appropriate.

Klosterman and Pettit [Klosterman 2005] provide a comprehensive review of other similar urban modeling strategies and planning/decision support systems.

The United States EPA [USEPA 2000] evaluated 22 land use change models to determine the state of this type of modeling. One aspect they investigated was the usability of these models as interactive decision making tools. A noteworthy comment from their summary was "Many of the more user-friendly models are integrated with GISs to become spatially explicit decision-support systems with relational database technology." This is a red flag, as traditional GIS packages are generally not lightweight interfaces, are often expensive, and require extensive training. If these are the "more user-friendly models", it is likely that none approach the kind of user-friendliness that the probe-based Urban Growth DSS provides. Indeed, in looking at the criteria they used to rate models one can see many of the desired qualities are aspects in which a probe-based interface excels, including: "Technical Expertise, Temporal Capabilities, Versatility, Linkage Potential, Public Accessibility, and Third-Party Use".

What if? [Klosterman 2001] and UrbanSIM [Waddell 2008] are popular urban growth modeling packages. However, both are single perspective and do not offer the interactive multi-focused inspection and comparison abilities found in probe-based interfaces. Nor are either particularly appropriate for exploratory analysis in the way the Urban Growth DSS is.

CommunityViz [Kwartler 2001] provides a highly interactive visualization interface for exploratory analysis of different land use possibilities. However, the user is generally restricted to a single perspective with a main map view and some coordinated information visualizations. Comparison is possible between different scenarios, however, it is essentially just two single-perspective interfaces side by side. In their sketch tool, used to design land use plans, they provide functionality similar to a probe in which the user can have a sketch monitor window open while defining a polygonal region. In this monitor, it reports values for that polygon as it is being manipulated, such as the amount of acres it contains. By assigning a land use type to this polygon, the sketch monitor window can also give total for values such as estimated energy usage of that proposed development. This sketch monitor seems to only be used to assist the user in creating polygonal regions of a size and shape that suits their purposes, and disappears after the region is done being defined. There does not appear to be any method for using the sketch monitor to semi-automatically adjust the polygonal region to change the values within it to conform to a specific range, as in the MAUP tools presented in this work.

INDEX [Allen 2001] is a community planning and modeling package that allows for the comparison between different policies/scenarios. Again, their comparison abilities are not as flexible or dynamic as those in the probe-based Urban Growth DSS.

The Urban Growth DSS was also designed for co-located collaborative usage on a large multi-touch table-top display. DTLens [Forlines 2005] is a table-top application that allows multiple users to explore a map through zooming lenses. It allows the user to define a region-of-interest and then zoom into it. This is much different than the probe-based interaction in the Urban Growth DSS, in that it does not let one pull up visualizations reflecting that region, but only to look at a detailed fisheye version of the region. It provides “folding” as a method of comparing regions, by distorting the map to place the two zoomed in regions next to each other for visual comparison. This again, does not approach the multi-focus inspection and comparison abilities of the probe-based interface of the Urban Growth DSS.

Forlines [Forlines 2006] also presents a multiple display collaborative system in which a single user application, e.g. Google Earth [Google 2010], is run on a table-top as well as wall mounted displays. Users navigate within a view on the table-top, while the surrounding displays provide other views of the same location from different angles or with different data being displayed. This is arguably still a single perspective interface at this point. However, they do allow the user to lock a display onto a particular location, and then navigate a second display to another location for comparison between the two locations. These are merely duplicated single perspectives and this type of indirect visual comparison again does not approach the direct comparison abilities of the probe-based Urban Growth DSS. Further, it does not seem that their system provides contextual awareness of the spatial relationships between the regions being compared.

2.3: The Modifiable Areal Unit Problem

The modifiable areal unit problem (MAUP) is a long standing, unsolved problem in geography sciences. It refers to the fact that when point data is aggregated into areal units, the variation in how the units, or regions, are delineated can cause significant variation in the aggregated values at any point. The issue itself has been long known, but the term MAUP was coined and the problem described in detail by Openshaw [Openshaw 1984]. It has primarily been studied in regard to its effects on geospatial analyses of aggregated data in fields such as of socio-economics, politics, and epidemiology. [Fotheringham 1991] [Openshaw 1999] [Armhein 1995]

Traditionally, the MAUP is split into two components. The first, the scale problem, relates the choice in the number of regions being compared to its effects on the variation in the results of numerical analysis between those regions, especially when the source data was initially aggregated at a different resolution. This component is not directly addressed in the Urban Growth Decision Support System, as it is more of an issue with how the underlying datasets are generated from data at different granularities. (This is however, discussed in Chapter 5.4) Further, to address its slight appearance on the interaction side, it would require drastic changes to the user's freedom to select and compare any number of regions in an explorative manner. This is more applicable to situations in which the map's area is completely distributed into non-overlapping, space-filling regions, and not the disconnected and sparsely covering region selections commonly made in the Urban Growth DSS interface. However, in the future, it might be worth considering the addition of automatic "split region" and "combine regions" behaviors if a sufficiently elegant method is devised to ensure these actions to not compromise the user's analytical tasks.

This work is primarily concerned with the second component of the MAUP, the aggregation problem. This problem relates the choice of where and how boundary lines are drawn between regions to the effect such choices have on variation in the resulting values for numerical analysis within those regions. A good example of this problem arises when working with census derived data. Due to privacy concerns, the individual household point data is never revealed. Instead, average values are given for "census blocks", which can be apartment complexes, city blocks, or arbitrary delineations of rural tracts of land. The choice in how to delineate these blocks has a direct and significant impact on the aggregated values. If the individual point data was instead aggregated into regions delineated by different methods, say a regular grid, or by postal code, the values available at any particular point on the map point would likely show significant variation from the "census block" method. Thus, the MAUP problem is closely related to another often encountered problem in geography, the ecological fallacy, which states that it is wrong to make inferences as to the values of individuals in a region based on the aggregated values of that region.

Research into the MAUP problem in geospatial analysis fields tends to focus on either understanding the variance or error that can be generated through different scales and aggregations so as to understand the effects that the MAUP can have on analyses performed on the aggregated data [Cutter 1995], or on developing methods to calculate optimal aggregation zones [Nakaya 1998]. In contrast, the MAUP helper tools in the Urban Growth DSS are primarily concerned with monitoring the ways in which the user chooses to define their own areal units, and then figuring out if these delineations could produce misleading results based on the differences across multiple dimensions.

One of the most important differences between the MAUP situations commonly encountered in probe-based interfaces and those studied in the geospatial analysis field is that the MAUP research in the geospatial analysis field seems to focus primarily on space-filling regions that cover the map's entire extent, and share boundaries. While the Urban Growth DSS does provide tools to deal with these conditions, its users are primarily concerned with disjunct regions, with large areas of unselected land, that are more common to probe-based interaction. These have more available space to expand into, and adjustments of multiple regions are rarely zero-sum cases.

A search of the available computer science literature reveals no similar visualization systems that attempt to find and alert users to potentially misleading dimensional inequalities between regions-of-interest being compared, and provide tools for the semi-automated adjustment of these questionable regions.

The Urban Growth DSS represents the next generation of probe-based interface, and the first to be put into the hands of actual end users. The considerations and tools for handling the MAUP detailed in this paper are one of the major new features that improve upon the original probe-based interaction groundwork [Butkiewicz, Dou 2008]. These improvements significantly strengthen the technique's power for geospatial analysis.

CHAPTER 3: APPLICATIONS

The main building blocks of this work are probes. Here, a probe is defined as a pair consisting of a user-defined region-of-interest and a pane containing any variety of information visualizations coordinated to depict and interact with the data within that region-of-interest. The region-of-interest and the visualization pane are linked either directly (e.g. by a line) or indirectly (e.g. the region-of-interest and the pane's background are shaded the same color).

To create a probe, the user selects a region-of-interest (e.g. specifying a central focal point and extent radius, or through manual selection for irregularly shaped regions) and a visualization pane is overlaid directly within the main geospatial visualization. Once created, a probe visually presents a focused, local view into the dataset/model along with a visual linkage back to the overall dataset/model.

Probe-based interfaces have been integrated into three existing applications. These example implementations are presented in this chapter, where they provide illustration and discussion of the limitations of the original application and the benefits gained by applying the probe concept.

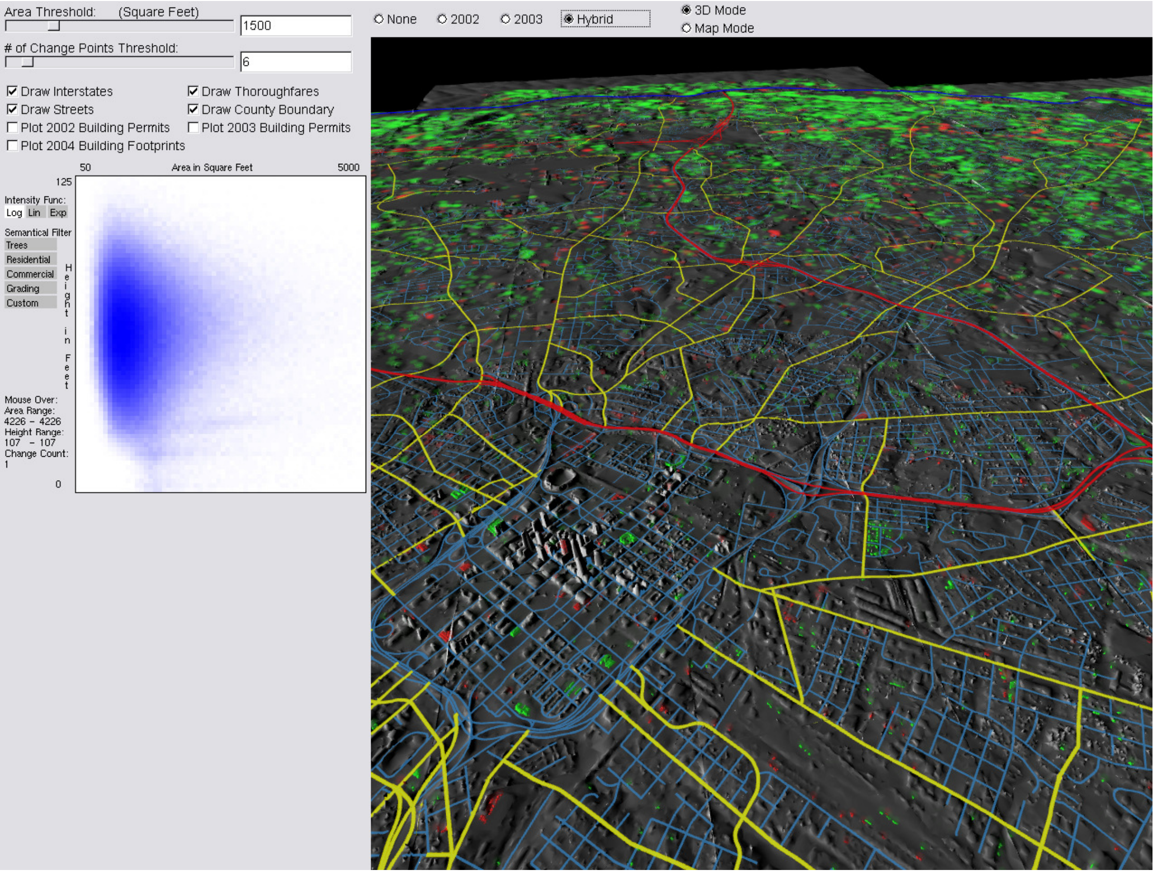
Notably, adding "on-demand" probes to an application will never remove or limit existing capabilities and functionality of the original application, but always adds benefits such as extending beyond a single-perspective, adding multi-focus and multi-scale inspection and interaction, and increased potential for collaborative use on large, shared displays. This is not to say that probes are perfect, as issues including occlusion and visual scalability can certainly arise. (Caveats such as these are discussed in further detail in Chapter 5.2.) However, their "on-demand" nature ensures that an application's original functionality can be maintained merely by restricting probe creation.

3.1: Urban LIDAR Change Detection

This section describes the process and results of integrating probes within a 3D GIS visualization. The primary function of this application is to detect changes such as construction, deforestation, etc. in an urban environment between annual aerial LIDAR scans and then to present the massive collection of changes to the user for exploratory analysis with a cognitively correct non-realistic level-of-detail scheme. [Butkiewicz, Chang 2008] An example view of the original application is shown in Figure 1. Aside from the primary 3D GIS view, a heatmap is presented on the side to depict the global distribution of the individually extracted 3D change models (in terms of height and area) across the entire urban environment. Filtering is allowed on the heatmap, which controls the visibility of changes in the 3D view, based on their area and height measurements.

Similar to most traditional GIS applications, this visualization allows for a single perspective that is directly tied to the viewable screen area. When the user zooms into a small region, it is difficult to maintain the global overview and context as the single perspective limits the user's spatial awareness. Conversely, as the user zooms out, local details become suppressed and difficult to see. Furthermore, since the heatmap is tied directly to the user's perspective, there is no easy way to compare the trends and patterns of two or more regions without running two instances of the application simultaneously or saving the images to files and comparing them offline.

Probes are introduced to this visualization to remedy these issues. A user defines a region-of-interest using a mouse, and a probe interface appears directly within the 3D view. Within the probe interface is the heat map visualization, now showing the distribution of only those changes within the region-of-interest. Also present are the filtering controls; here again their domain switches from global to local filtering. Multiple probes can be added on the same display, and they are differentiated based on the colors of the probes and the highlights of the regions-of-interest (Figure 2).



Area Threshold: (Square Feet) 1500

# of Change Points Threshold: 6

☑ Draw Interstates
☑ Draw Thoroughfares
☑ Draw Streets
☑ Draw County Boundary
☐ Plot 2002 Building Permits
☐ Plot 2003 Building Permits
☐ Plot 2004 Building Footprints

○ None ○ 2002 ○ 2003 ● Hybrid

● 3D Mode
○ Map Mode

Mouse Over:
Area Range:
4226 – 4226
Height Range:
107 – 107
Change Count:
1

Figure 1: Shown here is the original interface for the urban LIDAR change application. The main window (right) presents a 3D fly-around view of the county. To the side, a heatmap (upper left) shows the global distribution of all changes across the entire county. It is a density-shaded scatter plot with the vertical axis tied to the heights of change models and the horizontal axis tied to the projected 2D areas of change models. Different types of changes (e.g. newly built houses) generally fall in predictable areas of the heatmap. By selecting regions within the heat map, the user can filter the changes presented in the main window.

Figure 2 presents a scenario in which the analyst uses probes to conduct a search by example type interaction. They are interested in finding new residential housing developments around the region. If the analyst can determine what signature a standard two-story house has in the heatmap distribution, they can use that characteristic to filter the massive data collection to reveal the houses-of-interest. To determine what signature these developments would have in the heatmap, the analyst selects two regions-of-interest on the terrain. The first region, shown in blue, has changes consisting primarily of commercial and industrial buildings. The second region, shown in red, is a partially rural area that contains a number of new residential developments under construction at the time, each with many identical two-story houses. It is clearly visible that the distributions in these two regions are different by examining their corresponding heatmaps. The magenta arrow in Figure 2 shows a concentration of changes found only in the second, residential region. This corresponds to the two-story houses, which are generally the same height and vary slightly in area. This signature/region can then be applied to the main heatmap and used to filter the entire county, revealing all the similar new residential developments across the region.

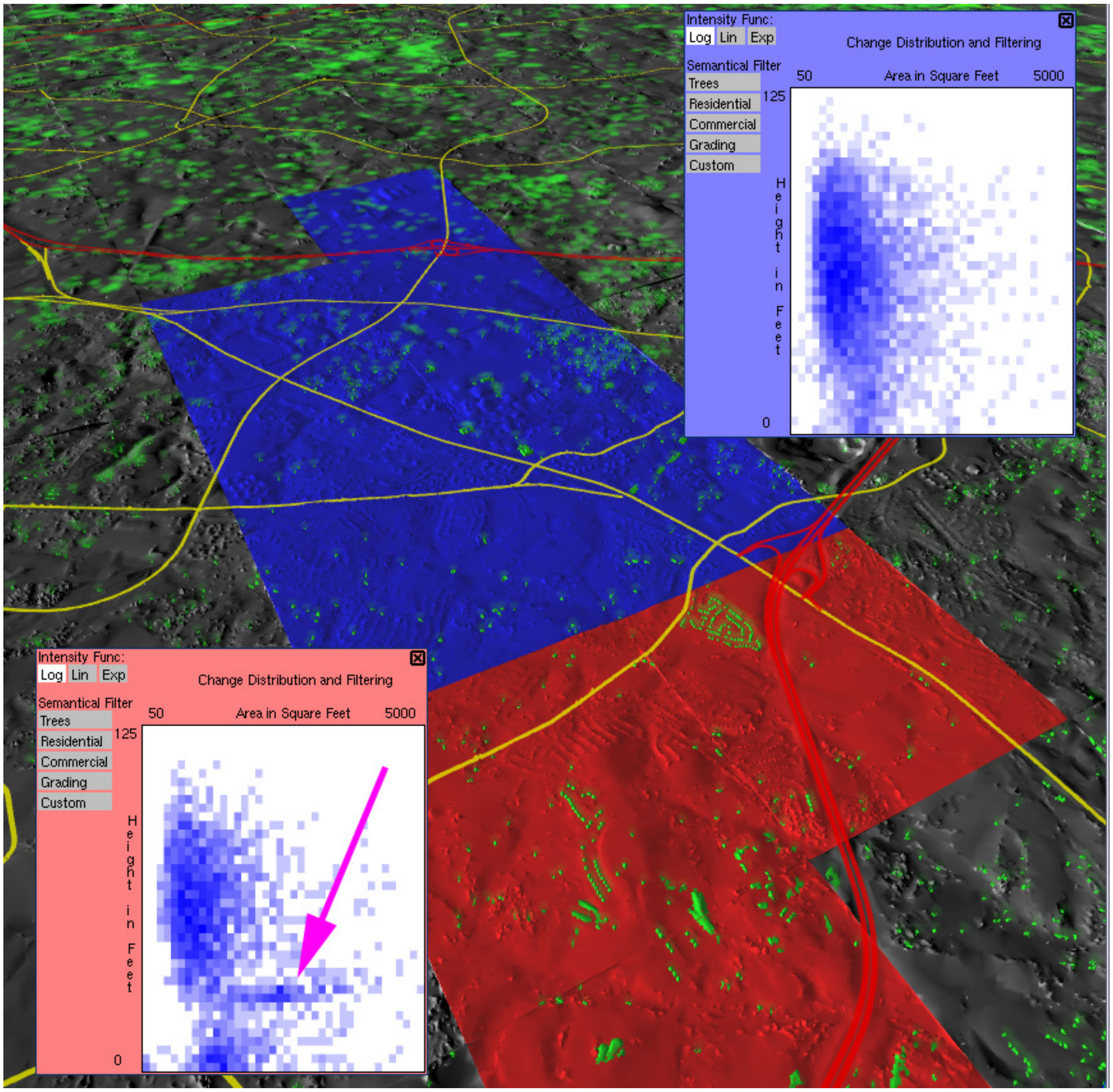


Figure 2: Shown here, the user has selected two regions-of-interest, the blue is a commercially zoned district, and the red mostly residential. The visualizations for each probe present a heat map showing the distribution (in height and area) of changes detected in the respective regions-of-interest. The magenta arrow points to a concentration of changes found only in the residential region. This region on the heat map can then be used as a global filter, revealing other similar new residential developments elsewhere.

Even in this simple example, the power of the probe interface is apparent. The user can now examine regions from afar so as to maintain spatial awareness in relation to the surrounding regions. With the heatmaps displayed directly in the 3D view, the user can easily relate the abstract information visualizations with their corresponding spatial locations. More importantly, comparison between locations is now trivial as the heatmaps can be juxtaposed for immediate comparisons. This can be done either visually, as shown in Figure 2, or directly, by requesting a comparison pane of the two probes. By selecting two probes, a third pane can be requested, which simply presents a difference image of the two probes' heatmaps. In a previous paper [Butkiewicz, Chang 2008], a manually created difference image had to be created outside the application to illustrate the differences in changes between two regions. The ability to do this type of comparison directly within the application is a powerful improvement. Further discussion of direct comparison abilities follows in Chapters 3.3, 3.4.4, and 4.1.2.

3.2: Census Data Exploration

UrbanVis [Chang 2007] is an application designed to explore an urban environment and its corresponding census information by combining a neighborhood-based 3D urban model view with an abstract information visualization view (Figure 3). With the use of the yellow sphere as a control, the user can interactively navigate an urban environment and explore relationships between spatial and abstract information in a multi-resolution manner. Neighborhoods closest to the focus point (the yellow sphere) subdivide to show more detail, while neighborhoods further from the focus point are aggregated together into larger districts.

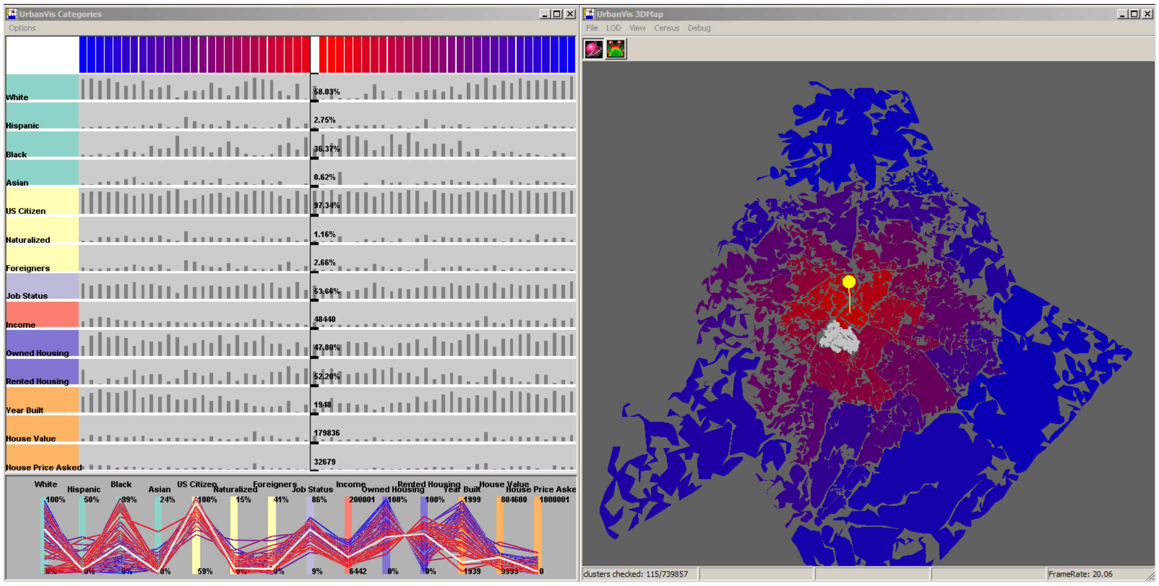


Figure 3: Shown here is the original interface of the UrbanVis application. It consists of both a 3D map view (right), used to select a region-of-interest, and a coordinated visualization (left).

Unlike the LIDAR system described in the previous section, UrbanVis already separates the region-of-interest (as denoted by the yellow sphere) from the visible screen area. This view independence allows the user of UrbanVis to explore the urban model while retaining spatial awareness. However, similar to the LIDAR system, UrbanVis allows for only a single perspective and therefore cannot support comparison of different localized regions. This shortcoming can be seen in their paper [Chang 2007], where to compare two regions, they had to navigate to the first region, take a screen capture, navigate to the second region, take another screen capture, and compare the images outside the application.

By applying the probe concept to UrbanVis, the user can now interact with multiple regions-of-interest simultaneously within the 3D model view. As shown in Figure 4, each region-of-interest is now accompanied by an information panel exactly like the one shown on the left of Figure 3. The information panels can be moved around directly within the 3D model view but are always connected to the yellow spheres by a (white) line to maintain a clear relationship between the two. When two information panels are placed next to each other, the differences and similarities between the two local regions become apparent.

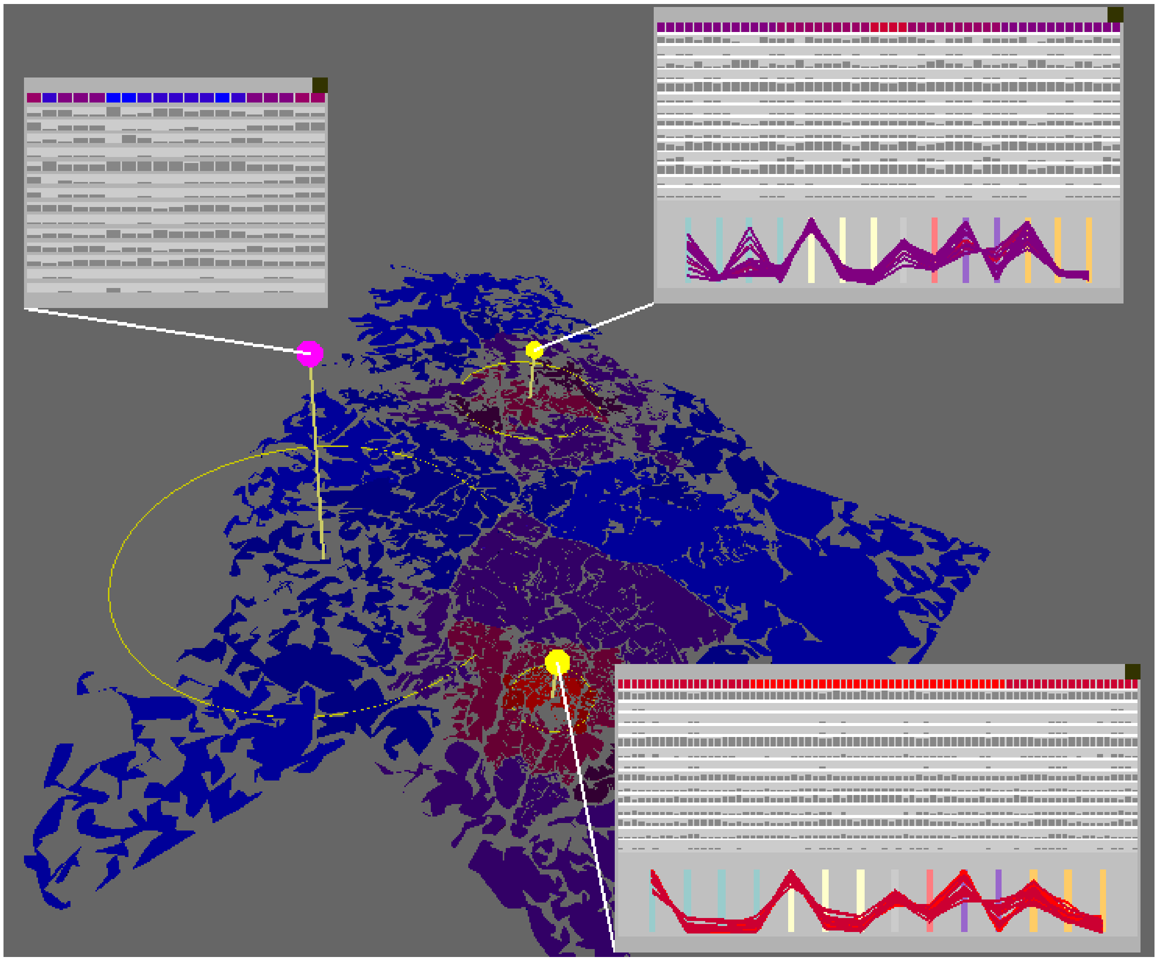


Figure 4: After modification to utilize probes, the user can now select multiple, variably sized regions-of-interest within the 3D view. Each region-of-interest is then linked to a resizable version of the original coordinated information visualization. By resizing the panels for each probe, the user can control the granularity/abstraction of the depiction of the data from the probe. Resizing is extended further in Figure 5.

