Design Study of LineSets, a Novel Set Visualization Technique

Basak Alper, Nathalie Henry Riche, Gonzalo Ramos, and Mary Czerwinski

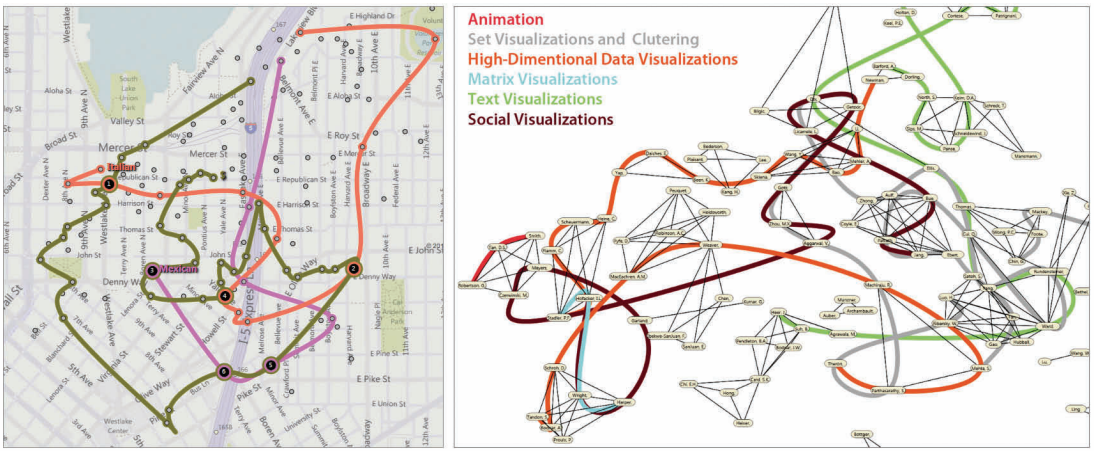

Fig. 1. LineSets showing restaurant categories on a map (left), LineSets showing communities on a social network (right).

Abstract—Computing and visualizing sets of elements and their relationships is one of the most common tasks one performs when analyzing and organizing large amounts of data. Common representations of sets such as convex or concave geometries can become cluttered and difficult to parse when these sets overlap in multiple or complex ways, e.g., when multiple elements belong to multiple sets. In this paper, we present a design study of a novel set visual representation, LineSets, consisting of a curve connecting all of the set's elements. Our approach to design the visualization differs from traditional methodology used by the InfoVis community. We first explored the potential of the visualization concept by running a controlled experiment comparing our design sketches to results from the state-of-the-art technique. Our results demonstrated that LineSets are advantageous for certain tasks when compared to concave shapes. We discuss an implementation of LineSets based on simple heuristics and present a study demonstrating that our generated curves do as well as human-drawn ones. Finally, we present two applications of our technique in the context of search tasks on a map and community analysis tasks in social networks.

Index Terms—Set visualization, clustering, faceted data visualization, graph visualization.

1 Introduction

Organizing and analysing large data collections often involves visualizing data elements in ways that reveal their properties and relationships. For example, biologists seek to understand the relationships between groups of genes in the human genome; social scientists study the interactions between people through the identification of communities in social networks and machine learning experts try to understand how their data has been categorized. Often, visualization tools are used to better explore this data. There are many types of visual representations of sets, each of which influences how people perceive an element's properties and relationships to those around it.

One of the most common representations of sets is the Euler or Venn diagram [3][7]. Both types of diagrams are limited to the use of simple convex shapes such as ellipsoids or rectangles. However, the principles of these representations can be generalized to more complex shapes such as arbitrary concave hulls that enclose elements sharing a particular attribute [7] [16]. While often effective, these set representations can become cluttered when many such sets intersect. In these cases, parts of the representation can become difficult to read [15]. Simonetto and Collins use color, transparency and texture to better convey the connectivity of spatially fragmented sets and help users to distinguish between different sets and their overlapping regions [4][17]. However, it is unclear whether these variations on the fill properties of set shapes alleviate the reading and comprehension difficulties that arise when many sets intersect.

To limit the visual clutter and increase the readability of complex set representations, we propose the use of geometrically continuous lines, called LineSets (Figure 1). Each set is represented with a curve, connecting all of the set elements. LineSets minimize the clutter of intersecting sets by producing line crossings instead of geometry overlaps. To better understand the advantages and drawbacks of this concept, we performed a quantitative user study assessing its readability compared to state of the art, Bubble Sets [4]. Results show that LineSets help users for certain tasks where traditional set representations do not scale well. We provide an implementation of LineSets using simple heuristics validated through a second user study. Finally, we present two applications of

- *Basak Alper is with Microsoft Research and UC Santa Barbara, E-Mail: basakalper@umail.ucsb.edu.*
- *Nathalie Henry Riche is with Microsoft Research, E-Mail: Nathalie.Henry@microsoft.com.*
- *Gonzalo Ramos is with Microsoft Research, E-Mail: gonzalo@microsoft.com.*
- *Mary Czerwinski is with Microsoft Research, E-Mail: marycz@microsoft.com.*

Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

1077-2626/10/$26.00 © 2010 IEEE Published by the IEEE Computer Society

LineSets in the context of search tasks on maps and community analysis in social networks.

2 Related Work

We use the term "set" to indicate a group of elements sharing a given property, present in the data or generated by an algorithm. We use the terms set, group, cluster and categories interchangeably.

Sets are common data structures in information visualization, especially when analyzing categorical data. Several works have focused on representing categorical data with frequency-based representations. TreeMaps [16], Parallel Sets [11] are examples of such techniques. They represent the frequency distribution of all data elements across categories. However, the membership of individual elements is not explicitly visualized. Commonly used Venn (or Euler) diagrams, bounding convex geometries (often ellipsoids) enclosing all members of a set, have their limitations.

Several variants exist that differ in their definition of the set region shapes allowed. Depending on the constraints applied to the drawing, each technique has different instances that cannot be drawn [6][7][18]. For example, discontinuous sets or sets intersecting multiple times cannot be represented using shapes such as ellipses. Moreover, Venn diagrams are not sufficient in cases where there is a need to not only have a global sense of a set's size and supporting area, but also of the distribution and density of its elements. In addition, adequate space needs to be available to display both the elements and the sets. For example, in the case of several sets sharing multiple elements, the intersection area needs to be large enough to draw all elements.

A common approach to set visualization is to spatially arrange set members such that each set forms a spatial cluster [21]. This method is especially popular in detection of communities when performing social network analysis [14]. These communities are often indicated by overlaying convex hulls [9] and are usually drawn as an additional layer over the social network. The use of convex hull set representations can be problematic for scattered or fragmented data sets. In particular, when representing points of interest on a map, a hull is likely to misrepresent the true support area of a set since it covers a larger region than necessary, e.g., a set contains four elements located in the four corners of a map generates a hull that covers the whole map, and thus, erroneously intersects all possible sets on that map.

