Data Visualization (DSC 530/CIS 602-01)

Filtering & Aggregation

Dr. David Koop
Design Space of Composite Visualization

- Composite visualization views (CVVs)
  - Includes Coordinated multiple views (CMV)
  - + More!
- Design Patterns:
  - Juxtaposition: side-by-side
  - Superimposition: layers
  - Overloading: vis meshed with another
  - Nesting: vis inside a vis (recursive vis)
  - Integration: "merge" views + links

[W. Javed and N. Elmqvist, 2012]
Which strategy?

[ComVis, K. Matkovic et al., 2008]
Juxtaposition

ComVis - California meteo views setup.png

[ComVis, K. Matkovic et al., 2008]
Which strategy?

![Temperature Graph](#)

- Austin
- New York
- San Francisco

Temperature (°F)

- October
- November
- December
- 2012
- February
- March
- April
- May
- June
- July
- August
- September

M. Bostock
Superimposition

Temperature (ºF)

Austin
New York
San Francisco
Which Strategy?

[NodeTrix, N. Henry et al., 2007]
Nesting

[NodeTrix, N. Henry et al., 2007]
Which Strategy?

"best statistical graphic ever"

[Napoleon's March to Moscow, C. J. Minard, 1869]

(later known as a Sankey Diagram)
Integration

"best statistical graphic ever"

[Napoleon’s March to Moscow, C. J. Minard, 1869]
Which Strategy?

Superimposed views overlay two or more visual spaces on top of each other (Figures 6 and 7). The resulting visualization becomes the visual combination of the component visualizations, often using transparency to enable seeing all views. Superimposed views are generally used to highlight spatial relations in the component visualizations. In other words, the spatial linking present in these views is one-to-one, i.e., all the overlay visualizations share the same underlying visual space. Line graph visualizations with several data series, where more than one graph is superimposed in a single chart (e.g., [19]), is a very commonly used example of this design pattern.

The spatial linking in the superimposed views allows for easy comparison across different datasets because the user does not have to split their attention between different parts of the visual space. Furthermore, the fact that visualizations are stacked means that they can each use the full available space in the view. However, because the composition simply adds the component visualizations together, the visual clutter may become significant, and it is also likely to cause conflicts arising from one visualization occluding another.

5.1 Mapgets

Mapgets [38] is a geographic visualization system that allows users to interactively perform map editing and querying of geographical datasets. The maps generated using Mapgets are built on an underlying presentation stack that superimposes multiple dataset layers on top of each other. The users can dynamically select the dataset to use for each layer and the total number of layers to compose. Different layers in the presentation stack allow users to independently interact with each of the associated visualization and control the layer attributes. The technique also allows the users to reorder layers in the presentation stack to achieve the desirable map result. Figure 6 shows an example of a European map generated in Mapgets. The presentation stack associated with this map consists of three layers: the bottom layer visualizes rivers, the center layer is used to depict the country borders, and the topmost layer is used to display the country labels.

5.2 GeoSpace

GeoSpace [22] allows users to interactively explore complex visual spaces using superimposed views. It permits progressively overlaying different datasets, based on the user queries, in a single view. Beyond allowing users to explore datasets through dynamic queries, GeoSpace also supports pan and zoom operations for navigation. Figure 7 shows GeoSpace system being used for exploring crime around the Cambridge, MA area. The figure shows a 2D view of the visualization, where red dots that are spatially coupled to the underlying layer show the reported crime cases in the region.

Figure 8: SPPC [45] (Overloaded Views). This tool overloads points into the region bounded by two axes in the parallel coordinate plot.

Figure 9: Links on treemaps [14] (Overloaded Views). The tool identifies a tree structure in a graph and visualizes it using a treemap.