It is easy to see the comparison capability gained from using probes in UrbanVis; the user can now compare multiple local regions simultaneously within the application, without a navigational burden. However, another subtle but important advantage is that the resizable information panel allows the user to "annotate" regions using small information panels (Figure 5). These small information panels now act like glyphs in that they give an aggregated, high-level overview of the selected regions-of-interest without taking up much screen real estate. This "accidental" feature has since been developed into a more complete "To Glyph" command described in Chapter 3.5.5.

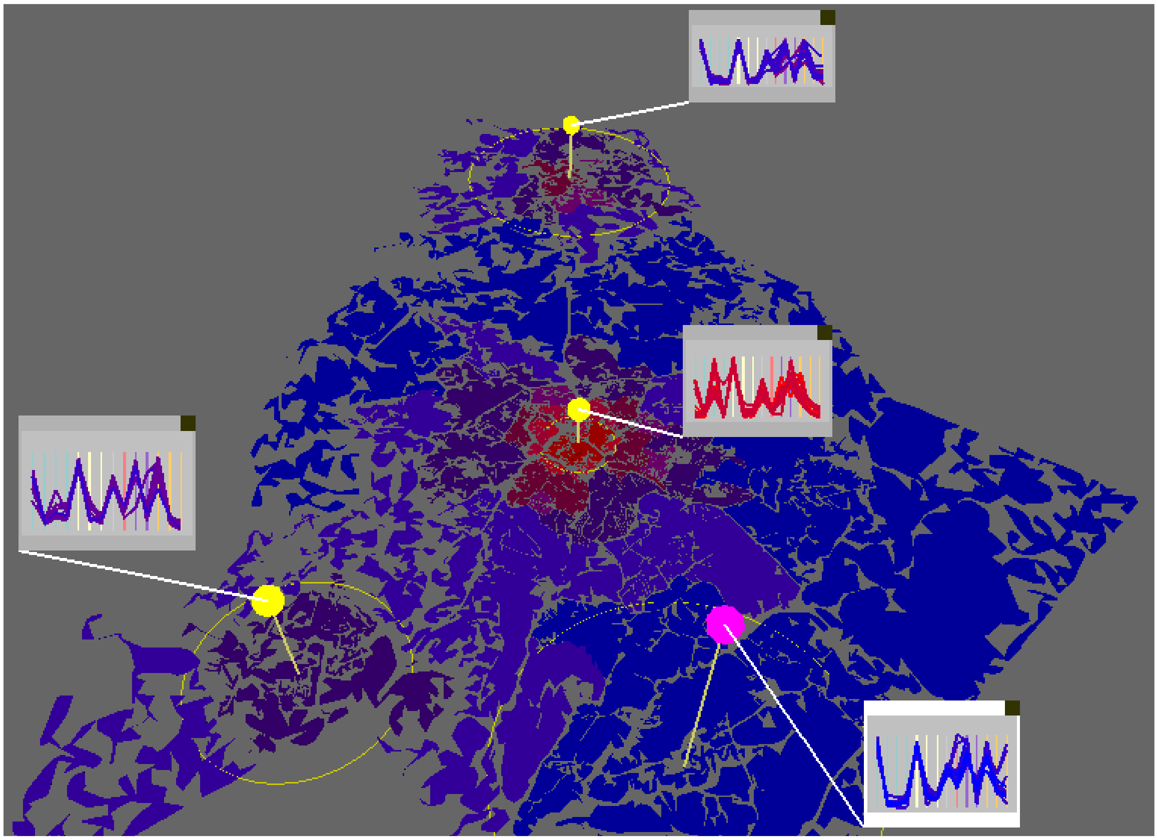
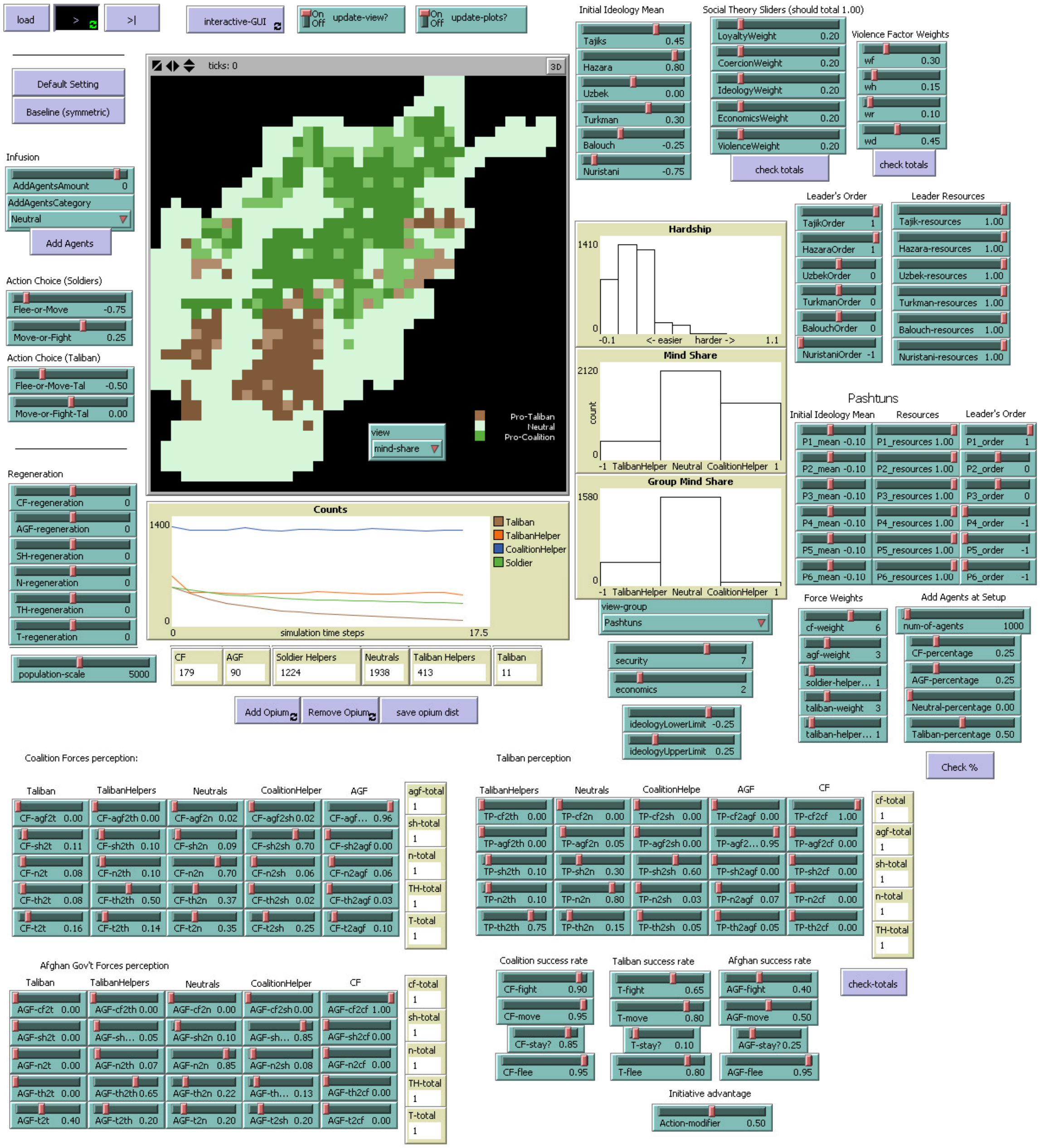


Figure 5: Shown here, the coordinated visualizations for each probe have been limited to solely the parallel coordinates view and resized to the point where each shows only the most general view of the associated data. Here the probes begin to resemble "flags" stuck in the map, giving a simple representation and allowing for quick visual comparison. (Assuming the user knows how to interpret them, of course.)

3.3: Agent-based Simulation

This agent-based simulation and visualization tool was created to visualize the results of a live, agent-based simulation that allows a user to experiment with different social theories and scenarios in Afghanistan. Like the two visualizations described in Chapter 3.1 and Chapter 3.2, the probe concept is applied to an existing visualization of the agent-based system (The original interface is shown in Figure 6). However, unlike the previous two visualizations, the introduction of probes transformed the agent-based tool nearly completely.

Like the original LIDAR system, the agent-based tool is also limited to a single perspective that is tied to the viewable map area. Similarly, the additional infovis views in the agent-based tool, such as the bar chart and the time-series view, are also tied to this single perspective. However, unlike the LIDAR system or UrbanVis, the main purpose of the agent-based tool is for the user to manipulate variables within the simulation and visualize the effects of the changes. Most of these variables are global, in that they affect the simulation of the entire country. Some are tied to fixed single locations or a specific, rigidly predefined regions. It is clear that without proper organization, an exponential number of controls are needed to capture all combinations of all the variables. In fact, Figure 6 shows some of the 150+ sliders that were needed to operate a few relatively simple social theories.



Infusion
AddAgentsAmount 0
AddAgentsCategory Neutral

Action Choice (Soldiers)
Flee-or-Move -0.75
Move-or-Fight 0.25

Action Choice (Taliban)
Flee-or-Move-Tal -0.50
Move-or-Fight-Tal 0.00

Regeneration
CF-regeneration 0
AGF-regeneration 0
SH-regeneration 0
N-regeneration 0
TH-regeneration 0
T-regeneration 0

population-scale 5000

view mind-share

CF	AGF	Soldier Helpers	Neutrals	Taliban Helpers	Taliban
179	90	1224	1938	413	11

Initial Ideology Mean
Tajiks 0.45
Hazara 0.80
Uzbek 0.00
Turkman 0.30
Balouch -0.25
Nuristani -0.75

Social Theory Sliders (should total 1.00)
LoyaltyWeight 0.20
CoercionWeight 0.20
IdeologyWeight 0.20
EconomicsWeight 0.20
ViolenceWeight 0.20

Violence Factor Weights
wf 0.30
wh 0.15
wr 0.10
wd 0.45

Leader's Order
TajikOrder 1
HazaraOrder 1
UzbekOrder 0
TurkmanOrder 0
BalouchOrder 0
NuristaniOrder -1

Leader Resources
Tajik-resources 1.00
Hazara-resources 1.00
Uzbek-resources 1.00
Turkman-resources 1.00
Balouch-resources 1.00
Nuristani-resources 1.00

Pashtuns

Initial Ideology Mean	Resources	Leader's Order
P1_mean -0.10	P1_resources 1.00	P1_order 1
P2_mean -0.10	P2_resources 1.00	P2_order 0
P3_mean -0.10	P3_resources 1.00	P3_order 0
P4_mean -0.10	P4_resources 1.00	P4_order -1
P5_mean -0.10	P5_resources 1.00	P5_order -1
P6_mean -0.10	P6_resources 1.00	P6_order -1

view-group Pashtuns
security 7
economics 2
ideologyLowerLimit -0.25
ideologyUpperLimit 0.25

Force Weights
cf-weight 6
agf-weight 3
soldier-helper… 1
taliban-weight 3
taliban-helper… 1

Add Agents at Setup
num-of-agents 1000
CF-percentage 0.25
AGF-percentage 0.25
Neutral-percentage 0.00
Taliban-percentage 0.50

Coalition Forces perception:

Taliban	TalibanHelpers	Neutrals	CoalitionHelper	AGF	
CF-agf2t 0.00	CF-agf2th 0.00	CF-agf2n 0.02	CF-agf2sh 0.02	CF-agf… 0.96	agf-total 1
CF-sh2t 0.11	CF-sh2th 0.10	CF-sh2n 0.09	CF-sh2sh 0.70	CF-sh2agf 0.00	sh-total 1
CF-n2t 0.08	CF-n2th 0.10	CF-n2n 0.70	CF-n2sh 0.06	CF-n2agf 0.06	n-total 1
CF-th2t 0.08	CF-th2th 0.50	CF-th2n 0.37	CF-th2sh 0.02	CF-th2agf 0.03	TH-total 1
CF-t2t 0.16	CF-t2th 0.14	CF-t2n 0.35	CF-t2sh 0.25	CF-t2agf 0.10	T-total 1

Taliban perception

TalibanHelpers	Neutrals	CoalitionHelpe	AGF	CF	
TP-cf2th 0.00	TP-cf2n 0.00	TP-cf2sh 0.00	TP-cf2agf 0.00	TP-cf2cf 1.00	cf-total 1
TP-agf2th 0.00	TP-agf2n 0.05	TP-agf2sh 0.00	TP-agf2… 0.95	TP-agf2cf 0.00	agf-total 1
TP-sh2th 0.10	TP-sh2n 0.30	TP-sh2sh 0.60	TP-sh2agf 0.00	TP-sh2cf 0.00	sh-total 1
TP-n2th 0.10	TP-n2n 0.80	TP-n2sh 0.03	TP-n2agf 0.07	TP-n2cf 0.00	n-total 1
TP-th2th 0.75	TP-th2n 0.15	TP-th2sh 0.05	TP-th2agf 0.05	TP-th2cf 0.00	TH-total 1

Afghan Gov't Forces perception

Taliban	TalibanHelpers	Neutrals	CoalitionHelper	CF	
AGF-cf2t 0.00	AGF-cf2th 0.00	AGF-cf2n 0.00	AGF-cf2sh 0.00	AGF-cf2cf 1.00	cf-total 1
AGF-sh2t 0.00	AGF-sh… 0.05	AGF-sh2n 0.10	AGF-sh… 0.85	AGF-sh2cf 0.00	sh-total 1
AGF-n2t 0.00	AGF-n2th 0.07	AGF-n2n 0.85	AGF-n2sh 0.08	AGF-n2cf 0.00	n-total 1
AGF-th2t 0.00	AGF-th2th 0.65	AGF-th2n 0.22	AGF-th… 0.13	AGF-th2cf 0.00	TH-total 1
AGF-t2t 0.40	AGF-t2th 0.20	AGF-t2n 0.20	AGF-t2sh 0.20	AGF-t2cf 0.00	T-total 1

Coalition success rate	Taliban success rate	Afghan success rate
CF-fight 0.90	T-fight 0.65	AGF-fight 0.40
CF-move 0.95	T-move 0.80	AGF-move 0.50
CF-stay? 0.85	T-stay? 0.10	AGF-stay? 0.25
CF-flee 0.95	T-flee 0.80	AGF-flee 0.95

Initiative advantage
Action-modifier 0.50

Figure 6: The agent-based social simulation's original interface. Notice the large portion of screen-space dedicated to sliders and other control elements, which are ambiguous in terms of their scope. The single map view allows for only one variable to be seen at a time. Likewise, the four graphs can only depict global statistics across the entire simulation.

In addition to the issue of over-crowding from excessive sliders shown in Figure 6, the agent-based tool suffers another equally severe interaction issue, in that the sliders offer no spatial context in terms of their relationships to the corresponding geographical regions. Users and observers of this agent-based simulation are often left wondering what slider to operate in order to affect a specific region-of-interest. This incongruity between the visualization and its controls greatly diminishes the effectiveness of the simulation as an experimental platform for testing social theories.

The introduction of probes can greatly increase the effectiveness of the interface. As can be seen in Figure 7, multiple instances of maps are now allowed, with each map colored based on a particular dimension in the data (e.g., ethnic group, loyalty, etc.). However, most importantly, the 150+ sliders can now be replaced by an “on-demand” tabbed control panel of sliders directly associated with each probe (as shown in Figure 8). This combination of sliders with geo-located probes makes the effect of each slider clearer, in that interaction with a slider now only (locally) effects the region tied to its corresponding probe (i.e. whatever portion of the simulation the user has circled). It should be noted that the original, global controls can be replicated by simply creating a region-of-interest encompassing the entire simulation, as shown in the bottom-left of Figure 7.

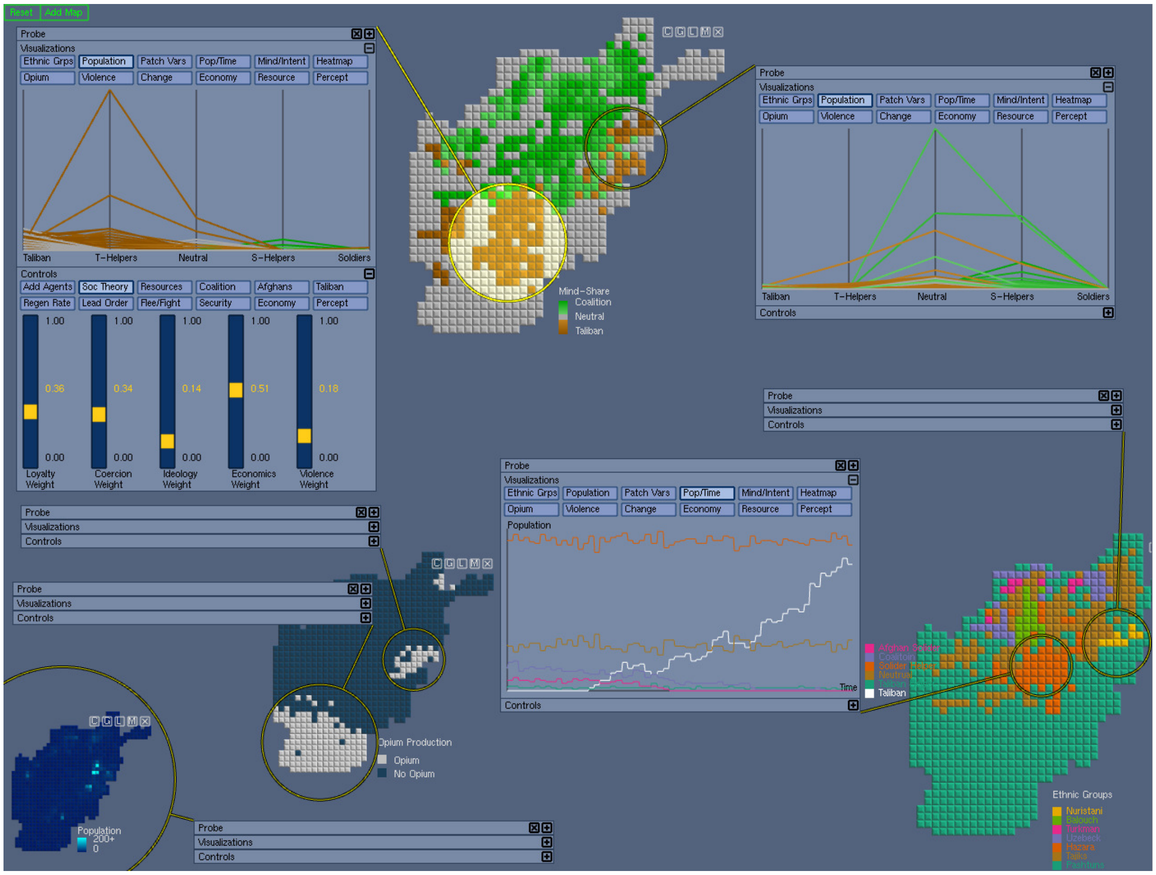


Figure 7: An example workspace in the new, probe-based interface. Notice that the user can add any number of different overview maps. Probes can then be inserted into these maps, spawning linked coordinated visualization/interaction panes. This extends observation and interaction across all levels, from global to individual cells.

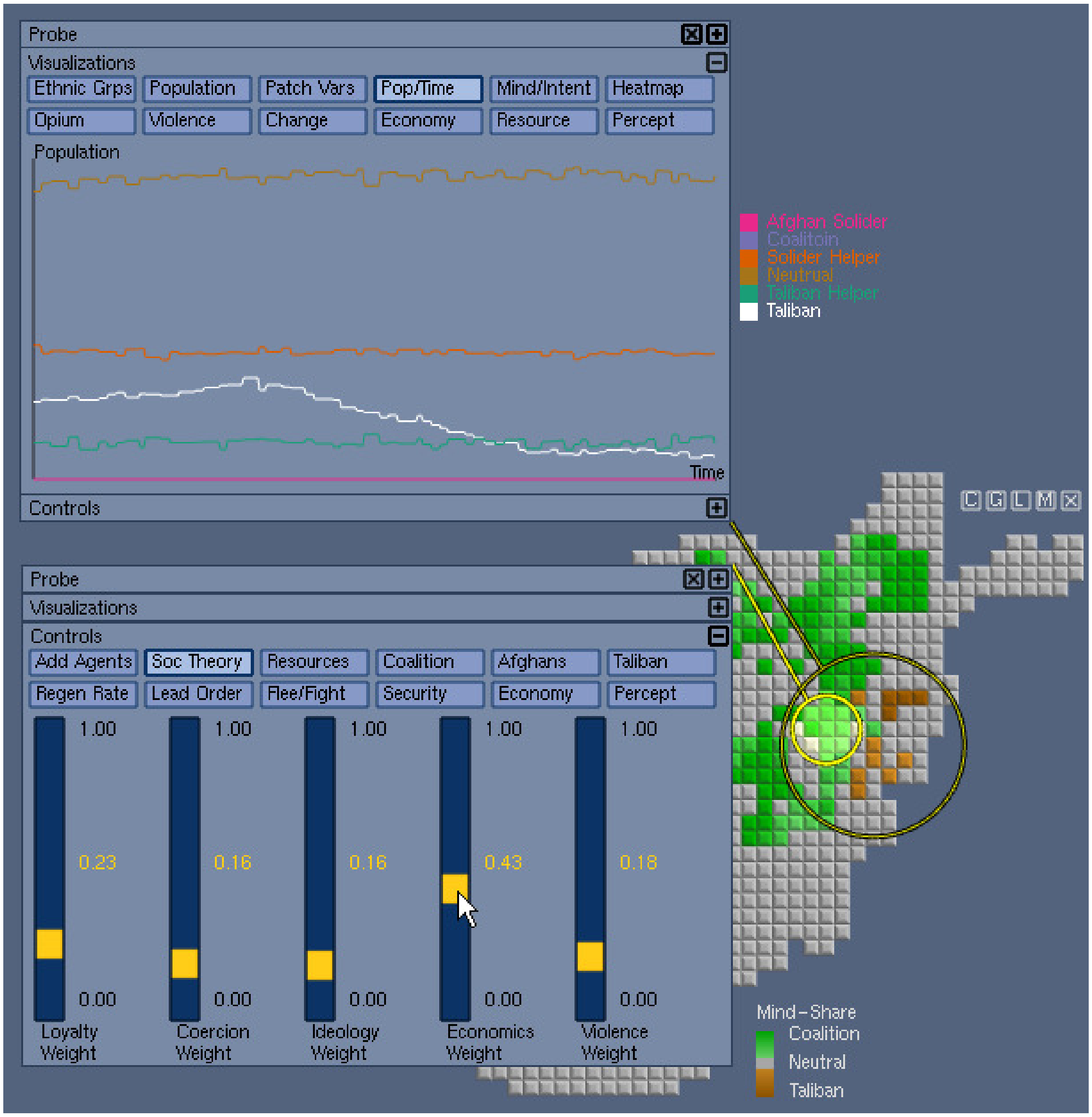


Figure 8: The use of localized control capabilities is shown in this scenario. Here, the user places a probe (the smaller circle) over the city of Kabul and expands the control section of that probe's interface and manipulates local variables to test out a new social theory within the city limits. A second probe (the larger circle) has been added to encompass the surrounding region, which has some pockets of Taliban loyalists (brown cells). This probe is setup to graph the relative populations of various factions over time. The user can easily see that after the new social theory is enacted within Kabul, the number of Taliban agents (white line) in the surrounding decreases.

The implication of a visualization that has capabilities for both passive inspection and active manipulation is striking. As shown in Figure 8, the user has selected two nested regions to test the impact of an economic change in a small selected region and its effect in the surrounding areas. With the probe interfaces, the user can directly modify the economy of the small selected region and observe its effects in the probe associated with the surrounding areas. In this example, it appears that as the residents of the selected small region increase in wealth, the population of Taliban agents diminishes in the surrounding area.

A common task for analysts testing social theories is to directly compare two regions-of-interest. With the probe interface this task becomes trivial. The analyst is no longer limited to strictly visual comparison, but instead can directly compare multiple regions-of-interest through the use of a comparison pane. As shown in Figure 9, a comparison pane can be created between multiple existing probes to visualize the relationships between the regions-of-interest there are tied to. In this example, the user has selected a “union” operation, combining the two selected regions into a single view, immediately revealing the differences in population characteristics, while preserving (using color) the distinction between the two data sources. Numerous possible operations are possible within this framework, including the previously mentioned difference image of two heatmaps and the intersection between overlapping regions-of-interest. This type of direct comparison capability has been further developed and refined, as presented and discussed in Chapters 3.4.4, 3.5.7, and 4.1.2.