Recent work in information visualization has explored these issues and the use of concave boundaries for drawing set regions. Simonetto et al. [17] computes polygonal shapes and Bubble Sets [4] computes smooth bubble-like shapes that can be drawn over arbitrary element layouts. Although these approaches scale well for arbitrary distributions and numbers of sets, they do not scale well for a larger number of set intersections. When a large number of sets intersect, the resulting set shapes and overlaps can become quite complex, and interpreting such geometries with occlusion becomes challenging. In order to overcome set membership ambiguities, Simonetto et al. manipulates the color, transparency and texture of the displayed sets. However, low-level tasks such as assessing the set membership of an element or counting the number of elements of a set still remain error-prone when many sets intersect (Figure 2).

The Bubble Sets technique presented in [4] does not consider cases where an element belongs to multiple sets. However it may still produce geometries that overlap each other when elements are scattered in space (Figure 2, middle). In addition, although the algorithm strives to minimize false inclusions, in some cases sets may still appear to include elements from other sets. Like previous techniques, identifying distinct sets becomes difficult (Figure 2) when many regions intersect. To improve some of these shortcomings and convey set regions as continuous units, Bubble Sets utilizes smooth line contours, color and transparency. Interactivity also helps disambiguating overlaps as users can bring a given set to the foreground of the visualization. However, when applying the technique to multiple intersecting sets, we show through a user study that it is still challenging to perform simple tasks such as identifying the set membership of an individual element.

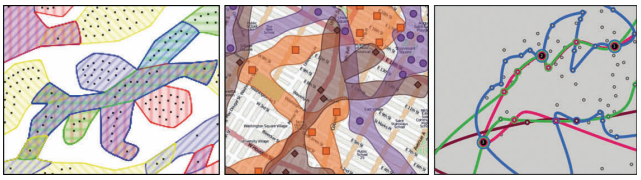

Fig. 2. Close up view of Simonetto et al. technique (left), Bubble Sets (middle) and LineSets (right) when many sets intersect.

To our knowledge, very few studies focus on the readability of the aforementioned set visualizations. Benoy [1] studied how the smoothness of the line contour, the sizes of the set regions and the closeness of contours affected the complexity and the readability of Euler diagrams, yet his results did not draw clear conclusions from these factors. A recent user study from Riche and Dwyer [15] showed that when multiple sets intersect, concave overlapping boundaries were difficult to assess. In this case, Riche and Dwyer proposed duplicating elements, an approach that also suffers from clutter when the number of elements to duplicate is large. In this paper, we present LineSets, a novel technique that attempts scaling for more complex set arrangements by improving the readability of set intersections and avoiding shape overlap.

3 LineSets

3.1 Design Goals

Our main motivation when designing LineSets was to support simple readability tasks where other techniques do not scale well (*i.e*, when many sets intersect with each other). We focused on four goals:

Allow users to efficiently identify elements belonging to a set (G1). We choose to represent each set by a single continuous curve connecting all its elements. This choice was inspired by results from Gestalt theory [19][13] and from the perception of graphs' paths [20]. From these works we know that connecting elements with a line is a strong visual cue for association, and that making this line continuous and smooth helps users visually traverse the path of connected elements easily. In addition, we believe that the global shape of LineSets has the potential to serve as a "signature" allowing for better recollection and identification of individual sets.

Allow users to efficiently identify the set membership of each data element (G2). There are several ways to connect a set of points, e.g., using tree-like or graph structures, and while these can minimize the overall line-lengths drawn, we chose to use a single connecting line because it provides a direct method for both scanning and browsing a set of elements. This quality can improve not only the identification of all elements of a set, but also users' ability to identify outlier elements that otherwise could be missed.

Allow users to identify how sets intersect with each other (G3). By using curves instead of two-dimensional surfaces, the intersection of sets is reduced to a common point or node, which provides a clear and compact representation.

Provide a visual metaphor that allows users to interact with a set and its elements (G4). Interactivity is key when analyzing large datasets. The linear design of LineSets provides a sequential arrangement of set elements, which facilitates browsing and filtering (as it resembles the familiar list representations). Since the shape of the curve can be constrained (e.g., transforming it into a straight line

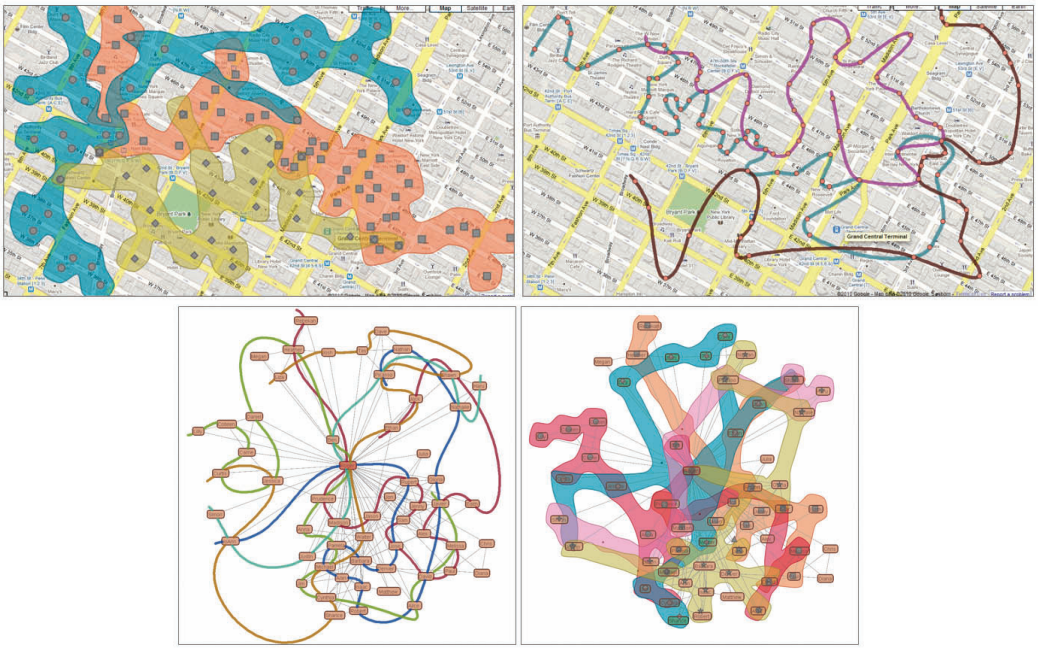

Fig. 3 Stimuli images from our study comparing LineSets and Bubble Sets on the medium size map (top) and difficult social network (bottom) datasets.

or a circle), LineSets also provide an effective way to manipulate the spatial organization of the set elements.