[Links on Treemaps, J.-D. Fekete et al., 2003]
Overloading

Superimposed views overlay two or more visual spaces on top of each other (Figures 6 and 7). The resulting visualization becomes the visual combination of the component visualizations, often using transparency to enable seeing all views. Superimposed views are generally used to highlight spatial relations in the component visualizations. In other words, the spatial linking present in these views is one-to-one, i.e., all the overlay visualizations share the same underlying visual space. Line graph visualizations with several data series, where more than one graph is superimposed in a single chart (e.g., [19]), is a very commonly used example of this design pattern. The spatial linking in the superimposed views allows for easy comparison across different datasets because the user does not have to split their attention between different parts of the visual space. Furthermore, the fact that visualizations are stacked means that they can each use the full available space in the view. However, because the composition simply adds the component visualizations together, the visual clutter may become significant, and it is also likely to cause conflicts arising from one visualization occluding another.

5.1 Mapgets

Mapgets [38] is a geographic visualization system that allows users to interactively perform map editing and querying of geographical datasets. The maps generated using Mapgets are built on an underlying presentation stack that superimposes multiple dataset layers on top of each other. The users can dynamically select the dataset to use for each layer and the total number of layers to compose. Different layers in the presentation stack allow users to independently interact with each of the associated visualization and control the layer attributes. The technique also allows the users to reorder layers in the presentation stack to achieve the desirable map result. Figure 6 shows an example of a European map generated in Mapgets. The presentation stack associated with this map consists of three layers: the bottom layer visualizes rivers, the center layer is used to depict the country borders, and the topmost layer is used to display the country labels.

5.2 GeoSpace

GeoSpace [22] allows users to interactively explore complex visual spaces using superimposed views. It permits progressively overlaying different datasets, based on the user queries, in a single view. Beyond allowing users to explore datasets through dynamic queries, GeoSpace also supports pan and zoom operations for navigation. Figure 7 shows GeoSpace system being used for exploring crime around the Cambridge, MA area. The figure shows a 2D view of the visualization, where red dots that are spatially coupled to the underlying layer show the reported crime cases in the region.

Figure 8: SPPC [45] (Overloaded Views). This tool overloads points into the region bounded by two axes in the parallel coordinate plot.

Figure 9: Links on treemaps [14] (Overloaded Views). The tool identifies a tree structure in a graph and visualizes it using a treemap.

[Links on Treemaps, J.-D. Fekete et al., 2003]
Which Strategy?

Fig. 13. A software system and its associated call graph (caller = green, callee = red). (a) and (b) show the system with bundling strength $\beta = 0.85$ using a balloon layout (node labels disabled) and a radial layout, respectively. Bundling reduces visual clutter, making it easier to perceive the actual connections than when compared to the non-bundled versions (figures 2a and 11a). Bundled visualizations also show relations between sparsely connected systems more clearly (encircled regions); these are almost completely obscured in the non-bundled versions. The encircled regions highlight identical parts of the system for (a), (b), and figure 15.

Fig. 14. Using the bundling strength $\beta$ to provide a trade-off between low-level and high-level views of the adjacency relations. The value of $\beta$ increases from left-to-right; low values mainly provide low-level, node-to-node connectivity information, whereas high values provide high-level information as well by implicit visualization of adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes.

Changing the bundling strength $\beta$ and by switching between different tree layouts. The participants from academia were our fellow researchers, PhD students and MSc students from the Computer Science department of the Technische Universiteit Eindhoven. They all had experience with either software development, software visualization, or information visualization in general. Participants from industry were representatives of the Software Improvement Group (SIG) in Amsterdam, which delivers insight in the structure and technical quality of software portfolios, and representatives of FEI Company Eindhoven, which produces software to operate with FEI’s range of electron microscopes.

The majority of the participants regarded the technique as useful for quickly gaining insight in the adjacency relations present in hierarchically organized systems. In general, the visualizations were also regarded as being aesthetically pleasing. SIG and FEI Company Eindhoven are currently supporting further development by providing us with additional data sets and feedback regarding the resulting visualizations.

More specifically, most of the participants particularly valued the fact that relations between items at low levels of the hierarchy were automatically lifted to implicit relations between items at higher levels by means of bundles. This quickly gave them an impression of the high-level connectivity information while still being able to inspect the low-level relations that were responsible for the bundles by interactively manipulating the bundling strength.

