AURORAMAP: A BOUNDARY-HOMOGRAPHIC VISUALIZATION FOR MAPPING MULTIVARIATE 2D SPATIAL DISTRIBUTIONS

by

Guojun Han

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

Department of Computer Graphics Technology

West Lafayette, Indiana

May 2020

THE PURDUE UNIVERSITY GRADUATE SCHOOL

STATEMENT OF COMMITTEE APPROVAL

Dr. Yingjie Victor Chen, Chair

Department of Computer Graphics Technology

Dr. Fenggang Yang

Department of Sociology

Assistant Professor Nancy Rasche

Department of Computer Graphics Technology

Approved by:

Dr. Nicoletta Adamo

To my family, advisors, and friends, who supported me throughout my two years at Purdue.

ACKNOWLEDGMENTS

The author would like to thank her family, especially her parents, for their unconditional love and support. The author would like to sincerely appreciate and show her gratitude to her academic advisor, Dr. Yingjie Victor Chen, for the expertise and consistent guidance and help throughout the author's entire graduate study, her committee member Dr. Fenggang Yang, for providing insightful advice and meaningful datasets, and Associate Professor Nancy Rasche, for the generous guidance and support. The author owes gratitude to Junhan Zhao, who inspired and encouraged the team's final effort despite the challenges, Jiankun Bi for providing constant and valuable technical support, and Jiahui Dong for contributing to the data collection, and analysis. The author thanks Dr. Chen Guo for guidance on literature review and Hariharan Sezhiyan who helps with proof reading. The author thanks the Center on Religion and Chinese Society for support under the project grant and other participants for their valuable time and feedback.

TABLE OF CONTENTS

LIST OF TABLES 7
LIST OF FIGURES 8
ABSTRACT 9
CHAPTER1.INTRODUCTION 11
1.1 Background 11
1.2 Research significance 12
1.3 Statement of Purpose 14
1.4 Research Questions and Hypothesis 14
1.5 Assumptions and Limitations 15
1.6 Delimitations 16
CHAPTER2.LITERATURE REVIEW 17
2.1 Perceptual Color Theories for Effective Density Visualizations 18
2.2 Off-screen Visualization 20
2.3 Multivariate Data Visualization 22
CHAPTER3.VISUALIZATION ALGORITHMS 25
3.1 Boundary and Offset Acquisition 26
3.2 Segmentation and Density Computation 28
3.3 Color Encoding and Smoothing 32
CHAPTER4.EXPERIMENTS 35
4.1 Participants 35
4.2 Color scale and displays 36
4.3 Training 37
4.4 Experiment 40
4.5 Experiment Data Analysis 43
CHAPTER5. CONCLUSION 47
5.1 Summary 47
5.2 Application of Work 48
5.3 Contributions 51
5.4 Limitations 52

REFERENCE 55
APPENDIX. RESEARCH PARTICIPATION CONSENT FORM 63

LIST OF TABLES

Table 1: Featured Statistical Results for Cluster-Detected Valid Rate.. 44

Table 2: Results for Spearman Correlation Test 44

LIST OF FIGURES

Figure 1: AuroraMap visualizes the spatial distributions of religious infrastructures in different provinces (Simplified provincial boundaries; Data source: Online Spiritual Atlas of China. The colors on the map encode the densities of the regions along the normal vectors of vertices on the boundary (black line)). The graph on the right shows the congruent relationship of plotting the distribution as a dot map and the AuroraMap. 13

Figure 2: Different Simplified factors applied to the same boundary: 0, 0.15, 0.3 and 0.4 from left to right........ 26

Figure 3: Different “stroke-width” value options........ 27

Figure 4: Geometry abstraction and boundary offsetting process........ 27

Figure 5: Maximum circle segmentation method. The adjacent dashed max-inscribed circles with respect to division knots on the boundary define the region for computing the densities. The diagram shows all essential elements. 29

Figure 6: (a) flow diagram shows how to divide and encode the subdivision densities accordingly; dots of original distribution in blue region (b) are homographically projected to the red region in (c). In this example, distribution has been divided into 20 slices along normal direction. 30

Figure 7: AuroraMap algorithm pseudocode 32

Figure 8: Single-hue color scheme we used for early research phase and the Blue/Red scheme. 33

Figure 9: Single Maximum circle projection 37

Figure 10: Regular geometric outlines and an irregular geometric outline as a part of training materials........ 38

Figure 11: Single Experiment materials 40

Figure 12: Participants working on two experiment trials........ 42

Figure 13: Overlapping Dot Density Map with User Inputs........ 45

Figure 14: Apply AuroraMap to Visually Compare Two Religions; Two offset areas are visualizing Buddhist and Non-Buddhist infrastructure locations in Hunan Province. Same technique could be applied to resolve off-screen visualization, where the outer offset area visualizes off-screen objects and the inner area shows the samples inside the boundary. 48

Figure 15: Apply AuroraMap to Visualize Religious Data of Three Adjacent Provinces; Three adjacent boundaries are visualized by AuroraMap. The observers are able to compare the densities between subregions. Top: Hubei; Bottom-Left:Hunan; Bottom-Right: Jiangxi. 49

ABSTRACT

Visualizing multidimensional spatial data is an essential visual analysis strategy, it helps us interpret and communicate how different variables correlate to geographical information. In this study, we proposed an abstract contextual visualization that encodes data on the boundaries of spatial distributions and developed a new algorithm, AuroraMap. AuroraMap projects the spatial data to the boundaries of the distributions and color-encodes the densities continuously. We further conducted the user experiments, and the results show users can detect the relative locations and scopes of the clusters. Furthermore, users can quantitatively determine the peak value of each cluster's density. The method provides three contributions: (1) freeing up and saving the graphical visualization space; (2) assisting the users to quantitatively estimate the clusters inside distributions; (3) facilitating the visual comparisons for multiple and multivariate spatial distributions. In the end, we demonstrated two applications with real-world religious infrastructural data by AuroraMap to visualize geospatial data within complex boundaries and compare multiple variables in one graph.

CHAPTER 1. INTRODUCTION

1.1 Background

The growing amount of spatial data enables us to capture sufficient geospatial information effortlessly, but impedes obtaining a flexible scope for data interpreting tasks. This places a huge challenge for decision-makers who analyze spatial data in a particular 2D space and its surrounding area simultaneously. For example, the task is to identify areas with densest spatial points in a small county while keeping track of its surrounding counties. A flexible scope restrains the amount of data users perceive at a time, as well as the geographical range to prevent overwhelming visual information.

Existing research has led to techniques and systems to visualize aggregated spatial data, including raw data aggregation over groups, clustering data in the map, presenting summative statistical data along with geographical information, and developing abstract visualization. For example, SungYe Kim et al. proposed a data encoding scheme called Bristle Map to support multi-attribute event aggregation within a specific space. However, the user can be cognitively overloaded if they need to compare spatial distributions at different geographical levels. Willmott et al. presented a geographic box plot that computes a spatially weighted diagram to describe a wide range of geographic variables and associates the spatial elements' area to the box plot's area. It enables a broad view of the overall spatial distribution pattern, whereas, it loses geographical correlation to the raw data significantly.

The limitations of these related works inspire us to propose a simple abstract visualization in this domain - AuroraMap. AuroraMap focuses on presenting the density and centroid of spatial distribution clusters (an area with distinct gathering points) on a flexible geographical scope and enables spatial distribution encoding upon any sophisticated geographic typology.

1.2 Research significance

In the context of the enormous growth of spatial data, efficient visualization methods are expected to deal with the request of displaying multiple distributions and multivariate distributions. Since boundaries and centroids are the core features of localizing spatial distributions, we present a novel contextual-based approach in this domain.

We derived visualization design from Phoenixmap, which is an abstract visualization technique that encodes densities as varying thicknesses across the distributions' boundaries. This work has proven that users can effortlessly understand and quantitatively estimate the density of spatio-temporal distribution through distributions' boundaries. Utilizing boundaries to visualize spatial data could fundamentally alleviate the difficulty of visualizing multiple 2D spatial distributions in one graph by only overlapping multiple Phoenixmaps in different colors.

We validated our design based on a quantitative evaluation with human subjects. The results of this study indicate that our design effectively facilitates users interpretation and decision-making process on 2D spatial data. Based on the results, we did discover that abstracting the distribution on the boundary could result in the information loss, which is the change of densities from the centroid to the boundary inside the distribution.

Nevertheless, Phoenixmap retains a reasonably better quantitatively estimable trait to observers compared to conventional heatmaps and dot maps assisted by additional legends. To avoid the discouragement of losing details while maintaining the advantages of Phoenixmap like space-saving and data-abstracting, we adapted the algorithm and developed a color-based visualization, AuroraMap, which uses the area offset from the boundary to display the distribution.

We further designed and conducted experiments to test the usability of AuroraMap and the users' capability of localizing the original clusters given AuroraMaps. The user test has directly examined the AuroraMap visualizations created from the real-world dataset, which consists of the geospatial data of religious infrastructures like Buddhist temples and Christian churches, located in different provinces of China. Observing given AuroraMaps, the participants were asked to draw the boundary in free-hand and estimate the peak value of each cluster within the distribution. The statistical results suggest that the participants were able to address the relative locations and scopes of the clusters for the given spatial distribution produced by AuroraMap. Additionally, the results also show the accuracy of users' estimations about the peak values for clusters. The number and locations of the clusters in users' answers correlated to the truths computed by a density-based clustering method, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and confirmed by a hierarchical clustering method, Chameleon. This test reveals the fact that the hierarchical relationship of the density distributions could be gained by the observers accurately. The application further shows the capability of applying AuroraMap to visual analysis tasks comparing between multiple variables for the same base-map and for resolving off-screen visualization.

This method can be conceptually considered equivalent to tearing, compressing, and projecting a conventional heatmap to the boundary of a spatial distribution, similar to homography. Therefore, the patterns encoded in the AuroraMap remain the transformed shapes of clusters and the colors indicate the density changes. The main contributions of this visualization method are: (1) freeing up the display properties for other uses, like overlapping the base-maps for references, (2) allowing the observers to quantitatively estimate the clusters or subsets of data and determine their relative locations and shapes, (3) enabling the visualization of multivariate data on one single map.

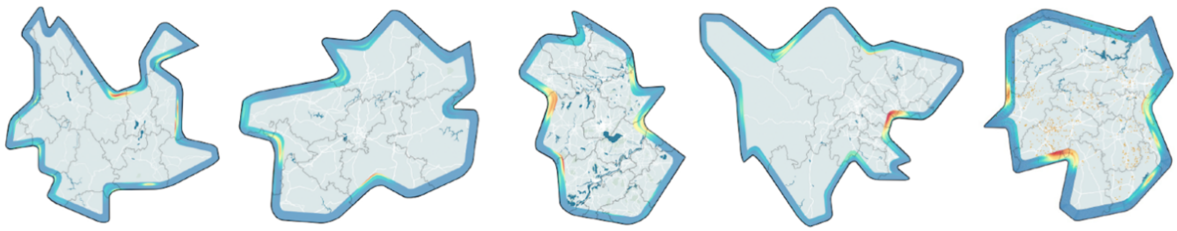


Figure 1: AuroraMap visualizes the spatial distributions of religious infrastructures in different provinces (Simplified provincial boundaries; Data source: Online Spiritual Atlas of China. The colors on the map encode the densities of the regions along the normal vectors of vertices on the boundary (black line)). The graph on the right shows the congruent relationship of plotting the distribution as a dot map and the AuroraMap.

1.3 Statement of Purpose

The purpose of the research is to address challenges in visualizing spatial distributions by proposing a novel visualization technique with the following characteristics: 1) reveals essential spatial distribution properties, including: the density and centroid of point clusters with the color-encoding method, 2) enables flexible geographical scope, with the ability to project density changes in spatial distributions on an arbitrary boundary, potential boundaries to use for encoding including a concave/convex hull outline of spatial points or any level of a political boundary, 3) resolves computation difficulties to encode distribution information with a variety of colors on a sophisticated boundary shape. We presented an algorithm to simplify, abstract and aggregate spatial data that works efficiently on most boundary typology types, including polygon, Bezier curve, etc. The research validates the efficiency of AuroraMap by conducting human subject tests. The results prove that AuroraMaps are a competitive alternative to traditional spatial distribution analysis techniques.

1.4 Research Questions and Hypothesis

The human subject study is conducted following the above three questions:

1. Does AuroraMap effectively reveal key spatial distribution properties, including: density and centroid of a point cluster?
2. How well can users perceive Auroramp in tasks in terms of accuracy?
3. Is AuroraMap a competitive visualization technique compared to ground truth visualizations, including: dot distribution map and heatmap?

We made the hypothesis on each questions respectively:

1. The AuroraMap effectively presents spatial distribution properties with color encoding method
2. Users performs well understanding AuroraMap in task-based evaluations
3. AuroraMap is a competitive visualization to abstract and present spatial data properly compared to dot distribution map and heatmap.