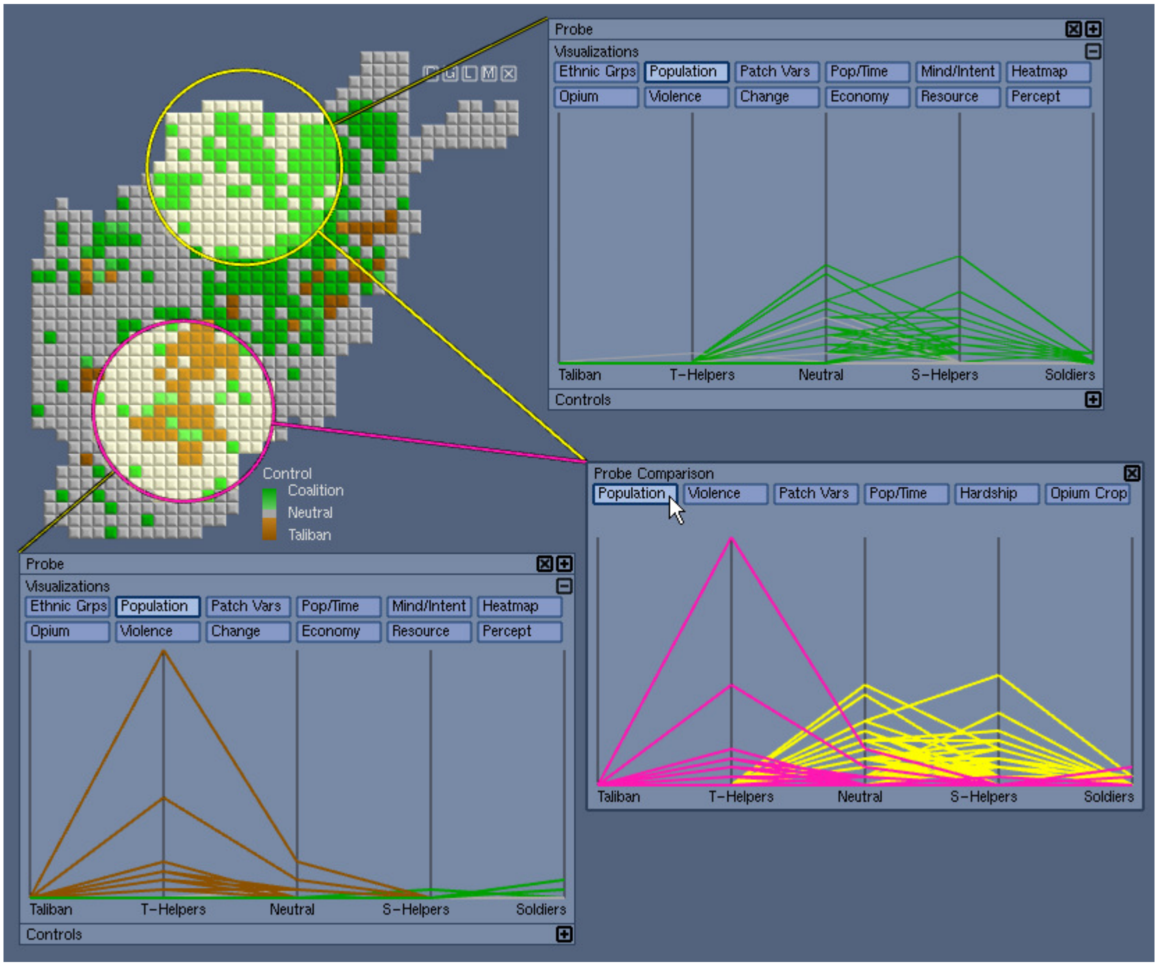


Figure 9: Shown here, the user has created two probes, one over a Taliban-controlled area (magenta circle/brown cells) and one over a Coalition-controlled area (yellow circle/green cells). Each is set to display the relative populations of each type of agent using parallel coordinates, scaled locally. Then by choosing to create a direct comparison (bottom right pane) of the two probes, the user can see the values from each region-of-interest together in a single visualization, scaled with respect to both.

The most obvious benefit gained by adding probes to this application is the ability to inspect multiple local regions at once. Whereas the original fixed coordinated visualizations once reflected the global model's values, the analyst is now empowered with the ability to have any number of dynamically created visualizations, each able to reflect the values in regions-of-interest of every size and shape imaginable. This technique also provides superior comparison capabilities by directly presenting regions-of-interest together or against each other in their own visualizations.

The probe interface also allows for geospatial-based manipulation of the simulation and visualization; the user gains the freedom to choose regions-of-interest of any size and shape and interact with their properties directly, allowing for easy experimentation of complex social theories and immediate visualization of their effects. Finally, by replacing the static interface with an unrestrained workspace, enabling and encouraging the user to add and remove "on-demand" interface elements as needed, not only is the original clutter and wasted screen space removed, but the single-user application is extended into one that has potential to support multiple users when run on a high resolution, shared display.

3.4: Urban Growth Decision Support System

This section describes the design of and features found in the Urban Growth Decision Support System (DSS). This system was built from the ground up, based on the lessons learned during the previous three implementations as described in the last two sections. The Urban Growth DSS represents the next generation of probe-based interface, and the first to be released into the hands of actual end users. In addition to refined versions of the previously presented features, it contains new considerations and tools for handling modifiable areal unit problems. These improvements considerably improve the probe technique's suitability for geospatial analysis.

3.4.1: Simulation Details

The Urban Growth Decision Support System is designed to provide a highly interactive interface for policy makers, urban planners, etc. to explore and analyze both 30 years of historical urban growth and 25 years of predicted future growth. It focuses on a 240 km (150 miles) wide region around a major metropolitan area characterized by significant urban sprawl.

Collaborators in The Center for Applied Geographic Information Science used satellite imagery to classify historical land coverage as developed or undeveloped (e.g. natural vegetation versus impervious surfaces). Protected lands such as forests and parks were recorded as well. The currently remaining undeveloped land was then ranked by its attractiveness to new development. This was done by considering positive factors, e.g. distance to major employment centers, percentage of surrounding parcels already developed, and established infrastructure such as road density, as well as negative factors, e.g. slope of terrain. Then, by using forecasts of population growth for each region, and knowing how much land is used per person in each type of area (i.e. high density urban core, suburban fringe, etc), the appropriate amount of land was converted from undeveloped to developed for that particular time step, and the model was recalculated for the next time step. The results of this simulation process are highly detailed land coverage maps for multiple time steps ranging from 1976 to 2030.

The application was designed to run on a desktop for standard single-analyst usage, a laptop with or without a projector for presentations to policy makers in the field, as well as on a multi-touch table for simultaneous collaborative use between multiple analysts and domain experts. A sample view of the application being used on the desktop is shown in Figure 10.

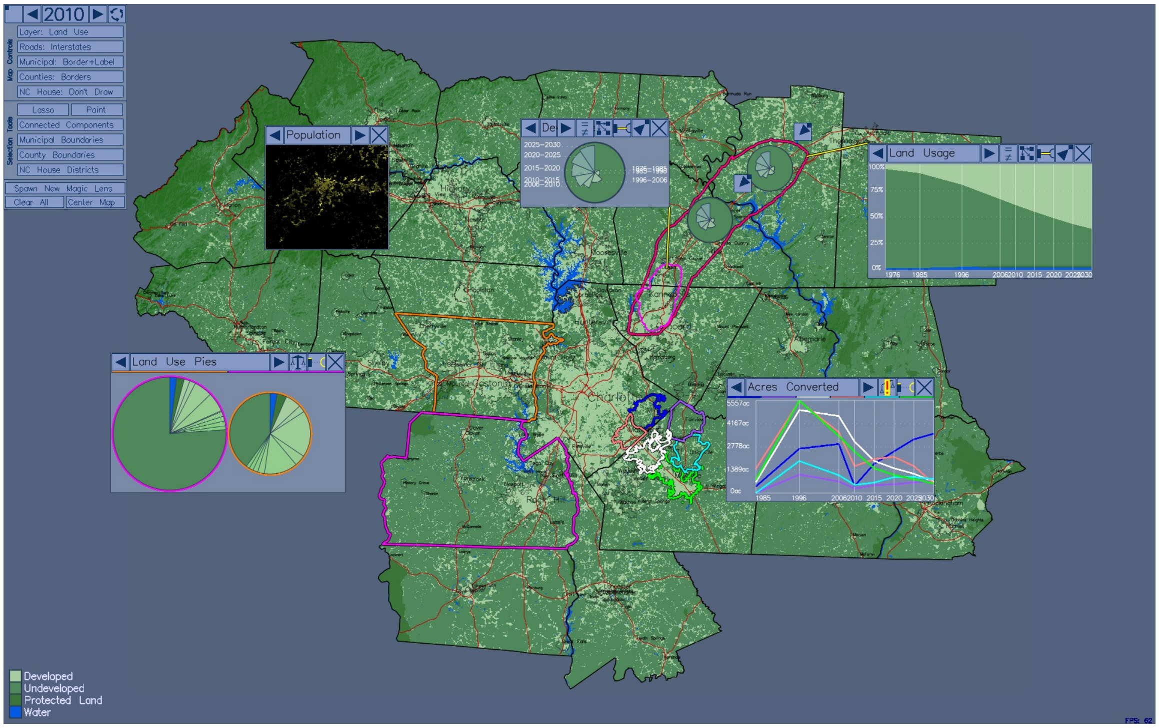


Figure 10: An example workspace in the Urban Growth Decision Support System. The control panel can be seen in the upper left. Various selections have been made with the different selection tools. Also shown are the comparison panels and Magic Lens tool.

3.4.2: Interface Design

The existing method of presenting the results of the urban growth model was to either to present static maps, or to form them into a simple slide show animation. The Urban Growth Decision Support System was to have an entirely new interface that could be used to both interactively explore and present these results to politicians, community groups, policy makers, and the general public. For small audiences, the goal was not only to be able to present the results, but also to engage them with the ability to play with the data themselves, leading to the discovery of findings of personal interest. A multi-touch table seemed promising to help accomplish these objectives, and so this application was developed specifically for simultaneous deployment on both a multi-touch table, with approximately 4 foot x 6 foot screen size, as well as a traditional mouse and keyboard desktop interfaces. For the multi-touch table, probe-based interaction was important to allow multiple users to collaboratively operate the system simultaneously while also giving personal “work-spaces” to each user.

The starting interface on the desktop is minimalist, with a central map, map legend, simple control panel. All other commands and controls are provided “on demand” as needed to reduce clutter and conserve valuable screen real estate.

When run on the multi-touch table, the application automatically switches to a less text and orientation dependent interface. Many buttons have pictographs that show what the button does. These can be understood from any viewing angle, as opposed to orientation-depended text. Multiple instances of the selection tools are placed at each side of the table to allow quick access, eliminating the need for users to reach across the shared workspace. The touch-table interface can be seen in Figure 11.

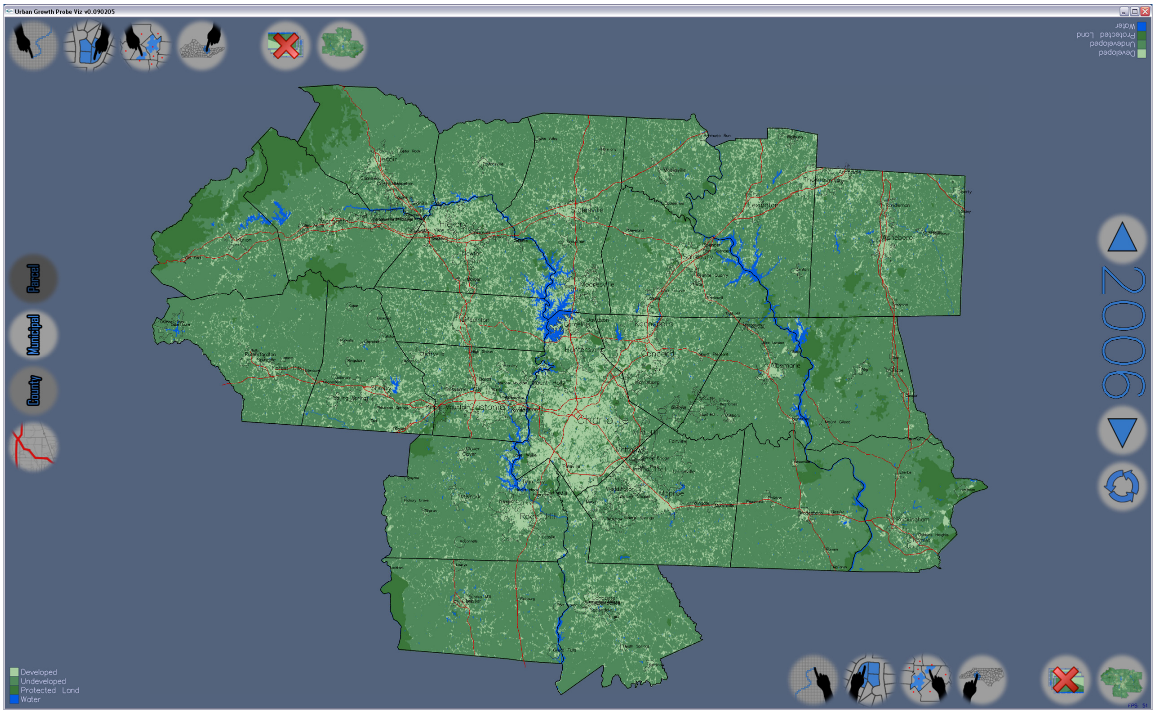


Figure 11: The default view of the system's interface when run on the multi-touch table. Notice that the commonly used selection toolbox buttons are conveniently presented both in the lower-right for the Southern users, and the upper-left for the Northern users. Map legends are also replicated for each side in the opposite corners. Less commonly used buttons for controlling map drawing parameters are presented midway on the rarely (due to table shape) occupied East and West sides of the table, putting them easily within reach of both Southern and Northern users.

3.4.3: Selection Tools

The Urban Growth DSS provides users with a powerful array of different methods to select regions-of-interest on the map. While these tools are not themselves particularly novel, the interactivity they provide as a collection is beyond that found in traditional GIS packages. The first set of selection tools are free-form selections, including lasso and paint.

The lasso tool allows one to use the mouse, or their fingers on a touch table, to circle an area for selection. As shown in Figure 12, it can be used to select irregular regions, such as proposed development areas, around features (roads, waterways, etc), and any other areas not already delineated by predefined geospatial records. When the free-form lasso selection mode is entered, the user can drag the mouse or finger anywhere on the map, acting like a pen (with visual feedback in the form of a bold red line). The points that are recorded in between the down and up events are transformed into a list of lines, and checked for intersections that create valid polygons. A check is also made to see if the addition of a short connecting line could "close up" a near-polygon. This ensures that the user does not need to fully complete/connect a circling gesture around a region in order for it to register. The polygonal regions generated are then fed to a new region-of-interest object for data extraction, as detailed in Chapter 3.5.4.

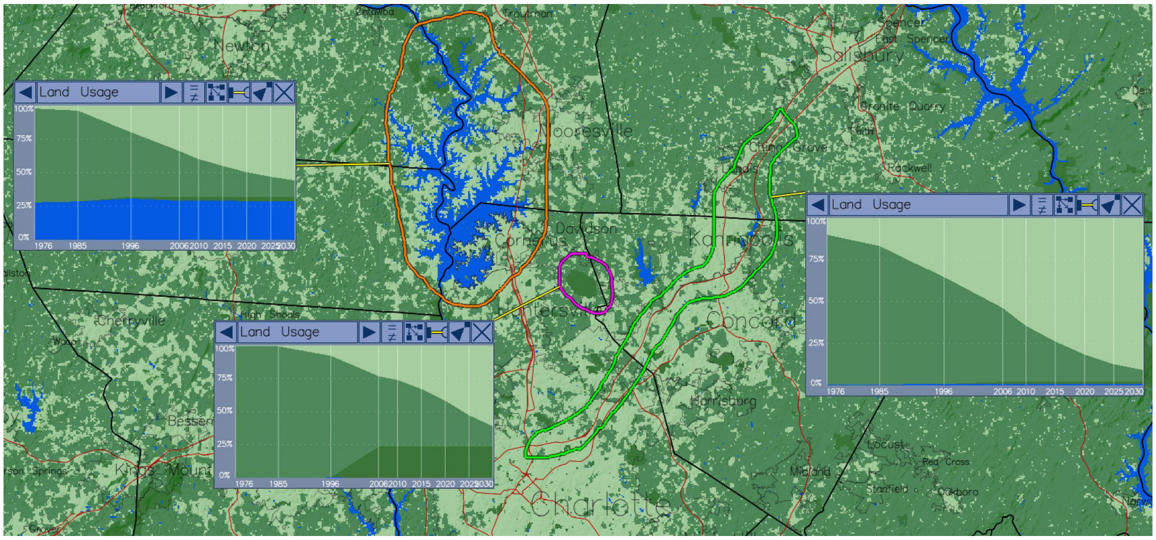


Figure 12: Example regions-of-interest generated through the use of the free-form lass selection tool. This method of selection allows for the selection of irregular regions-of-interest and those not easily defined by pre-loaded geospatial records. In orange, the user has traced a region roughly around a lake's shoreline. In magenta, the user has quickly circled a protected area (a park). In green, the user has selected a region along either side of an interstate highway.

The free-form paint tool allows the user to "paint" regions directly onto the map. This is useful not only for selecting oddly shaped areas and features, but also for tracing along linear features, such as roads, as show in Figure 13. In this selection mode, when the mouse or finger is applied to the map a thick line is left behind on the map. The thickness of the line is static in screen space, so zooming in allows for more precise selection and zooming out allows for faster selection of large areas. When the user completes their selection, the binary mask being generated by the paint tool is fed to a new region-of-interest object for data extraction.

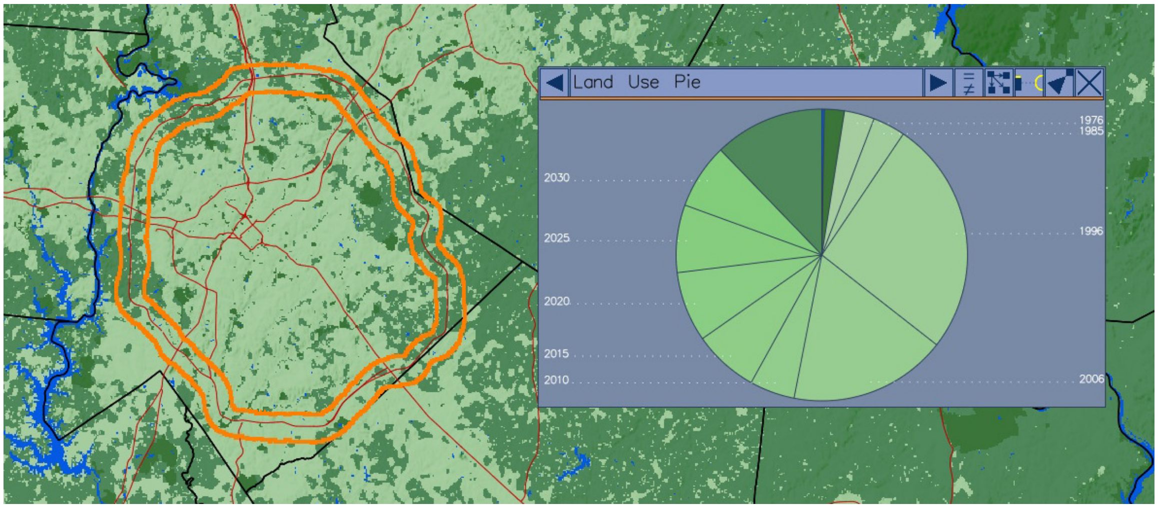


Figure 13: Example of a region-of-interest generated with the free form paint selection tool. Here the user dragged the paint tool over the interstate highway that circles this metropolitan area. The result is a region-of-interest that contains everything within roughly one mile of the interstate. The visualization shows how much land in this region was developed during each timestep (lighter greens) as well as what amount remains (darker greens).

The next set of selection tools take advantage of the predefined hierarchical geospatial data the system has associated with the survey extent. These tools allow the user to select these pre-defined sub-regions at multiple granularities. The Urban Growth DSS is concerned primarily with municipal and county boundaries, but depending on the application domain, many different types of predefined boundaries are advantageous, e.g.

watersheds, voting districts, sales routes, etc. A selection tool is provided for each category of predefined regions. In the Urban Growth DSS, the municipal selection tool, for example, allows the user to quickly and easily select the area within a municipal (city, town, etc) boundary, including any unconnected outparcels. When entering a selection mode with one of these tools, the geospatial boundaries for the regions associated with the desired category are highlighted on the map. One can then either make a single mouse click (or finger tap), which will select the encompassing region, or one can drag the mouse (or finger) across multiple regions, which will select all regions that encompass any portion of the mouse's (or finger's) trail. This is accomplished by taking the list of points generated by the interaction and checking to see if they fall within the bounding circles of valid boundaries, then checking any matching points against the actual boundaries with a simple point-in-polygon test. All of the polygons from any matching regions (not just those from the matching boundary, which is a subset) are then compiled into a single region-of-interest object. An example of the results of using the city selection tool is shown in Figure 14.

An auxiliary, slightly modified selection mode is possible for these particular tools to aid in the quick selection of multiple regions at once, without the need to return to the control panel each time. Here the completion of a selection does not automatically exit the selection state; instead each selection spawns a separate region-of-interest until the selection state is manually exited. In this manner, one is able to, for example, quickly click or tap a number of counties and immediately have probes spawned for each region without the iterative process of entering the selection state each time.

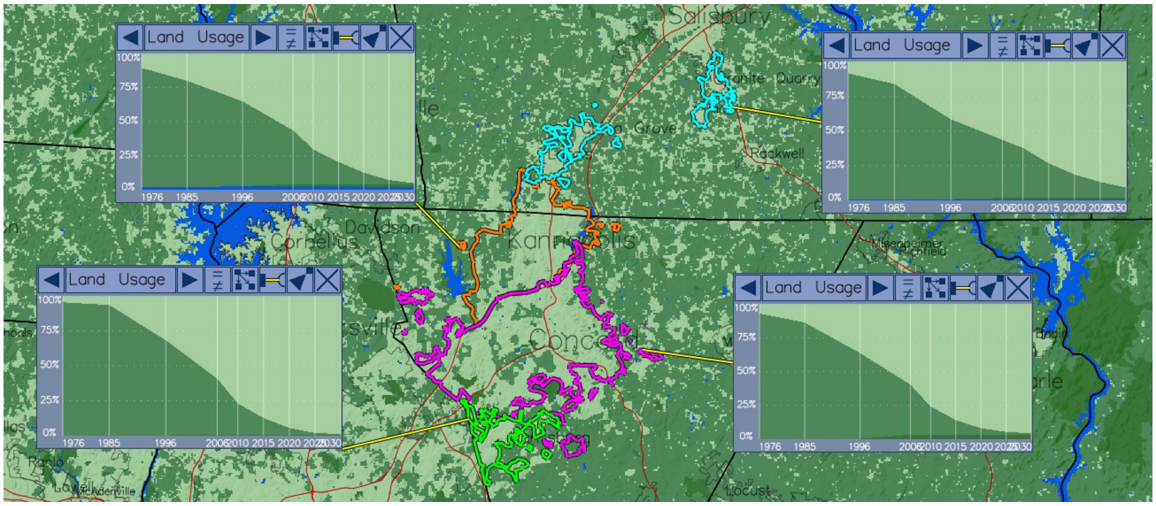


Figure 14: Example regions-of-interest generated through the use of the municipal boundary selection tool. This method allows for quick and easy selection of complicated political boundaries including unconnected outparcels. Here the user has separately selected three adjacent cities (orange, magenta, green), as well as a collection of smaller cities (all cyan).

The Urban Growth DSS also leverages image-processing techniques to provide another powerful method for creating regions-of-interest: the connected components selection tool. This tool allows one to select features extracted directly from the map. The user enters the connected components selection mode and then clicks a seed point on the map. The application then expands the selection outward from this seed point, following the edges of the features in the data layer that is currently displayed. For example, this can be used to select a feature (e.g. a park, urban core, etc) on the land-use type map. This type of use is shown in Figure 15, where the tool is used to select the extremely complex shape of a lake. This tool can also be used to select features within the underlying data layers as well. Figure 16, for example, shows it being used on the footprint (acres developed per person) data layer to select an area of extremely high values. The ability to probe the data layers themselves is a particularly effective enhancement for geospatial analysis.