This is illustrated in figure 14, which shows visualizations using different values for the bundling strength $\beta$. Low values result in visualizations that mainly provide low-level, node-to-node connectivity information. High values result in visualizations that provide high-level information as well by implicit visualization of adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes.

Another aspect that was commented on was how the bundles gave an impression of the hierarchical organization of the data as well, thereby strengthening the visualization of the hierarchy. More specifically, a thick bundle shows the presence of two elements at a fairly...
Superimposition: Hierarchical Edge Bundles

Fig. 13. A software system and its associated call graph (caller = green, callee = red). (a) and (b) show the system with bundling strength $\beta = 0.85$ using a balloon layout (node labels disabled) and a radial layout, respectively. Bundling reduces visual clutter, making it easier to perceive the actual connections than when compared to the non-bundled versions (figures 2a and 11a). Bundled visualizations also show relations between sparsely connected systems more clearly (encircled regions); these are almost completely obscured in the non-bundled versions. The encircled regions highlight identical parts of the system for (a), (b), and figure 15.

Fig. 14. Using the bundling strength $\beta$ to provide a trade-off between low-level and high-level views of the adjacency relations. The value of $\beta$ increases from left-to-right; low values mainly provide low-level, node-to-node connectivity information, whereas high values provide high-level information as well by implicit visualization of adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes.

The majority of the participants regarded the technique as useful for quickly gaining insight in the adjacency relations present in hierarchically organized systems. In general, the visualizations were also regarded as being aesthetically pleasing. SIG and FEI Company Eindhoven are currently supporting further development by providing us with additional data sets and feedback regarding the resulting visualizations.

More specifically, most of the participants particularly valued the fact that relations between items at low levels of the hierarchy were automatically lifted to implicit relations between items at higher levels by means of bundles. This quickly gave them an impression of the high-level connectivity information while still being able to inspect the low-level relations that were responsible for the bundles by interactively manipulating the bundling strength.

This is illustrated in figure 14, which shows visualizations using different values for the bundling strength $\beta$. Low values result in visualizations that mainly provide low-level, node-to-node connectivity information. High values result in visualizations that provide high-level information as well by implicit visualization of adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes.

Another aspect that was commented on was how the bundles gave an impression of the hierarchical organization of the data as well, thereby strengthening the visualization of the hierarchy. More specifically, a thick bundle shows the presence of two elements at a fairly
# Multiple Views

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Data</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All</td>
<td>Subset</td>
<td>None</td>
</tr>
<tr>
<td>Same</td>
<td>Redundant</td>
<td>Overview/Detail</td>
<td>Small Multiples</td>
</tr>
<tr>
<td>Different</td>
<td>Multiform</td>
<td>Multiform, Overview/Detail</td>
<td>No Linkage</td>
</tr>
</tbody>
</table>

[Munzner (ill. Maguire), 2014]
Multiform

[Improvise, Weaver, 2004]

D. Koop, DSC 530, Spring 2018
Overview-Detail View
Small Multiples

- Same encoding, but different data in each view (e.g. SPLOM)
Partitioned Views

• Split dataset into groups and visualize each group
• Extremes: one item per group, one group for all items
• Can be a hierarchy
  - Order: which splits are more "related"?
  - Which attributes are used to split? usually categorical
Example: Grouped Bar Chart

Population

65 Years and Over
45 to 64 Years
25 to 44 Years
18 to 24 Years
14 to 17 Years
5 to 13 Years
Under 5 Years

[CA] [TX] [NY] [FL] [IL] [PA]
Matrix Alignment

In Figure 2 there are 6 panels, 1 column, 6 rows, and 1 page. Later, we will show a Trellis display with more than one page. We refer to the rectangular array as the trellis because it is reminiscent of a garden trelliswork.

Each panel of a trellis display shows a subset of the values of panel variables; these values are formed by conditioning on the values of conditioning variables. In Figure 1 the panel variables are variety and yield, and the conditioning variables are site and year. On each panel, values of yield and variety are displayed for one combination of year and site. 