For the listed research hypothesis, we designed a human subject test that asked users to perform a series of tasks using AuroraMap. We validated whether users are able to interpret centroid of point a cluster by asking them to draw the clusters with the maximum density above a particular range; we tested if users can perceive density variation from AuroraMap by asking them to estimate the maximum density of each cluster they drew.

1.5 Assumptions and Limitations

We admit the following limitations in this research:

1. All participants are from Purdue University, and the number of participants is not significantly large. Therefore, they may not fully represent all users for AuroraMap
2. As an abstract visualization, the visualization omits detail about each exact location of spatial points; however, the users can implement interaction technique such as focus-plus-context to enhance detail exploration
3. AuroraMap encodes spatial distribution with a continuous projection function using relative densities. Hence, it has limitations in capturing significantly small density variations, as the color for those variations are less evident for human eyes
4. It introduces the possibility of visualization clutter in visualizing multi-categories data.

We admit the following limitations in this research:

1. All participants are from Purdue University, and the number of participants are is not significantly large. Therefore, they may not fully represent all users for AuroraMap
2. As an abstract contextual visualization, AuroraMap omits details about each exact location of spatial points; however, the users can implement interaction technique such as focus-plus-context to enhance detail exploration
3. AuroraMap encodes spatial distribution with a continuous projection function using relative densities. Hence, it has limitations in capturing significantly small density variations, as the color for those variations are less evident for human eyes
4. It introduces the possibility of visualization clutter in visualizing multi-categories data

1.6 Delimitations

We present the following delimitations in this research:

1. The study uses a Blue-Red sequential color scale to encode spatial distribution properties, and does not provide other color options
2. AuroraMap provides an abstract visualization approach and uses simplified boundary for visual encoding, and does not offer a solution for boundaries with over complexity

CHAPTER 2. LITERATURE REVIEW

Spatial distribution data is widely used in census data, bioinformatics, and urban planning, to name a few. However, visualizing distribution data can be challenging due to the requirement to analyze and compare various spatial situations and gain an understanding of temporal changes.

The spatial distribution data sets can be characterized as multidimensional data, which refers to the spatial attributes of the raw data (eg., 0D that represents individual "dot," 1D, 2D, 3D), it may also include other dimensions such as time variation. If the data present only at an individual "dot," the visualization can be relatively straightforward using statistical approaches, such as box plots or bar charts. Box plot is a standardized visualization method of representing data distribution and can be used to visualize geographic or spatial variables. Willmott's Geographic box plot [1] computes a spatially weighted diagram to describe a wide range of geographic variables and associates the spatial elements' area to the box plot's area. Another alternative method of traditional box plot in visualizing spatial statistics is the "Oriented Spatial Box Plot," introduced by Laurent et al. [2] that extends the classical one-dimensional box plot in 2D point clusters summarizing and visualizing. The accuracy of box plots in displaying statistical patterns makes it easy and straightforward to understand spatial distribution density, pattern, etc. However, it often fails to represent multidimensional data or display complex geographical shape.

There is much research that has been done to visualize spatial distribution data — mainly utilizing map-based visualizations for spatial information transferring. A thematic map visualization represents the distribution of a specific phenomenon with visual components; it contains information found in a normal topographic map. A visualization system is created by distinguishing six basic visual variables: size, value, texture, color, and shape to identify the proper symbology for a thematic map visualization, which is used widely in representing data attributes with cartographic components. Examples including proportional symbol grid maps that include symbols that vary in size based on the value they represent, using either absolute scale or apparent magnitude scale to visualize data. Another widely used visualization method is grid choropleth, it abstract regions on a map into a series of shapes; the shapes allow for the inclusion of different sort of data displaying visualizations. Dang et al. introduced Dynamaps [3] as a generalized map-based visual analysis tool for dynamic queries and brushing on choropleth maps. However, as relatively abstract data representations, some of these solutions lack of good data representations for distribution details. We further conducted literature reviews in the following aspects: 1) Off-screen Visualization, 2) Perceptual Color Theories for Effective Density Visualizations and 3) Multivariate Data Visualization.

2.1 Perceptual Color Theories for Effective Density Visualizations

It is prevalent to use color to convey categorical and quantitative differences in geospatial visualization. A discrete color map defines discrete data items, and a continuous color map is used to present a continuous data range. Color has three perceptual dimensions – hue, saturation, and lightness. Variations in the three channels can encode differences in the data. Hue, which represents the color name, is more appropriate to show the nominal or categorical differences. On the other hand, lightness, the brightness of the color, and saturation, the vividness of the color, is more suited to visualize the ordinal differences [14][15]. Depending on the design, data can be comprehended in a better or worse way. A well-designed color scheme can provide good insight into the data, while an ill-designed color scheme can cause confusion and mistake.

Color perception has been studied extensively in the visualization community [16] [17] [18]. Silva et al. [16], Bernard et al. [17] and Zhou & Hansen [18] review the most advanced work in colormap design and color perception and further provide guidelines about which color scale or colormap to select or create concerning to the type of data and analysis tasks to be performed. ColorBrewer.org [19] is an excellent tool to select and generate color scales for cartographic visualization based on users' specific mapping needs. Inspired by the ColorBrewer tool, Dykes and Brunsdon [15] propose a series of geographically weighted interactive maps and use both sequential and diverging color schemes to show the variations of spatial patterns. Similarly, Lampe and Hauser [20] adopt the concept of kernel density estimation to visualize streaming data of maritime trajectories and commercial air traffic. A sequential colormap is designed to help explore the variations in the continuous spatial domain and compare different trajectory regions.

Although there have been numerous research studies conducted to generate effective color mapping techniques, it is still a difficult task to design effective colors that provide good discrimination between data values. One of the most important tasks in spatial data analysis is to detect the degree of similarity among global and local spatial distributions [21]. As one of the most popular visualizations, a geospatial heatmap is useful to represent an overview of the spatial distribution on a map as well as estimate the difference across maps. A warm-to-cool spectrum is widely adapted by researchers to colorize the density data in a heatmap [22]. Following the convention, "Dark equals more" [19], regions with higher densities are colored darker compared with those with lower densities.

Despite the popularity of the rainbow color scheme, research has shown several weaknesses and criticized the rainbow color scheme through its lack of perceptual ordering, has a difficulty of distinguishing small saturation variations, falsely segments data, and hinders accurate reading of data due to banding effects at hue boundaries [23][24][25]. To provide guidelines and designs for quantitative color encoding, researchers have conducted a series of comparative analyses of different colormap types. Liu and Heer (2018) [26] analyzed the speed and accuracy of four single-hue and five multi-hue colormaps in which participants judged similarity across varying scale locations and value spans. They proposed that multi-hue colormaps may be preferable to single-hue in heatmaps because multi-hue colormaps can provide improved resolution. Reda, Nalawade, and Ansah-Koi (2018) [27] conducted three crowdsourced experiments to measure participants' ability to estimate quantities and perceive longitudinal patterns in nine commonly-used colormaps. They found that the rainbow color scheme is the most accurate colormap for quantity estimation irrespective of spatial complexity, while divergent colormaps excel in tasks requiring the gradient perception of high-frequency patterns. ColorBrewer [19] also proposes that the diverging color scales could effectively highlight the contrasting luminance at high and low extremes. It is more suitable to emphasize a critical data class or break point for scalar fields.

All of the previous work on color perception and color mapping provides a deeper understanding of the nature of data and tasks as well as how to create aesthetically pleasing yet effective colormaps. Given the advantages of the multi-hue color scheme as well as its popularity and similarity in the information visualization community, we decided to implement the multi-hued color scheme to encode the distribution and variance of spatial data in AuroraMap.

2.2 Off-screen Visualization

As a data-driven approach, the general idea of off-screen visualization is to preserve overview while maintaining the information about the data topology and characteristics.

According to Cockburn, Karlson, and Bederson [6], "Cue-based techniques […] modify how objects are rendered and can introduce proxies for objects that might not be expected to appear in the display at all." The off-screen visualization techniques provide visual cues located at the display border to indicate the position and the distance of elements clipped from the viewport.

Aligned arrows[7][8] have been applied to map visualizations, while it only encodes the direction, not distance. Halo [9] uses translucent arcs on the border of the display window to indicate the location of off-screen elements. The arc length encodes the distance from the viewing space. Although Halo successfully uses the orthographic strategy to visualize off-screen objects, it suffers from visual clutter at the border of the screen when showing a large number of off-screen objects. To overcome some of the limitations of Halo, Hopping [10] is designed to quickly and easily navigate to an interesting area in a large-scale context. It combines oval halos with a "laser beam" to show proxies of off-screen targets and a teleportation mechanism to navigate to the target location and context. However, Gustafson et al. [11] further find the oval approach is not sufficiently accurate to locate off-screen objects. Wedge uses an overlap-avoidance algorithm to reduce the amount of overlap, maximize the location accuracy, and provide sound distance cues. The Wedge layout algorithm offers significant improvements over Halo and shows substantial accuracy advantages. Some techniques improve Halo by preserving topology and overcome clutter through aggregation [11][12]. HaloDot [12] aggregates points-of-interest with color and transparency to represent the relevance and the distance of each object. The objects with warmer and more visible colors represent more relevant ones, while the less relevant ones are represented with colder and less visible colors. Ambient Grids [13] proposes a grid coloring approach to project and visualize the point and shape data in the border region. Points and shapes are rasterized using the grid and then projected to the border region. Ambient

Grids also assigns a radial cell-based color gradient to the cells mapped within the eight border regions.

Although these off-screen solutions aim to aggregate an overview of off-screen objects, there is a high potential for applying the approach to in-screen data items. Jäckle, Kwon, and Keim [14] propose the area of off-screen visualization as a pioneering approach and uncover the potential of off-screen visualization through general applications. One of the most critical challenges is to preserve the overall topology of objects through the appropriate design of visual proxies. Inspired by the projection method and visual proxy design in off-screen visualizations, AuroraMap projects the spatial data to the geometric boundaries of the distributions and uses colors to encode the density of the distributions. This makes it feasible for researchers to maintain the overall topology of data items, while still being able to use the inside screen real estate for more important information.

2.3 Multivariate Data Visualization

Multivariate data analysis involves examining patterns and relations in three or more data variables. Various visual mapping techniques have been presented to convey the high-dimensional information to the user through pixel-oriented techniques, glyphs, geometric projections, hierarchy-based techniques, and animation [29]. All of these techniques try to reduce dimensionality while preserving the main structure visually. However, Chan [30] have shown that each approach has its advantages and disadvantages. One of the limitations is that it may result in a cluttered representation for more massive data sets, which can impede the user's perception capabilities. A straightforward approach is to apply aggregation in visualization to reduce clutter. The binned aggregation has been employed to reduce the data record and query processing time. For example, imMens [31], Nanocubes [32], and Hashedcubes [33] design data cubes to explore and analyzes large multidimensional spatial datasets. Although these systems successfully solve the scalability problem, the structures only support heatmaps. Heatmaps are effective in presenting the overall spatial distribution patterns. However, they are hard to detect the locations of data values in each cluster precisely, and a high-density area may be indicated where there are actually few data points.

Compared to existing heatmaps or binned plots, a cluster-based data cube has been utilized to support interactive visualization of large-scale multidimensional spatial data [34]. Li et al.[35] propose ConcaveCubes to apply a novel concave hull construction method for boundary-based cluster map visualization. The color of the concave hull represents an aggregated value of properties in each cluster. The experimental evaluation of ConcaveCubes shows that the boundary-based cluster maps present more precise geographical information with semantic meanings and are suitable to visualize geographical points on a map. TopoGroups [35] applies a boundary distortion algorithm to enable effective context-preserving navigation and identify different spatial distribution patterns at varying scales. The design space of different visual encodings for the boundaries are fully explored. Color, transparency, shading, and shapes can be used to convey the hierarchical information of the clusters across multiple scales. Phoenixmap [36] uses a similar approach to visualize multiple spatial distribution datasets. This approach adapts the concave-convex hull algorithm to divide the outline into many segments with various widths to convey the range and density information on a map. It frees up the space inside the boundary and can overlap many Phoenixmaps as needed on the map to visualize multiple spatial distributions. However, Phoenixmap is hard to precisely the shape of each cluster within the distribution. We further adapt the algorithm and develop Aurormap to preserve the contextual information as colored areas mapped to the border region.

CHAPTER 3. VISUALIZATION ALGORITHMS

In this chapter, we proposed a novel algorithm to tackle the challenges in visualizing spatial data, including 1) traditional methods like heatmap and dot distribution map required multiple visual representations for multidimensional spatial data, 2) numerous graphs cause difficulties for user perception tasks such as lookup, comparison, and relation seeking, 3) existing contextual-based algorithms such as Bristle Map, Ambient Grid have limitations in providing a flexible geographical range.