3.2 Design Methodology

Our approach to the design of LineSets differs from the general methodology used by the InfoVis community. Similar to the graph layout problem, the number of criteria to take into account when drawing a single curve connecting all the set elements can be large. For example, Brandes et al. [2] present some mathematical considerations for drawing paths in hypergraphs such as the monotony or planarity of the path. However, they do not discuss the quality and impact of these criteria on readability. The implementation for drawing these curves can also impose some constraints.

To assess the potential of the technique without any constraints, we decided to generate sketches of the concept manually. We decided on a subset of them, exhibiting interesting aesthetic properties and performed a controlled experiment, comparing these "ideal" hand-drawn visualizations to actual results from state-of-the-art, Bubble Sets [4]. We then implement LineSets based on simple heuristics and evaluate its effectiveness with a follow up user study. Finally, we demonstrate a prototype application of the technique for visualizing sets of restaurants on maps and social networks.

4 Study 1: Evaluating LineSets' Readability

The details of our controlled experiment are as follows. We asked 12 users to answer readability questions for both representations for varying data set complexity and size in two contexts (map and social network). We collected time, error and user preference. Our experiment was a within-subjects design:

2 Visualizations (LineSets, Bubble Sets) × 2 Data type (map, social network) × 3 Difficulty levels (number of elements, sets and intersections) × 4 Tasks of varying complexity

4.1 Participants and Apparatus

We recruited 12 researchers (6 females) from within a large technology research organization. The age of our participants ranged between 24 and 31 years, with a mean age of about 27 years. The study computer was a 3Ghz Dual-Core equipped with a standard 19-inch screen of resolution of 1280 x 1024 pixels.

4.2 Visualizations and Datasets

We selected two types of datasets: hotels and social networks. For each, we defined the level of difficulty by the number of sets, set sizes and the number of intersecting sets. The easy dataset had 50 elements, 3 sets and 5 intersections, the medium dataset had 100 elements, 4 sets and 10 intersections, and the difficult dataset had 200 elements, 5 sets and 30 intersections (Figure 3).

For each dataset, we generated two static pictures visualized with each technique using Photoshop. For the Bubble Set technique, we generated the pictures using the implementation generously provided by the authors of the Bubble Sets as described in [4]. All of the materials for the LineSets visualization were manually prepared using Photoshop. While creating the paths, we aimed to minimize the length and curvature and avoid self-crossings. To ensure that our tasks were isomorphic with both LineSets and Bubble Sets, we used the same data. However, to avoid any memorization, we rotated the datasets horizontally between techniques.

4.3 Tasks

We selected four generic tasks in an attempt to capture both overview (e.g., "how many sets?") and detail questions (e.g., "which sets does a particular element belong to?"). Tasks and associated examples are listed in Table 1.

4.4 Procedure

We performed the study with one participant per session; each session lasted about 60 minutes. A session was divided into four phases. During phases 1 and 4, we gave the participants color pencils and sheets of white paper with set elements printed on it. We labelled elements with numbers indicating which set(s) they belonged to. Participants were asked to draw sets as well as they could. Before the controlled experiment, the experimenter gave participants a tutorial explaining LineSets and Bubble Sets to make sure that participants understood the concepts. Next, participants used a computer system that showed images corresponding to datasets represented with either Line or Bubble Sets and asked a question about it. The system then

Table 1. Tasks used in our controlled experiment

	Task Type	Task Text
T1	Overview: number of sets	"How many groups of hotels are shown?"
T2	Overview: size of a set	"Which one is tagged more in users profiles, Matrix or Pulp Fiction?"
T3	Membership	"Which bands do Alan and Tim both like?"
T4	Intersection	"How many hotels have free parking and breakfast?"

recorded the time taken to complete a task and the accuracy of the answers provided. Participants answered a multiple-choice questionnaire of 24 questions per technique.

The order of tasks was fixed, from easy to difficult. To account for learning effects, we counter-balanced the order of the visualizations. Other conditions (data type and complexity) were randomized to control for memorization. After completion of all tasks, participants filled out a satisfaction questionnaire. They were then debriefed and provided with software gratuity.

4.5 Hypotheses

(H1) Bubble Sets will outperform LineSets in completion time for overview tasks (T1, T2). We believe that LineSets require more time for these types of tasks, as users have to scan across the entire curve.

(H2) LineSets will outperform Bubble Sets in accuracy and completion time for set membership and set intersection tasks (T3, T4). We believe that LineSets' footprint is smaller than Bubble Sets', which should make set intersection more readable.

4.6 Results

We considered the most important measure for evaluation to be accuracy. This is because it does not matter if participants perform twice as fast with a given visualization if in the end the analysis of their data is inaccurate. We used a 2 Visualizations (LineSets, Bubble Sets) × 2 Data type (map, social network) × 3 Difficulty levels (easy, moderate, difficult) × 4 (Tasks T1-T4) Repeated Measures Analysis of Variance (RM-ANOVA) to analyse the accuracy and completion time results using a within subject design. We used the logarithm of the task times to normalize the skewed distribution, as is standard practice with reaction time data. Post-hoc comparison reported are significant at the p<.05 level.

4.6.1 Accuracy

On average, 84% of the answers were correct (SD=4). We found a significant effect of accuracy for Technique, F(1,11)=13.5, p=.004. Overall, participants had 88.5% (SD=1.9) accuracy with LineSets and 80% (SD=2.7) with Bubble Sets. It is interesting to note that only one participant had more accurate answers with Bubble Sets (95.8%) than with LineSets (87.5%). Our analysis showed a significant effect of accuracy for Task, F(3,33)=3.132, p<.04, Data type F(1,11)=9.52, p<0.01 and Difficulty, F(2,22)=12.34, p<.0001. Unsurprisingly, as the difficulty of tasks and datasets increased, participants committed more errors. Post-hoc comparisons also showed that participants made significantly more errors with the map than with the social network.

RM-ANOVA also revealed significant interactions Technique x Task, F(3,33)=3.64, p<0.02 and Technique x Difficulty, F(2,22) = 13, p<0.001. Post-hoc comparisons did not show any significant difference between techniques in accuracy for tasks T1, T2, and T4. However, they revealed a significant difference between Techniques for task T3 - comparing set membership of elements. For this task, LineSets were 25% more accurate than Bubble Sets. On average, 88.9% (SD=3.7) of the answers were accurate with LineSets and 66.7% (SD=5) with Bubble Sets. Post-hoc comparisons also revealed that LineSets were 25% more accurate than Bubble Sets in the case of difficult datasets (datasets with more intersections). On average 86.5% (SD=3.9) of the answers were accurate with LineSets and 65.6% (3.8) with Bubble Sets. (See Figure 4, right).