Barley Yield (bushels/acre)

Downloaded by [50.148.12.36] at 21:36 08 January 2013

[Becker et al., 1996]
Recursive Subdivision

[Slingsby et al., 2009]
Project Designs

- Use interaction
- Be creative!
- Next step: Designs
  - 3 Designs (Either iterations or three separate ideas)
  - At least one must be prototyped (if others are not prototyped, turn in detailed sketches)
  - Due Thursday, April 5
- Inspiration: mbtaviz.github.io
Test 2

• Details
• Next Wednesday, April 11
• Similar Format
• Covers material since the beginning of class but with an emphasis on the material covered since Test 1
Reducing Complexity

- Too many items or attributes lead to visual clutter
- Interaction and Multiple Views can help, but often lose the ability to start understanding an entire dataset at first glance
- **Reduction** techniques show less data to reduce complexity
- Can reduce items or attributes (both are elements)
- **Filtering**: eliminate elements from the current view
  - "out of sight, out of mind"
- **Aggregation**: replace elements with a new element that represents the replaced elements
  - summarization is often challenging to design
- Another method is **focus+context**: show details in the context of an overview
Overview

Reducing Items and Attributes

Filter
- Items
  ![Filtered Items Diagram]
- Attributes
  ![Filtered Attributes Diagram]

Aggregate
- Items
  ![Aggregated Items Diagram]
- Attributes
  ![Aggregated Attributes Diagram]

Reduce

Filter
- Items
  ![Reduced Items Diagram]
- Attributes
  ![Reduced Attributes Diagram]

Aggregate
- Items
  ![Aggregated Items Diagram]
- Attributes
  ![Aggregated Attributes Diagram]

Embed

[Munzner (ill. Maguire), 2014]
Filtering

- Just don't show certain elements
- Item filtering: most common, eliminate marks for filtered items
- Attribute filtering:
  - attributes often mapped to different channels
  - if mapped to same channel, allows many attributes (e.g. parallel coordinates, star plots), can filter
- How to specify which elements?
  - Pre-defined rules
  - User selection
Example: NYC Health Dept. Restaurant Ratings

Restaurant locations are derived from the New York City Department of Health and Mental Hygiene database. Due to the limitations of the Health Department’s database, some restaurants could not be placed.
Dynamic Queries

• Interaction need not be with the visualization itself
• Users interact with **widgets** that control which items are shown
  - Slides, Combo boxes, Text Fields
• Often tied to attribute values
• Examples:
  - All restaurants with an "A" Grade
  - All pizza places
  - All pizza places with an "A" Grade
Scented Widgets

Occupation
- Artisan / Art Teacher
- Athlete
- Auctioneer
- Author
- Baggage
erman
- Baker
- Bank Teller
- Barber / Beautician
- Bartender
- Bill Collector
- Blacksmith
- Blaster
- Boarding House Keeper
- Boatman
- Boilermaker
- Bookbinder
- Bookkeeper
- Bootblack
- Building Manager
- Bus Driver
- Buyer - Farm
- Buyer - Store
- Cabinet Maker
- Car Washer
- Carpenter

Scale
- Total People Count
- % of Work Force
- # of times viewed

Figure 3: The number and variety of jobs in the United States have changed significantly over time. For example, the occupation of Bartender has become less common in recent decades.

[Willett et al., 2007]
Scented Widgets

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hue</td>
<td>Varies the hue of the widget (or of a visualization embedded in it)</td>
<td>Option A Option B</td>
</tr>
<tr>
<td>Saturation</td>
<td>Varies the saturation of the widget (or of a visualization embedded in it)</td>
<td>Option A Option B</td>
</tr>
<tr>
<td>Opacity</td>
<td>Varies the saturation of the widget (or of a visualization embedded in it)</td>
<td>Option A Option B</td>
</tr>
<tr>
<td>Text</td>
<td>Inserts one or more small text figures into the widget</td>
<td>(2) Option A (10) Option B</td>
</tr>
<tr>
<td>Icon</td>
<td>Inserts one or more small icons into the widget</td>
<td>Option A Option B</td>
</tr>
<tr>
<td>Bar Chart</td>
<td>Inserts one or more small bar chart visualizations into the widget</td>
<td>Option A Option B</td>
</tr>
<tr>
<td>Line Chart</td>
<td>Inserts one or more small line chart visualizations into the widget</td>
<td>Option A Option B</td>
</tr>
</tbody>
</table>