Our visualization algorithms address the above challenges, in terms of 1) proposing an abstract visualization method that aims at presenting two essential properties, density and centroid of points clusters in spatial distribution data, 2) free up space for potential overlapping and introduces contour layers to present multiple category spatial data at one time to prevent graph clutter, 3) projecting spatial distribution information on a geographical boundary and enables easy comparison for different geographical range, 4) using smoothed boundary and color range to provide a beautiful visualization solution.

For example, given a set of 2D spatial points presented in any particular region, we define the region boundary that encloses all target objects, then we simplify and round the edges before assigning an offset width to the outline which provides space for spatial density projection; then, we transform and compress the position of the points in the area to the offset space using a segmentation technique, we segment the offset space into segments for color filling; lastly, we smooth and average the color value in each cell with a continuous projection function. Our approach offers an overview of the density properties within any geographical range and helps users to conduct quantitative estimation on density values accurately.

3.1 Boundary and Offset Acquisition

The first step on spatial data aggregation is to define a boundary that encloses the investigated points on a map. There are two typical boundary types, including 1) a pre-defined boundary, such as political province outline (province, state, nation), and natural geographical outline (rivers, mountain, lanes, etc.), 2) a computational spatial boundary, mainly derived from data processing domain (such as a concave hull that embraces all the points but with minimal area.) For our AuroraMap algorithms, we choose to use a pre-defined boundary as a restricted border for the investigated. Our raw spatial data records spatial points as pairs of longitude and latitude in a map within a Euclidean coordinate; each pair of data describes a unique point location without duplication. For computational consistency, we describe the outline as a group of points that locates on the outline to embrace all spatial points. There are, on average, hundreds of points since sufficient adjacent points can determine the geometry shape as polygon accurately. We use GeoJSON for encoding the boundary geographical data structure.

In AuroraMap visualization, we intend to use a simplified outline shape to provide an abstract overview of spatial distribution properties. Therefore, we introduce polygon simplification algorithms: Douglas-Peuker and Radial Distance algorithms as a combination to reduce the number of points. We implemented the polygon simplification and decreased 80% additional points while retaining its shape by keeping track of simplified tolerance. There are averagely 60 points to define the adjusted geographical boundary without losing shape details.

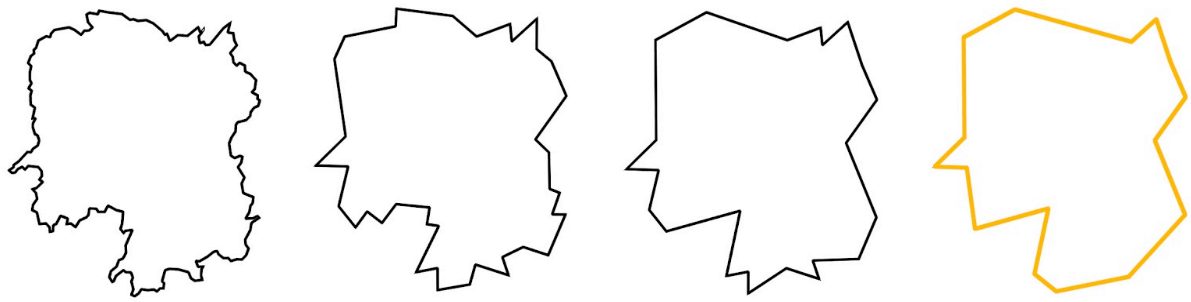


Figure 2: Different Simplified factors applied to the same boundary: 0, 0.15, 0.3 and 0.4 from left to right

Based on the simplification results, we computed an outline abstraction by connecting outline points with Bezier Curve. We chose to use the Bezier curve algorithm because it produces curves with reasonably smooth at all scales and passes through all the control points on the outline. The Bezier curve also provides a better aesthetical representation of any geometry shapes. Afterward, we rounded up the edges to prevent sharp corners appearing on each Bezier Curve outline due to drastic changes in each segment of the curve, causing difficulties in segmentation (our second step), which we will elaborate on in the following chapter.

To offer space to color encoding spatial densities, we offset the Bezier Curve as an outer boundary and intended to produce a “parallel” outline as an inner boundary using quadratic bezier offsetting with selective subdivision. The outer boundary and inner boundary provides a “stroke” naturally where we put color information to present density variation and cluster position. We set the “stroke-width” with a practical value that prevents lines from self-intersection while providing enough space for color information. The values vary from different geometrical typology based on the complexity of the shape.

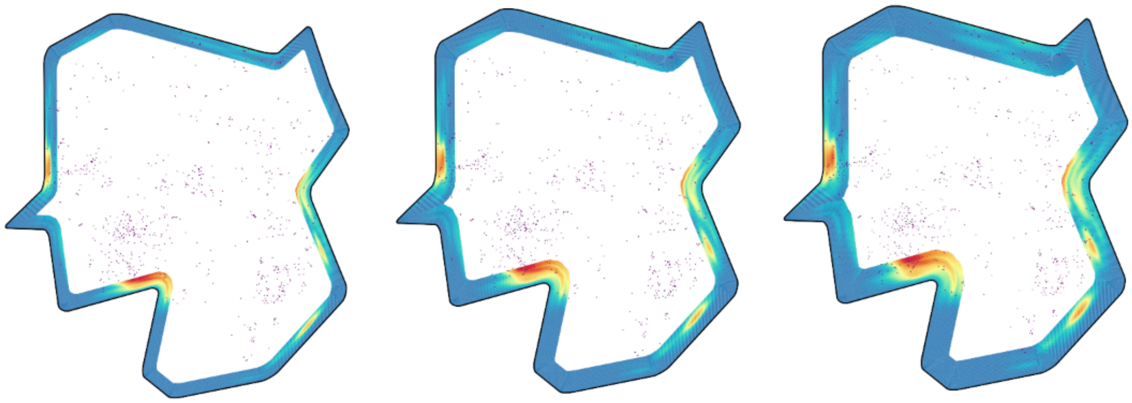


Figure 3: Different “stroke-width” value options

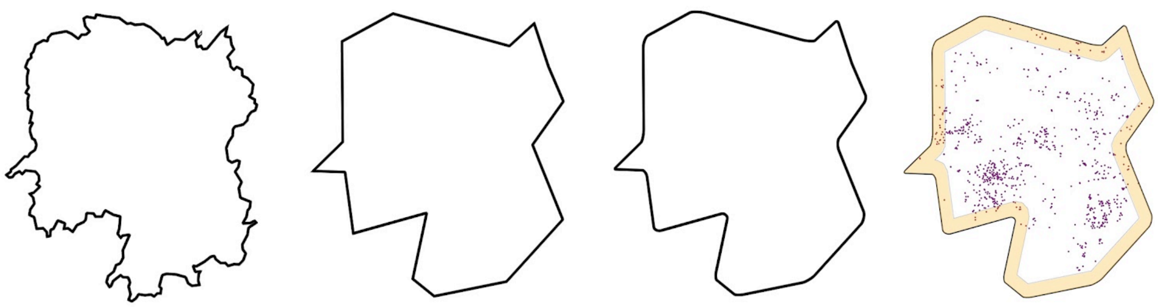


Figure 4: Geometry abstraction and boundary offsetting process

3.2 Segmentation and Density Computation

AuroraMap uses a range of colors to encode density variation within a particular region. In the real-world scenario, spatial distribution densities vary in different locations in the region. Therefore, we segment the regional geometry shape into multiple minor pieces, calculate the density in each segment, and transform density information as color variants to be filled in the offset stroke. There are various geometric segmentation algorithms in computer graphics that attempt to segment a 2D surface (typically closed area) into meaningful pieces as expected from human observers. Medial axis transformation [37] and Voronoi diagram [38] of a closed bounded area are basic entities associated with the natural properties of that area. The medial axis is a set of points of maximum-radius circles (bounded by the area in at least two points), and the medial axis transformation technique is frequently used in geometry area segmentation. The Voronoi diagram describes a series of curve segments such that each point constrained in a particular region is at least as close to its nearby boundary segment as to other segments.

In our visualization, we intend to segment a given domain as well as its corresponding boundary stroke and transform spatial properties enclosed in the domain to stroke spaces and inspired by Phoenixmap, which breaks the outline into multiple segments and assigns a different thickness to encode density information. We chose to segment the region denoted as S with the maximum circle sampling method introduced as a part of Phoenixmap algorithm. We first break the outline denoted as C into n segments by interpolating points on the Bezier curve outline, each sampling points have an average distance between each other to ensure an even segmentation. For any line segment ci ∈ C, we defined a segment ri ∈ R to measure density variation. On any line segment ci, we extracted two adjacent points as a pair denoted as (pi, pi+1). Then we calculated a maximum inscribed circle (bounded by the outline) for each point to get two circle centers denoted as (oi, oi+1). We constructed a quadrangle region denoted as Ri from (pi, pi+1, oi, oi+1) sequentially until reaching the last point pn-1 on C. Since we use a closed geometry-shape to encode spatial data, we extracted the last point pn-1 and the first point p1 as (pn-1, p1) to compute the last segment. This method provides an approximation segmentation solution with the same mechanism of medial axis transformation. It is mathematically provable that i=1nRi S. Moreover, the maximum circle method avoids segmenting irregular polygons other than a quadrangle, which decreases algorithm complexity significantly. In our algorithm, we used n=5000 as an empirical number of segments according to the region geometry entities such as perimeter, complexity.

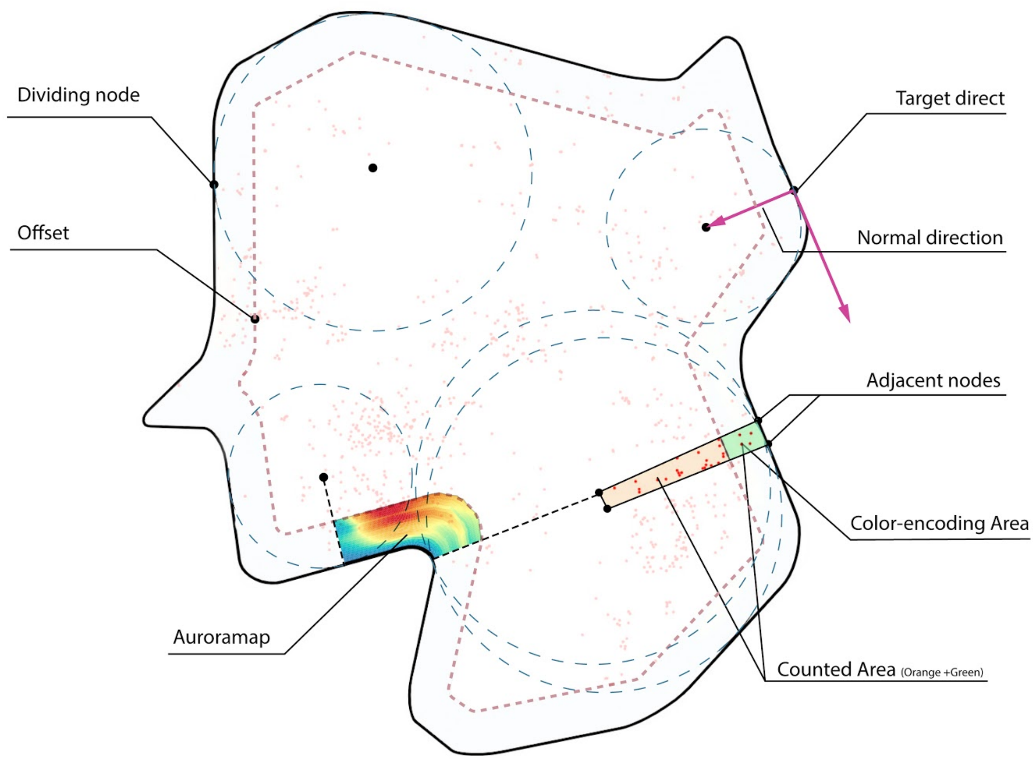


Figure 5: Maximum circle segmentation method. The adjacent dashed max-inscribed circles with respect to division knots on the boundary define the region for computing the densities. The diagram shows all essential elements.

Aiming to measure the distribution in either vertical (one can imagine a vector represented by a directed line segment on the outline) or perpendicular (another vector perpendicular to the vertical vector), we differentiated the domain segments into n sub-segments averagely. Given a quadrangle segment Ri which is constructed by four points (pi, pi+1, oi, oi+1), we segmented two lines defined by two pairs of points (pi, o1) and (pi+1, oi+1) into m sub line segments, and constructed perpendicular a set of sub-segments denoted as {(oi, oi+1, opi, opi+1), (op1i, op1i+1, op2i, op2i+1), … (opmi, opmi+1, pi, pi+1)}. We named the vertical sub-segment as a region cell as an easier reference for the next step.

To encode density variation into the "stroke" that we computed from the step of "boundary and offset acquisition," we differentiated the stoke into sub-segment following the above region segmentation process. We divided the "stroke" into a set of cells sequentially, the cell is small enough, and each fits in less than one pixel, we further fulfill color information into each cell to present the density variation in its corresponding domain segments. We named the stoke cell as a stroke cell as a more natural reference for the next step.