4.6.2 Completion Time

On average, tasks took 15.3 seconds (SD=4) to complete. We found a significant effect for Technique, F(1,11)=5.09, p=.045. Participants took an average of 13.4 seconds (SD= 3.8) with LineSets and 17.1 seconds (SD=3.3) with Bubble Sets. Our study revealed a significant effect for Tasks, F(3,33)= 92.6, p<.0001 and Difficulty, F(2,22)=136.58, p<0.001. Post-hoc comparison showed that LineSets were about 20% faster than Bubble Sets. Unsurprisingly, as the difficulty of tasks and datasets increased, participants required more time to answer.

RM-ANOVA also showed an interaction between Technique x Task F(3,33)=15.98, p<0.001. Post-hoc comparisons revealed no significant difference between Techniques for the overview tasks T1 and T2. However, they revealed a significant difference for Task 3 – comparing set member-ship of elements and Task 4 – identifying set intersections. For T3, participants performed about 25% faster with Line-Sets than with Bubble Sets (resp. 17.9s (SD=1.5) on average with LineSets and 23.6s (SD=2) on average with Bubble Sets). For T4, participants performed about 40% faster with LineSets than with Bubble Sets (on average 13.8s (SD=1) with LineSets and 23.5s (SD=1.5) with Bubble Sets). (See Figure 4, left).

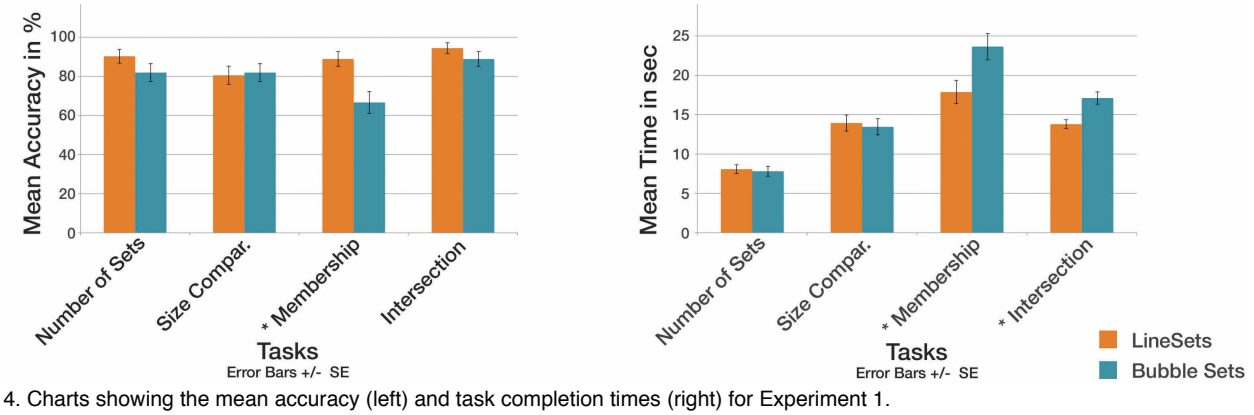

Fig. 4. Charts showing the mean accuracy (left) and task completion times (right) for Experiment 1.

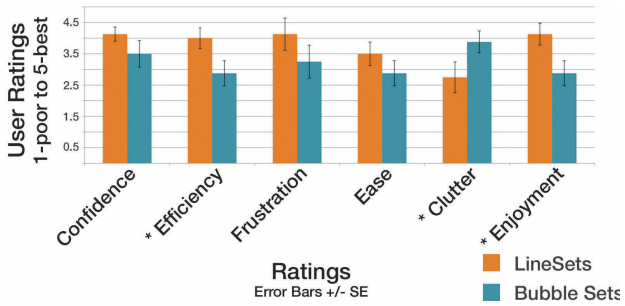

Fig. 5. Subjective user ratings from the readability study.

4.6.3 User Preference

Users filled out satisfaction surveys after completing all tasks. Users rated their satisfaction scores on multiple criteria: preference, confidence, accuracy, and frustration, using a Likert scale from 1 to 5. Figure 5 provides the average ratings for each technique. Using RM-ANOVA, we found that LineSets rated significantly higher than Bubble Sets ($F(1, 11) = 28.3$, $p<0$). In addition, 11 out of 12 participants preferred LineSets. A single participant preferred Bubble Sets. It is interesting to note that this was the only participant who performed more accurately using the Bubble Sets technique.

We also collected written comments from our participants. Eight participants expressed that LineSets were better for identifying intersections of sets and looking for common attributes between two elements. Ten participants expressed Bubble Sets as not being suitable for these tasks.

Participants were divided over the size comparison task: 6 expressed their preference for LineSets, 6 for Bubble Sets. We observed that participants who actually counted the set elements preferred LineSets, while participants who tried to estimate the size of sets preferred Bubble Sets. Two participants expressed that Bubble Sets gave them a better sense of "grouping", yet one still preferred LineSets for the tasks they performed in the experiment.

4.6.4 Sketch Results

We asked our participants to sketch sets prior to and after the experiment. Our goal was to discover the types of visual set representations the participants produced. The experimenter did not guide them towards any particular representations. By repeating the drawing exercise at the end of the experiment, we wanted to observe whether the participants would incorporate the Bubble Sets or LineSets technique into their repertoire.

The analysis of the sketches produced before the study revealed that six participants used only color to indicate set membership. Three participants used straight lines connecting elements along with color. Two participants used continuous smooth concave enclosing geometries (similar to Bubble Sets). One participant used a hybrid of enclosing geometries and connecting straight lines.

The analysis of the sketches produced after the experiment revealed that only two participants persisted in using only color to indicate set membership. Two participants used enclosing geometries along with texture. Four participants used lines similar to LineSets. Four participants used enclosing geometries and connecting lines in concert. They enclosed elements spatially grouped in simple concave geometries and used lines to connect outliers to the rest of the elements (Figure 6).

4.6.5 Discussion

Results from our controlled experiment revealed three things: a) participants were able to both understand and use LineSets effectively as a visualization technique, b) Participants performed differently with LineSets and Bubble Sets, and c) participants rapidly

Fig. 6. Examples from post experiment sketch studies utilizing bubble like geometries (left), LineSets (middle) and hybrid solutions (right).

incorporated the idea of LineSets as a way to represent collections of related elements. We had hypothesized that Bubble Sets would perform better in completion time for overview tasks (H1). While participants commented that Bubble Sets provided a better sense of grouping, our quantitative results did not indicate significant differences between techniques in accuracy or completion time (H1). Against our expectations, these results indicate that for the presented tasks LineSets seem to perform as well as Bubble Sets.