[Willett et al., 2007]
Star Plots

Aberfeldy
- Fruity
- Malty
- Nutty
- Spicy
- Honey
- Smoky
- Body
- Sweetness
- Floral

Aberlour

AnCnoc

Ardbeg

Ardmore
- Fruity
- Malty
- Nutty
- Spicy
- Honey
- Smoky
- Body
- Sweetness
- Floral

ArranIsleOf

Auchentoshan

Auchroisk
Attribute Filtering on Star Plots

[Yang et al., 2003]
Attribute Filtering

• How to choose which attributes should be filtered?
  - User selection?
  - Statistics: similarity measures, attributes with low variance are not as interesting when comparing items

• Can be combined with item filtering
Aggregation

- Usually involves **derived** attributes
- Examples: mean, median, mode, min, max, count, sum
- Remember expressiveness principle: still want to avoid implying trends or similarities based on aggregation

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>x</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>10.0</td>
<td>8.04</td>
<td>10.0</td>
<td>9.14</td>
<td>10.0</td>
</tr>
<tr>
<td>8.0</td>
<td>6.95</td>
<td>8.0</td>
<td>8.14</td>
<td>8.0</td>
</tr>
<tr>
<td>13.0</td>
<td>7.58</td>
<td>13.0</td>
<td>8.74</td>
<td>13.0</td>
</tr>
<tr>
<td>9.0</td>
<td>8.81</td>
<td>9.0</td>
<td>8.77</td>
<td>9.0</td>
</tr>
<tr>
<td>11.0</td>
<td>8.33</td>
<td>11.0</td>
<td>9.26</td>
<td>11.0</td>
</tr>
<tr>
<td>14.0</td>
<td>9.96</td>
<td>14.0</td>
<td>8.10</td>
<td>14.0</td>
</tr>
<tr>
<td>6.0</td>
<td>7.24</td>
<td>6.0</td>
<td>6.13</td>
<td>6.0</td>
</tr>
<tr>
<td>4.0</td>
<td>4.26</td>
<td>4.0</td>
<td>3.10</td>
<td>4.0</td>
</tr>
<tr>
<td>12.0</td>
<td>10.84</td>
<td>12.0</td>
<td>9.13</td>
<td>12.0</td>
</tr>
<tr>
<td>7.0</td>
<td>4.82</td>
<td>7.0</td>
<td>7.26</td>
<td>7.0</td>
</tr>
<tr>
<td>5.0</td>
<td>5.68</td>
<td>5.0</td>
<td>4.74</td>
<td>5.0</td>
</tr>
</tbody>
</table>
Aggregation

- Usually involves **derived** attributes
- Examples: mean, median, mode, min, max, count, sum
- Remember expressiveness principle: still want to avoid implying trends or similarities based on aggregation

<table>
<thead>
<tr>
<th></th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td>x</td>
<td>y</td>
<td>x</td>
</tr>
<tr>
<td>10.0</td>
<td>8.04</td>
<td>10.0</td>
<td>9.14</td>
<td>10.0</td>
</tr>
<tr>
<td>8.0</td>
<td>6.95</td>
<td>8.0</td>
<td>8.14</td>
<td>8.0</td>
</tr>
<tr>
<td>13.0</td>
<td>7.58</td>
<td>13.0</td>
<td>8.74</td>
<td>13.0</td>
</tr>
<tr>
<td>9.0</td>
<td>8.81</td>
<td>9.0</td>
<td>8.77</td>
<td>9.0</td>
</tr>
<tr>
<td>11.0</td>
<td>8.33</td>
<td>11.0</td>
<td>9.26</td>
<td>11.0</td>
</tr>
<tr>
<td>14.0</td>
<td>9.96</td>
<td>14.0</td>
<td>8.10</td>
<td>14.0</td>
</tr>
<tr>
<td>6.0</td>
<td>7.24</td>
<td>6.0</td>
<td>6.13</td>
<td>6.0</td>
</tr>
<tr>
<td>4.0</td>
<td>4.26</td>
<td>4.0</td>
<td>3.10</td>
<td>4.0</td>
</tr>
<tr>
<td>12.0</td>
<td>10.84</td>
<td>12.0</td>
<td>9.13</td>
<td>12.0</td>
</tr>
<tr>
<td>7.0</td>
<td>4.82</td>
<td>7.0</td>
<td>7.26</td>
<td>7.0</td>
</tr>
<tr>
<td>5.0</td>
<td>5.68</td>
<td>5.0</td>
<td>4.74</td>
<td>5.0</td>
</tr>
</tbody>
</table>