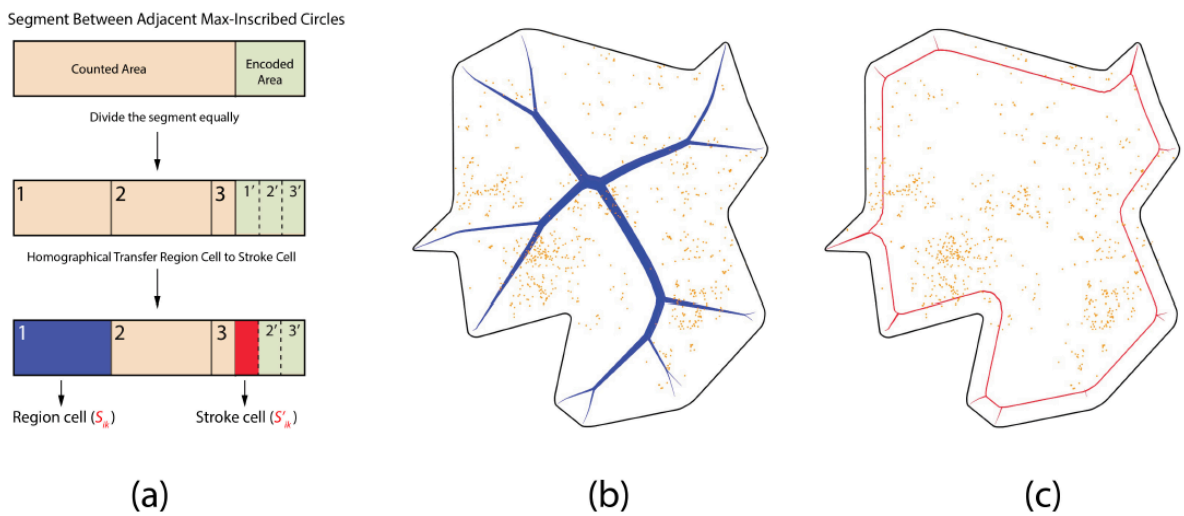


Figure 6: (a) flow diagram shows how to divide and encode the subdivision densities accordingly; dots of original distribution in blue region (b) are homographically projected to the red region in (c). In this example, distribution has been divided into 20 slices along normal direction.

Lastly, we transformed the density of the spatial points in each region cell to its corresponding stroke cell. One can imagine a scenario that all points enclosed by a region boundary are compressed into the stroke. For each region cell, we counted the number of the dots and divided it by region cell area to calculate the absolute density value. We stored the absolute density value in a dictionary, each with a unique "key" value that represents which region cell it is calculated from. In the program, we looped through the dictionary and calculated the absolute density range. To alleviate the drastic change in adjacent region cell, we averaged and smoothed the absolute density values to relative density values by two means: 1) for a set of region cell densities that derived from an identical vertical segment, we defined a step k and weighted factors to average adjacent cell densities. The weighted factors are presented as float numbers in an array A0 with a length of k. For each region cell, we extracted the cells forward and backward with k/2 steps and stored their density values sequentially in another array A1. By traversing through A0 and A1 in the same time, we calculated the weighted arithmetic mean(WAM) for the cell; 2) for a set of region cell densities that derived from an identical perpendicular index, we calculated the weighted arithmetic mean for each cell as the relative density value.

3.3 Color Encoding and Smoothing

The algorithm accurately calculates the relative density values for each region segments and presents a continuous density variation in its geographical context. We used color to encode spatial distribution properties within the minor region pieces without losing details. In the previous step, we calculated the absolute density values for all region cells, and projected region cell densities to stroke cell spaces. According to the testing, most users can imagine the scenario that spatial points are transformed and compressed into the stroke area. After this transformation, we averaged the density value for each cell according to its neighboring segments to convert the discrete density variation into a continuous weighted value. We use a sliding window m = 50 and a step m' = 7 or vertical and perpendicular segments, respectively.

Algorithm 1 *Auroramap*

Require: A set of points $P = \{p_1, p_2, \ldots, p_k\}$ where p_i is the coordinates for $i = 1, \ldots, k$; number of tangent direction segments x; number of normal direction segments y; size of tangent direction sliding window t; size of normal direction sliding window n; offset distance b; a color scaler SC; ***Optional:*** a boundary B.

Ensure: *Auroramap*

if B is Ø **then**
  $B \leftarrow$ Compute distribution boundary based on P (e.g. user defined or using an algorithm e.g. concave hull)
end if
$B', A \leftarrow B + b$ where $b \neq 0$ ($b > 0$, outside offset; vice versa). B' is the offset boundary. A is the enclosed area between B' and B.
Divide B into y segments to get V, a set of dividing vertices.
$W \leftarrow \emptyset$
for $i \in [0, x)$ **do**
  $v_i \leftarrow$ a point from V
  $c_i, o_i, r_i \leftarrow$ the **maximum** inscribed circle inside C with respect to v_i; the center of the circle c_i; the radius of the circle c_i.
  v_{i+1}, $c_{i+1} \leftarrow$ adjacent point of v_i and its corresponding maximum inscribed circle. Note, v_y is v_0.
  Define a region S_i through the points $(v_i, v_{i+1}, o_i, o_{i+1})$.
  v'_i, $v_{i+1} \leftarrow$ intersections between polygon S_i and offset B'.
  Define a region S'_i from A, cropped by segments (v_i, v'_i), (v_{i+1}, v'_{i+1})
  Equally divide S_i and S'_i into x pieces, along (v_i, o'_i) and (v_i, v'_i) respectively.
  for $k \in [0, y)$ **do**
    $Sum_{ik} \leftarrow$ the number of points that belong to P inside the region S_{ik}.
    $A_{ik} \leftarrow$ the total area of S_{ik}.
    $d_{ik} \leftarrow Sum_{ik}/A_{ik}$ (d_{ik} is the density).
    Append d_{ik} to W_i
  end for
  Append W_i to W
end for
$W' \leftarrow \emptyset$
for each $d_{ik} \in W (i \in x, k \in y)$ **do**
  Apply weighted arithmetic mean (WAM) calculation on all density outputs, using the area A_{ik} as weight factor for WAM. $d'_{ik} \leftarrow \sum_{i-t}^{i+t}(d_{ik} \times A_{ik}) / \sum_{i-t}^{i+t} A_{ik}$
  Normalize d_{ik} with the two sliding windows as size $2t$ in tangent direction and $2n$ in normal direction.
  $d'_{ik} = d'_{ik} \times c \leftarrow$ constant scalar to scale the number up or down.
  Append d_{ik} to W'
end for
Scale W' according to color scaler SC and assign to the set S'.
return *Auroramap*

Figure 7: AuroraMap algorithm pseudocode

There are numerous color encoding scheme options in data visualization including 1) Categorical color scheme which is frequently used to present discrete data, 2) Diverging color schemes as continuous color interpolator and as discrete schemes, 3) Sequential, single-hue color schemes which use hue and shade to present value as continuous interpolators. In the research early phase, we chose single-hue red color ranges to show the density variation. The color encoding is straightforward and intuitive that darker colors present higher density among their corresponding areas than lighter colors. We later transferred to a Red/Blue multi-hued color scheme as it captures density that changes slightly in different places better.

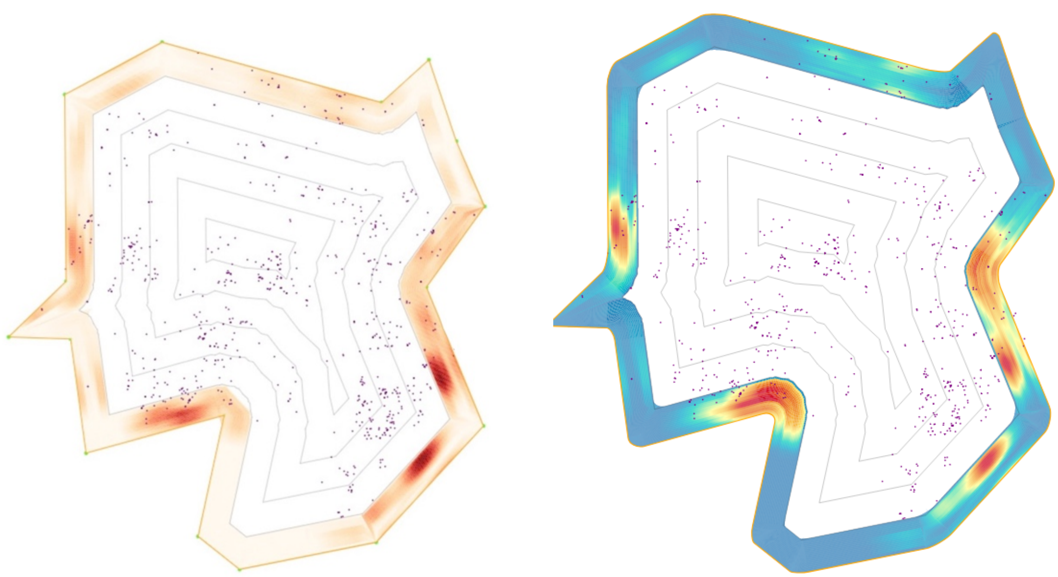


Figure 8: Single-hue color scheme we used for early research phase and the Blue/Red scheme

CHAPTER 4. EXPERIMENTS

We performed a human subject experiment to investigate how users perceive 2D density distributions with AuroraMap. We choose heatmap and dot distribution map as the benchmark visualizations and provide a summative comparison among the heatmap, dot distribution map, and AuroraMap. In the experiment, we investigated the effectiveness of AuroraMap by asking users to perform a series of evaluation tasks.

Firstly, we recorded users' demographic information and educational background. However, we did not record any identifiable information; secondly, we offered a training session to introduce the basic computational algorithm about AuroraMap as area projection, followed by multiple visualization examples of typical distributions constrained within conventional geometric boundaries (circles and rectangular), and one visualization demonstration of a geographical outline visualizing a set of real-world data. Lastly, the users were asked to identify the position and mark the peak density values of circled clusters in the distribution for two given AuroraMaps.

4.1 Participants

We recruited 42 participants with an average age of 23, who were undergraduate students or graduate students at Purdue University. There are 18 female and 24 male participants. The study protocol is approved by the Purdue University IRB Human Research Protection Program. All participants have a normal color vision and are able to perceive color information confidentially (listed as a requirement in the study consent form). All these participants have essential prior knowledge of computer graphics and visualization.

4.2 Color scale and displays

Color is utilized in AuroraMap visualization design in order to convey the attributes of 2D distribution: density values, cluster patterns, and sample locations. We generated heatmap and AuroraMap with a Red-Blue diverging color scheme with continuous interpolators, since it is a widely accepted color encoding scheme method to represent low to high values, and is commonly used in mainstream visualization software including Tableau, d3 in JavaScript, matplotlib in Python, etc. A heatmap in a multi-hued color scale is proven to be more effective in task-driven visualization [60].

In our experiment, we implemented the visualizations for the purpose of training and testing on the participants in a web-based graphical rendering environment. We utilized SVG (Scalable Vector Graphics) as the visual representation format. D3 in JavaScript provides access to a variety of practical color schema options that are derived from ColorBrewer by Cynthia et al. In our experiment, there are two types of visualization trials: 1) six regular geometric boundaries, in which a set of AuroraMap with over 100 points randomly distributed and visualized, 2) three geographical boundaries, in which the religious facilities spatial distribution data (three provinces) are visualized with AuroraMaps. The testing aims at investigating whether users are able to localize the position of point clusters and estimate the maximum density value accordingly. Each visualization trial consists of two parts: 1) Visual representation of AuroraMap, regular geometric boundary visualization was a 3× 3 grid; 2) Color legend with two types of illustrations in a fixed height, one displayed as a set of color blocks with 9 typical values and colors, the other used constant color gradient without value label.

We conducted a pilot study before the formal experiment process to determine a better display method for our experiment. We recruited two participants from Purdue Intelligent Visualization and Interaction lab; both users have normal color perceiving capacity.

4.3 Training

In our experiment, we conducted a training session after collecting participants' basic information and before the testing procedure, as an essential part of the experimental design. Due to the fact that heatmap and dot distribution is widely accepted by the public as an efficient method to present spatial distribution data, some users have prior knowledge in interpreting visualizations such as heatmap and dot distribution. Considering AuroraMap might be new to users, we introduced a training session to help participants understand how AuroraMap works in presenting distribution data within any geographical boundary.

The training session mainly aims at delivering the projection mechanism and color-to-density perception methods in 2D distribution pattern recognition using AuroraMap. There are two major parts in the training sessions, including maximum circle projection demonstration and three visualization perception tasks. We extracted and simplified a geographical outline based on a typical provincial boundary of China, and selected 5 points on the outline with average distance. The users were asked to draw the maximum circles which start at the points individually after users draw on each trial, an experimenter explained to the users how AuroraMap algorithm projects spatial points to the outline spaces, and the grand truth result for the maximum circle was displayed to the users for the betterment of their learning process.