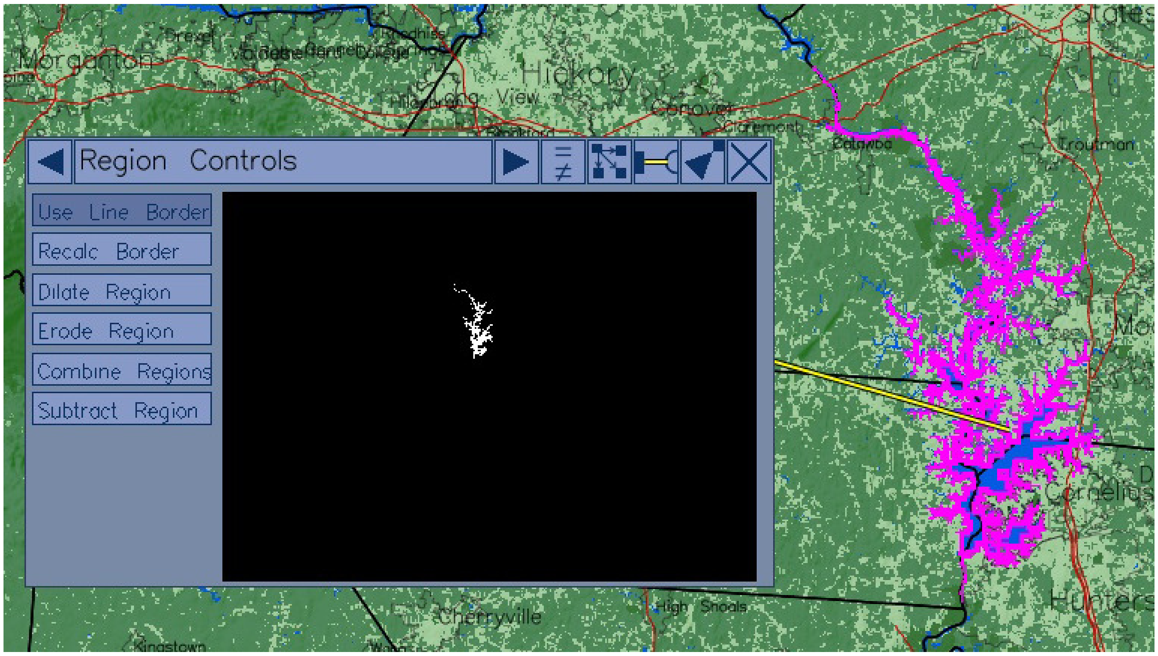


Figure 15: An example of a region-of-interest generated using the connected components selection tool. Here the user has selected a seed point within the lake, and the selection automatically expanded to fill the entire lake. Also shown within the associated probe interface is the region control panel.

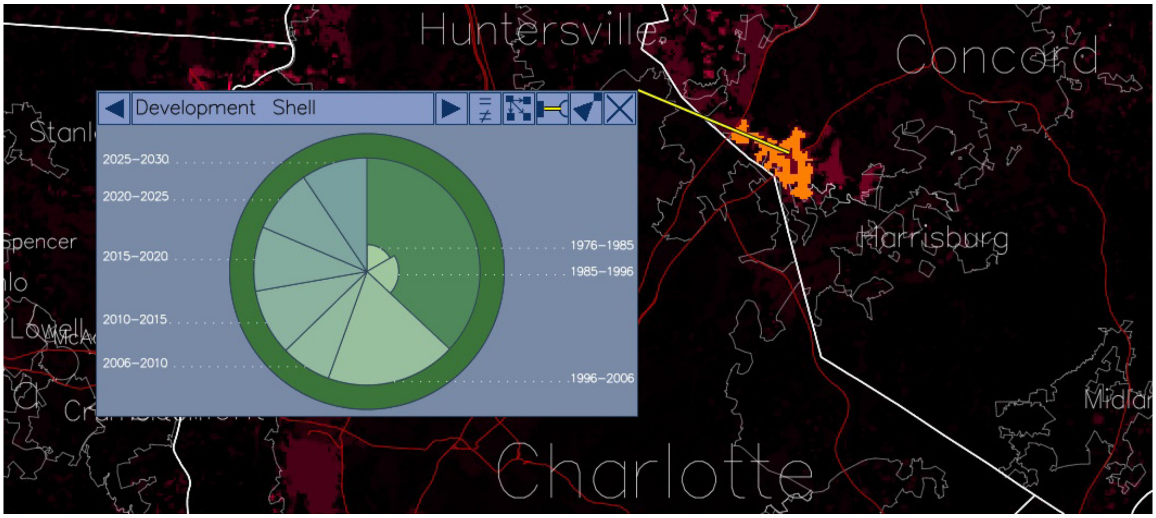


Figure 16: A scenario in which the user was examining the footprint (acres used per person) data layer, and noticed an area of extremely high values. (This area contains a huge mall and a race track, with minimal residential housing.) They used the connected components selection tool to select the area (orange). In the associated probe-interface, the development shell diagram shows a significant amount of protected land (the dark outer ring), which is a golf course/park, and that all undeveloped land (medium green) was completely paved over and developed between 1996 and 2006. (This was the time period in which the mall was constructed.)

Tools to refine already selected regions are provided in a region control panel, located within the probe-interfaces. The region control panel can be seen in Figure 15. Tools provided here include morphological dilation and erosion operations, and Boolean combination and subtraction of regions with one another. An example of the usefulness of these tools is that often the regions-of-interest resulting from feature based selection tools, like connected components, are too complicated and/or contain many holes. As shown in Figure 17 this problematic characteristic can be quickly and easily corrected with the use of the tools provided in the region control panel.

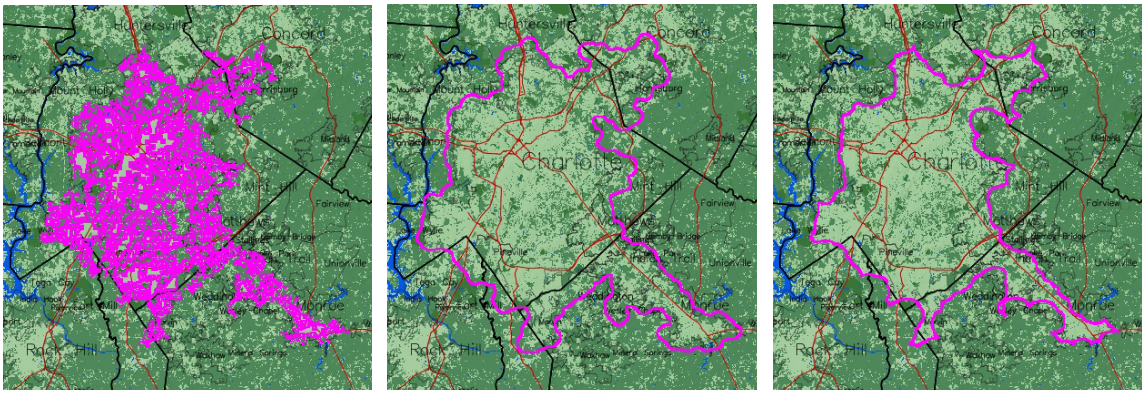


Figure 17: An example of boundary refinement for a region-of-interest generated with the connected components selection tool. In the first image, the region is very complicated and contains many holes. The second image shows the same region after the Dilate Region tool is applied. The third image shows the results of performing a Dilate Region operation followed by an Erode Region operation. In this final result, the simplified boundary tightly conforms to the mass of developed pixels that the user was originally interested in selecting.

The wide assortment of free-form, pre-defined, and feature based region selection methods available provides great freedom in how the user can query the data. However, it also means that the user can easily select unequal (in terms of size, composition, etc) regions for comparison, exacerbating the modifiable areal unit problem, which inherently arises in this type of analytical situation.

3.4.4: Comparisons

After the user has selected multiple regions-of-interest, each spawning its associated probe-interface, they can choose to combine these interfaces with each other to form comparison interfaces. In these interfaces, the visualizations pull the data from the individual regions-of-interest and plot it directly against each other. Upon the creation of a comparison interface, the statistical distribution of the regions across all relevant dimensions is calculated. If it is determined that any of the regions being compared are potentially significant outliers within a particular dimension, then the user is alerted to this by displaying a large flashing exclamation mark on that comparison window's toolbar. A comparison interface with this alert icon is shown in Figure 18.

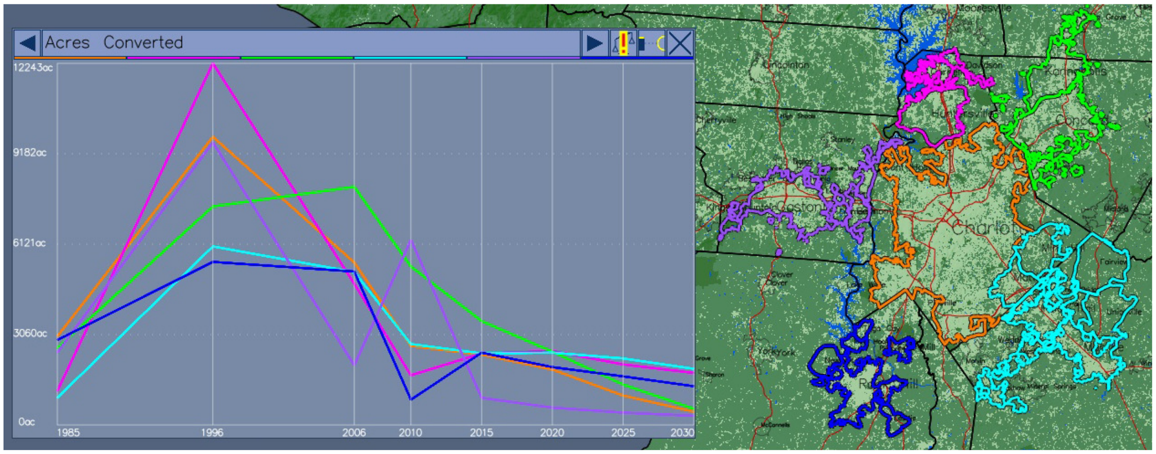


Figure 18: An example of a comparison interface in the Urban Growth DSS. Here the user has selected a major metropolitan area (orange) as well as several groups of suburb cities. The comparison panel is plotting the number of acres converted from undeveloped to developed in each timestep of the simulation. Notice the yellow and red alert icon ('!') in the upper right of the comparison interface, alerting the user that potentially significant outliers have been detected.

3.4.5: MAUP Overview Panel

From within a comparison interface, pressing the MAUP interface icon switches the interface to the MAUP overview panel. The purpose of this panel is to allow the user to evaluate any potentially problematic inequalities and choose which to take corrective action upon.

In the MAUP overview panel, each dimension has its own one-dimensional plot and action button, as shown in Figure 19. The plot itself is centered at the mean value for the dimension and expands three standard deviations above and below the mean on each side. Each region being compared is then plotted as a vertical line color coded to match the region. Regions beyond three standard deviations of the mean are plotted at the appropriate end of the plot. Any regions that were determined to be outliers are highlighted with a yellow indicator above the plot, and that dimension is automatically selected for corrective action. Under each dimension's plot, there is a scale/measuring tool that allows the user to drag across the plot to quickly measure the actual range of values across a cluster of regions, as well as the actual value by which outliers deviate from these clusters.

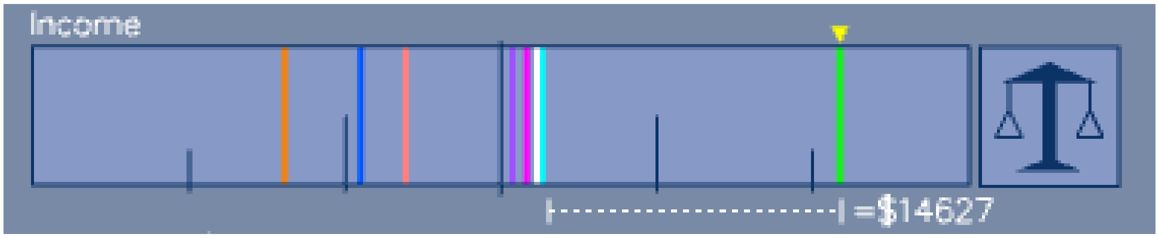


Figure 19: An example plot from the overview panel showing the distribution of eight regions in terms of average income. Notice that the green region has been flagged as a potentially problematic outlier, and that the user has measured its deviation from the main cluster.

One shortcoming of this technique is that there is a limit to how many regions can differentiated from each other with any color coding scheme. There are only so many distinctive colors, and after about ten regions, it becomes hard to distinguish which lines correspond to which regions. The limit on the number of unique colors can be overcome, however, with schemes such as labeling or highlighting regions on the map upon selection.

Upon entering the MAUP overview panel, the user can quickly assess the situation by viewing the highlighted dimensions with potentially problematic outliers and choose whether to either accept the suggested and automatically selected dimensions, or select and deselect dimensions at will. In practice, the user will rarely want to simply accept all of the suggested selections, as they are usually interested in looking at the differences between regions in at least one dimension. The choice of which dimensions are to be adjusted relies heavily on the domain knowledge of the analyst, specifically in knowing whether or not inequalities in a particular dimension will affect a particular analytical task, and how much inequality is required for there to be a significant effect.

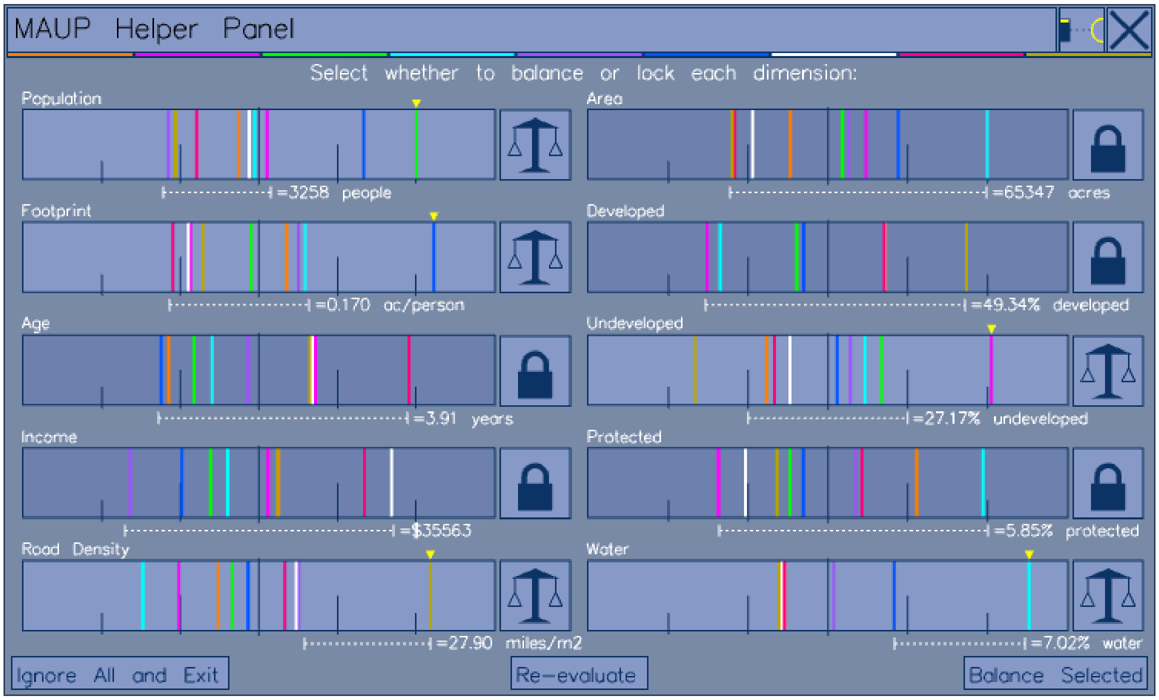


Figure 20: An example view of the MAUP overview panel. Notice that the dimensions “Population”, “Footprint”, “Road Density”, “Undeveloped”, and “Water” have all been automatically selected because the software has detected outliers in each of them. The outliers have been highlighted with a yellow triangle above them. Also notice that the user has used the measuring tool to determine the range of the main cluster of regions in “Population” and the deviation of an outlier from the main cluster in “Road density.”

Once selections have been made, pressing the “Adjust selected dimensions” button transfers the user and the selected dimensions to the MAUP adjustment panel.

3.4.6: MAUP Adjustment Panel

In the MAUP adjustment panel, each dimension that was selected in the MAUP overview panel is once again presented as a one-dimensional plot of the statistical distribution of the regions being compared. However, now the purpose of this graph is to adjust minimum and maximum values for the boundaries to be used as targets during the region adjustment procedures.

For each dimension, the user can move the ends of the selection box to encapsulate an existing cluster of regions in the plot, or to define a new range, within which they would like all regions to fall. The actual value range of the selection is presented below the plot. A target value is also indicated by an upward pointing green triangle. Outliers outside the selected range are those that the adjustment algorithms will adjust until they either reach the target value, get as close as possible, or fall within the desired range, depending on which adjustment method is being used.

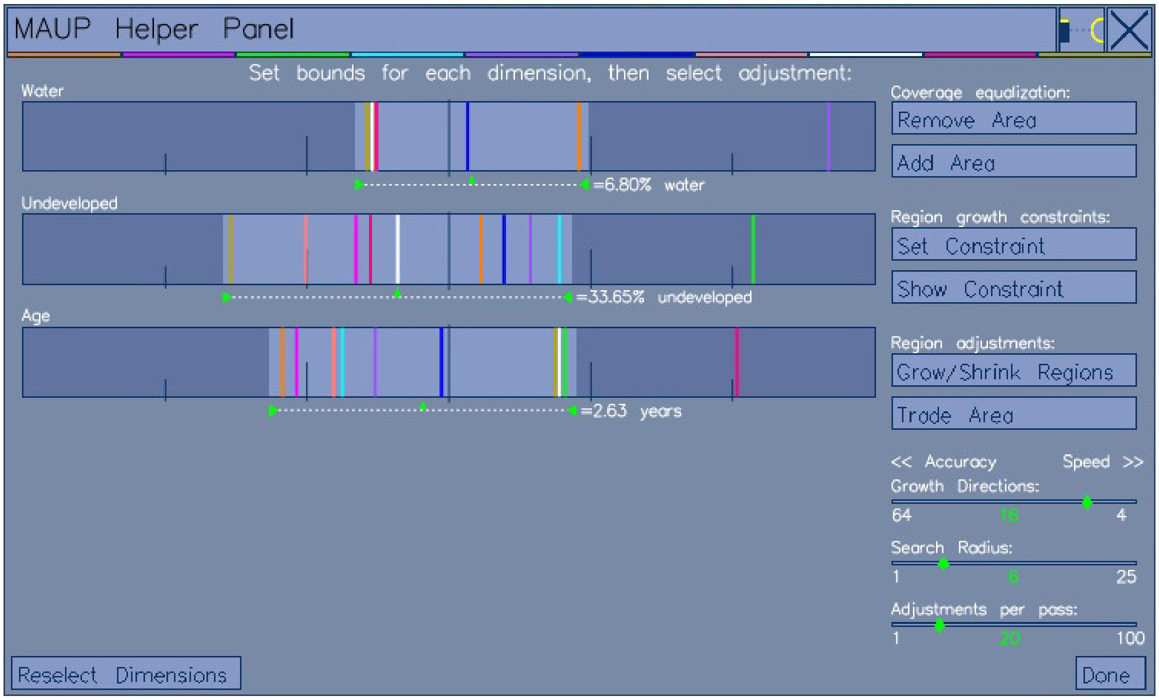


Figure 21: An example view of the MAUP adjustment panel. In this example, the user has selected three dimensions to balance. They have selected bounds around the main clusters of regions in each of the dimensions. The purple, green, and red outlier regions are those that will be adjusted when the tools available to the right are used.

Regions are adjusted multi-dimensionally with respect to the bounds set for all selected dimensions. As such, one might wish to select a dimension, not to actually adjust the regions within it, but solely to enforce an existing range within which those regions must stay while other dimensions are being adjusted. (E.g. if all regions had populations within a certain range and one wanted to preserve this maximum range during adjustment of total area.) This can be done simply by stretching the desired bounds for a dimension to encompass all regions.

Once the desired boundaries are set for each dimension, the user can choose from an assortment of adjustment tools, which are enabled or disabled based on the dimensions that have been selected for adjustment. However, before adjustments are initiated, the user has the opportunity to use the region-of-interest selection tools to create constraints around regions, which they will not be allowed to grow beyond. For example, as shown in Figure 22, one might want to adjust the boundaries of local political jurisdictions which must always remain a subset of a larger political jurisdiction.

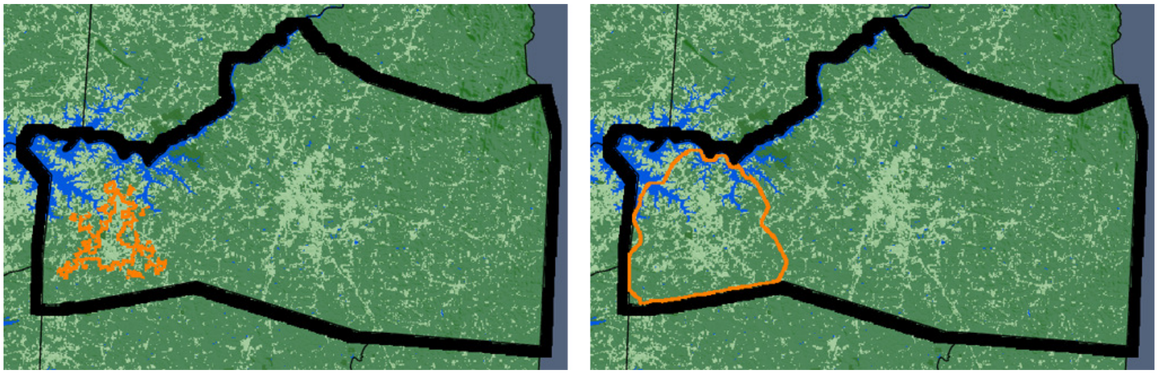


Figure 22: Before and after an add area adjustment of a city boundary (orange) within a constraint (thick black line) set for the county boundary that the city must remain inside.

The first, and most simple, adjustments available are “Add area” and “Remove area”. These are available for dimensions with categorical data, such as land coverage types, e.g. water, protected, etc. “Add area” attempts to expand regions that are below the minimum bound outwards into matching land types until either the target value is reached or until there is no available land within a reasonable distance. (“Reasonable” in this case is defined as how far a domain expert wants to allow any added, non-contiguous regions to stray from the main region.) "Remove area" erodes the boundaries of regions that are above the maximum bound inwards, removing matching land types until either the target value is reached or there is no more available land of that particular type to remove. Both of these methods can be easily adjusted to maintain the existing connectivity of regions, however in practice this greatly reduces its effectiveness and ability to reach target values, and provides little more than an aesthetic benefit in analyses that do not require contiguous regions.

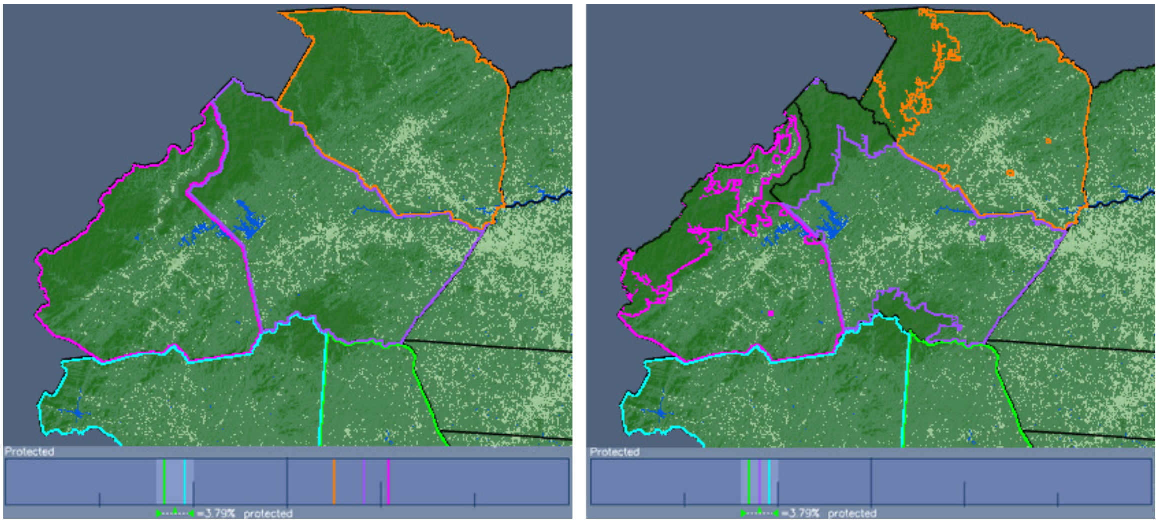


Figure 23: Before (left) and after (right) a remove area adjustment is made to remove protected wild lands (darkest green on map). The plot below on the left shows the selected cluster and desired bounds with the three outlier regions (orange, purple, magenta) outside the desired bounds. The plot below on the right shows the same distribution of regions, with the three outlier regions now situated at the target point (upwards green arrow).

The other adjustment tools are more complicated, but are able to adjust dimensions with continuous data. The first is "Grow / Shrink regions", which manipulates the boundaries of the regions both inward and outward at the same time, in an attempt to bring their values within the desired bounds. This is done through an iterative process consisting of simultaneous combinations of both removing and adding area at the edges of the regions to maximize movement towards the desired bounds while not exceeding the bounds set on other dimensions. The process completes when either the values for the selected dimensions fall within the desired bounds, or no more possible progress is achievable, e.g. no appropriate area is left available for removal. When using this tool, regions can both initially overlap, as well as overlap after adjustments are made. If overlapping results are not desirable, regions can be prohibited from growing into each other.