Our results confirm (H2): LineSets outperform Bubble Sets for set membership and set intersection tasks. This result validates two of our design goals: facilitating the identification of set membership (G2) and assessing set intersections (G3). Subjective ratings confirmed that most participants (11 out of 12) preferred LineSets.

Results from the sketching session indicate that participants integrated the concept of LineSets in a short period of time. It is encouraging to observe the ease with which most of our participants integrated and used LineSets into their expressive repertoire. The fact that four of them drew hybrid representations of Bubble Sets and LineSets opens interesting future directions that we discuss in the general discussion section of this paper.

5 An Implementation of LineSets

Our experimental results encouraged us to find a way to algorithmically compute LineSets. Given an arbitrary distribution of points in the space, there are many ways in which to draw a line visiting all points at once. The main criteria we identified were geometric simplicity (linearity) and smoothness, in order to enable users to follow the path easily [20] and support recall. Geometric simplicity requires elimination of self-crossings and minimization of bends. One heuristic we considered was to start with fitting a linear curve to the general distribution of the data and then locally modifying the curve so that it goes through all the data points. Soon we realized that such an implementation generates long curves with a lot of zigzags. Trying to eliminate these bends approaches computing the shortest path. We adopted the Lin-Kernighan's travelling salesman heuristic (LKH) [12] in our current implementation for its close to real-time computation time for the given data size. Our formal and informal observations with this implementation generated relatively simple paths with little or no self-crossing effects. In order to ensure geometric smoothness, Lines were drawn using piecewise Bezier splines with virtual control points to make sure that the spline visits all set members. For each element that is required to be traversed by the LineSet, we computed two control points with continuous second and first order derivative constraints.

Elements on a LineSet are represented as circular nodes, in a visual style inspired by subway map representations [10]. As two or more LineSets intersect, we decorate the participating elements with concentric rings color-coded corresponding to the color key of the participating sets (Figure 9, bottom right). We allow LineSets to have a selected and deselected state. In their deselected state, a LineSet is shown as a thin line to reduce clutter on the canvas. When a LineSet becomes selected, e.g., by a user clicking over it, it grows in width and makes it salient in comparison with unselected ones.

6 Study 2: Evaluating Paths for LineSets

As noted before, it is possible to draw the path connecting set elements in many ways. Although we believed LKH yielded acceptable results, we wanted to investigate further how human drawn paths differ from paths generated by LKH, and in what conditions people prefer human drawn paths to computer generated ones. To answer these questions and as a first attempt to characterize aesthetic criteria for optimal paths, we designed a two-stage study.

6.1 Stage 1: Generate Paths

In the first part of the experiment, we gave users 15 sheets of letter size papers, each with a certain number of gray points on them. The simplest had 9 points, and the most crowded had 50 points. We used the map dataset from the first study. The users had no knowledge about the meaning of the data.

We recruited 6 participants (2 females) between ages 24 and 34 with a mean age of 29. We instructed them to connect all the points on the paper with a single continuous path using a pen. We told them to produce paths that, in their opinion, were "simple", "aesthetic" and felt easy to follow. We ordered the data sets from simple to more complex and gave them to all the users in the same order. The presentation order was intended to progressively improve participants' drawing skills. In the complex data sets, the main challenges were number of nodes and distribution, uniform distributions being more challenging.

Subjects completed the tasks in 25 minutes average. All subjects revised their drawings at least once. Two out of six participants missed an outlier on a medium size data set, we asked them to correct their drawing and incorporate the dot they missed in the final drawing. After they completed all the drawings we asked them if they were pleased with their drawings, and whether they thought it could be drawn better. All six of them were satisfied with their drawings. Two of them indicated that they were not confident about the drawings for crowded data sets. Four of them indicated that there was room for improvement.

None of them could give a precise definition of the strategy they adopted for drawing the curves. Three of them commented that starting from outliers helped creating simpler paths. Two of them indicated that they aimed for minimizing the curvature of the path by avoiding sharp turns.

6.2 Stage 2: Evaluate Paths

In the second part of the study, our goal was to compare user-generated LineSets to the computer generated ones using LKH. We collected 90 drawings from 6 subjects for 15 datasets in the previous stage. We scanned these images (Figure 7, left) and recorded the order of points visited by each user. We used this order to generate the LineSets using Bezier splines. (Figure 7, middle). For each dataset, we also generated the LineSets using the LKH heuristic (Figure 7, right).

We recruited 8 participants (3 females) with ages 22-35 (mean of 27.4) and provided them with experimental software displaying

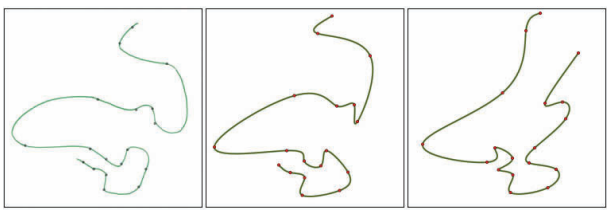

Fig. 7. Path drawings for a data set of 18 points (D15). (Left) human drawn path, (middle) digitized version of the path, (right) shows the computer generated path for the same data set.

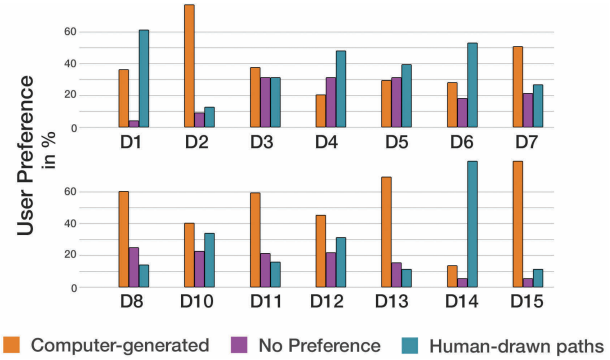

Fig. 8. User preference results from the second stage of Study 2 for all data sets.

one of the user-generated digitized drawing adjacent to the computer-generated one for the corresponding dataset. The subjects were asked to pick one of the drawings by clicking 'Left' or 'Right' buttons displayed on top, or state a lack of preference by clicking the 'No Preference' button displayed on the top centre of the screen. We told them the decision criteria were "simplicity", "aesthetics" and what felt easy to follow. The subjects answered 90 questions in about 15 minutes. We randomized the left right order of the computer and user-generated paths and the order of datasets.