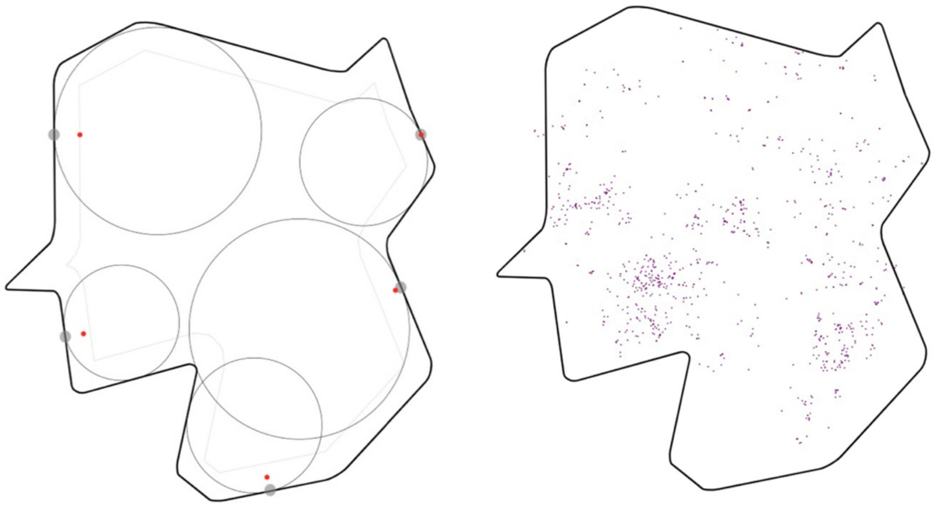


Figure 9: Single Maximum circle projection

Afterward, the participants saw three visualizations, sampled from different data sets, and with varying types of the boundary, they were asked to perform tasks with each visualization trial. We generated distribution density changes inside two types of regular geometric outlines: circle and rectangle. The spatial dots are produced with a normal distribution pattern but randomly using Geopandas in Python. There are three different distribution patterns presented for the first and second visualizations, and we overlapped a dot map with the AuroraMap accordingly. The users can perceive the grand truth visualization together with AuroraMap to help decision-making in the training session. In the third visualization, we chose an irregular geometric outline as a sophisticated example.

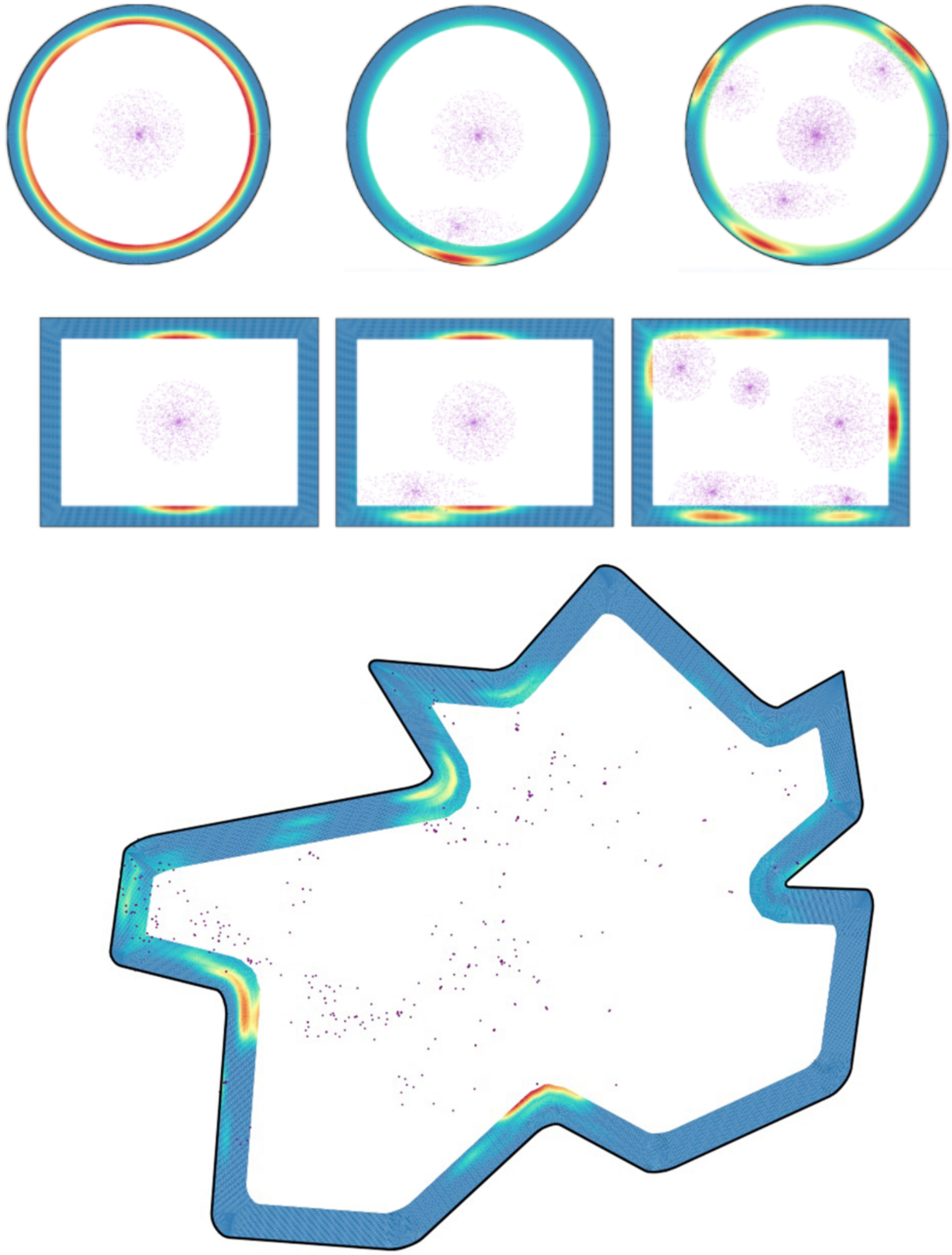


Figure 10: Regular geometric outlines and an irregular geometric outline as a part of training materials

The training process took 30 minutes on an average per user and involved 42 participants. 92% users have significant improvement throughout each training phase; we determined the success of a training session based on each participant's performance over the last training task. We filtered three results due to significant errors made during the training session, collected and analyzed 37 test data.

4.4 Experiment

In the experiment, we designed two representative scenarios that used different geographical boundaries, which enclosed spatial dots sampled from a real-world dataset. We provided two visualizations trials in the experiment procedure, Trial A and Trial B. Each trial presents as an SVG that consists of two parts: 1) an AuroraMap, 2) color legend with two types of illustrations in a fixed height, one displayed as a set of color blocks with 9 typical values, the other used constant color gradient without value labels.

We generated two different distributions with the religious facilities distribution data and China geographic data. We chose two typical provinces in China: Hunan and Yunnan, both contain over 500 religious sites to avoid sparse distribution without regular and identifiable distribution pattern. We then extracted the geolocation of each religious site constrained by two different provinces' political boundaries separately and transformed each site latitude and longitude into a pair of value presented in the SVG space. By simplifying the boundary accordingly to the algorithm discussed in the previous chapter, we generated the contextual visualization with 2D spatial distribution encoded adequately.

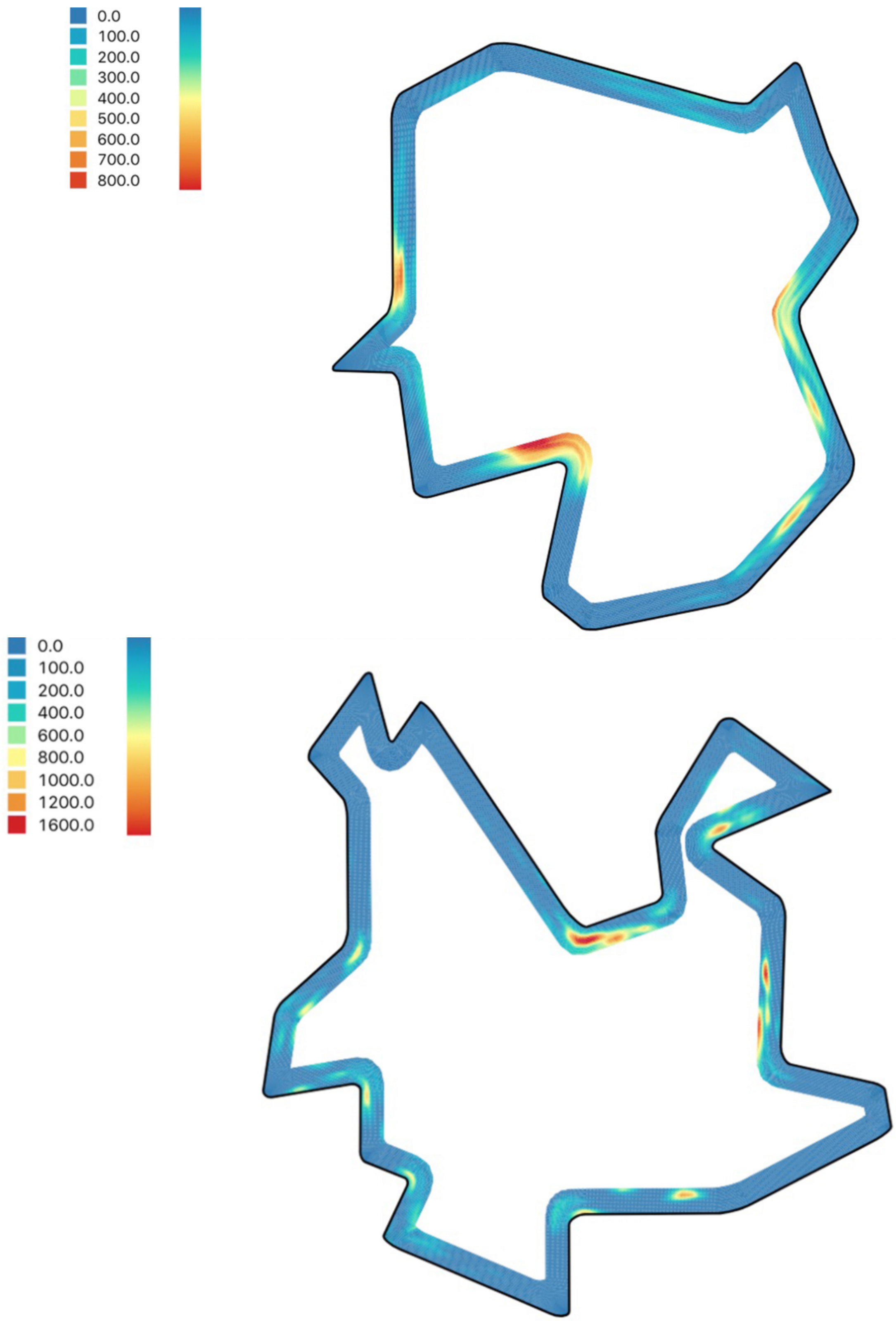


Figure 11: Single Experiment materials

The experiment aims at investigating whether users can identify spatial distribution pattern (density, the centroid of a point cluster, etc.) at different locations, and how densities vary at a particular area (such as points distribute from sparse to dense). The users can imagine a scenario that investigated items (which are religious sites in our experiment) are enclosed in a fixed area where the densities change in different spaces. The participants saw two AuroraMap in a sequence, which is "Hunan" first and "Yunnan" second, due to Hunan has less spatial distribution peak clusters, but Yunnan has more clusters that is harder to perceive. There are two tasks for each scenario, 1) identifying several peak point clusters with maximum density higher than a specific value within the given boundary, 2) estimating the maximum value for each peak point cluster according to the legend. After each trial, we presented the dot distribution map as ground truth visualization, which the participants could refer to in their decision-making process for the second trial. By observing the AuroraMap and legend provided for each visualization trial, the users draw outlines with least perimeters that enclosed several clusters that have a maximum density higher than a specific value. Our experiment also tested the effectiveness of AuroraMap in representing quantitative data. Therefore, we asked our participants to estimate the maximum density as a value according to the legend with value labels.

The experiment process took 15 minutes on an average per user and involved 42 participants. We removed 5 participant's results from analysis due to their extreme error. It seems they do not understand our method correctly. We made a summative comparison about heatmap and AuroraMap, to investigate their performance in revealing spatial distribution patterns by analyzing the position and area of each cluster that users draw and the maximum estimation result.

Figure 12: Participants working on two experiment trials

4.5 Experiment Data Analysis

As the aforementioned procedures of the experiments, we collected 42 participants' responses in total. We filtered 5 participants' data due to extreme error they made during training session, thus, we believed they did not understand AuroraMap visualization. In the experiment session, the participants were asked to determine (free-hand drawing) the clusters inside the distributions, which were visualized by AuroraMaps. Two different AuroraMaps ("Hunan" and "Yunnan") were given to each participant. Considering the boundary of "Yunnan" is more complicated than "Hunan" that could cause more perceptual and cognitive load, all participants were tested on "Hunan" first, then "Yunnan." After testing on each graph, the participants were allowed to check the dot maps overlapping on top of the AuroraMaps for the corresponding distributions. It aims to improve their understanding of AuroraMap. Therefore, the users are expected to perform better on "Yunnan" than "Hunan."