Mean of x: 9
Variance of x: 11
Mean of y: 7.50
Variance of y: 4.122
Correlation: 0.816
Anscombe's Quartet

[F. J. Anscombe]
Aggregation: Histograms

- Very similar to bar charts
- Often shown without space between (continuity)
- Choice of number of bins
  - Important!
  - Viewers may infer different trends based on the layout

[Munzner (ill. Maguire), 2014]
Binning Scatterplots

- At some point, cannot see density
- Blobs on top of blobs
- 2D Histogram is a histogram in 2D encoded using color instead of height
- Each region is aggregated
Hexagonal Binning

- Hexagonal bins are more circular
- Distance to the edge is not as variable
- More efficient aggregation around the center of the bin
Fig. 8. The left side shows the discrete scatterplot of the "b-lunt-fin" data set, whereas the continuous version is shown on the right. Both types of scatterplots visualize the scalar data value along the horizontal axis and the magnitude of the gradient along the vertical axis. Choosing these data dimensions, material and boundary identification is possible by finding arc-like structures.

Fig. 9. In the upper part, both types of scatterplots are shown for the "tornado" data set. The upper-left images show the discrete scatterplot, to the right is the continuous version. Both scatterplots employ the same color lookup table as the previous example. The lower part shows three volume visualizations of the data set. The lower-left image (a) shows the tornado visualized by a representative isosurface of velocity magnitude. The image in the middle (b) shows highlighted voxels (yellow) that were marked in the continuous scatterplot. This highlighting corresponds to the upper-right selection rectangle in the continuous scatterplot. The other selection rectangle in the lower-middle part of the continuous scatterplot highlights different voxels, as shown in the lower-right volume-visualization image (c). In image (c), highlighted voxels (yellow) and the velocity magnitude are simultaneously visualized by rather transparent volume rendering in order to show selected features at different depths. Therefore, we can see that different voxels than in (b) are highlighted, especially not the ones in the center of the tornado.

CONCLUSION AND FUTURE WORK

We have presented continuous scatterplots as a generalization of conventional scatterplots. One aspect of generalization is the support of any dimension of the domain of the data set and of the scatterplot. The other aspect of generalization is the extension to data defined on continuous domains. The basis for continuous scatterplots is provided in the form of a generic mathematical model. This mathematical model maps an arbitrary density value defined on an \( n \)-D input data set to \( m \)-D scatterplots. We have shown how continuous scatterplots are related to conventional discrete histograms and to histograms of isosurface statistics. In particular, the 2-D version of continuous scatterplots is, by construction, identical to conventional discrete scatterplots in the limit process of infinitely dense sample points. Therefore, continuous scatterplots lead to the same basic visual mapping as traditional histograms, scatterplots, or other frequency plots, utilizing their proven visualization power. We have provided typical examples of multi-attribute visualization—such as 2-D transfer function specification and flow visualization—to demonstrate the applicability of our approach. The difference to discrete scatterplots is especially visible for low-resolution data sets and for data sets defined on grids with largely different cell sizes.

The main advantage of continuous scatterplots is that they are directly designed for input data defined on continuous domains. Therefore, this paper adds one missing piece to the general approach of applying...
Aggregation: Continuous Scatterplot

Fig. 8. The left side shows the discrete scatterplot of the "b-lunt-fin" data set, whereas the continuous version is shown on the right. Both types of scatterplots visualize the scalar data value along the horizontal axis and the magnitude of the gradient along the vertical axis. Choosing these data dimensions, material and boundary identification is possible by finding arc-like structures.