6.3 Study Results and Discussion

Overall, users preferred computer-generated paths 44.3% of the time, user-generated paths 32.9% of the time, indicating no preference 22.7% of the time. We analyzed the data using Pearson Chi-Square tests. Figure 8 summarizes the preference results for each dataset. Users significantly favoured computer-generated path for datasets 2 and 15, and user-generated ones for dataset 14. Since there were not enough data sets with significant differences, we could not formally identify criteria for optimal LineSets. However, we made two notable observations: a) people prefer straight lines to circular paths; and b) a circular path is preferred to a zigzagging path. Both these observations validate our initial criterion of geometric linearity for LineSets. It is important to note that LKH exhibits a tendency to make circular paths for uniform like distributions where users are able to produce more linear paths. Indeed, these were the cases where human generated paths were preferred.

Our understanding of an optimal heuristic for LineSets is a shortest path algorithm with an additional constraint of linearity. In some cases, linearity and the shortest path might be contradictory. A weighted heuristic favouring linearity over shortest path should be evaluated extensively. Also, such an algorithm can become computationally very costly. Given all these considerations, we concluded that LKH is a reasonable heuristic to compute LineSets. Thus, we used it in our subsequent two prototypes.

7 Applications

Among the factors that impact a set representation, one is the possibility to adjust the spatial layout of the data elements. For example, the location of points of interest on a map (Figure 9) should not be modified to improve the representation of the existing sets as it would destroy an important set of properties. Conversely, when representing a social network such as the one depicted in Figure 10, right, the nodes' position can be adjusted, as their location in space has no direct semantic meaning. We support this interaction by drag-and-drop. We applied LineSets to maps and social networks, as they are fair representatives of many types of dispersed point visualizations.

7.1 LineSets on Maps

Maps are representations that often integrate multiple layers of dense information including colors, textures, icons, glyphs, labels encoding roads, points of interest, topological features, etc. Visualizing sets of geo-located elements on a map can introduce visual clutter, in particular when elements within a small set are located far apart. In order to investigate the benefits of using LineSets in the context of set of elements on a map we created a prototype application for exploring and searching for restaurants.

Our system uses a dataset of 120 restaurants from downtown Seattle. We collected the data using the Yelp for Developers API [21]. The restaurants are categorized into 13 types of cuisine such as Indian or American, 4-point price ranges, and 5-point average rating. Overall, there were 21 sets defined, many set intersections occurred, and each restaurant always belonged to three sets.

7.1.1 Interface

Our prototype's interface consists of three panels (Figure 9): 1) a map canvas displaying set elements in their location, 2) a category selector region where users can turn categories on and off, and 3) a list displaying the elements users select through the map and category selector areas.

Users select a set by clicking on its label in the category selector region. This action makes the corresponding LineSets selected, i.e., the LineSets appear on the map and the list area shows all restaurants belonging to the set. We also added labels to the end point of LineSets on the map. When users select multiple sets, corresponding LineSets appear on the map. Concentric rings indicate their intersection. Restaurants matching all criteria are shown using black numbered circles, those that do not match all criteria are indicated by small grey circles. Figure 9, bottom right, shows close-up figure of the concentric rings. The list shows the elements from the selected sets as well as flattened version of the selected LineSet(s) on its left margin (Figure 9, upper right). This flattened LineSets representation aims at reinforcing the connection between the list's contents and the elements on the map. In addition, when users place their mouse over a restaurant, on either the map or the list, a highlighting visual halo appears around the corresponding visual elements in the map and in the list.

When a user selects multiple sets from the category selector, our system follows a behavior similar to services such as Yelp [1] i.e., we perform a union operation within each level of a category (food type, price range and rating) and an intersection between categories. This allows users to express selections that indicate "Indian or American restaurants with a medium price range and a 3.0 rating".

7.1.2 Informal User Feedback

We observed six male users (21 to 27 years of age) search for restaurants using the yelp.com Internet service. Our task was very informal, simply asking them to pick various restaurants from different criteria requirements and to comment on their decisions. For example, we asked participants to select "a restaurant to impress a date" or "a restaurant for a working lunch meeting". Then, we had them perform similar tasks using the LineSets prototype.

As initial feedback, all six participants were able to understand LineSets and use the interface within the few minutes long study. Half of our participants showed the same searching behaviour when using Yelp and LineSets: they decided on a set of criteria, clicked on them and picked their favourite restaurant in the results returned. These participants generally preferred Yelp, commenting that LineSets brought visual clutter and missed the reviews information provided by Yelp. However, we observed the remaining participants adopting a noticeably different searching behaviour with LineSets. Instead of selecting strict criteria, these participants were willing to "compromise" in terms of the type of food or location, and they explored multiple possibilities before picking a specific restaurant. In particular, they commented that one benefit of LineSets was the possibility of obtaining partial results for a given query. One participant explained that with LineSets, it was easier to see why particular results were returned, since set membership was immediately visible. This participant explained that LineSets were especially useful when performing queries using an "or" criteria. He also mentioned that partial results led to more discoveries.

Our observations highlight the tradeoffs of introducing a set visualization in a map canvas. While the visualization may prove particularly useful for exploratory tasks requiring iterative refinements and improve the browsing experience, it can increase the visual clutter.

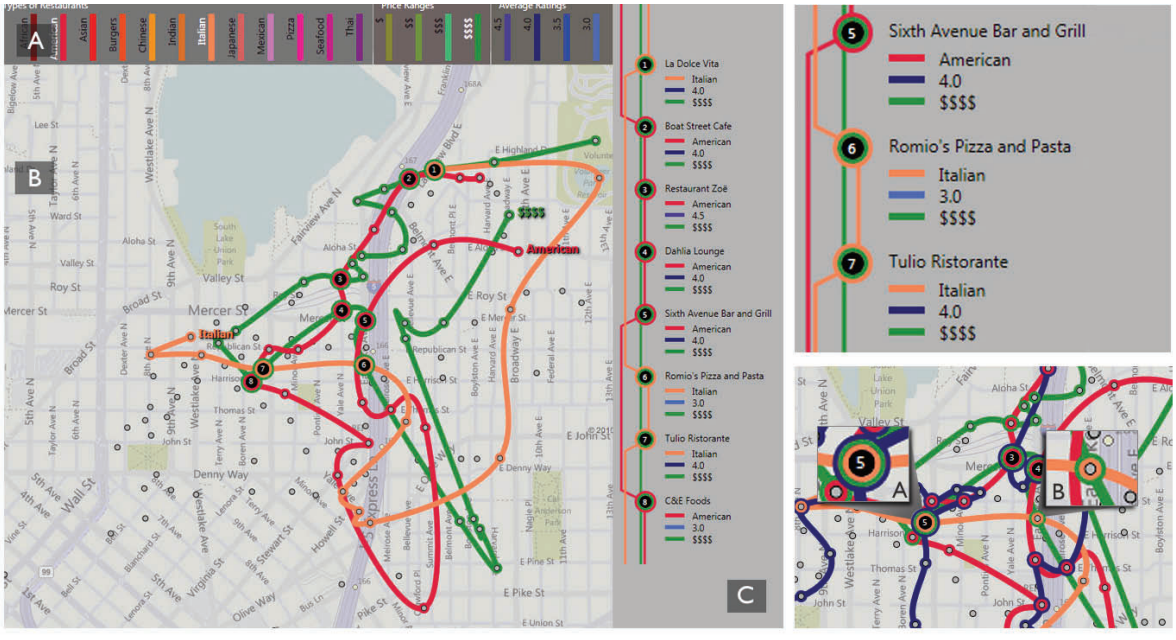

Fig. 9. Application with category selector area (A), map visualization (B) and list view (C) sections. Detail of the list view (upper right) shows the flattened versions of LineSets. In the close up view (bottom right), element A is an exact match to user specified criteria with 3 concentric rings around black dot, whereas element B matches only a subset of the criteria.