The participants were also required to mark down the peak density values of each clusters circled by the users according to the given color legends. The tests are designated to test whether the users are able to determine two crucial features of the clusters: (1) relative location of each cluster inside the boundary, (2) peak density value of each cluster. There are six clusters for each map and the peak density values are rounded to integers upon the computational results from the algorithm. We used a density-based clustering method, Density-Based Spatial Clustering of Applications with Noise (DB-SCAN) [1], to automatically identify the clusters' positions and outlined the boundary for each cluster.

In Fig. 2, we overlapped the users' drawn areas (highlighting in black) with the algorithm-detected clusters (highlighting in green) and the original dataset (orange dots). Visually judging by the intensity of the darkness, we can comprehend that the users were able to detect the important clusters, especially for those clusters with clear separations. Comparing Fig.2a to Fig.2b, even though "Yunnan" has a fairly more complicated boundary, the users performed better. It may be because the users tested "Hunan" first and gained more experience and a more profound understanding of AuroraMap.

We counted how many clusters are successfully covered by the enclosed curves drawn by each user and their valid rate (covered number versus total clusters, ranging from 0 to 1). As the table shows in 1, participants are able to detect the clusters with fairly precise localization (0.831 ± 0.182). The result also suggests that "Yunnan," which was tested later, performed slightly better. Additionally, we conducted a Pearson correlation [6] to calculate the correlation between user estimations and the correct peak density values for the detected clusters. This statistical method tests the linear correlation between user estimations and the right answers, which ranges from -1 to 1 (1 means positive relationship, vice versa).

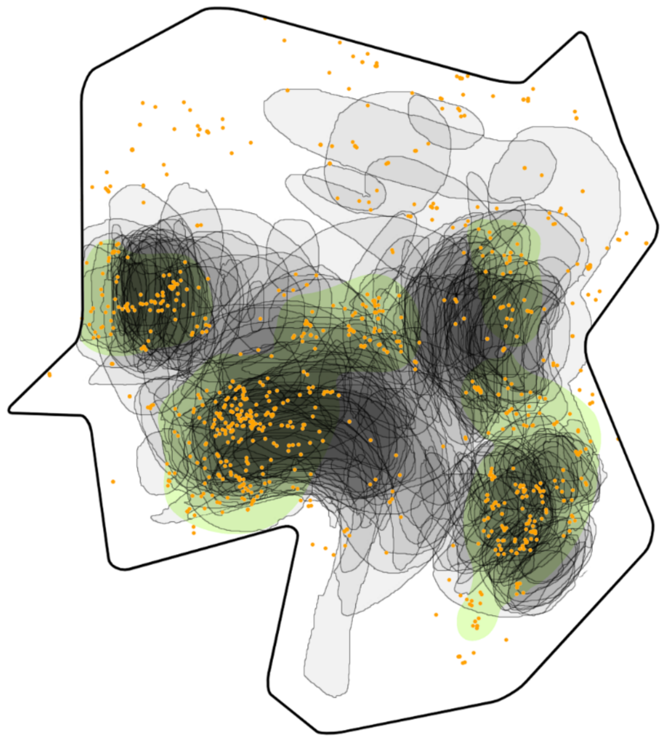
In table 2, we can see most of the users can estimate the answers correctly. Notice p-value is more as a reference since our sample size is less than 500. However, if we observed the coefficient values and set 0.75 as a threshold to filter the data, we can summarize that most of the users (29 and 37 respectively out of 37 in total) can estimate the peak density values accurately with legends provided.

Table 1: Featured Statistical Results for Cluster-Detected Valid Rate

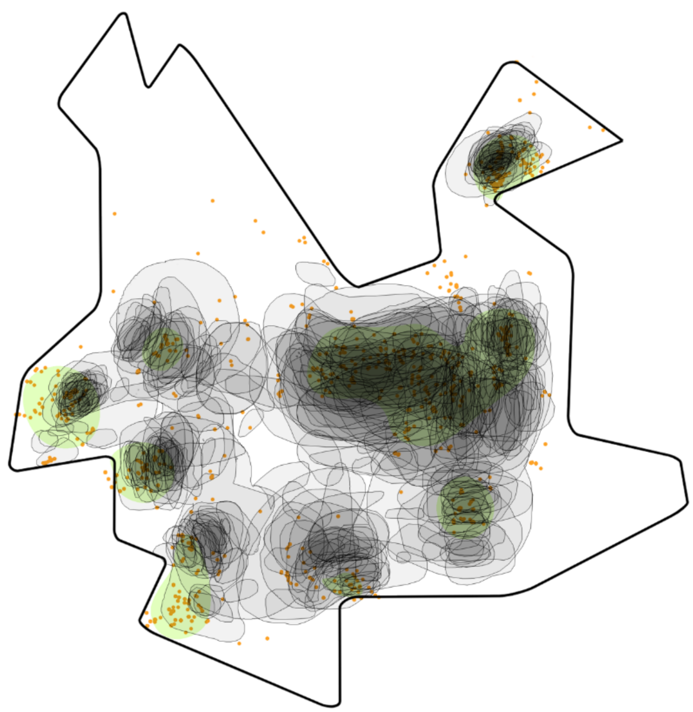
	Hunan	Yunnan	Total
Mean	0.779	0.883	0.831
STD	0.192	0.156	0.182

Table 2: Results for Spearman Correlation Test

	Hunan	Yunnan
P-value (<0.05)	14/37	35/37
Correlation coefficient (>0.75)	29/37	37/37



(a) Hunan Province



(b) Yunnan Province

Figure 13: Overlapping Dot Density Map with User Inputs

CHAPTER 5. CONCLUSION

5.1 Summary

In this thesis, we present a highly versatile abstract visualization technique: AuroraMap for 2D spatial distribution data. Our technique uses a pre-defined boundary to segment a region, count densities in each segment area, and apply a variety to colors to encode density variations. We offset the boundary that yields a certain width that generates a “stroke-width” and later segments the stroke-width into cells that provide spaces for color encoding. In AuroraMap visualization, we utilized a multi-hued color scheme to present density values. Hence, the higher the density among a segment, the darker the cell is. The visualization reveals two essential distribution attributes: 1) density, 2) centroid of density peak clusters.

To validate the effectiveness of AuroraMap, we conducted human subject experiments to investigate whether users can perform well on tasks. The test was approved by the Purdue University Human Research Protect Program; we recruited 42 participants from Purdue University who have normal color perception ability. We started from a training session to help participants understand the fundamental visualization mechanism and filtered 37 participants’ data due to the extreme error they made. We involved 37 participants in the testing phase; they were asked to locate, draw the density peak points within two given regions, and estimate the maximum density value with density value legends given aside. To evaluate the accuracy of the cluster localization, we implemented ground truth visualization using a density-based clustering method DB-SCAN. The DB-SCAN provides a valid reference to determine the accuracy of user-draw clusters centroid and area. We overlapped the area of ground truth visualization clusters and user-draw clusters and calculated a valid rate determined by whether the user-draw area covers the ground truth visualization clusters (ranging from 0-1). To evaluate user estimate density values, we conducted a Spearman correlation by analyzing the p-value and correlation coefficient. From the experiment, we can tell the users can easily perceive and interpret the geographical boundary of the distribution and can accurately estimate the position and orientation of each peak clusters within the region. According to the hypothesis and assumptions we made previously, the AuroraMap effectively presents spatial distribution properties, since 92% users are able to localize and estimate the distribution peak points using AuroraMap; the summative comparisons indicate that AuroraMap is a competitive visualization to traditional visualization including dot distribution map and heatmap.

5.2 Application of Work

AuroraMap is designed for two major visualization issues, multivariate and multiple 2D spatial visualization in one graph. This implementation enables the visual comparison for the users. They could easily perceive the difference between different variables or different distributions. Here we introduce two applications, in which AuroraMap demonstrates its advantages. Additionally, we also demonstrated how to implement AuroraMap for off-screen visualization

problem.

Since data is getting more and more accessible, one data entry could contain multiple variables or attributes other than geospatial information. Therefore, visualizing multi-variate distribution is always demanded. Analyzers expect to get more insights from the visual comparison. In Figure14. we created two offset areas, inside the boundary and outside the boundary respectively for visualizing both Buddhist (inside) and Non-Buddhist (outside) infrastructure locations in Hunan Province. Considering there are a larger amount of Buddhists in China, it is not surprising that we can observe that there are more colors projected to the inner offset area than the outer offset. However, one may realize that at the right side of the AuroraMap, there are more non-Buddhist facilities than the Buddhists'. We can locate it back to the original map and check the specific city. We can even read the legend and get the density difference. Notice the legend could be scaled down to the reasonable domain to represent infrastructure numbers. The users can adjust the legend according to the usage and the data types.

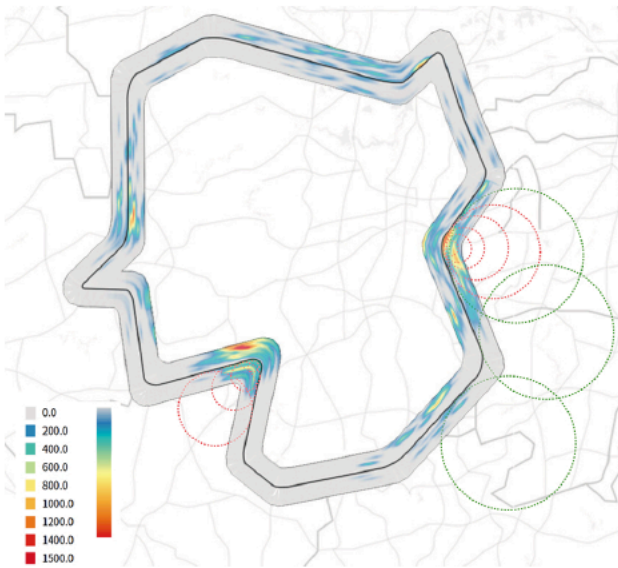


Figure 14: Apply AuroraMap to Visually Compare Two Religions; Two offset areas are visualizing Buddhist and Non-Buddhist infrastructure locations in Hunan Province. Same technique could be applied to resolve off-screen visualization, where the outer offset area visualizes off-screen objects and the inner area shows the samples inside the boundary.

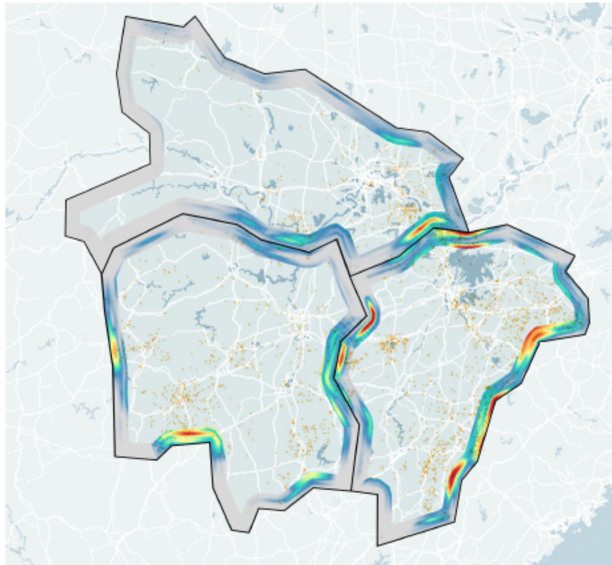


Figure 15: Apply AuroraMap to Visualize Religious Data of Three Adjacent Provinces; Three adjacent boundaries are visualized by AuroraMap. The observers are able to compare the densities between subregions. Top: Hubei; Bottom-Left:Hunan; Bottom-Right: Jiangxi.

Another handful application is visualizing multiple distributions simultaneously. In Figure 15. we applied AuroraMap to a case that three provinces are adjacent. Observing the AuroraMap combination, the users can easily find out the different amount and relative locations of the clusters inside each province. The hues also reveal the peak density values for the clusters. Tracking the sharing boundaries of different provinces, we can realize there are very few religious infrastructures along the provincial boundaries. We can also see the increasing trend from left to right, geographically from west to east. AuroraMap could relief the overplotting issue compared to dot-map. With the legend introduced, the users can easily quantify the relative densities' difference between different clusters, instead of misleading by the overlapping dots.