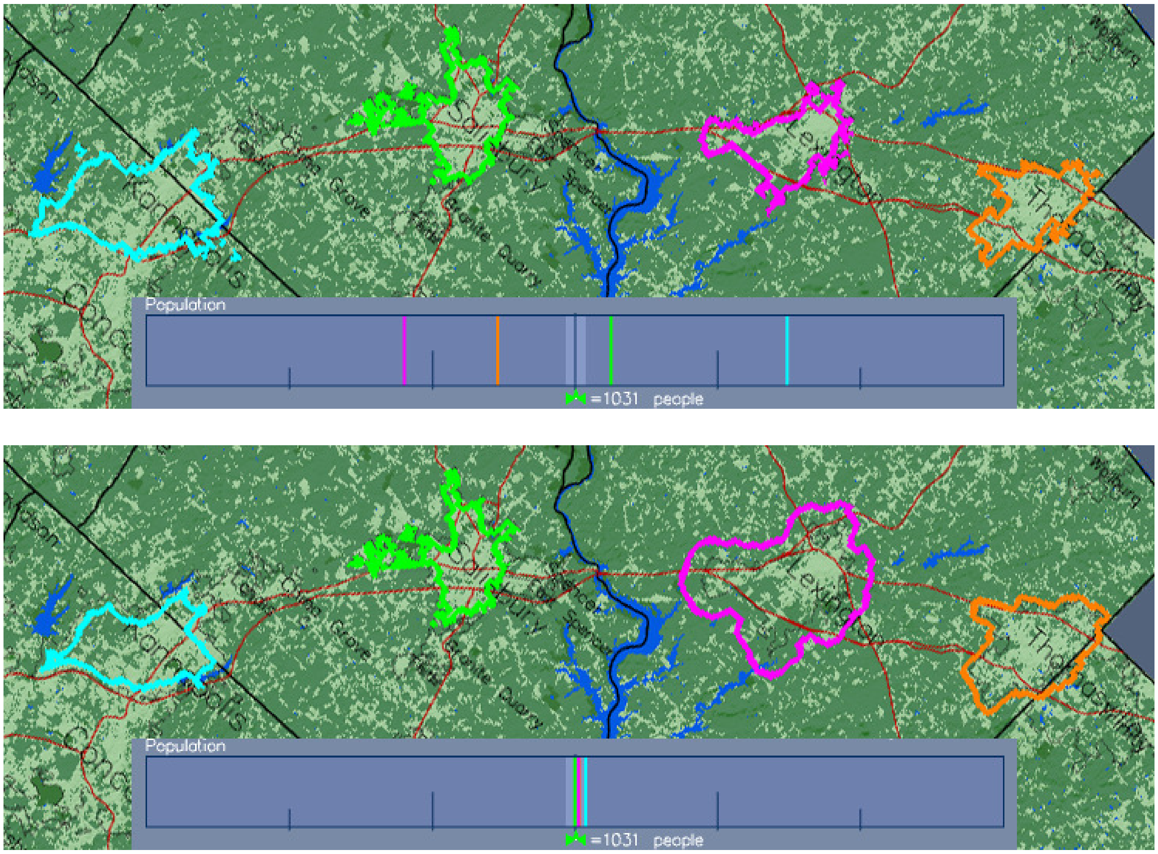


Figure 24: A grow/shrink adjustment of four regions to bring the population of each to be within ~1000 people of the mean. Above are the original predefined city boundaries and below are the results of the adjustment.

The final adjustment available, “Trade area”, is the most complicated. It is used to adjust border-sharing and space-filling regions, such as political jurisdictions, which cannot overlap and must collectively cover a certain area completely, as opposed to the collections of disjunct and/or overlapping regions adjustable by the previous methods. It behaves much like the “Grow / Shrink regions”, in that it attempts to both grow and shrink portions of regions’ boundaries to bring values for selected dimensions within the desired ranges, but now it considers the costs and benefits of each boundary adjustment to the regions of each side of the boundary. Thus it is actually weighing the benefits of trading bits and pieces of area between the regions. Using a greedy algorithm, it iteratively executes the most advantageous trades of area between regions, redrawing the boundaries of multiple regions in the process, until it achieves its goal or runs out of valid adjustments.

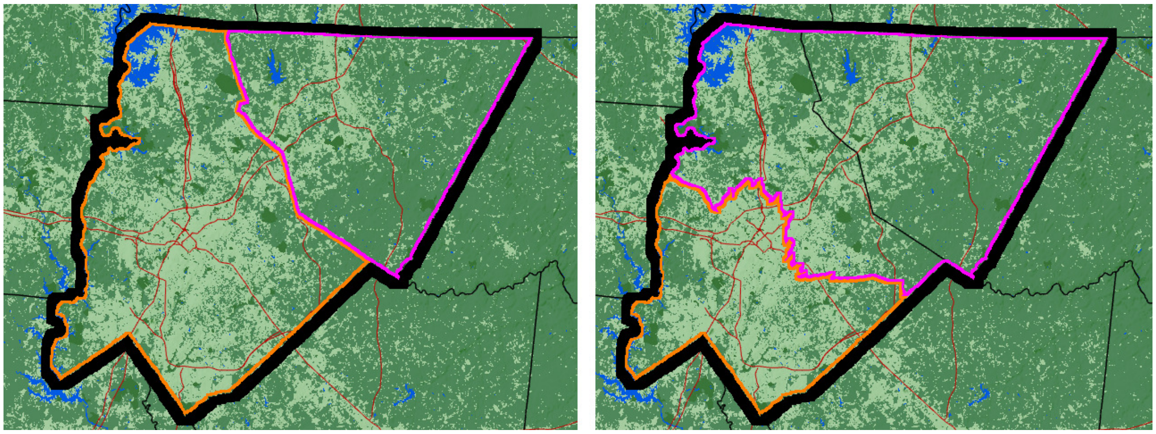


Figure 25: Before and after a trade area adjustment of two regions to make their populations equal. The thick black line is a constraint used to force the regions to stay within their non-shared boundaries.

The MAUP adjustment panel also provides sliders for expert users to adjust tradeoffs between region adjustment accuracy/aesthetics and speed of calculation. These can be seen in the lower right of Figure 21. The behavior and effects of these variables is detailed in Chapters 3.4.9 through 3.4.11.

Careful consideration and domain knowledge is still required by the user as to choosing which dimensions to adjust, target bounds, and adjustment methods. However, the MAUP helper panel attempts to assist the user in making these choices through both helpful intuitive visualizations and enabling only those adjustment methods relevant for the selected dimensions.

3.4.7: Implementation Details

The Urban Growth Decision Support System software accepts two main types of data. The first is vector data that is used to provide both reference, e.g. roads, city names, as well as semi-automated assisted selection techniques, e.g. "Select City Bounds." It utilizes the ESRI shapefile format for this type of data, which is ubiquitous in the GIS community.

The second data type is raster based data layers, in .tif format. These raster images provide the raw data for the application, such as land coverage and demographic information. For most variables, conversion from existing GIS formats to this raster based format is fairly straightforward. However, for many household based demographic variables, such as median income, consideration must be made with regard to ensuring the most accurate distribution of aggregate data to individual pixels, so as to minimize ecological fallacy effects.

When collaborators in The Center for Applied Geographic Information Science created the underlying population maps, for example, instead of merely dividing the population of a census block by the number of pixels within it to get a population value for each pixel, supplementary data was utilized, including satellite imagery, to perform dasymetric mapping (using the method described by Mennis and Hultgren [Mennis 2006]). In this manner, if a census block contains farmland as well as an urbanized area, the pixels in the urbanized area would contain the majority of the population, while the farmland areas with no impervious surfaces would have near zero population values. This was very important for this particular application, where users are interested in the differences between developed and undeveloped areas, and can select their own regions cutting through census blocks.

The Urban Growth DSS was written in C++ and uses OpenGL for all onscreen graphics. OpenCV [OCV 2009] is used to perform all image processing operations.

3.4.8: Statistical Evaluation

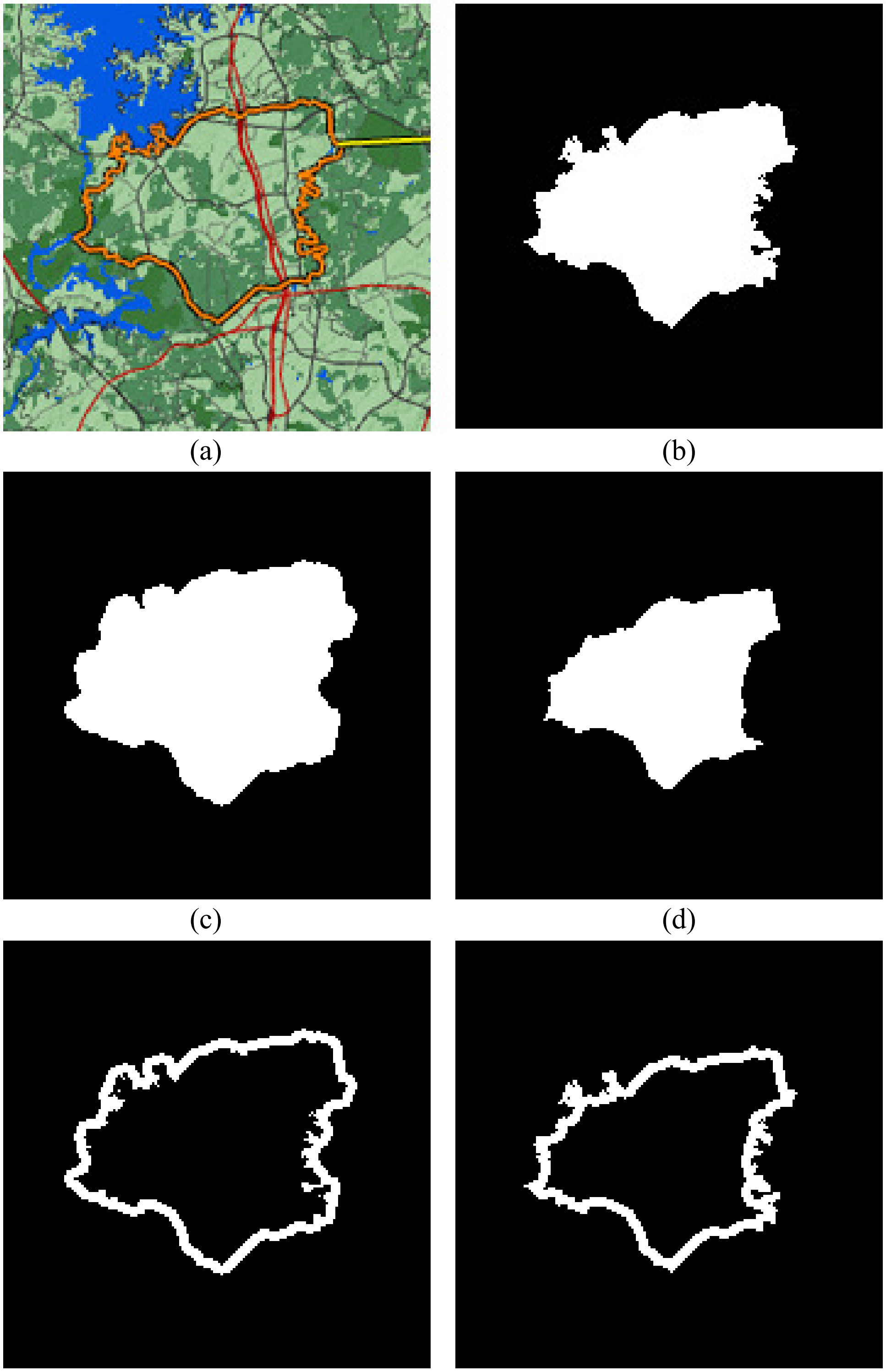
The statistical evaluation method used to detect outliers in the Urban Growth DSS is quite simple, but is sufficient for its purposes. Upon comparison interface creation, the mean and standard deviation for each dimension is calculated by examining the precomputed values for all regions that are being compared. The number of standard deviations from the mean value is used to detect outliers. Greater than two standard deviations from the mean is used as a threshold, over which the user is alerted to the detected outlier and that dimension is automatically selected for adjustment. A more rigorous statistical evaluation could easily be substituted here if deemed necessary.

3.4.9: Adding and Removing Area

The “add area” and “remove area” functions, which expand or contract a region’s boundary to include more or less of a particular categorical value, behave as follows: First a search mask is generated that is used to find candidate pixels to either add to or remove from the region. This process is visually explained in Figure 26.

A binary image mask (b) is extracted from the region-of-interest, describing what areas make up in the current region (a). If adding area is desired, morphological dilation is performed on this mask, resulting in an expanded mask (c). A search mask (e) is then generated, which equals (c AND (NOT b)). The search mask is a ring around the outside of the original mask containing all pixels within the chosen kernel size (how to choose the size is discussed later) of, but not within, the original mask.

Likewise, if removing area is desired, morphological erosion is performed on the original mask, resulting in a shrunken mask (d). A search mask (f) is then generated as ((NOT d) AND b). This results in a ring around the inside of the original mask, with all pixels within the original mask’s boundary by no more than the kernel size.



(a) (b)

(c) (d)

(e) (f)

Figure 26: The process of calculating search masks for adding or removing area from a region. (a) is the region-of-interest to be adjusted on the map, (b) is the binary image mask for the area inside this region, (c) is the dilated mask, (d) is the eroded mask, (e) is the dilation search mask, and (f) is the erosion search mask. Notice that (e) = ((c) - (b)) and that (f) = ((b) - (d)).

After generating a search mask, the pixels within are examined individually to see if they match the categorical type the adjustment pertains to. If trying to add area, these pixels are set as true in the original mask defining the region. If trying to remove area, they are set as false in the original mask. This process is continued until either the desired number of pixels has been added or removed, or there are no more candidate pixels in the search mask. In the former case, adjustment for this region is done. In the latter case, the process is repeated, generating a new, further reaching, search mask.

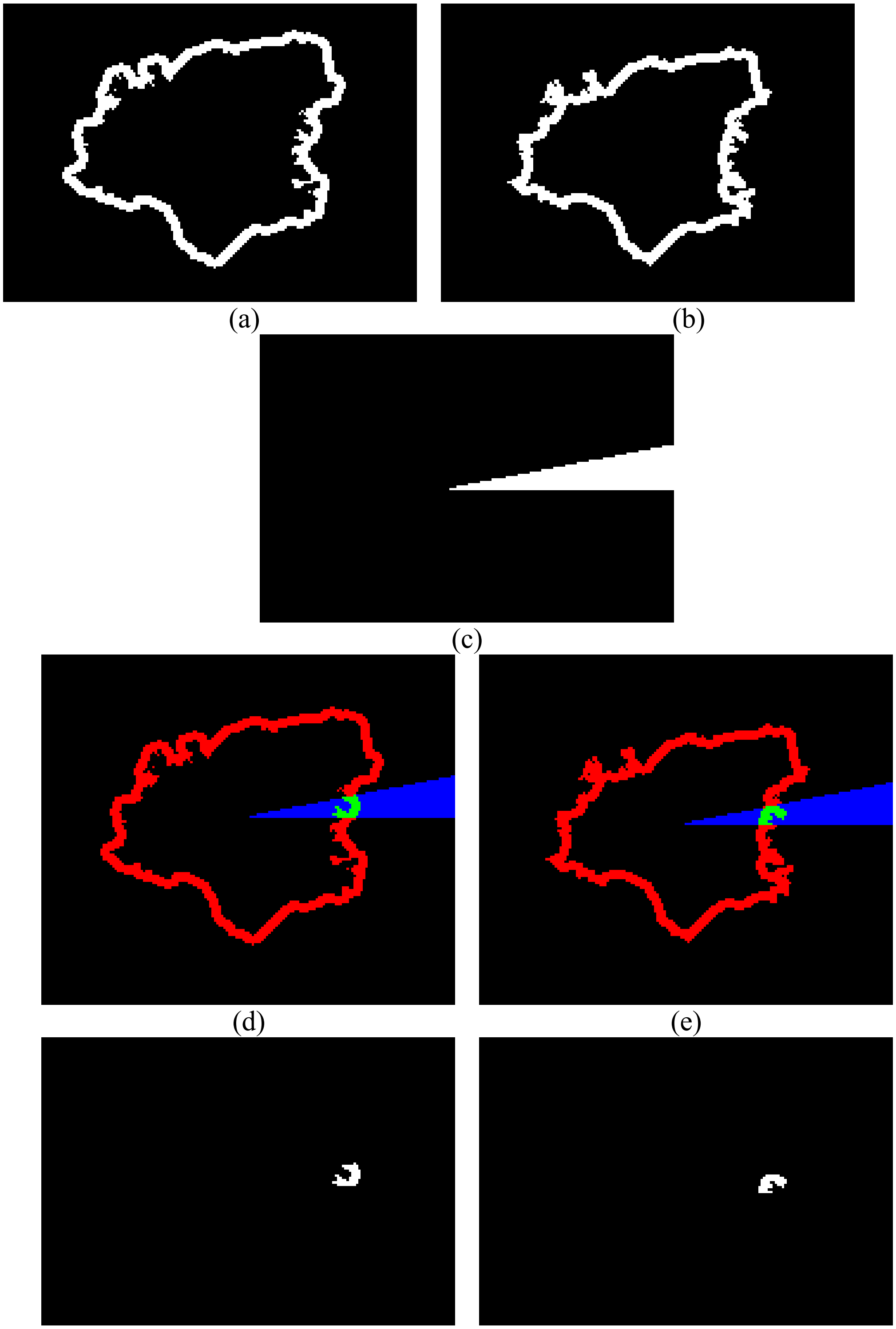
Aside from achieving the target goal, there are two other stopping conditions: When removing area, the adjustment algorithm stops if there are no longer any new candidate pixels being generated, i.e. all possible pixels that can be removed have been removed. When adding area, it will also stop if a certain number of dilations have failed to unearth any candidate pixels that match the specific categorical type of interest. The number of fruitless dilations allowed before stopping dictates how far away new disjunct sub-regions can stray from the original region.

The choice of kernel size for these morphological operations is a tradeoff between speed (less iterations required) and even growth (or reduction) patterns. Larger kernel sizes have a tendency to provide more candidate pixels than needed. The algorithm converts candidate pixels in a scanning pattern from the top left, and so this can result in growth mostly in the northern direction when kernel sizes are too large. Lower kernel sizes ensure that multiple concentric rings of candidates will be evaluated, resulting in a more even, outward growth. Anecdotally, a 7x7 kernel is a good balance. By using a 3x3 kernel one can ensure that only those pixels that are directly connected to the edges of the region will be added or removed, and hence no new disconnected islands or holes will be generated. Additional enhancements, such as converting pixels in order of local concavity, could be added to increase the smoothness/aesthetics of resulting boundaries. These variables can be adjusted by expert users in the MAUP adjustment panel through the use of several sliders, as shown in Figure 21.

3.4.10: Growing and Shrinking Regions

The "grow and shrink regions" function attempts to automatically augment the size and shape of regions, independently of each other, in order to adjust outlying values in selected dimensions to be within the specified value range. Each region is checked to see if it has at least one value outside the desired range in any of the dimensions selected for adjustment. If so, the algorithm attempts to adjust this region, then move on to evaluate the next region.

The adjustment process for individual regions, which is visually explained in Figure 27, begins with the generation of search masks from both dilation and erosion operations on the region's mask, as detailed in Chapter 4.2. After calculating these two masks, which form rings both inside and outside of the region's current boundary, they are cut up into a number of candidate sub-masks. This is done by finding the center of the region, and then generating a number of "pie slice" shaped masks emanating outwards from the center point (using OpenCV's cvEllipse function). A collection of candidate adjustment masks is then generated by computing the binary AND of each slice mask and the erosion and dilation masks. If it is desirable to restrict regions from growing into each other, the other regions' masks can be subtracted from the candidate masks.



(a) (b)

(c)

(d) (e)

(f) (g)

Figure 27: The process of creating candidate adjustment masks. (a) & (b) are the dilation and erosion search masks generated as in Figure 9. (c) is a sample of one of the many slice masks that are used to divide up the search masks. (d) & (e) show the slice mask superimposed on each search mask to show the Boolean operations. (f) & (g) are the resulting masks, representing candidate adjustments.

The number of slices to cut the original erosion and dilation masks into is a tradeoff between speed of computation and accuracy. By making too few, and thus larger, slices, the regions are very restricted in their choice of growth directions, will not add or remove area as efficiently, and are less likely to reach their dimensional value goals. A reasonable solution is to choose a number of slices based on the current size of the region. Small regions (< 3km wide) may require as few as eight slices for sufficiently pleasing results, while larger regions (~30km wide) can benefit from as many as 30-40 slices. Another option here is to vary the number of slices on each pass, as the size of the region changes. By varying the number of slices in each pass, one also lessens the chance of unnatural looking radial region boundaries.

All non-zero candidate masks are processed as temporary regions-of-interest and evaluated to determine the values it contains for each dimension of interest. The temporary region-of-interest and mask are discarded, and the values are stored in a candidate adjustment object along with details including which operation type (erosion or dilation) and slice number was applied to generate it.

After all candidate masks are processed into a list of candidate adjustments, they are sorted in descending order according to the progress they would make in bringing the values for the dimensions that still need adjustment within their desired ranges. A subset of this list is then chosen by starting at the top and evaluating if the adjustments would result in the region moving outside any of the bounds for the other dimensions, or overshooting the target values. By deciding how far down the list to evaluate on each pass, one can make a tradeoff between speed and closer-to-optimal results. Selecting only the single best adjustment from the list results in only the locally optimal choice being made on each pass. Conversely, selecting all valid adjustments produces quick results, but they may be far from the optimal solution. This, again, is adjustable within the MAUP adjustment panel.

Once a subset of adjustments has been selected, two new slice masks are redrawn, one containing all the slices that correspond to selected adjustments that were erosions, and one for the selected dilation slices. Each slice masks is AND'ed with the corresponding erosion and dilation masks. The resulting sliced erosion mask is then used to remove pixels from the regions mask, and then the sliced dilation mask is used to add pixels to the regions mask.

This whole process repeats itself until either the values for the selected dimensions are all within their desired boundaries, or no candidate adjustments are found to be acceptable, and thus there are no more adjustments that can be made. In the latter case, a refinement can be made which increases the number of slices and searches again, with the candidate areas now of smaller area.

3.4.11: Trading Area

The trade area function is similar to the "grow and shrink regions" function, but instead of adjusting the regions independently of each other, it adjusts regions with respect to each other. This is used for cases where regions border each other, and the user does not want them to overlap, but is willing to allow the boundary between them to move.

To accomplish the adjustment of multiple regions at once a modified greedy algorithm is employed. This solution makes the optimal choices on each pass but does not guarantee the best possible solution. It can however be fast enough to return results within a short enough amount of time (< 5 minutes) to maintain interactivity, whereas finding the optimal solution could take hours or days. It is merely a proof of concept implementation at this point, and future work should be done to make this adjustment as efficient and effective as possible.

The trade area algorithm begins by generating a list of candidate adjustments in the manner described in Chapter 4.3, but this time they are generated for each region. This time, not only does it record the effects the adjustment would have on the region it was generated from, but also its converse effects on any other regions that either currently contain, or are proposed to contain it.

For each region and dimension that needs adjustment, the list is sorted by how far the adjustments would move the outlying value into the desired bounds. The algorithm then starts at the top of the list and looks for adjustments that are advantageous (they move the value towards the target) and do not bring the values in other dimensions outside those bounds. Matching candidate adjustments have a preference value incremented each time they are chosen to be made by a region or dimension.

After all regions and their dimensions have been considered, the list of candidate adjustments is sorted by preference value. The top N adjustments from this list are executed, as long as they have a preference value of at least one. The choice of N, how many of the top requested adjustments to make, is another trade-off between speed and how close the results will be to the locally optimal solution. A value of 5% of the total number of candidate adjustments works well, different values are appropriate across various situations. Again, this value is adjustable within the MAUP adjustment panel.

When executing the top N adjustments, the same process as in Chapter 3.4.10 is followed. However, when using these sliced dilation and erosion masks, not only are the pixels added or removed from the region the mask was generated from, but the opposite operation is performed on the same pixels in the neighboring region(s). In this manner, the area/pixels are transferred between regions.

Once these adjustments are complete, the regions are checked to see if they still have values outside the desired bounds. If they do not, then this adjustment process has finished successfully. If so, then another pass is made. If another pass still results in no acceptable candidate adjustments, the adjustment process can be stopped, or the number of slices per region could be increased in an attempt to find smaller valid adjustments.

3.4.12: Scenario

In this example scenario, the analyst's goal is to compare the growth patterns, both historical and predicted, for a number of cities, and clusters of smaller cities, that are all suburbs of Charlotte, North Carolina. The analyst wants to examine the relationship between the amounts of land that change to a developed state with each time step, and during what times development rates peaked, for each region relative to the others. As shown in Figure 28, they have selected regions-of-interest using the city selection tool.

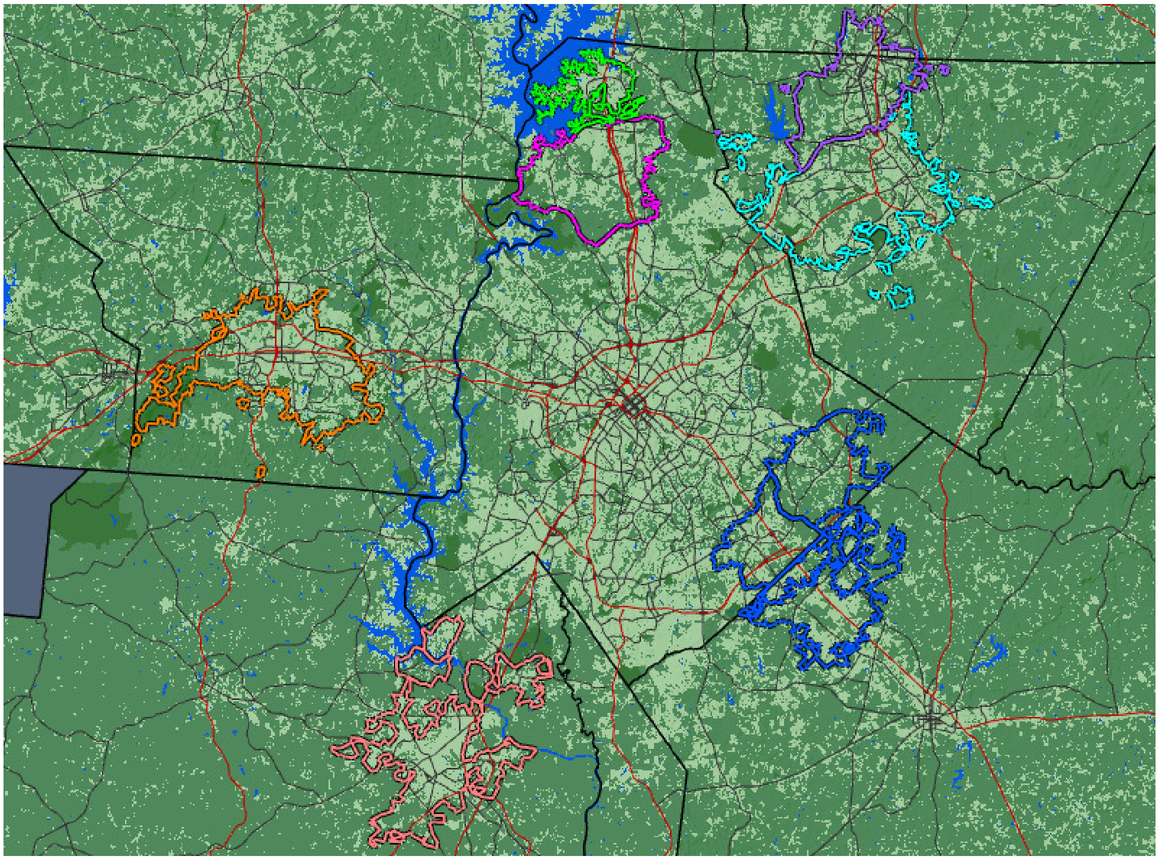


Figure 28: The seven regions selected by the analyst in the scenario. Each corresponds to a city, or group of cities (e.g. the blue region), that are all suburbs of a major metropolitan area (center of map).