7.2 LineSets on Social Networks

We are interested in the application of LineSets in cases other than elements fixed in space. In the general case of points drawn on a canvas, one can apply self-organizing algorithms to distribute and cluster elements, based on a particular distance metric. The members of a social network fall within this type of dataset. We present an initial prototype visualizing communities in social networks as LineSets.

Our prototype loads a social network and renders it using a force-directed layout. The system then draws LineSets for a predefined attribute within the dataset. Figure 10 presents an example of co-authorship networks in which sets are communities of author working on the same topic. Unlike our map application, users are able to reposition elements over the canvas, an action that causes LineSets to be recomputed. We also implemented simple interactions to apply constraints to a selected LineSet, allowing users to make it circular (Figure 10 bottom) or straight. We also envision additional interactions, allowing users to change the drawing of the curve, create new sets, add or remove elements from existing ones or merge multiple sets together. Such interactions could feel very direct by using touch- or pen-based input devices.

Applying constraints to LineSets can also serve to highlight a particular set and showcase its intersections with others. For example, Figure 10, bottom shows a network with a LineSet drawn as a circle, the other LineSets crossing this circle represent set intersections. The reduction of visual clutter compared to traditional set representations also makes it easier to discern set membership when an element is member of a large number of sets as illustrated in Figure 10, bottom.

8 Scalability

To investigate how the effectiveness of LineSets is affected by the set's size, we visualized two larger datasets borrowed from [15]. Figure 11 shows the top 100 movies and their 1174 actors from the International Movie Data Base (IMDB) and Figure 12 shows the top 200 works in the ten tragedies of Shakespeare.

The issue of scale for LineSets is algorithmic if one wishes to optimize the lines to draw the shortest path between elements of the sets. In this case, the technique depends on the heuristic used to compute the travelling salesman problem. As an indication, Figure 11 was generated in a few seconds using LKH and would be rendered in few milliseconds without computing the shortest paths. As an indication of the impact of computing shortest paths, LKH has not been used in Figure 1. In terms of visual clutter, Figure 11 shows that thousands of elements may be easily visualized if the set intersections are relatively simple.

The dataset represented in Figure 10, left has only 61 nodes but the complex intersections of its ten sets tests the limits of readability of any set representation. We compare our technique with the state-of-the-art presented in [15] (Figure 12). Limitations include the representation of exact same sets, which are currently difficult to identify in LineSets as curves are superposed. These are hierarchically enclosed in [15], which makes them easy to read but may mislead users into assessing wrong inclusions. A possible solution we are investigating is to offset the LineSets for same sets so all curves become parallel, making similar sets more salient.

9 Conclusion and Future Work

In this paper, we described LineSets, a novel set visualization technique that represents sets as smooth curves. LineSets add another information visualization tool to the arsenal available to both data analysts and casual users alike, thus amplifying their ability to understand and reason about categorical datasets.

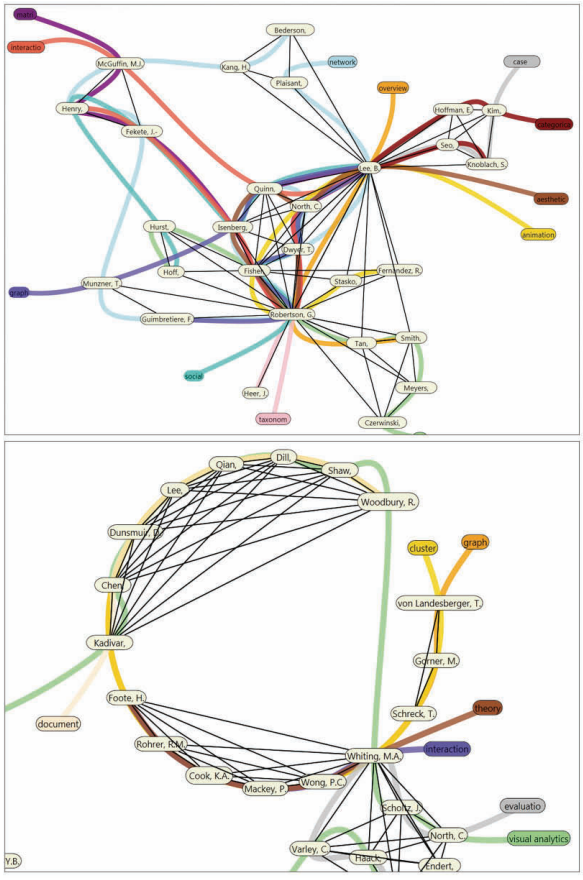

Fig. 10. LineSets applied to the InfoVis co-authorship network. (top) complex set intersections; (bottom) LineSets drawn as a circle.

Contrary to general practice, we validated our visualization idea through a controlled experiment prior to the development of the technique. Our experiment showed that LineSets improve the readability of set membership and set intersection tasks compare to traditional bubble set techniques. While the design principles behind LineSets are simple to state, there are many ways to create them. We have implemented LineSets using two simple heuristics: generate paths that are as linear as possible and geometrically smooth. Through our observations of people drawing LineSets, we found that these heuristics were reasonable if not ideal.

The problem of computing a "best" path for a LineSet is a non-trivial one, and an initial solution to it might involve users' actions to tweak initial paths. Another way to produce good LineSets paths might involve using a hybrid bubble/line set representation as some of our participants sketched (Figure 6). By strategically using bubble representations to create cluster elements (using metrics such as point density), one can greatly simplify the underlying path connecting a set of elements. We plan to investigate this solution in the future.

We have implemented two interactive prototypes that apply the LineSets technique to different families of data, e.g., map and social network data, thus bringing the technique closer to real-world use scenarios. These prototypes pointed us to several areas of improvements and inspired novel interactions. We believe that LineSets are compelling entities to be manipulated in multi-touch environments, where they become "tangible threads" with many physical analogues available for exploration.