Due to the restriction of the availability of the digital display property, visualizing off-screen objects and allowing the comparison between inbound objects and off-screen ones are difficult to accomplish. Utilizing the boundary of the distribution, AuroraMap provides a basis where to project off-screen objects to the visible area. In Figure14, assuming that the outer offset area is reserved for the off-screen samples, the inner area can still display the distribution. Here, we suggest a simple method to transfer off-screen samples to the offset area. Similar to the real-world scenario, the users firstly define a scope for spatial visualization. Assuming we take a constant distance away from the boundary, one can see green circumscribed circles in Figure14. in the same diameters. However, the diameters of the circumscribed circles would change while the circles are approaching concave polygons. Like the red circles show, the circles would gradually become smaller till vanishing. For each dividing node on the boundary, we have constructed an associated circumscribed circle. Therefore, we can apply a similar algorithm to compute the densities for sub-regions assembled by two circumscribed circle centers and two dividing nodes on the boundary. Ultimately, the outer offset area can visualize AuroraMap for the off-screen (off-boundary) objects. The users can flexibly adjust the initial diameter of the circumscribed circle so that the considering scope can be redifined.

5.3 Contributions

AuroraMap provides an abstract visualization solution that is highly applicable to visualizing 2D spatial points dataset in the geographical context. This technique has several advantages: 1) yield space compared to other visualization and enables multidimensional data representation. For example, overlaps dot distribution map to the AuroraMap, each presents different raw data to enhance geospatial information, 2) enables users to quantitatively estimate the position, orientation, and relative maximum density value for each peak cluster in raw spatial data, 3) provide more flexible scope for users to view data that presents in different geographical context. For example, users can compare data aggregated inside, and outside a region, and in different geographical levels, such as a nation, state, province, etc.

Except for these following advantages, the visualization algorithm resolves the problem in projecting density values to a sophisticated typology without omitting distribution details. We provided computational methods to color encode spatial data, including 1) extract, simplify and round up a given geographic boundary, 2) implement offset acquisition to yield place for color encoding, 3) region segmentation using maximum circle computation, 4) diverging color scheme encoding on a set of boundary segments.

Compared to other visualization. Phoenixmap encodes density as the width (thickness) of the outlines.The width ensures that users ideally can have a better quantitativeestimation using Phoenixmap compared to color-based methods.However, Phoenixmap is lack of sensitivity to the distribution de-tails, which makes it hard to be implemented for certain tasks likelocalizing the clusters inside the distribution. Heatmap is a density-based method, which could fairly reflect the accurate locations, sizesand densities for a given distribution. However, heatmap requires tooccupy a lot display properties, which restricts heatmap to a singleusage. The users can not engage heatmap visualization with otherprocesses or methods.

In light of a large spatial distribution dataset, AuroraMap allows users to compare distribution patterns in different areas of a region without much effort, and to discern changes in various of geographical scope easily.

5.4 Limitations

Even though the benefits of carrying out Auroramap for 2D spatial distribution visualization, Auroramap is a concise color-based visualization technique. It means that for earning the advantages like saving space, some details are abstracted. It leaves some limitations of Auroramap.

Boundary is a critical base for generating Auroramap. However, when we scale and project a real-world boundary to the display area, often the projected boundary is complex with lots of geometry variations. Computed on the complex boundary, Auroramap may be break into multiple small pieces overlapping together. One way could mitigate negative effect like this is increasing the dividing number of the boundary. In other word, the more divisions we create, the smaller each fan-shaped region and the variation would be, which could reduce the inequality. On the other hand, for those interior angles sunken towards the centroids, we optimally rounded the angles to convert a sharp corner to a small arc because the folding corner could make a sudden color change. The side effect of rounding the corner is that some samples may happen to locate at the corner exactly. In this case, the algorithm has to remove the missing sample points back to the curve, which locate in the center of the curvature.

Additionally, a cluster, whose center locates around the centroids of the distribution, may be break down into multiple projections on the offset area. This requires an advanced understanding and degree of proficiency of implementing Auroramap. The users need to combine them together in use of space imagination. In our user tests, a small batch of the users who volunteered to experience more testing graphs (after experiments) performed very well on assembling the cluster.

Moreover, when applying Auroramap to multi-variate visualization task, the method is limited by the computation and the number of the most allowed offsets. Technically, the offset could be generated layer after layer. However, the mismatch would occur when the offset area steps deeply away from the boundary. Also, the simplification of the boundary will be dramatically increased for deeper offset area, which basically lose the original shape of the distribution.

Judging from the user experiment on Hunan Auroramap, the new users were lack of accuracy about localizing multiple clusters in the same normal direction. This could be alleviated after enough training and understanding about Auroramap. Additionally, if necessary, the users can also utilize interaction like toggle to display the skeleton/centroids of the distribution or the centers of the max-inscribed circles to assist themselves in localizing the clusters.

REFERENCE

[1] Rogowitz, B.E, and L.A Treinish. “Data Visualization: the End of the Rainbow.” IEEE Spectrum, vol. 35, no. 12, 1998, pp. 52–59.

[2] Willmott, Cort J, et al. “Geographic Box Plots.” Physical Geography, vol. 28, no. 4, 2007, pp. 331–344.

[3] Laurent Etienne; Thomas Devogele; Gavin McArdle. “Oriented Spatial Box Plot, a New Pattern for Points Clusters.” Int. J. of Business Intelligence and Data Mining, vol. 9, no. 3, 2014, pp. 233–253.

[4] Dang, G, et al. “Dynamic Queries and Brushing on Choropleth Maps.” Proceedings Fifth International Conference on Information Visualisation, vol. 2001, 2001, pp. 757–764.

[5] Yang, F., & Pettit, J. (2018). Atlas of religion in China : Social and geographical contexts.

[6] A. Cockburn, A. Karlson, and B. B. Bederson, “A review of overview+detail, zooming, and focus+context interfaces,” ACM Comput. Surv., vol. 41, no. 1, pp. 2:1–2:31, Jan. 2009, doi: 10.1145/1456650.1456652.

[7] B. L. Harrison, G. Kurtenbach, and K. J. Vicente, “An experimental evaluation of transparent user interface tools and information content,” in Proceedings of the 8th annual ACM symposium on User interface and software technology, Pittsburgh, Pennsylvania, USA, 1995, pp. 81–90, doi: 10.1145/215585.215669.

[8] D. A. Cox, J. S. Chugh, C. Gutwin, and S. Greenberg, “The usability of transparent overview layers,” in CHI 98 Conference Summary on Human Factors in Computing Systems, Los Angeles, California, USA, 1998, pp. 301–302, doi: 10.1145/286498.286777.

[9] P. Baudisch and R. Rosenholtz, “Halo: a technique for visualizing off-screen objects,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Ft. Lauderdale, Florida, USA, 2003, pp. 481–488, doi: 10.1145/642611.642695.

[10] P. Irani, C. Gutwin, and X. D. Yang, “Improving selection of off-screen targets with hopping,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Montréal, Québec, Canada, 2006, pp. 299–308, doi: 10.1145/1124772.1124818.

[11] “Wedge | Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.” https://dl.acm.org/doi/10.1145/1357054.1357179 (accessed Mar. 20, 2020).

[12] T. Gonçalves, A. P. Afonso, M. B. Carmo, and P. Pombinho, “HaloDot: Visualization of the Relevance of Off-Screen Objects,” 2011. https://www.semanticscholar.org/paper/HaloDot%3A-Visualization-of-the-Relevance-of-Objects-Gon%C3%A7alves-Afonso/0ffdc011ced3e5b2e316fd1cb83a6a4014ef9dc0 (accessed Mar. 27, 2020).

[13] D. Jäckle, F. Stoffel, B. C. Kwon, D. Sacha, A. Stoffel, and D. A. Keim, "Ambient Grids: Maintain Context-Awareness via Aggregated Off-Screen Visualization," in EuroVis, 2015, doi: 10.2312/eurovisshort.20151125.

[14] D. Jäckle, B. C. Kwon, and D. A. Keim, "Off-Screen Visualization Perspectives : Tasks and Challenges," 2015.

[15] J. Chuang, D. Weiskopf, and T. Möller, "Energy Aware Color Sets," Computer Graphics Forum, vol. 28, no. 2, pp. 203–211, 2009, doi: 10.1111/j.1467-8659.2009.01359.x.

[16] J. Dykes and C. Brunsdon, "Geographically Weighted Visualization: Interactive Graphics for Scale-Varying Exploratory Analysis," IEEE Transactions on Visualization and Computer Graphics, vol. 13, no. 6, pp. 1161–1168, Nov. 2007, doi: 10.1109/TVCG.2007.70558.

[17] S. Silva, J. Madeira, and B. S. Santos, "There is More to Color Scales than Meets the Eye: A Review on the Use of Color in Visualization," in 2007 11th International Conference Information Visualization (IV '07), 2007, pp. 943–950, doi: 10.1109/IV.2007.113.

[18] J. Bernard, M. Steiger, S. Mittelstädt, S. Thum, D. Keim, and J. Kohlhammer, "A survey and task-based quality assessment of static 2D colormaps," in Visualization and Data Analysis 2015, 2015, vol. 9397, p. 93970M, doi: 10.1117/12.2079841.

[19] L. Zhou and C. D. Hansen, "A Survey of Colormaps in Visualization," IEEE Transactions on Visualization and Computer Graphics, vol. 22, no. 8, pp. 2051–2069, Aug. 2016, doi: 10.1109/TVCG.2015.2489649.

[20] M. Harrower and C. A. Brewer, "ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps," The Cartographic Journal, vol. 40, no. 1, pp. 27–37, Jun. 2003, doi: 10.1179/000870403235002042.

[21] O. Daae Lampe and H. Hauser, "Interactive visualization of streaming data with Kernel Density Estimation," in 2011 IEEE Pacific Visualization Symposium, 2011, pp. 171–178, doi: 10.1109/PACIFICVIS.2011.5742387.

[22] A. Dasgupta, J. Poco, B. Rogowitz, K. Han, E. Bertini, and C. T. Silva, "The Effect of Color Scales on Climate Scientists' Objective and Subjective Performance in Spatial Data Analysis Tasks," IEEE Trans Vis Comput Graph, vol. 26, no. 3, pp. 1577–1591, Mar. 2020, doi: 10.1109/TVCG.2018.2876539.

[23] S. Silva, B. Sousa Santos, and J. Madeira, "Using color in visualization: A survey," Computers & Graphics, vol. 35, no. 2, pp. 320–333, Apr. 2011, doi: 10.1016/j.cag.2010.11.015.

[24] "Rainbow Color Map (Still) Considered Harmful," 01-Mar-2007. https://www.computer.org/csdl/magazine/cg/2007/02/mcg2007020014/13rRUxYrbOE (accessed Mar. 28, 2020).

[25] B. E. Rogowitz and L. A. Treinish, "Data visualization: the end of the rainbow," IEEE Spectrum, vol. 35, no. 12, pp. 52–59, Dec. 1998, doi: 10.1109/6.736450.

[26] C. Ware, “Color Sequences for Univariate Maps: Theory, Experiments and Principles.” IEEE Computer Society Press, 01-Sep-1988, Accessed: 28-Mar-2020. [Online]. Available: https://doi.org/10.1109/38.7760.

[27] Y. Liu and J. Heer, “Somewhere Over the Rainbow: An Empirical Assessment of Quantitative Colormaps,” in Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Montreal QC, Canada, 2018, pp. 1–12, doi: 10.1145/3173574.3174172.

[28] K. Reda, P. Nalawade, and K. Ansah-Koi, “Graphical Perception of Continuous Quantitative Maps: the Effects of Spatial Frequency and Colormap Design,” in Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Montreal QC, Canada, 2018, pp. 1–12, doi: 10.1145/3173574.3173846.

[29] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, “Visualizing High-Dimensional Data: Advances in the Past Decade,” IEEE Transactions on Visualization and Computer Graphics, vol. 23, no. 3, pp. 1249–1268, Mar. 2017, doi: 10.1109/TVCG.2016.2640960.

[30] W. W.-Y. Chan, “A Survey on Multivariate Data Visualization,” 2006.

[31] Z. Liu, B. Jiang, and J. Heer, “imMens: Real-time Visual Querying of Big Data,” in Proceedings of the 15th Eurographics Conference on Visualization, Chichester, UK, 2013, pp. 421–430, doi: 10.1111/cgf.12129.

[32] “Nanocubes for Real-Time Exploration of Spatiotemporal Datasets,” 01-Dec-2013. https://www.computer.org/csdl/journal/tg/2013/12/ttg2013122456/13rRUytF41z (accessed Mar. 29, 2020).

[33] C. A. L. Pahins, S. A. Stephens, C. Scheidegger, and J. L. D. Comba, “Hashedcubes: Simple, Low Memory, Real-Time Visual Exploration of Big Data,” IEEE Transactions on Visualization and Computer Graphics, vol. 23, no. 1, pp. 671–680, Jan. 2017, doi: 10.1109/TVCG.2016.2598624.

[34] M. Li, F. Choudhury, Z. Bao, H. Samet, and T. Sellis, “ConcaveCubes: Supporting Cluster-based Geographical Visualization in Large Data Scale,” Computer Graphics Forum, vol. 37, no. 3, pp. 217–228, 2018, doi: 10.1111/cgf.13414.