However, some of the regions contain significant amounts of water, and others significant amounts of protected land. The simulation is programmed to ignore both of these land-cover types, and they will never get developed. As such, their presence can cause misleading results for analyses or visualizations that rely on ratios involving developing land. This effect can be seen in the “land use pies” visualization shown in Figure 29, where water (blue) and protected land (dark green) slices squeeze the developed (lightest greenish-yellow, one for each time step) and undeveloped (light green) land slices into remaining degrees of the circle. Comparing the angles between regions to see the relative amounts of growth that occurred in each time step is now misleading. The same amount of growth will appear smaller in the region with excess water.

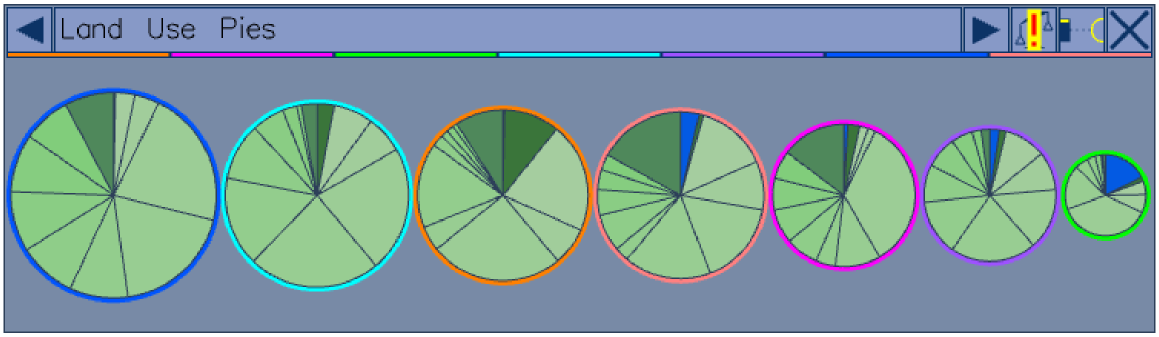
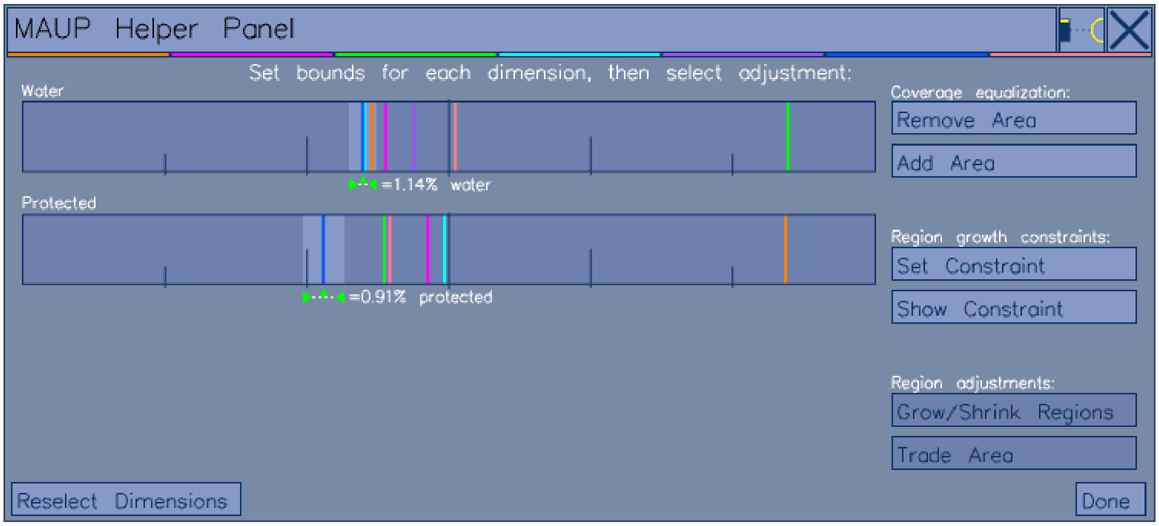


Figure 29: Pie charts of the amounts of land developed over each time step Notice the amount of water in the 7th region and protected land (darkest green) in the 3rd region. Also notice the MAUP alert icon (yellow and red '!') is active in the upper right hand corner of the interface, informing the user that it has detected some potentially problematic outliers.

The user is alerted by the flashing MAUP alert icon, and enters the MAUP overview panel. Here, as shown in Figure 30, the footprint (land developed per person), road density, undeveloped, protected, and water dimensions have been automatically selected due to outliers being detected within them. Not concerned with footprint or road density, the user unselects those dimensions. The user also unselects the undeveloped dimension, as the variations within that dimension are one of the aspects of the data they are interested in.

Figure 30: The MUAP overview panel from the scenario. Notice that the application has automatically detected outliers in the footprint, road density, undeveloped, protected, and water dimensions. These dimensions have also been automatically selected for adjustment.

The user advances to the MAUP adjustment panel. As shown in Figure 31, they set the target bounds for each dimension around the regions with the least amounts of water and protected land. They then select the “remove area” tool to bring the other regions within those bounds. Figure 32 shows the water and protected space being removed from the regions with excesses. Finally, Figure 33 shows the pie charts, now free of the misleading distortions from excess water and protected land.



Set bounds for each dimension, then select adjustment:

Figure 31: The MAUP adjustment panel from the scenario. Notice that the user has put tight desired bounds around the region with the lowest values for both land use types.

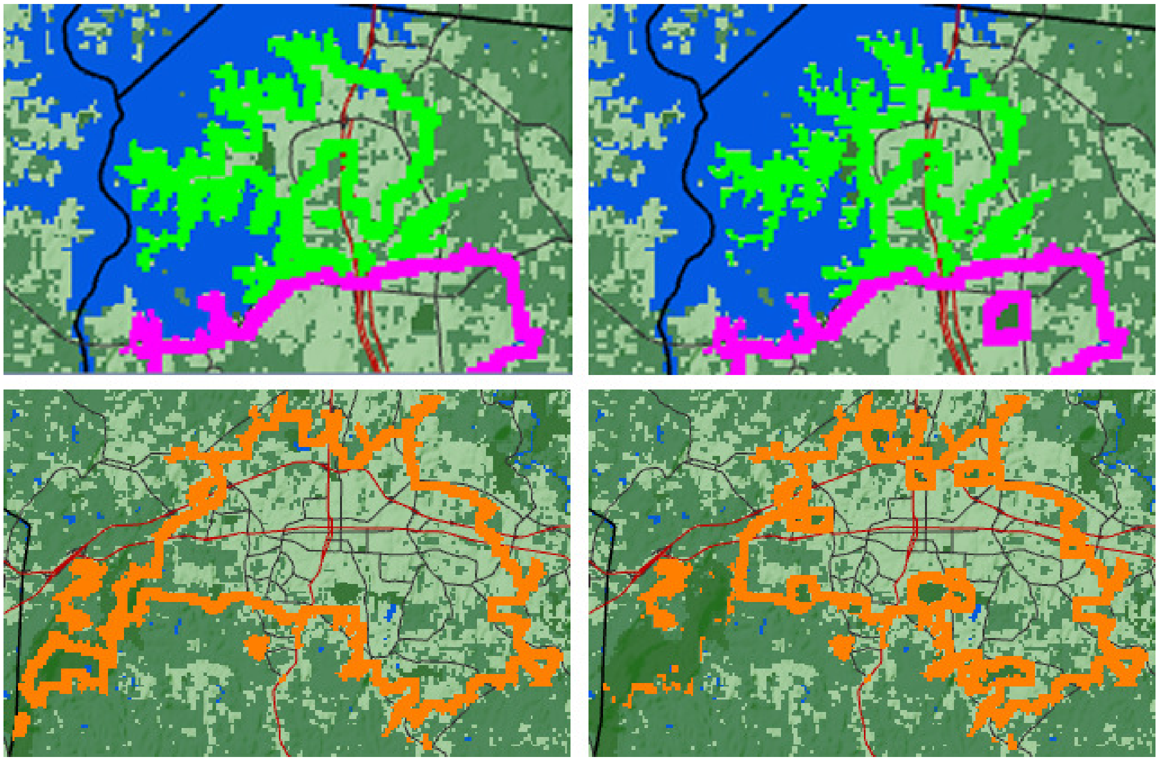


Figure 32: Some of the regions before (left) and after (right) the adjustments made in the example scenario. Notice the removal of water (top, green region) and protected land (bottom, orange region).

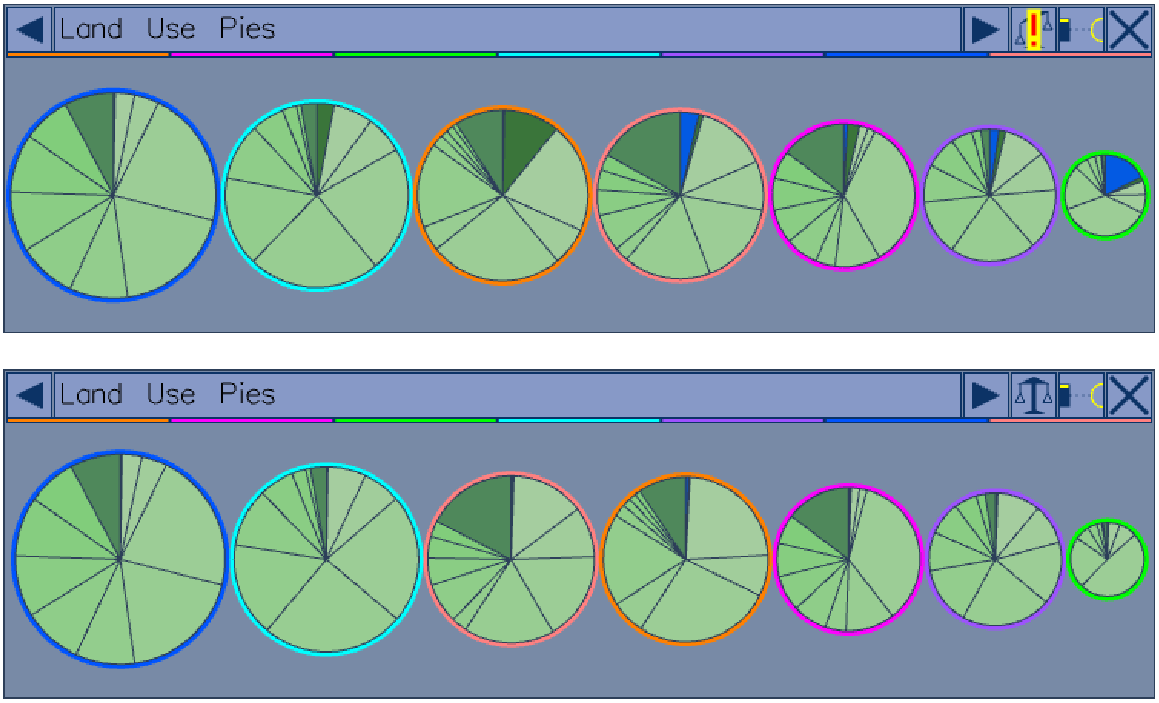


Figure 33: The same pie charts as in Figure 12 before (above) and after (below) adjustment to remove excess water and protected land. Notice that the 3rd and 7th regions are no longer distorted.

3.5: General Implementation

This section provides a detailed description of the implementation methods required to reproduce a fully-functional probe-based interface within a new or existing simulation or other geospatial visualization application. It describes each of the (not necessarily novel) major components and their behaviors, as well as what each is tasked with, how these components interact, and the system's behavior as a whole.

3.5.1: Panel / Window Management System

Before one can implement the individual components of a probe-based system, one should create a robust system to handle multiple panes or windows within a single graphical space. This system should allow the creation of any number of panels of any size. In addition to storing the basic data structure for each window, containing information such as window position and size, it should also manage the drawing order of these panels in such as method as to keep the most recently interacted with panels in front of those that have not been used recently. It should also allow you to lock particular panels either always in front or always in back of the drawing list. This allows one to set a base map (Chapter 3.5.2) to a background position, to ensure that all other components only get drawn on top of it. It also allows one to set a control panel (Chapter 3.5.3) to always be drawn last, ensuring that the buttons there are always visible and clickable.

This management system is the initial receiver for all user input, be it from a mouse/keyboard, or a more exotic interface, such as a multi-touch table. The system must take input events, determine which panels they fall within, translate the coordinates, and pass the event on to the individual panels themselves. Each panel should thus provide a "Handle Clicks" method, which receives interaction events and checks to see if they correspond to its interactive components or whitespace. If they correspond to interactive components, the panel will execute the consequent behaviors and return TRUE. If they correspond instead to only whitespace, such as the title bar, then the panel will return FALSE. This Boolean response will alert the management system to the event being satisfactorily handled. In the case that it is not satisfactorily handled, the management system will attempt to process the interaction as a panel repositioning or resizing event.

It is also useful to implement a simple animation system for panels. One way to do this is to have a function which receives as input a new position and size for a panel, as well as delay and length of animation times. Once called, it waits the delay period (useful for coordinating multiple complex animations,) and then begins to morph the panel from its original size/shape into the newly specified size/shape. For both aesthetic and perceptual reasons it is useful to do this animation non-linearly. For example, in the case in which a new panel is spawned from an existing interface element, the panel should start moving very fast, to attract attention. It should quickly travel the majority of they way to its destination and then slow down as it begins to come to rest. Avoiding an abrupt stop is both visually appealing and also simulates actual physical movement. In the Urban Growth DSS, animation is accomplished with a simple physics-type simulation in which the panel is given an initial velocity and then slowed by a frictional force. This was initially used in the collaborative table top environment where the user could not reach all the way across the table; to pass a panel to a user on the other side, the user could "flick" it, imparting a large velocity, and it would quickly move across the table, then slowly come to rest as if it was a physical tile one had slid across a real tabletop. By varying the initial velocity imparted on the panel, one could control how far it slid out of their reach.

3.5.2: Base Map / World

The base map or world is the most application/domain specific interface element in a probe-based interface. If converting an existing geospatial visualization to include probes, this will already be implemented, and may only require minimal changes to accommodate other elements. In the agent based simulation (Chapter 3.3) and Urban Growth DSS (Chapter 3.4) the base map was a 2D map that was always perpendicular to the camera. In the LIDAR change detection application (Chapter 3.1), and the census data exploration tool (Chapter 3.2) the base map was actually a 3D world, in which there was a terrain and the user could fly the camera around freely and examine the world from any angle. In the 2D case, the base map implementation is much simpler than the 3D case, due to the ease in selection, projecting points from screen space, and the displaying of vector data and raster images on terrain.

It is helpful to implement the base map within a panel, like all other components. This allows the panel management system to coordinate interactions with it, allowing one to switch input methodology easily (e.g. mouse to multi-touch table). It also means that any animation and other panel manipulation code can apply to the base map as well. In the 2D case, the base map can be scaled to fit entirely within said panel, in which case resizing and repositioning the panel is the method of map navigation. In the 3D case, the panel would need to contain a viewport into the 3D world, in which navigation would then occur directly, and resizing and repositioning the panel would only change the size and shape of the viewport.

3.5.3: Control Panel

There are a number of options for the implementation of interface controls. The first option is to overlay the controls on top of the rest of the system, such that the buttons are always on top and in the same location. This is seen in the table top interface, shown in Figure 11, where large pictographic buttons are arranged along the outer edges of the table, with repeating “toolboxes” on each side of the table to avoid issues involving reach limitations. In this case, all interactions are first checked against these buttons before being passed onto the panel management system.

The other option is more complex but is potentially far more rewarding. In this case, the control panel is implemented as another panel, as seen in the upper right of Figure 10. It can then be repositioned around the interface as needed, to best suit the user’s needs. If one allows the user to dynamically spawn these control panels, interesting new scenarios can be envisioned. For example, on a large-format multi-touch display, e.g. a table top the size of a conference table, or a room sized touch-wall, there is sufficient space for each user to have their own fully functional personal workspaces. See Chapter 5.3 for a discussion of how these dynamic workspaces could be handled and leveraged for collaborative usage.

3.5.4: Regions-of-Interest

The region-of-interest object is the core data structure utilized during probe-based interaction. It is what is created by the user's selections, its extracts and stores information, knows how to draw itself, and shares data between other interfaces.

When a user makes a selection, a region-of-interest object is created. The region-of-interest is provided with a link to the map it corresponds to, as well as a boundary, defined in map coordinates, that defines it. This boundary can be provided in vector form, as a collection of polygons, which is the usual result of a lasso or predefined geopolitical boundary selection. It can also be provided in a raster image based form, which is the result encountered when using selection tools such as painting or connected components selection. A list of edge coordinates (16 is usually enough) is also generated from the vertices of the polygonal boundary or the pixels of a raster based boundary. This list is then used by connectors, as described later in Chapter 3.5.6. At creation, a region also selects the next available color from a global pool of available colors. Brewer [Brewer 1994] provides helpful guidelines for choosing color schemes for categorical data, and carefully designed color schemes at colorbrewer.org. Since the map already used a modified colorbrewer.org color scheme for its categorical land use types, highly saturated, distinctive colors had to be used for region boundaries to ensure that they stood out visually from the map's shading.

After receiving a map and boundary descriptor, a binary mask defining the region-of-interest is generated. This can be created directly from provided raster boundaries, or generated through rasterizing a polygonal boundary. This mask is then stored for extraction and region manipulation purposes. Using image processing operations, a binary outline image is also generated from the mask image. This outline image is generated by conducting a morphological dilation of the region's mask by a few pixels and then subtracting the pixels from the original mask. This outline image is then converted to an alpha blended OpenGL texture and stored for use in displaying the region-of-interest directly on top of the base map.

Once a region mask has been generated, the region-of-interest object uses the mask to extract data from all of the different data layers containing both the output of the simulation and the underlying data layers from which it derives its results. This is done by first determining a bounding box for the mask, and cropping this portion out of each data layer for temporary assessment. For each of the data layers, all the pixels within the mask are examined and the total and average values for each dimension are recorded. These values are then stored within the region-of-interest object for retrieval from probe interface visualizations, comparison panels, MAUP evaluations and adjustments, etc. The cropped data layers are discarded, as there is no reason to return to them since all contents have been exhaustively totaled and averaged. One could also calculate additional statistics, e.g. standard deviation, during this extraction process if they are deemed worthwhile. It is important to extract everything you will ever want to know about this region in this one single pass, as to go back later to retrieve data values from the layers would come at a significant time cost. This scheme of pre-calculating all the possibly needed statistics at region creation time is crucial to the speed and fluidity of interaction and exploration in the probe-based system. Of course, for simulations that update their results as the user interacts, such as the agent based simulation in Chapter 3.3, the region-of-interest will need to query the new data either automatically as it is received, or on demand whenever a visualization needs to be updated with the newest data or an adjustment requires or influences the newest data.

A possible modification to increase the speed of this process is to communicate regions-of-interest directly to the simulation (which could be running on a separate computer), such that instead of updating the interface with entire data layers, it can simply update the selected regions and report the totals, averages, etc. for those regions. This technique would provide the dual benefits of offloading much of the updating efforts onto the simulation, as well as allowing it to concentrate its resources on the regions actually being examined.

Regions-of-interest are also tasked with being able to draw themselves directly onto the base map when provided parameters describing the base map’s position and size. This can either be done with an alpha blended overlay covering the entire region (using a texture generated from the binary region mask), with the precomputed raster based outline, or in vector form with the stored polygonal boundaries (if available). This duty could be offloaded to the base map, which would then retrieve region-of-interest data as needed, although such an implementation is less self-contained and portable.

Regions-of-interest are maintained as long as at least one other component, i.e. probe interface or comparison interface, is currently tied to them. When an associated interface is closed, it signals the region-of-interest, which then checks to see if any interfaces remain connected. If none remain it marks itself for deletion by the panel management system that stores all regions-of-interest.

3.5.5: Probe Interfaces

The probe interface is the main interface for examination and manipulation of regions-of-interest objects. For each probe interface there is exactly one corresponding region-of-interest, however not all regions-of-interest will have probe interfaces associated with them. (E.g. they may be merged into a comparison interface.)

The first behavior of a probe interface is to provide a range of visualizations that can be used to examine its associated region-of-interest. It can organize these visualizations into a paging system, such that the user can browse through a number of "pages" containing different visualizations until they find one that best suits their particular analytical goals. These can be cycled through by the use of a button or drop down box. The visualizations get their data by pulling the pre-calculated values (totals, averages, etc) directly from the associated region-of-interest. Of course, the particular choice of which types of visualizations to provide is a domain-specific issue and varies widely from application to application.

In addition to the pages containing visualizations, it is useful to include pages that can provide direct information about the region-of-interest selection itself, as well as methods to control and modify them. For example in the Urban Growth DSS (Chapter 3.4), a region control panel is provided that shows the binary mask for a region and allows the user to manipulate the mask directly using image processing operations, i.e. dilation and erosion, as well as to logically combine or subtract regions from each other to form more complex regions.

One should ensure that at all times there is a method of accomplishing a number of basic commands within a probe interface. The easiest way to do this is to implement a toolbar, or row of buttons, along one edge of the probe interface. The major commands that should be accessible at any time are:

- "Close", which removes the probe interface and signals the components (e.g. connector and region-of-interest) it is associated with that it has been deleted and if they have no other interfaces connected they should delete themselves as well.
- "To Glyph", which removes the entire probe interface leaving only a small "From Glyph" button and the currently displayed visualization. The purpose of this command is to produce a visualization that can be placed directly on the map as a sort of annotation. For further usefulness probe interfaces that are sent to glyph mode should be locked in position and size relative to the base map, such that they move and resize with the map as an attached feature.
- "Change linkage", which switches between different methods for visually linking the probe interface with its associated region-of-interest. The different methods of linking regions are described in the next section.
- "Apply to all", which causes all other probe interfaces to immediately switch to display the same visualization, with the same settings, as the one in which "Apply to all" was clicked. This is useful for doing indirect comparisons between regions. Or if when examining one area, the user finds a particular visualization

particularly helpful and wishes quickly to switch all interfaces to that visualization.

- “Compare with”, which is used to spawn comparison panels. When the user selects this option, the interface changes into a probe interface selection mode. In the Urban Growth DSS this is accomplished by darkening the entire interface except for any existing probe interfaces, which have an “Add to Comparison” icon displayed on top of them. Clicking on any of these will change the icon to “Selected” and dim the interface. Once the user has selected the probe interfaces they desire to compare, they click the original interface, marked with a “Done” icon, one more time to exit this selection mode. At this point a comparison panel is spawned with all the selected probe interfaces’ regions-of-interest. Note that it is helpful to add a shortcut command here that can be used to automatically select all available probe interfaces. In the Urban Growth DSS this was implemented as simply whenever the user entered selection mode and then selected the “Done” icon without first selecting any probe interfaces.

3.5.6: Connectors

Visually linking the probe (and comparison) interfaces with their associated regions-of-interest is an important component of a probe-based interface. Without some form of visual linkage, it is virtually impossible for the user to remember which interface corresponds to which region-of-interest.

The default method of linking a probe interface to its region-of-interest is with a connecting line. To ensure the line connects in the most efficient and aesthetically pleasing manner, each component keeps a list of possible connection points. Regions-of-interest maintain a list of points around their outlines. This is either a collection of pixel locations from the raster based outline, or the vertices from a polygonal outline, depending on how the region-of-interest was formed. All panel based interfaces maintain a list of the screen coordinates for their connecting points: one at each corner, and one halfway along each edge. Each time a frame is rendered, the connector object, which is linked to both the region-of-interest and the interface, checks to see if either the map (and thus the region-of-interest) or the interface has moved. If it has moved, it predetermines the closest pair of possible connecting points between the two objects. The connector object draws a line connecting these points each frame. Thus interfaces are graphically directly linked.

The other method of visually linking interface panels to their associated regions-of-interest is to use color coding. This is the default linkage mode for comparison panels, as having too many lines emanating outwards from a panel can be very confusing, and data items contained within are already color coded to the individual regions. For probe interfaces, or any interface that is only associated with a single region-of-interest, a solid colored (to match the region-of-interest) bar is drawn between the interface's toolbar and its contents. For comparison interfaces, or any interface that is associated with multiple regions-of-interest, the solid bar is replaced by a bar cut up into equally sized segments, each colored to match a particular region. Each of these different linkage methods can be seen in Figure 10.

3.5.7: Comparison Interfaces

Comparison interfaces are very similar in form and function to a probe interface, with a few significant differences. The most obvious, of course, is that instead of being associated with a single region-of-interest and connectors to such, they are associated with multiple regions-of-interest and connectors. It must also provide a button that can be used to access the MAUP overview and adjustment interfaces. Upon creation, a comparison panel is tasked with examining the values contained within all the regions-of-interest, and calculating general statistics, namely distributions and standard deviations of the regions in each dimension. It needs to store the mean value and standard deviation for each dimension, for use in the MAUP overview and adjustment interfaces. It must also check each dimension to see if any regions are more than a particular number of standard deviations from the mean value in that dimension. This threshold value can be adjusted either globally or by dimension to modify the sensitivity of outlier detection. If an outlying region has been detected, the comparison panel alerts the user to this by highlighting the button that is used to enter the MAUP interfaces.