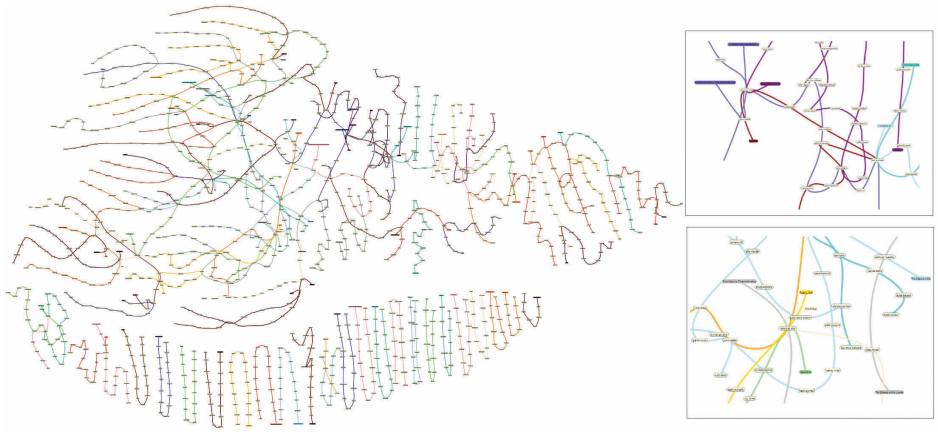

Fig. 11. Top 100 IBMD movies and close up view from the same visualization (on right).

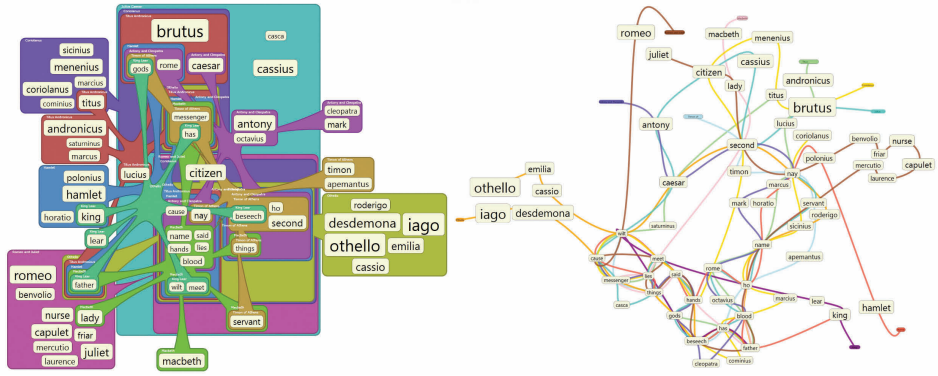

Fig. 12. Top 200 words in ten plays of Shakespeare. Left shows Euler Diagrams [15] version, Right figure shows the LineSet version.

REFERENCES

[1] F. Benoy, P. Rodgers, "Evaluating the comprehension of Euler diagrams", *In Proc. IEEE Conf. on Information Visualization*, pp. 771–780, 2007.

[2] U. Brandes, S. Cornelsen, B. Pampel, and A. Sallaberry, "Path-based supports for hypergraphs," *Proceedings of the 21st International Workshop on Combinatorial Algorithms (IWOCA 2010)*, C. Iliopoulos, W. Smyth, eds., Lecture Notes in Computer Science, Springer, 2010.

[3] S.C. Chow, "Generating and drawing area proportional Euler and Venn diagrams", PhD dissertation, Dept. of Computer Science, Univ. of Victoria, 2007.

[4] C. Collins, G. Penn, S. Carpendale. "Bubble Sets: Revealing set relations with isocontours over existing visualizations", *IEEE TVCG*, vol. 15, pp. 1009–1016, 2009.

[5] M. Czerwinski, E. Horvitz, E. Cutrell, "Subjective Duration Assessment: An Implicit Probe for Software Usability", *IHMHCI*, 2001.

[6] A. Fish, G. Stapleton, "Defining Euler diagrams: choices and consequences", *In Proc. Euler Diagrams Workshop*, 2005.

[7] J. Flower, A. Fish, J. Howse, "Euler diagram generation", *Journal of Visual Languages and Computing*, vol. 19, pp. 675–694, Dec. 2008.

[8] W. Freiler, K. Matkovi´c, H. Hauser, "Interactive visual analytics of set typed data", *In Proc. IEEE Conf. on Information Visualization*, vol. 14:6, pp. 1340–1347, 2008.

[9] J. Heer, danah boyd, "Vizster: Visualizing online social networks", *In Proc. IEEE Conf. on Information Visualization*, 2005.

[10] E. Jabbour, "Mapping information: redesigning the New York city subway map", *Beautiful Visualization: looking at data through the eyes of experts*, J. Steele, N. Illinsky, eds., O'Reilly, 2010.

[11] R. Kosara, F. Bendix and H. Hauser, "Interactive exploration and visual analysis of categorical data," *IEEE Transactions on Visualization and Computer Graphics*, pp. 558-568, 2006.

[12] S. Lin, B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem", *Operations Research*, vol. 21, pp. 498–516, 1973.

[13] D. Marr, "Vision: A computational investigation into the human representation and processing of visual information", Freeman, 1982.

[14] G. Palla, I. Derenyi, I. Farkas, T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society", Nature, vol. 435:7043, pp. 814–818, June, 2005.

[15] N.H. Riche, T. Dwyer, "Untangling Euler Diagrams", *In Proc. IEEE Conf. on Information Visualization*, 2010.

[16] B. Shneiderman, "Tree visualization with tree-maps: 2-d space-filling approach," *ACM Transactions on graphics*, vol. 11:1, pp. 92-99, 1992.

[17] P. Simonetto, D. Auber, D. Archambault, "Fully automatic visualization of overlapping sets", *Comput. Graph. Forum*, vol. 28:3, pp. 967–974, 2009.

[18] A. Verroust, M.-L. Viaud, "Ensuring the drawability of extended euler diagrams for up to 8 sets", *In Proc. Diagrams*, LNAI vol. 2980, pp. 128–141, Springer Verlag, 2004.

[19] C. Ware, Information Visualization: Perception for Design, Morgan Kaufmann, 2nd edition, 2004.

[20] C. Ware, H. Purchase, L. Colpoys, M. Mcgill, "Cognitive measurements of graph aesthetics", *Information Visualization*, vol. 1:2, pp. 103-110, 2002.

[21] N. Watanabe, M. Washida, T. Igarashi, "Bubble clusters: An interface for manipulating spatial aggregation of graphical objects", *In Proc. of ACM Symp. on User Interface Software and Technology*. ACM, Oct. 2007.

[22] http://www.yelp.com/developers/.