[35] J. Zhang, A. Malik, B. Ahlbrand, N. Elmqvist, R. Maciejewski, and D. S. Ebert, “TopoGroups: Context-Preserving Visual Illustration of Multi-Scale Spatial Aggregates,” in Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, Colorado, USA, 2017, pp. 2940–2951, doi: 10.1145/3025453.3025801.

[36] J. Zhao, X. Liu, C. Guo, C. Qian, and Y. V. Chen, “Phoenixmap: An Abstract Approach to Visualize 2D Spatial Distributions,” IEEE Trans Vis Comput Graph, Oct. 2019, doi: 10.1109/TVCG.2019.2945960.

[37] Chen, Z., & Fu, C. (2014). An efficient, accurate approach to medial axis transforms of pockets with closed free-form boundaries. Engineering with Computers, 30(1), 111-123.

[38] Choi, S., Moon, H., & Wee, N. (1997). New algorithm for medial axis transform of plane domain. GRAPHICAL MODELS IMAGE PROCESS, 59(6), 463-483.

[39] Hsu, K., & Li, S. (2010). Clustering spatial–temporal precipitation data using wavelet transform and self-organizing map neural network. Advances in Water Resources, 33(2), 190-200.

[40] Prasannakumar, V., Vijith, H., Charutha, R., & Geetha, N. (2011). Spatio-Temporal Clustering of Road Accidents: GIS Based Analysis and Assessment. Procedia - Social and Behavioral Sciences, 21(C), 317-325.

[41] Rocha, J., Oliveira, G., Alvares, L., Bogorny, V., & Times, V. (2010). DB-SMoT: A direction-based spatio-temporal clustering method. 2010 5th IEEE International Conference Intelligent Systems, 114-119.

[42] Brewer, C., Hatchard, G., & Harrower, M. (2003). ColorBrewer in Print: A Catalog of Color Schemes for Maps. Cartography and Geographic Information Science, 30(1), 5-32.

[43] Luo, A., Kao, D., & Pang, A. (2003). Visualizing spatial distribution data sets. ACM International Conference Proceeding Series; Vol. 40: Proceedings of the Symposium on Data Visualisation 2003; 26-28 May 2003, 40, 29-38.

[44] Von Landesberger, Tatiana, Brodkorb, Felix, Roskosch, Philipp, Andrienko, Natalia, Andrienko, Gennady, & Kerren, Andreas. (2016). MobilityGraphs: Visual Analysis of Mass Mobility Dynamics via Spatio-Temporal Graphs and Clustering. IEEE Transactions on Visualization and Computer Graphics, 22(1), 11-20.

[45] Karypis, G., Eui-Hong Han, & Kumar. (1999). Chameleon: Hierarchical clustering using dynamic modeling. Computer, 32(8), 68-75.

[46] Vathy-Fogarassy, &., & Abonyi, J. (2013). Graph-Based Clustering and Data Visualization Algorithms (1st ed. 2013. ed., SpringerBriefs in Computer Science).

[47] Mammen, M., Pimgate, C., Koenraadt, C., Rothman, A., Aldstadt, J., Nisalak, A., . . . Riley, S. (2008). Spatial and Temporal Clustering of Dengue Virus Transmission in Thai Villages (Clustering of Dengue in Thai Villages). PLoS Medicine, 5(11), E205.

[48] Eccles, R., Kapler, T., Harper, R., & Wright, W. (2008). Stories in GeoTime. Information Visualization, 7(1), 3-17.

[49] Tominski, C., Fuchs, G., & Schumann, H. (2008). Task-Driven Color Coding. 2008 12th International Conference Information Visualisation, 373-380.

[50] Tonini, M., Pereira, M., Parente, G., & Vega Orozco, J. (2017). Evolution of forest fires in Portugal: From spatio-temporal point events to smoothed density maps. Natural Hazards, 85(3), 1489-1510.

[51] Xiao-Ting, H., & Bi-Hu, W. (2012). Intra-attraction Tourist Spatial-Temporal Behaviour Patterns. Tourism Geographies, 14(4), 625-645.

[52] Kraak, M.J., Longley, P.A, [et al], ..., Faculty of Geo-Information Science Earth Observation, Ut-I-Itc-Stamp, & Department of Geo-information Processing. (2005). Visualising spatial distributions.

[53] Laurent Etienne; Thomas Devogele; Gavin McArdle. (2014). Oriented spatial box plot, a new pattern for points clusters. Int. J. of Business Intelligence and Data Mining, 9(3), 233-253.

[54] Schloss, K., Gramazio, C., Silverman, A., Parker, M., & Wang, A. (2019). Mapping Color to Meaning in Colormap Data Visualizations. IEEE Transactions on Visualization and Computer Graphics, 25(1), 810-819.

[55] Tominski, C, Schumann, H, Andrienko, G, & Andrienko, N. (2012). Stacking-Based Visualization of Trajectory Attribute Data. IEEE Transactions on Visualization and Computer Graphics, 18(12), 2565-2574.

[56] Giesen, J., Miklos, B., Pauly, M., & Wormser, C. (2009). The scale axis transform. Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry, 106-115.

[57] Jäckle, Dominik, Stoffel, Florian, Kwon, Bum Chul, Sacha, Dominik, Stoffel, Andreas, & Keim, Daniel A. (2015). Ambient Grids: Maintain Context-Awareness via Aggregated Off-Screen Visualization. The Eurographics Association.

[58] Smoothing of Multivariate Data: Density Estimation and Visualization. (2009). Biotech Week, 3228.

[59] Brewer, C., & Pickle, L. (2002). Evaluation of Methods for Classifying Epidemiological Data on Choropleth Maps in Series. Annals of the Association of American Geographers, 92(4), 662-681.

[60] Jacquez, G. (2010). Geographic boundary analysis in spatial and spatio-temporal epidemiology: Perspective and prospects. Spatial and Spatio-temporal Epidemiology, 1(4), 207-218.

[61] "Wedge | Proceedings of the SIGCHI Conference on Human Factors in Computing Systems." https://dl.acm.org/doi/10.1145/1357054.1357179 (accessed Mar. 20, 2020).

[62] T. Gonçalves, A. P. Afonso, M. B. Carmo, and P. Pombinho, "HaloDot: Visualization of the Relevance of Off-Screen Objects," 2011. https://www.semanticscholar.org/paper/HaloDot%3A-Visualization-of-the-Relevance-of-Objects-Gon%C3%A7alves-Afonso/0ffdc011ced3e5b2e316fd1cb83a6a4014ef9dc0 (accessed Mar. 27, 2020).

[63] D. Jäckle, F. Stoffel, B. C. Kwon, D. Sacha, A. Stoffel, and D. A. Keim, "Ambient Grids: Maintain Context-Awareness via Aggregated Off-Screen Visualization," in EuroVis, 2015, doi: 10.2312/eurovisshort.20151125.

APPENDIX

RESEARCH PARTICIPANT CONSENT FORM

Enhancing Contour based Visualization for Massive spatial Distribution Data with Color Encoding

Principle Investigator: Yingjie Victor Chen
Researcher: Guojun Han

Computer Graphic Technology
Purdue University

KEY INFORMATION

Please take time to review this information carefully. This is a research study. Your participation in this study is voluntary which means that you may choose not to participate at any time without penalty or loss of benefits to which you are otherwise entitled. You may ask questions to the researchers about the study whenever you would like. If you decide to take part in the study, you will be asked to sign this form, be sure you understand what you will do and any possible risks or benefits.

- The study is about investigating the efficiency of a new data visualization approach. The new approach aims at revealing dot density within a given region, and uses different colors to represent different dot density levels.
- For example, regions with higher dot density has darker color, while those with lower dot density have lighter color.
- The total duration for the research study will be 30 minutes, on February 25th.

What is the purpose of this study?

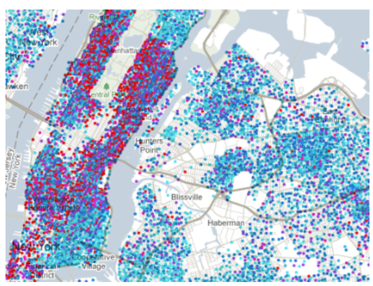
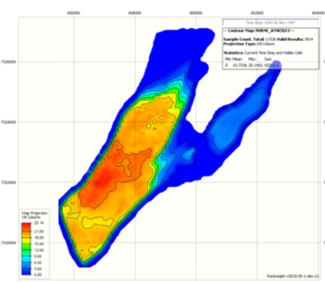
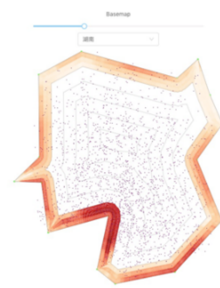
- The purpose of this research study is to measure the efficiency of a new data visualization method, which uses different color ranges to represent dot density in an area.
- To be more specific, we separate an area into multiple pieces and calculate total amount of dots per piece, then use color to represent different dot amounts. For example, a piece with a higher dot amount has a darker color.
- We would enroll 30 participants in this study.

WHAT WILL I DO IF I CHOOSE TO BE IN THIS STUDY?

- The whole study will take you 30 minutes to accomplish.
- Your demographic information and education background will be recorded. However, we will not record any of your identifiable information.
- For your reference, the experiment has a user survey, 2 experimental sessions, followed by a short interview.

1. First, you will fill a brief user survey about your gender, familiarity with data visualization, frequency of using web-based software;

2. Second, you will be briefed about the three types of visualization that are designed to show distribution patterns.
   The three visualization types are: dot distribution map, contour visualization and our visualization approach, please check the form below as reference.

	Dot distribution map	Contour visualization	Our visualization approach
Definition	A dot distribution map, or dot density map, is a map type that uses a dot symbol to show the presence of a feature or a phenomenon.	Contour map is used to determine elevations and are lines on a map that are produced from connecting points of equal elevation.	Our visualization approach uses color to encode spatial distribution patterns within a given boundary.
Example			

3. Then, we set up the software environment for you. A computer monitor and a mouse will be provided, you will need to implement the experimental session in this setting.
4. After that, you will complete two experimental sessions: You will be provided with one contour based visualization that encoded spatial information with color, you will need to circle the area enclosed by the contour that shows certain distribution pattern. Your input and completion time will be recorded; You will be provided with multiple graphs, you will need to compare the distribution density level. Your input and completion time will be recorded.
5. After the two experiment sessions, you will be interviewed about your experience with the contour based visualization.

Interview:
Interview Questions:

1. In the below visualization, identify the position of area which has most dot density;
2. In the below visualization, identify the position of area which has least dot density;
3. In the below visualization, identify the potential pattern you discovered;
4. In the below regions, identify the potential patterns you discovered;
5. In the below regions, identify the region that has most dot density;
6. In the below regions, identify the region that has least dot density.

HOW LONG WILL I BE IN THE STUDY?

- The study will be completed in one session sequentially.
- To estimate completion time for you will be 30 minutes.

WHAT ARE THE POSSIBLE RISKS OR DISCOMFORTS?

THIS STUDY WILL NOT BE ANY DISCOMFORTS OR RISKS FOR USERS TO INTERPRET STATIC VISUALIZATION.

ARE THERE ANY POTENTIAL BENEFITS?

YOU WILL BE PROVIDED WITH INSIGHTS ON HOW TO UNDERSTAND COLOR-BASED VISUALIZATION.

Will I receive payment or other incentive?

You will be compensated $10.

Are there costs to me for participation?

No cost is required for you in this study.

WILL INFORMATION ABOUT ME AND MY PARTICIPATION BE KEPT CONFIDENTIAL?

- Your research records collected for research purposes will be labeled with unique ID and will be stored for 3 years. No Personal or identifiable information will be collected.
- Research results will be stored electronically and analyzed at Purdue University will be kept in a secured area in PI's Office.
- In the event of any publication or presentation resulting from the research, no personally identifiable information will be shared.
- Only the research team will have access to identifiable research records, data, and the purpose of that access.

WHAT ARE MY RIGHTS IF I TAKE PART IN THIS STUDY?

- You do not have to participate in this research project.
- If you agree to participate, you may withdraw your participation at any time without penalty.
- To opt out of participation or withdraw your consent please notify.

Who can I contact if I have questions about the study?

To report anonymously via Purdue's Hotline see www.purdue.edu/hotline

If you have questions about your rights while taking part in the study or have concerns about the treatment of research participants, please call the Human Research Protection Program at (765) 494-5942, email (irb@purdue.edu) or write to:
Human Research Protection Program - Purdue University
Ernest C. Young Hall, Room 1032
155 S. Grant St.
West Lafayette, IN 47907-2114

Documentation of Informed Consent

I have had the opportunity to read this consent form and have the research study explained. I have had the opportunity to ask questions about the research study, and my questions have been answered. I am prepared to participate in the research study described above. I will be offered a copy of this consent form after I sign it.

____ __________________________

Participant's Signature Date

____

Participant's Name

____ __________________________

Researcher's Signature Date