Fig. 9. In the upper part, both types of scatterplots are shown for the "tornado" data set. The upper-left image shows the discrete scatterplot, to the right is the continuous version. Both scatterplots employ the same color lookup table as the previous example. The lower part shows three volume visualizations of the data set. The lower-left image (a) shows the tornado visualized by a representative isosurface of velocity magnitude. The image in the middle (b) shows highlighted voxels (yellow) that were marked in the continuous scatterplot. This highlighting corresponds to the upper-right selection rectangle in the continuous scatterplot. The other selection rectangle in the lower-middle part to the left in the continuous scatterplot highlights different voxels, as shown in the lower-right volume-visualization image (c). In image (c), highlighted voxels (yellow) and the velocity magnitude are simultaneously visualized by rather transparent volume rendering in order to show selected features at different depths. Therefore, we can see that different voxels than in (b) are highlighted, especially not the ones in the center of the tornado.

Conclusion and Future Work

We have presented continuous scatterplots as a generalization of conventional scatterplots. One aspect of generalization is the support of any dimension of the domain of the data set and of the scatterplot. The other aspect of generalization is the extension to data defined on continuous domains. The basis for continuous scatterplots is provided in the form of a generic mathematical model. This mathematical model maps an arbitrary density value defined on an $n$-D input data set to $m$-D scatterplots. We have shown how continuous scatterplots are related to conventional discrete histograms and to histograms of isosurface statistics. In particular, the 2-D version of continuous scatterplots is, by construction, identical to conventional discrete scatterplots in the limit process of infinitely dense sample points. Therefore, continuous scatterplots lead to the same basic visual mapping as traditional histograms, scatterplots, or other frequency plots, utilizing their proven visualization power. We have provided typical examples of multi-attribute visualization—such as 2-D transfer function specification and flow visualization—to demonstrate the applicability of our approach. The difference to discrete scatterplots is especially visible for low-resolution data sets and for data sets defined on grids with largely different cell sizes.

The main advantage of continuous scatterplots is that they are directly designed for input data defined on continuous domains. Therefore, this paper adds one missing piece to the general approach of applications. 

[Bachthaler & Weiskopf, 2008]
Spatial Aggregation

in cartography, changing the boundaries of the regions used to analyze data can yield dramatically different results.
Spatial Aggregation

In cartography, changing the boundaries of the regions used to analyze data can yield dramatically different results.
Spatial Aggregation

In cartography, changing the boundaries of the regions used to analyze data can yield dramatically different results.

[Penn State, GEOG 486]
Modifiable Areal Unit Problem

- How you draw boundaries impacts the type of aggregation you get
- Similar to bins in histograms
- Gerrymandering

[Wonkblog, Washington Post, Adapted from S. Nass]
Boxplots

- **Show distribution**
- Single value (e.g. mean, max, min, quartiles) doesn't convey everything
- Created by John Tukey who grew up in New Bedford!
- **Show spread and skew** of data
- Best for **unimodal** data
- Variations like vase plot for multimodal data
- Aggregation here involves many different marks
Boxplot Example

(a) Overall Activity

(b) Structural Activity

(c) Parameter Activity

(d) Layout Activity

[L. Lins et al., 2008]
Hierarchical Parallel Coordinates

Figure 4: This image sequence shows a Fatal Accident data set of 230,000 data elements at different levels of detail. The first image shows a cut across the root node. The last image shows the cut chaining all the leaf nodes. The rest of the images show intermediate cuts at varying levels of detail. (See Color Plates).

Figure 6: Left image shows Iris data set without proximity-based coloring. Right image shows Iris data set with proximity-based coloring revealing three distinct clusters depicted by concentrations of blue, green and pink lines. (See Color Plates).

[Fua et al., 1999]
D3 Multiple Views and Interaction

• https://codepen.io/dakoop/pen/jBQrYp