The comparison panel also contains, displays, and conducts the functions of the MAUP overview and adjustment interfaces. The design and behavior of these interfaces are described in detail in Chapters 3.4.7 through 3.4.11.

CHAPTER 4: EVALUATION

In this chapter, the advantages of using a probe-based interface are explored through illustrative examples and informal evaluations with target users and GIS professionals. The general probe-based technique is first discussed in relation to other techniques. Then the focus shifts specifically to cover the full-fledged implementation in the Urban Growth Decision Support System.

4.1: Benefits of Applying Probes

Informal evaluations of the three original probe-enhanced applications were conducted to solicit feedback regarding the usefulness over their previous single-perspective interfaces. Each of the three systems were presented first in their original form, and then with probes to two audiences, each with a mixture of both faculty members and graduate students. The first group consisted of thirteen participants from the Center for Applied Geographic Information Science at UNC Charlotte, both GIS professionals and educators, as well as advanced GIS students. The second group consisted of eight participants from the College of Architecture at UNC Charlotte, who all had experience with computer aided modeling. A few of these participants had actual previous experience in designing and working with the original UrbanVis census data exploration tool described in Chapter 3.2.

4.1.1: View Independence

Using probes removes the burden of having to change zoom-levels to inspect local data. By preserving the global view, it is ensured that the user can always perceive the overall dataset. Visually depicting the selected regions-of-interest directly on the global view ensures that the user always knows the context of the local region. By using multiple probes, distant local regions can now be simultaneously inspected and directly compared onscreen, alongside the global view. This preserves maximum spatial awareness and decreases the navigation required to switch between zoom-levels.

Many participants identified the issue of loss of spatial awareness when constrained to a single perspective as one they have encountered in their work. Even though some of their existing applications have the ability to present multiple camera views (e.g. 3D modeling suites), one participant noted that “when trying to navigate in true 3D space, you often lose track of where something is in [the overall] space,” while another elaborated that “where there’s a lot of data….its important to be able to drill down” but that “sometimes you dive into detail on something and its not easy to navigate your way back out again.” They saw the probe-based interface as being a solution to this problem, in that “all [the probes] are organized by the overall metaphor of the map, so it really does help a lot [to know that] this window relates, in this way, to the other windows” and that this linkage “allows you to navigate more fluidly between different parts.”

4.1.2: Multi-Focus Inspection

Probes allow the user to dynamically specify regions-of-interest and select from a wide variety of “drop-in” information visualizations. An assortment of methods are appropriate for selecting regions-of-interest: circular regions can be generated from a focal point and an extent, irregular regions can be selected with various free form tools, etc. Extending the target of a coordinated visualization beyond what is merely onscreen at the time to these more flexible and precise regions improves both relevancy and accuracy. By allowing different visualizations to be tied to each probe, one can perform a wider range of inspections at one time than if limited to a traditional coordinated visualization interface.

By using multiple probes, the user can select multiple local regions and view their values side-by-side, or directly together in a comparison pane, always along with a global reference for overall context. This removes cognitive memory requirements, avoids change blindness, and speeds up comparison (in regards to navigation requirements).

All participants appreciated the view independence and multi-focus aspects allowing them to access lower-level information about multiple local regions, while preserving the higher-level overview. The ability to visualize a multiplicity of scales simultaneously was also well received, with one participant specifically commenting that “having multiple scales is incredibly interesting, because at different scales you may be starting to visualize different processes.”

The comparison abilities were also identified as attractive by the participants, one noted that having that capability in her application would make it “a lot easier to compare all my variables [while looking at it] quickly.” Being able to investigate multiple regions without “having to go through the steps of selecting them and then opening up attribute tables” was described as “fast and intuitive.”

4.1.3: Location-specific Manipulation

The creation of probes at a multitude of different shapes and sizes not only enhances inspection capabilities, but interaction capabilities as well. The user is no longer limited to applying adjustments and controls specific to predetermined scales. This extends the previously global controls into one that can be used to control specific local regions, empowering the user to more precisely interact with the data.

Several participants expressed enthusiasm about the probe-based technique's potential to enhance their own existing projects with location-specific interaction. Their projects included a landslide hazard analysis application, an interactive disease outbreak map, and a cellular urban growth simulation. (This particular urban growth simulation was not the same one as later utilized in the Urban Growth Decision Support System) The cellular urban growth simulation was the center of much discussion, as it had many parallels to the agent-based social simulation discussed in Chapter 3.3. In particular they saw the probes as an attractive method of being able to "change parts of the simulation…and affect the simulation locally," and "a really exciting opportunity to take to the decision makers."

4.2: Urban Growth Decision Support System

One exciting aspect of the Urban Growth Decision Support System project was the collaborators' focus on outreach and getting the tool into actual usage. The Urban Growth DSS has been formally and informally demonstrated to many target users and GIS professionals over the last two years, and it is currently being used by collaborators to explore and analyze the simulation results. Furthermore, they use the tool as a presentation aid to communicate the Urban Growth issues to decision makers and community groups. The feedback from target users and collaborators has been beneficial in understanding the strengths and weaknesses of the probe-based approach, as well as provided new directions for future development and refinement of the techniques.

4.2.1: Demonstrations to Target Users

The Urban Growth Decision Support System was presented, running on the RENCI built multi-touch table, to members of the target audience (urban planners, politicians, etc) at the 2009 Institute for Emerging Issues Forum, which was focused on growth and infrastructure needs. For two days, individuals across a wide range of familiarity with computers were encouraged to use the system to investigate their “home” districts. Most found the probe-based interface straightforward and had no problems understanding the multiple tools available to select regions-of-interest at various granularities. Users were primarily interested in selecting regions-of-interest corresponding to either a municipal boundary or the more general “around” a particular feature (a request the free-form selection tools are invaluable for). Thanks to the multi-touch capabilities of the table interface, selecting large numbers of bounded regions, which previously (with a mouse) required either repeated clicking or a tiresome back and forth “scribbling” motion, was now possible for people to do with literally a wave of the hand. The novelty of the touch-based interaction was effective at initially drawing people’s interest into the application, but on a more meaningful level, it allowed people to immediately step up and start using the application, without the social barrier of asking to have a turn using the mouse and/or keyboard. The collaborative nature and large shared display made accommodating new users easy, as more experienced users could take a moment to guide them along or demonstrate a feature without having to abandon their own analyses already in progress.

Reset buttons were included to quickly and conveniently clear out the no longer wanted regions-of-interest and probe interfaces that were left over after a user had walked away from the table, but if the system was to be left out in public unattended, say as part of a museum exhibit, some sort of automatic clean up routine would be most beneficial. This routine would seek to clear deserted personal workspaces while maintaining interesting analyses distributed in the shared workspace (as defined by Scott [Scott 2004])

One of the most obvious conclusions that this demonstration provided was that this interface had great potential as a presentation tool. But not just for a rigid speaker-audience presentation model. Instead, it lends itself well to a more interactive, intimate collaboration between the person attempting to communicate ideas about land change and development patterns and those who need to understand these ideas. The audience is no longer a static observer; they can perform the same analyses, on their own region, in time with the presenter.

While there were some touch-table hardware issues related to IR illumination and direct sunlight, and software issues regarding finger tracking in the experimental touch-driver provided from RENCI, it was clear from the ease at which the general public was able to interact with the simulation, that the probe-based interface was indeed very intuitive in its design. This has implications not only for the broadness of the audience that can be reached with such a tool, but the costs involved with reaching those audiences as well. Traditional GIS tools that would be used for these types of analyses and presentations are expensive and have steep learning curves, creating an economic bottleneck to the communication process between researcher and information consumer. This issue is discussed in detail with GIS professionals in the next section.

Figure 34: The Urban Growth Decision Support system being demonstrated, and interacted with, by members of the target audience (urban planners, politicians, community leaders, etc) at the 2009 Institute for Emerging Issues Forum. It was made available on a multi-touch tabletop display over two days in the heart of the conference area.

4.2.2: Discussions with Expert Users

Throughout the development of the Urban Growth Decision Support System, I worked closely with collaborators in both The Center for Applied Geographic Information Science (CAGIS), who were producing the underlying data and programming the simulation, as well as those in The Urban Institute, who focused on outreach to policy and decision makers, community groups and leaders, and the general public. To better understand each groups perspectives, I talked with them regarding their goals, how they evaluate their success, how they have used the probe-based DSS, and the benefits and drawbacks of the probe-based interface over the more traditional GIS software packages they have used in the past.

I spoke with the outreach director, John Chesser, in The Urban Institute to get a better understanding of their group's goals in regards to this project. The project's original funding came from The Open Space Protection Collaborative, which commissioned The

Urban Institute and CAGIS to study and forecast the future impacts of urban growth upon the undeveloped lands in the region around Charlotte, North Carolina. Soon after, the Renaissance Computing Institute (RENCI) became involved, expanding the goals of the projects into utilizing its urban growth modeling results to provide policy and decision makers, community groups and leaders with a utility that would allow "them to understand and analyze what is happening in their regions." Chesser lists a number of questions they seek to answer with this project:

> "What important open space is being gobbled up? How can we get that message out to the general public? It's sort of grown beyond that and is now looking at that issue, and the overall pattern of development. How sprawling is it? How sustainable is it?"

Clearly being able to communicate both the historical and forecasted growth patterns is one of the Urban Institutes major goals. Whereas before, the best tool they had for these purposes was printed maps or an animation on a website, the probe-based interface has "introduced a whole new way to understand what's happened, and what could happen if trends don't change." Chesser elaborates,

> "The model itself gives you all kinds of interesting information and statistics that can be really powerful. But what's really clear, is that the visualization and the interactivity of this interface gives you a whole different perspective than just those pieces of data."
>
> "They could show these static maps in the original version of the model, and it was powerful stuff when you talked about how much we are consuming per person. But the key here, is that it becomes much more real for people to see how much visually is being consumed. To be able to step through it and interact with it, means that people understand it in an entirely different way than just hearing and seeing the static maps and the data. It's visceral to people when they see rather than hear."

Chesser uses the interface primarily for demonstrations to planners and land conservation groups, where it serves as a visual aid for communicating the results of CAGIS's research. The probe-based local inspection capabilities allow him to cater demonstrations to appeal directly to actual practitioners and policy makers, showing them their own particular jurisdictions. By allowing multi-focused inspection, the tool allows him to show them the surrounding region at different extents beyond their rigidly defined jurisdictions. As he puts its, "Everyone has only been looking within their municipal and county boundaries, and relatively few people were looking at it from a higher altitude." With the ability to look along and across boundaries, they are able to understand how borderlines develop, and begin conversations with peoples who don't normally look outside their own jurisdictions. He also identifies the strength of the probe-based comparison abilities to provide additional assistance in the form of conveying the inter-relationships within the overall trends: "You really begin to get people to understand how interconnected and interdependent [development] is, and how artificial their boundary lines are."

In preparing for his outreach efforts, Chesser uses the tool to find interesting features around the audience's particular region, as well as how their region's growth patterns compare to those elsewhere. It is with this usage that he identifies the strong benefit that the probe-based interface has in terms of facilitating exploratory analysis without the a priori knowledge of what exactly one is looking for, which is usually required in the more rigid traditional GIS software packages. He elaborates,

> "When you are working with GIS data, whether it's regionally or in a jurisdiction, you have to start with what you think is interesting. Basically, you take the data, and it may need to be manipulated [before

> you can do anything with it] and it's much more reliant on somebody having an idea of where they want to go. If you don't have an idea of where you want to go, you don't get there. The difference with [the probe interface] is that you see connections and understand dynamics in a different way, and you don't have to be able to go into a product like ESRI's and do those many steps to make it all happen. In the end, you may go back [and use those tools] but people don't have to have that expertise to see and understand the dynamics of what's happening over time. With GIS products you have to make the calculations you are interested in and get a picture of just that single point in time. This [probe-based interface] lets you not have to have that expertise, and not have that deep of an understanding of the data to lead to [an effective visualization for presentations]."

This is true, in that as an exploratory tool, there is far less of a requirement that one know what one is looking for before actually examining the data. Patterns and relationships are instead discovered as the user interacts with the system. For further, more rigorous analysis, one will still likely need to resort to the traditional GIS systems for their intense number-crunching abilities, generality, and reporting capabilities. But by including features that let one export their selections, regions, etc to these traditional tools, this transference can be facilitated.

To further investigate how the Urban Growth Decision Support System is used and perceived by GIS professionals, I spoke with Douglas Shoemaker, CAGIS's Associate Director of Research and Outreach, and a principal investigator on the Urban Growth Model project. Since 2007 he has been working to "develop core content and analyses to understand urban growth patterns in the Charlotte metro region, and ultimately across the entire state of North Carolina." He has developed the Urban Growth Model by turning "raw data from remote sensing into an understanding of historical growth patterns." Then, based on those trends and predicted population trends, he forecasts where development is going and how much is going there.

Shoemaker evaluates his model's predictions through measures of internal accuracy. This is primarily done by withholding data from a particular timestep, allowing the model to simulate that missing timestep, and comparing the predicted and actual values. Through backcasting they can determine what factors and policies have lead to the current land usage, and apply these findings to strengthen their forecasts. These analytical behaviors have traditionally been conducted within the usual popular GIS software packages, however Shoemaker sees great potential in using the probe-based interface for some of these analyses:

> "From an analytical standpoint [the probe-based interface] makes it a lot easier to do analysis without expensive specialized tools. This tool also does things that my fancy GIS can't do… The MAUP tool is an incredible innovation. And being able to freehand select pixels and have them added up and compared with other areas is something that a GIS just does not do well or easily."
>
> "I hope to actually use it as an analysis tool… To be able to go through and use some of these good tools and really support a decision about where things should go, or what sort of challenges, in terms of population or land change, that different municipalities will face. There's a lot of depth of analysis here that I'm looking forward to exploring."

It is not only that the probe-based interface provides these analytical functions, but that it provides them in an intuitive manner. Shoemaker identifies one of the biggest differences between the probe-based interface, which he describes as "light-weight", and traditional software packages, as being one of accessibility to the general public:

> "We are generating a lot of data. The data is spatial and visual, but it is underwritten by tabular data that is primed for analysis. This is not very accessible to those that are not specific GIS technicians. One of the ultimate goals of this project is to put this analysis into the hands of all different types of people that can use it. This interface goes a long way,

> and has been very successful at allowing people to access the data through intuitive approaches."
>
> "These days we have too much data. The difficulty is being able to put it into some form that allows you to support your decisions, and this does just that… in an intuitive fashion that doesn't require any expensive training."

The costs of training are indeed a concern when choosing an interface design. For public consumption ease of learning is essential. Traditional GIS software packages have steep learning curves and are too expensive for the general public. Shoemaker continues,

> "Right now, to do any sort of analyses, it requires very, very expensive software and a lot of expensive training. I would put a third year GIS student on these types of things. Three semesters of training is equivalent to what a person could walk in and do fairly quickly with this tool on first sight. I anticipate that even with the more advanced tools that this [interface] offers, people will be able to master it in minutes, rather than semesters or months."
>
> "The MAUP adjustments are probably going to require some sort of tutorial, but it seems like it would have a very short learning curve."

Aside from the modeling goals of trying to figure out historical growth trends and forecasting future development, he also stresses the need to "inform decision makers about these projections" and to identify and evaluate the impacts on municipalities and the environment. Shoemaker also uses the probe-based interface as a visual aid when conducting outreach events, where he praises its "wow factor" and "ability to get the message across without specialized software and specialized skills." He summarizes, "As a demonstration tool, I think it's superb."

In discussing collaborative usage, he remarked that the probe-based interaction "promotes collaboration," when the hardware to support such is available:

> “Assuming there is a hardware interface for it that allows multiple users, a bunch of people can look at it at the same time and run their own little personal analyses and then be able to compare things together. So it could be a huge collaborative tool.”

Shoemaker identified that one of the major drawbacks of the current interface was that there was currently no support for the specification of more rigid, database like queries, specifically,

> “You lose some of the tabular background, which is swapped for the intuitive visual interface. If we wanted to move away from the lay public and install this tool as a decision support system for someone who is doing hard science, its tracking, its output, and its query functions [would need improvement]. But it’s not really designed for that.”

Like John Chesser, he agreed that for these types of analyses, one needs to resort to the traditional GIS software packages with their computational abilities, and that some method of transferring regions, selections, and analyses from one interface to the other would beneficial.

An interesting feature that he felt the application lacked was the ability to log or track what the user is doing. He described how useful it would be to have a record of where you’ve been and what you’ve done and to be able to leverage this information to generate output, a “report you could take and show [to policy makers.]” Further, he envisions a way that one could run their own analyses, and then be able to duplicate the process exactly in a demonstration for someone else. This type of behavior would be especially helpful when selections are made with the free form tools, when the regions are often quite uniquely defined. He suggested the ability to provide some form of output that gives a textual form of your answer/analysis instead of a purely visual one. Further discussion of the planned improvements to satisfy these needs can be found in Section 5.4.

CHAPTER 5: DISCUSSION

This chapter contains discussions of the potentials of probe-based techniques in different applications domains and methodologies. Specifically it discusses the applicability of probes outside of strictly geospatial visualizations, caveats and limitations to the use of probes in various applications, usage in collaborative systems, and future improvements that can be made.

5.1: Applicability to Non-Geospatial Visualizations

Although the specific applications described here demonstrate the effectiveness of applying probe interfaces to geospatial visualizations and simulations, it is probable that this concept can be applied to more abstract data spaces as well. The most obvious visualizations that can benefit from this are tools that present a spatial layout in which the locations of data items are of importance, such as in an organizational chart or graph layout. However, it is also conceivable that this type of interface can be extended to any information visualization that presents an overview that can zoomed into further. In theory, this probe-based interface should be very generalizable, and a great deal of future research could be done to explore the possibility of applying this interface to other types of visualizations.

While this applicability holds true for data that can be held to some spatial coherency, it is unclear whether it would be at all effective in data which is inherently non-spatially stable. As probes by definition require regions-of-interest, what becomes of them when regions can only be perceived temporarily? Jeong et al. [Jeong 2009] presents iPCA, an interface in which the user can change the contributions of various dimensions to a principal components analysis (PCA) plot of multidimensional data items. As the user adjusts the eigenvectors, the data points move around the 2D space in various manners, which assists the user in determining the contributions of those dimensions. If one were to probe the data by selecting a region-of-interest in the PCA plot, after adjusting the dimensional sliders (the primary interaction in the interface), the region-of-interest would no longer be coherent. In fact, the points might be just as far apart as any others. At this point does the region-of-interest become meaningless? Or does it simply become a form of brushing and saving a selection? The adjacency of the points was relevant for a particular set of dimensional parameters, but once those change, the adjacency is likely no longer relevant, or accurate. These are unanswered questions, and the applicability of probes to this type of situation has yet to be explored.

5.2: Caveats

There are potential scalability issues that may arise when probes are implemented within existing applications which require significant processing to render their information visualizations. What may have been sufficiently fast to draw in a single inset view, may be too slow for deployment across multiple probes. This is especially true if the visualization requires extra calculation to aggregate information to condense itself to a smaller screen size. The UrbanVis application detailed in Chapter 3.2, for example, ran much slower under the strain of having to calculate multiple levels-of-detail and aggregations for each region-of-interest, something it was not originally designed to do efficiently. While this is primarily an issue that is encountered when one is extending existing applications and reusing existing visualizations, one must still consider issues of scalability when choosing visualization algorithms/techniques for use in new probe-based interfaces to avoid such issues.

Visual scalability can also quickly become an issue as the number of probes created increases, both in terms of screen real estate and overall cognitive load to the viewer. The two basic methods that can be used to help alleviate these issues are to make the probe interfaces collapsible (see Figure 7) and to make the probe interfaces resizable (see Figure 4 & Figure 5). Collapsing a probe interface reduces the screen space needed to display a probe interface, maintains visibility of the region-of-interest, and reduces the overall visual complexity of the application. Resizing probe interfaces also achieves these benefits, and has the additional advantage of allowing the user to customize the complexity of the associated visualization. As shown in Figure 5, this can help with a shortage of screen real estate, as it permits the user to fit more, smaller visualizations onscreen at once. However, consideration must of course be made in regards to how the visualizations are resized downward, into a more glyph-like form, in order to ensure that they are able to be correctly interpreted and meaningfully compared.

Another issue arises from overlapping regions-of-interest denoted by probes. This is particularly problematic and ambiguous if direct data manipulation is allowed on each probe, as is the case in the agent-based simulation. This overlap creates a one-to-many mapping issue, since there can be multiple controls affecting one area. There are some obvious solutions to alleviate this problem, such as prompting the user when a conflict arises. However, this problem is very application and domain-specific, and effective solutions may be found on a case-by-case basis.

Another issue that may arise with increasing scale, is that of the ratio between regions-of-interest and the whole simulation/world. In most of the applications described in this work, the regions-of-interest are generally a few to a dozen percent of the whole. However, one could envision a scenario with a national map, and the user wishing to select small, individual parcels in two distant states. In this scenario, there would be navigational requirements on the map to get to the zoomed-in scale needed for accurate selections, and return to a national view showing both regions. Inset, zoomed-in views could of course be presented within the probe interfaces to remove the need to return the main map to a zoomed in view of the region. One could also provide multiple maps, such as done in the agent-based simulation, which supports comparisons of probes inserted into different maps. A more complex solution might be possible, in which a multiple fish-eye type distortion is done to the overall map, showing more detail around selected regions-of-interest, and less in between. This assumes the user can cope with the distortion of the map.

5.3: Collaborative Usage

As best shown in Chapter 3.3, a probe-based interface can be implemented to remove fixed single perspective interfaces, and instead allow the user to dynamically insert interface elements anywhere they are needed. There is an immediate benefit of this style of interaction for co-located collaboration on a single common display, as there are no theoretical limits to the number of probes or map instances. Multiple users can interact with the same visualization at the same time without interfering with each other’s views. An attractive interface device for deploying this kind of probe-based visualization is a multi-touch table, which has been demonstrated to be an effective medium for a multi-user environment. [Shen 2006] [Wigdor 2006] As the popularity of these types of touch surfaces increases, the need for these probe-based interaction techniques and their future extensions will grow, hopefully finding widespread use and application on collaborative systems with these hardware methodologies.

Brewer et al [Brewer 2000] developed a prototype collaborative geovisualization environment and used it to perform interviews/informal evaluations with domain experts to ascertain what is expected/required of geospatial visualizations when they are to be used collaboratively. Some of their findings are particularly relevant to this work: the role of maps in a collaborative environment, drawing attention, and joint interface controls. Most of their participants mentioned that in a collaborative environment, the role of the maps were to provide the context in which discussion would take place. The importance of the map for conveying spatial characteristics and locations is vital when attempting to communicate a finding to a collaborator, thus the probe-based interface succeeds in this aspect; results (in the form of visualizations) always have a direct visual link back to the map that provides spatial context. The need to draw attention to areas on the map (via circling or pointing) was also raised by most of their participants. Again, the way in which probes link results back to their contextual locations explicitly draws attention from the results (the discussion topic) to the source on the map. Finally, while many of the participants in Chapter 4.1 saw the need for joint interface controls, they also raised the issue of potential conflicts. Solutions proposed included turn-based control and separate control panels for each user. A solution such as shown in Chapter 3.3, in which a single, global interface is replaced by on-demand controls, tied to local regions, has the potential to alleviate potential conflicts by allowing each user to manipulate variables for only their own specific regions-of-interest, (perhaps even in only their own map).

The main challenge when handling multiple simultaneous users in a collaborative application on a multi-touch table, such as with the Urban Growth Decision Support System, is determining the ownership of touch events. That is, when a finger touches the surface, the system needs to be able to distinguish which user has initiated the event.

Early systems, such as DiamondTouch [Dietz 2001] achieve this by having finger touches complete capacitive electrical circuits between the table's surface, through the user, and into the user's chair. This places restrictions on the surface's material construction and translucency, thus limiting the projection methods that can be used for the display. (Generally only projection from above is suitable here). By making the chairs an integral component of the tracking system, it limits the number, arrangement, movement, and positions of users. It is a reasonable argument that placing these limitations on users is unacceptable for most use real-life usage where people will unpredictably change stances and move about. Furthermore, the above projection systems suffer from the serious issue of occlusion, which worsens with each additional user.

The more promising option is to use computer vision methods to track users interacting with the table. By placing a camera above the table, one can segment the arms and hands of individual users and monitor their locations. When a touch event occurs, its location can be compared to the known positions of the users' hands, and thus the user who initiated the touch event can be determined. Vision based solutions generally have significant computational cost, however the tracking of arm and hand locations can be done at relatively low resolution due to the high ratio of size between the objects being tracked and the tracking area, and there is a much lower requirement for frame rates since one is not concerned with tracking each fine movement, only general locations. Some have found successful methods which allow collocating the user tracking camera(s) behind the screen with the touch tracking camera(s) through the use of DNP HoloScreen, as in the TouchLight [Wilson 2004] system, which allows projection onto transparent surfaces, or with a active surface material, such as used by Kunz [Kunz 2002] for CAVE walls, that can alternate its opacity fast enough to allow both projection of the display image and visual tracking of the users on the other side.

Another set of issues that arises when designing collaborative multi-user systems such as the Urban Growth Decision Support System is workspace territoriality. Decisions to be made here include what distinctions, if any, are to be made between “personal space” and “shared/group space.” Scott [Scott 2004] observed tabletop collaboration in both casual and formal settings and found participants would intuitively divide their workspaces into three general types of interaction areas: personal, group, and storage. These areas would expand and contract dynamically as the activities continued, but there were several constant themes that must be considered when designing a collaborative workspace. Areas directly in front of participants were always used as personal workspaces in which participants would individually execute their own tasks. Areas within in the center of the tabletop were used as a “shared” or “group” workspace. It was here that participants collaborated on the main tasks that contributed toward the overall group’s goals. It was also here that participants would help each other with their individual tasks. The peripheral edges of the table tended to be used as a sort of “storage area" in which items were loosely organized and moved further away from the active workspaces as they became less often used.

The Urban Growth Decision Support System was designed to allow user to come and go randomly, and was designed to not require protective personal space. As such, there was no enforcement of any strict limits on who can do what where. Returning to Scott's [Scott 2004] observations about intuitive division of collaborative space, one can recognize that the map in the Urban Growth DSS is obviously the center point and the "group workspace" area. It is here that users share their regions-of-interest, and where finished glyph-like probe interfaces are often overlaid atop their associated regions. The "personal spaces" for each user are the initially empty spaces around the map directly within their arms reach. It is here that users privately analyze the contents of their regions-of-interest within their associated probe interfaces, before moving them into the shared workspace for distribution. Also, as Scott predicts, the periphery of the table, areas along the edges and out of users' easy reach becomes a "storage space". This space tends to fill up over time with "monitor" probe interfaces that have been set to keep an eye on particular regions for long periods of time. Those "monitor" probe interfaces that concern a single user tend move from their personal workspace into the periphery to make way for new personal analysis tasks, and those that are of interest to the entire group tend to be moved to the periphery from the center as the data they monitor becomes stale or less relevant to newer analytical results being shared in the central "group workspace."

Relevant design implications identified in Scott's research correspond with design choices made in the Urban Growth DSS. Providing appropriate table space is a concern.

While the physical size of the table top is fixed and beyond the control of the software, included are simple methods of resizing all interface elements to allow more content in less space. There are of course limits on how much one can shrink and stack interface elements to fit more onscreen, but fortunately there is also an inherent limitation on how many users can physically fit comfortably around a table.

It is noted that one should provide functionality appropriate to each type of workspace. This is accomplish in probe-based interfaces by: tying all probe interfaces, in which analyses are conducted, back to their associated regions-of-interest, providing methods of reducing complexity and size for interfaces that one wishes to "store" on the periphery, and the ability to reduce probe interfaces to a glyph like state for overlay on a central map. Finally, the freedom of the probe-based interface allows for the casual grouping of items in the workspace to both allow users to customize their personal workspaces and freely share their findings with the group.

For the Urban Growth Decision Support System, formally tracking of specific users is not necessary; of more concern is the location of users as they casually interact with the table. The general location of users around the table is needed to determine how interface elements, probe interfaces, and visualizations are oriented with respect to the table. Given this straightforward requirement, and taking into account that many multi-touch tables are based on infrared (IR) illumination techniques, placing a camera with an IR-pass filter above the work surface is a very attractive solution. With an IR-pass filter installed, the camera, mountable on a telescoping arm or hanging from the ceiling will only see a brightly illuminated rectangle corresponding to the table's surface. It is then simply a case of automatically calibrating a homography from this rectangle to screen coordinates and then watching, with background subtraction, for shadows entering the rectangle. When a touch event spawns or modifies an interface element, the shadow nearest to it can be followed back to its intersection with the edge of the rectangle/screen and this point on the perimeter can then be used to determine the necessary orientation for that interface element. Thus, when a user interacts with an interface element, it can automatically rotate to face them properly. This process could be further expanded to allow more complicated actions that would be needed in a collaborative environment, such as the sharing of information with other users: One user could drag a probe interface “into the hands” of another user, at which point it would then orient itself to face its new observer.

By making assumptions about the maximum number of simultaneous users (realistically under 6 for most semi-mobile tables), this solution can also be used to restrict selection and refinement tools/modes to the touches of individual users. For example, in a time-critical situation, User A could be making rough selections of areas they know need particular resources as they are request them, while User B examines and refines these selections to be realistic in terms of resources available and continuously updated return-on-investment figures, simultaneously User C is following along, making final approval or denial decisions as to the actions to take with each of the refined regions. In such a manner, a collaborative decision making process can be continuously undertaken from start to finish in the same analytical environment, in which all users can see each others work in context with their own.

An interesting problem comes in managing multiple “private” workspaces on extremely large format touch interfaces, for example a large touch wall. In this case, individual workspaces could be spawned in such a manner as a unique gesture, e.g. placing the entire hand flush against the display for a few seconds. This could partition off a particular space of the wall for personal use, spawning a new base map and corresponding control panel that would be used to control it. In ways such as this, multiple users could dynamically enter and exit a large format collaborative space, each receiving their own controls and data spaces, while still being able to share findings and probe each others data. For more on this subject, see Peltonen's [Peltonen 2008] observations of the behaviors of over a thousand members of the public who interacted with a large format touch-display made available in a city-center.

In this type of setup, where users wish to interact between personal spaces on a display that is far wider than their physical reaching abilities, it might be helpful to add a "portal" feature. This is a known technique that has been implemented before, most notably in the Pad++ interface [Bederson 1996]. This would allow a user to designate a portion of their workspace as a window into another user's workspace. This portal would be used to see a distant user's workspace within one's own workspace. It could be used for more than just monitoring other workspaces, as probe based techniques could establish inter-workspace communication through these portals. One analyst could open a portal to another analyst, and then select a region-of-interest on a map the other has been working on. This would spawn a probe interface locally that reflects the contents of the remote map. Furthermore, a third analyst could open additional portals, pulling analytical results from multiple analysts' remote interfaces, for comprehensive comparison within a single local comparison interface. In this manner, the remote data spaces of multiple analysts could be accessed and explored simultaneously in a truly collaborative fashion.

5.4: Future Work

As identified in Chapter 2.3, one possible solution that could be explored to address the "scale problem" component of the Modifiable Areal Unit Problem, would be the introduction of tools that could be used to semi-automatically split and combine regions-of-interest. This would, of course, require a more thorough understanding of how much modification of the user's analysis is tolerable. For example, the current model lets the user ask and answer questions such as "How are areas A, B, and C like area D?", whereas a split operation might turn this into "How are areas A, B, and C like these similarly sized subsets of area D?"

It is also worth examining the processes that the GIS professionals use to de-aggregate data into the types of raster based input that are suitable for probe-based interfaces that require data in such a format (e.g. the Urban Growth DSS). This is an area in which both the scale component of the MAUP and the ecological fallacy are of supreme concern, as all analyses done within the interface rely on the accuracy of the underlying maps. This has been studied in spatial analysis literature, but there may be specific concerns or loopholes related to the particular usage of the derived rasters in probe-based interfaces. This is likely more a research question for those in the GIS community, although tools should be developed to facilitate these conversions, such that they can be done easily and with regard to the potential error they may introduce into datasets. These tools could prove to be invaluable if distributed along with probe-based applications.

As noted in Chapter 3.4.11, the “trade area” adjustment algorithm, which is currently still considered a “proof-of-concept” implementation, has much room for improvement. The problem is neither simple, nor straightforward, and many different approaches and solutions are possible. It is assumed that collaboration with those in image processing and geography backgrounds is needed to help improve both the speed and effectiveness of the current technique. It is likely that an efficient, fast, and accurate algorithm that can handle all the possible unique cases and domain specific demands would be an extremely deep and cross-discipline research topic.

One feature that GIS professionals specifically requested be added to the Urban Growth DSS was increased support to additional data input formats. While the Urban Growth DSS currently only accepts dimensional data in raster form (vector data is used for selection and reference), the ability to provide input in non aggregated or interpolated form was seen as important. Indeed, this is not only a good idea from a convenience standpoint, but also for ensuring accuracy is maintained. One example is point data. Currently point data would need to be aggregated or interpolated to the pixel level, spreading precise measurements out over wider areas and allowing for ecological fallacy. The ability to accept point data presents no substantial technical challenge and is slated to be included in the next major revision of the application.

As discussed in Chapter 4.2.2, another feature that was often requested by the users of the Urban Growth DSS was the ability to write out reports that would assist in justifying decisions and aid in presentations. This could include an interaction log, such that when a desirable analysis result is achieved, the analysis can be output, along with a path showing how the analyst arrived at the result. This could then be utilized in two important scenarios. The first would be when a user wishes to communicate their findings to another. Here the log can be used to recreate the analysis and ensure the other analyst can replicate their findings. This is an important concern when free form selection tools are used, and complicated region adjustments are repeatedly made, as the resulting regions are very difficult to describe and replicate. The second scenario would be in support of a decision to be made. Here the analyst, a decision maker would use the log or report to illustrate their research, as well as their thought-processes to an audience or their superiors. This is helpful because audiences are often interested not only in the results, but as to how the results were obtained, and in the justifications of its validity and how it is supported by the data.

This type of reporting/logging could be provided in a number of forms, internal playback formats, automatic movie/animation generation, or exporting analyses and data to other presentation tools. An example of this was requested recently. It involved being able to export data and findings in .kml format for importation and distribution through Google Earth. This would allow for integration of findings with its massive geospatial database and distribution worldwide within a freely available and widely supported interface. The potential benefits of such reporting features are huge. As such, some form of report generation and exportation to Google Earth will soon be added to the Urban Growth DSS.

Another planned addition to the Urban Growth DSS is the ability to steer the simulation interactively from within the interface. While the simulation is currently not able to recalculate its forecasts within the time required to maintain interactivity (generally regarded as less than 2-5 minutes), it is currently being rewritten with an aim of optimization for speed. Once this is complete, the DSS will be able to be used to explore the results of different scenarios on the fly. For example, one might wish to propose the construction of a new highway connecting two cities on the map. In the DSS they could then use a “add highway” tool to draw the proposed path of the new highway. The model would recalculate the surrounding region to show the effects of this new corridor. The user could then explore those results and compare them to the original forecasts.

This advancement prompts some interesting new questions for the probe-based technique. How does one handle probes across diverging timelines? Are different timelines treated separately, with comparisons of a single region between two timelines the same as two separate regions within the same map? How does one show linkage from a probe interface to a region-of-interest in another timeline? Perhaps these linkages will need to be moved from an internal-external, probe-map scheme to an internalized region-representation format. This seems all the more likely when one considers that in the example “new highway” scenario, the analyst would likely want to compare more than one proposed route, creating many possible proposed timelines. Representing all of these as distinct maps would leave the user struggling to make sense of a mountain of maps. It seems more attractive to instead link the probe interfaces to a representation of the differences between each timeline than the actual results of each. If regions are to remain on the map, how does it handle the fact that the region-of-interest exists in a singular instance then diverges into two different instances? The best solution to this is not clear. How does one differentiate within the visualizations which data items have come from which timeline? Introducing non-color based coding schemes (e.g. motion, stippling, etc) is a possibility, but quickly runs out of distinguishable representations. More likely, some sort of looping or user-controlled animation is required. While there are many unanswered questions, it is clear that this type of multi-temporal exploratory analysis has the potential to be one of the next major features for the probe-based technique and the possible research directions are numerous.

Scalability is currently an unaddressed issue within the Urban Growth DSS. This is primarily an issue with the way the underlying data layers are stored and accessed and not a shortcoming of the probe technique itself. Currently the system simply loads entire data layers into memory and queries them directly. When either the number or size of the data layers gets too large to keep them all within memory, some form of paging or tiling will need to be implemented. This could be done hierarchically as well, which would have a the benefit that precalculated values could be queried for a large region faster than one could query and evaluate all individual pixels in said region. Likewise, the OpenGL textures for onscreen display could be generated from the lower-resolution data, and only upon full zoom-in would the highest resolution data need to be accessed and converted to textures.

It is likely that in the future a more general probe-based GIS application will be developed. It would implement all the newest features found in the Urban Growth DSS, however it would not be domain-specific to urbanization issues. A major goal would be to increase support for dynamic loading of any number of data layers using generic industry standard formats. Scalability issues would need to be addressed, to allow the system to load local, national, and even global scale map sizes. This could be done entirely within 2D, or possibly even support for a 3D environment. The other option here is to write a plug-in or extension for one of the popular GIS packages. However, this would limit distribution to only those able to afford to buy and learn the underlying GIS package. Either way, having a more general implementation would allow researchers and analysts in wider ranges of application domains to benefit from probe-based geospatial analysis.

CHAPTER 6: CONCLUSIONS

This work has introduced probe-based interfaces that can be used to replace or supplement the single perspective, fixed interfaces of traditional geospatial visualization applications. Coordinated information visualizations, linked to user defined regions-of-interest, become directly integrated within the main view. Interaction controls are also relocated within dynamic "on-demand" interfaces, reducing clutter and allowing for local control across the entire range of scales. Together, these changes bring many benefits including view independence, multi-focus inspection, and location-specific manipulation across the entire range of scales, and increased potential for co-located collaboration on large, shared displays.

The usefulness and applicability of the probe-based interface is demonstrated through the modification of three unique geospatial applications to utilize probes. In each case, one can see the benefits gained by moving away from traditional single-viewpoint interfaces. Informal evaluations with experts in GIS and architecture confirm that with the addition of probes, the three geospatial visualization and analysis tools become more useful and intuitive.

Collaborative visual analytic environments require consideration of a number of factors. One must consider the specific requirements of the problem domain, and what the overriding and imperative questions to be answered are. In the case of geospatial analysis, the analyst is primarily concerned with Where? These questions are predominantly answered spatially in the form of regions-of-interest, one of the major building blocks of a probe-based interface. Selecting these regions-of-interest is thus a main focal point when designing an interface for a geospatial analysis application. This work describes many techniques that allow for fast, efficient, and intuitive selection of regions-of-interest. The combination of selection and adjustment tools provided in the Urban Growth DSS empower analysts with the ability to create and reshape regions interactively in an instinctive and powerful manner.

By understanding the ways in which users divide territory in table-top collaborative spaces, one can predict how the spaces provided will be used. This allows one to provide features that will promote more efficient collaborative behaviors, such as sharing and storing of results. The probe-based interface is well suited for collaborative applications, due to user's interactions and use of space mimicking the real life behaviors observed in tabletop collaboration.

The combined result of these considerations is a system that is powerful, but non-intimidating; a system that allows and encourages complex combinations of actions, but can be operated with minimal training.

From talking with target users and GIS professionals, it is clear that the Urban Growth Decision Support System is a powerful tool that allows people of a wide range of familiarity, both with computers in general and the domain material, to interactively explore and analyze the results of a complex geospatial simulation. The intuitive interface minimizes economic barriers of expensive software and training. The system is practical not only for conducting complex exploratory analysis, but also serves as an effective and engaging presentation vehicle for the results of such analysis.

By exploring the origins of the modifiable areal unit problem (MAUP), and based on understanding its challenges, this work identifies the ways in which probe-based geospatial applications are particularly susceptible to the MAUP. The user can probe the data by selecting their own regions-of-interest using a wide range of selection tools operating at a range of scales. When combined with the underlying raster-mapped data, generated from sources with different aggregation scales, the opportunities for the MAUP to affect the user's analysis are infinite.

While the MAUP is not easily solved, by planning for its appearance in geospatial analysis applications, its effects can be alleviated and potentially avoided. By alerting the user to any potential issues with the regions-of-interest they select to compare, much of the possibility that the comparisons they make will be misleading or misinterpreted is removed. Simple visualizations can provide quick indication of outliers in the distributions, allowing one to see at a glance what dimensions might become problematic in their analyses. Finally, semi-automated tools are provided to help the user understand these inequalities, and then correct their selections, minimizing the impact of these unintended problems that are inherent to probe-based interfaces, with their great freedom in region-of-interest selection choices.

REFERENCES

[Allen 2001] E. Allen, ``INDEX: software for community indicators", in Planning Support Systems: Integrating Geographic Information Systems, Models and Visualization Tools, ESRI Press, Redlands, CA, pp. 229-262, 2001

[Andrienko 1999] G.L. Andrienko, N.V. Andrienko, “Interactive maps for visual data exploration,” International Journal of Geographic Information Science, vol. 13, no. 4, pp. 355–374, 1999.

[Armhein 1995] C. Armhein. “Searching for the Elusive Aggregation Effect: Evidence from Statistical Simulations” Environment & Planning A, vol. 27, pp 105-120. 1995.

[Bederson 1996] B. Bederson, J. Hollan, K. Perlin, J. Meyer, D. Bacon, G. Furnas, “Pad++ A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics” Journal of Visual Languages and Computing, vol. 7, pp. 3-31, 1996.

[Bier 1993] E. Bier, M. Stone, K. Pier, W. Buxton, T. DeRose, “Toolglass and magic lenses: the see-through interface,” In Computer Graphics (Proceedings of SIGGRAPH 93), pp. 74-80, 1993.

[Bier 1994] E. Bier, M. Stone, K. Pier, K. Fishkin, T. Baudel, M. Conway, W. Buxton, T. DeRose, “Toolglass and magic lenses: the see-through interface,” Proceedings of CHI ‘94, pp. 445-446, 1994.

[Brewer 1994] Cynthia Brewer. “Color Use Guidelines for Mapping and Visualization” in Visualization in Modern Cartography. edited by A.M. MacEachren and D.R.F. Taylor, Elsevier Science, New York, pp 123-147, 1994.

[Brewer 2000] I. Brewer, A. M. MacEachren, H. Abdo, J. Gundrum, G. Otto, "Collaborative Geographic Visualization: Enabling Shared Understanding of Environmental Processes," IEEE Symposium on Information Visualization (InfoVis 2000), p. 137, 2000.

[Butkiewicz, Chang 2008] Thomas Butkiewicz, Remco Chang, Zachary Wartell, William Ribarsky. “Visual Analysis and Semantic Exploration of Urban LIDAR Change Detection" EuroVis 2008 / Computer Graphics Forum, vol. 27, num. 3, 2008.

[Butkiewicz, Dou] Thomas Butkiewicz, Wenwen Dou, Zachary Wartell, William Ribarsky, and Remco Chang. “Multi-Focused Geospatial Analysis Using Probes” InfoVis 2008 / IEEE Transactions on Visualization and Computer Graphics (TVCG) vol. 14, no 6. pp. 1165-1172, Nov/Dec, 2008.

[Butkiewicz 2009] Thomas Butkiewicz, Dong Hyun Jeong, Zachary Wartell, William Ribarsky, and Remco Chang. "Hierarchical Multi-touch Selection Techniques for Collaborative Geospatial Analysis" SPIE Volume 7346: Visual Analytics for Homeland Defense and Security 2009.

[Chang 2007] R. Chang, G. Wessel, R. Kosara, E. Sauda, W. Ribarsky, "Legible Cities: Focus-Dependent Multi-Resolution Visualization of Urban Relationships," IEEE Trans. Visualization and Computer Graphics, vol. 13, issue 6, pp. 1169-1175, Nov-Dec 2007.

[Cutter 1995] S.L. Cutter, D. Holm, and L. Clark. "The role of geographic scale in monitoring environmental justice" Risk Analysis, vol. 16, no. 4, pp 517-525, 1995.

[Dietz 2001] Dietz, P., and Leigh, D., "DiamondTouch: a Multi-user Touch Technology," in Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology, pp. 219-226, 2001.

[Donelson 1978] W. C. Donelson, "Spatial management of information," In Computer Graphics (Proceedings of SIGGRAPH 78), pp. 203-209, 1978.

[Dykes 1997] J. A. Dykes, "Exploring Spatial Data Representation with Dynamic Graphics," Computers and Geosciences, vol. 23, no. 4, pp. 345-370, 1997

[Edsall 2003] R. M. Edsall, "The parallel coordinate plot in action: design and use for geographic visualization," Computational Statistics & Data Analysis vol. 43, issue 4, pp. 605-619, 28 August 2003.

[ESRI 2010] ESRI, ArcGis website, http://www.esri.com/software/arcgis/, 2010.

[Forlines 2005] Clifton Forlines, Chia Shen, "DTLens: Multi-user tabletop spatial data Exploration" in Proceedings of User Interface Software and Technology (UIST), ACM, New York, NY, USA, pp 119-122, 2005.

[Forlines 2006] Clifton Forlines, A. Esenther, Chia Shen, D. Wigdor, K. Ryall, "Multi-user, multi-display interaction with a single-user, single-display geospatial application" in Proceedings of User Interface Software and Technology (UIST), ACM, New York, NY, USA, pp 273-276, 2006.

[Fotheringham 1991] A. S. Fotheringham, D.W.S. Wong, "The modifiable areal unit problem in statistical analysis" Environment and Planning A, vol. 23, pp 1025-1044. 1991.

[Furnas 1986] G. W. Furnas, "Generalized fisheye views," SIGCHI Bulletin, vol. 17, no. 4, pp 16-23, 1986.

[Furnas 2006] G. W. Furnas, “A fisheye follow-up: further reflections on focus + context,” Proceedings of CHI ‘06, pp. 999-1008, 2006

[Google 2010] Google Inc., Google Earth website, http://earth.google.com/, 2010.

[Jeong 2009] Dong Hyun Jeong, Caroline Ziemkiewicz, Brian Fisher, William Ribarsky, Remco Chang, “iPCA: An Interactive System for PCA-based Visual Analytics” Computer Graphics Forum (EuroVis 2009). pp. 767-774, 2009.

[Kadmon 1978] N. Kadmon, E. Shlomi, “A polyfocal projection for statistical surfaces,” Cartography J., 15, 1, pp. 36,-41, 1978.

[Klosterman 2001] Richard E. Klosterman, R. K. Brail, ``The What if? Planning support system'', in Planning Support Systems: Integrating Geographic Information Systems, Models and Visualization Tools, ESRI Press, Redlands, CA, pp. 263-284, 2001

[Klosterman 2005] Richard E. Klosterman, Christopher J. Pettit. “An Update on Planning Support Systems” Environment and Planning B: Planning and Design, vol. 32, no. 4, pp 477-484, July, 2005.

[Kunz 2002] Kunz, A. M., and Spagno, C. P., “Technical System for Collaborative Work,” in Proceedings of the Workshop on Virtual Environments 2002, pp. 73-80, 2002.

[Kwartler 2001] M, Kwartler, and R. N. Bernard, ``CommunityViz: an integrated planning support system'', in Planning Support Systems: Integrating Geographic Information Systems, Models and Visualization Tools, ESRI Press, Redlands, CA, pp. 285-308, 2001.

[Leung 1994] Y. K. Leung, M. D. Apperley, “A review and taxonomy of distortion-oriented presentation techniques” ACM Trans. Computation Human Interaction, vol. 1, no. 2, pp. 126-160, 1994.

[Mennis 2006] Jeremy Mennis and Torrin Hultgren. “Intelligent Dasymetric Mapping and its Application to Areal Interpolation” Cartography and Geographic Information Science, vol. 33 no. 3, pp 179-194, 2006.

[Nakaya 2000] Tomoki Nakaya. “An Information Statistical Approach to the Modifiable Areal Unit Problem in Incidence Rate Maps” Environment and Planning A, vol. 32, pp 91-109, 2000.

[OpenCV 2010] Open Computer Vision Library, Open Computer Vision Library website, http:/sourceforge.net/projects/opencvlibrary/, 2010.

[Openshaw 1984] Stan Openshaw. “The Modifiable Areal Unit Problem” Concepts and techniques in modern geography, issue 38, Geo Books, 1984.

[Openshaw 1999] Stan Openshaw, Seraphim Alvandies. “Applying Geocomputation to the analysis of spatial distributions” in Geographic Information Systems: Principals and Technical Issues. vol. 1, 2nd ed., 1999.

[Peltonen 2008] P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta, P. Saarikko, “It's Mine, Don't Touch!: interactions at a large multi-touch display in a city centre” in Proceedings of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems 2008. ACM, New York, NY, pp. 1285-1294, 2008.

[Perlin 1993] K. Perlin, D. Fox, “Pad: an alternative approach to the computer interface,” In Computer Graphics (Proceedings of SIGGRAPH ‘93), pp. 57-64, 1993.

[Scott 2004] Scott, S. D., Sleelagh, M., Carpendale, T., and Inkpen, K. M., “Territoriality in collaborative tabletop workspaces,” in Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, pp. 294-303, 2004.

[Shen 2006] C. Shen, “Multi-User Interface and Interactions on Direct-Touch Horizontal Surfaces: Collaborative Tabletop Research at MERL” in First IEEE International Workshop Proceedings of the Horizontal Interactive Human-Computer Systems 2006, pp. 53-56, 2006.

[Unwin 1996] A. R. Unwin, G. Hawkins, H. Hofmann, B. Siegl, “Interactive graphics for data sets with missing values – MANET,” Journal of Computational and Graphical Statistics, vol. 5, no. 2, pp. 113-122, 1996

[Unwin 1998] A. Unwin, H. Hofmann, “New interactive graphics tools for exploratory analysis of spatial data,” Innovations in GIS 5, Taylor & Francis, London, 1998.

[USEPA 2000] United States Environmental Protection Agency, “Projecting land-use change: a summary of models for assessing the effects of community growth and change on land-use patterns”, EPA/600/R-00/098, Cincinnati, OH, 2000.

[Viega 1996] J. Viega, M. J. Conway, G. Williams, R. Pausch, “3D magic lenses,” Proceedings of ACM UIST ’96, pp. 51-58, 1996.

[Waddell 2008] Paul Waddell, Liming Wang, Xuan Liu, “UrbanSim: An Evolving Planning Support System for Evolving Communities” in Planning Support Systems for Cities and Regions, edited by Richard Brail. Cambridge, MA: Lincoln Institute for Land Policy, pp. 103-138. 2008.

[Wigdor 2006] D. Wigdor, C. Shen, C. Forlines, R. Balakrishnan, “Table-Centric Interactive Spaces for Real-Time Collaboration” in Proceedings of Advanced Visual Interfaces, ACM Press, New York, NY, pp. 103-107, 2006

[Wilson 2004] Wilson, A. D., “TouchLight: an imaging touch screen and display for gesture-based interaction,” in Proceedings of the 6th International Conference on Multimodal Interfaces, pp. 69-76, 2004.

VITA

Thomas Butkiewicz was born to Dr. Bernard Butkiewicz, MD and Dr. Barbara Heere, MD in Pittsburgh, Pennsylvania on April 20th, 1983. He was raised in Williamsport, PA where he attended Loyalsock Township High School, graduating in 2001. In 2005 he received a Bachelor of Science cum laude in Computer Science with a minor in mathematics from Ithaca College in Ithaca, New York. In 2007 he received a Master of Science degree from The University of North Carolina at Charlotte in Charlotte, North Carolina.