Data Visualization (CIS 490/680)

Data

Dr. David Koop
JavaScript in one slide

• Interpreted and Dynamically-typed Programming Language

• Statements end with semi-colons, normal blocking with brackets

• Variables: `var a = 0; let b = 2;`

• Operators: `+`, `-`, `*`, `/`, `[]`

• Control Statements: `if (<expr>) {...} else {...}, switch`

• Loops: `for`, `while`, `do-while`

• Arrays: `var a = [1,2,3]; a[99] = 100; console.log(a.length);`

• Functions: `function myFunction(a,b) { return a + b; }`

• Objects: `var obj; obj.x = 3; obj.y = 5;`
  - Prototypes for instance functions

• Comments are `/* Comment */` or `// Single-line Comment`
Including JavaScript in HTML

• Use the script tag
• Can either inline JavaScript or load it from an external file
  - `<script type="text/javascript">
    a = 5, b = 8;
    c = a * b + b - a;
  </script>`
  ```javascript
  <script type="text/javascript" src="script.js"/>
  ```
• Script tag can reference local or remote external javascript files
• The order the javascript is in is the order it is executed
• Example: in the above, `script.js` can access the variables a, b, and c
JavaScript Features

• Any object can serve as an associative array

```javascript
states = {"AZ": "Arizona", "MA": "Massachusetts"};
```

• Array functions: map, filter, reduce, forEach

```javascript
- Object.keys(states).filter(d => d.startsWith("A"));
```

• Function chaining is common (sometimes the original object is returned, others another object is returned)

```javascript
- $('#myElt').css("color", "blue").height(200).width(320)
```

• Closures are functions that "remember their environments" [MDN]

```javascript
- function makeAdder(x) {
  return function(y) {
    return x + y;
  };
}
var add5 = makeAdder(5);
```
JavaScript Objects

- var student = {name: "John Smith", id: "000012345", class: "Senior", hometown: "Peoria, IL, USA"};

- Objects contain multiple values: key-value pairs called **properties**

- Accessing properties via dot-notation: `student.name`

- Always works via bracket-notation: `student["name"]`

- May also contain functions:
  - var student = {firstName: "John",
                 lastName: "Smith",
                 fullName: function() { return this.firstName + " " + this.lastName; }};
  - student.fullName()
Functional Programming in JavaScript

• Functions are first-class objects in JavaScript
• You can pass a function to a method just like you can pass an integer, string, or object
• Instead of writing loops to process data, we can instead use a map/filter/reduce/forEach function on the data that runs our logic for each data item

• **map**: transform each element of an array
• **filter**: check each element of an array and keep only ones that pass
• **forEach**: run the function for each element of the array
• **reduce**: collapse an array to a single object
Using Array Functions

- var a = [2, 4, 7, 11, 22, 84];

- Named function:
  - function isEven(d) {
      return (d % 2 == 0);
    }
    a.filter(isEven);

- Anonymous function
  - a.filter(function(d) { return (d % 2 == 0); });

- Arrow function
  - a.filter(d => (d % 2 == 0));
Manipulating the DOM with JavaScript

- Key global variables:
  - `window`: Global namespace
  - `document`: Current document
  - `document.getElementById(...)`: Get an element via its id
- HTML is parsed into an in-memory document (DOM)
- Can access and **modify** information stored in the DOM
- Can add information to the DOM
Example: JavaScript and the DOM

• Start with no real content, just divs:

```html
<div id="firstSection"></div>
<div id="secondSection"></div>
<div id="finalSection"></div>
```

• Get existing elements:
  - `document.querySelector`
  - `document.getElementById`

• Programmatically add elements:
  - `document.createElement`
  - `document.createTextNode`
  - `Element.appendChild`
  - `Element.setAttribute`

• Link
Assignment 2

• Link
• Three parts: table, horizontal bar chart, vertical bar chart
  - data processing
  - highlighting (CS 680)
• Vertical chart can be tricky
• Start early!
• Questions?
Creating SVG figures via JavaScript

• SVG elements can be accessed and modified just like HTML elements
• Create a new SVG programmatically and add it into a page:
  - var divElt = document.getElementById("chart");
    var svg = document.createElementNS(
      "http://www.w3.org/2000/svg",
      "svg")
    divElt.appendChild(svg);
• You can assign attributes:
  - svg.setAttribute("height", 400);
    svg.setAttribute("width", 600);
    svgCircle.setAttribute("r", 50);
Manipulating SVG via JavaScript

• SVG can be navigated just like the DOM

• Example:

```javascript
function addEltToSVG(svg, name, attrs) {
    var element = document.createElementNS(
        "http://www.w3.org/2000/svg", name);
    if (attrs === undefined) attrs = {};
    for (var key in attrs) {
        element.setAttribute(key, attrs[key]);
    }
    svg.appendChild(element);
}
mysvg = document.getElementById("mysvg");
addEltToSVG(mysvg, "rect", {
    "x": 50, "y": 50,
    "width": 40,"height": 40,
    "fill": "blue"});
```

• **Notebook**
SVG Manipulation Example

• Draw a horizontal bar chart
  
  ```javascript
  var a = [6, 2, 6, 10, 7, 18, 0, 17, 20, 6];
  ```

• Steps?
SVG Manipulation Example

• Draw a horizontal bar chart
  - `var a = [6, 2, 6, 10, 7, 18, 0, 17, 20, 6];`

• Steps:
  - Programmatically create SVG
  - Create individual rectangle for each item

• Link:
  - [https://codepen.io/dakoop/pen/mdbxQKe](https://codepen.io/dakoop/pen/mdbxQKe)
“Computer-based visualization systems provide visual representations of **datasets** designed to help people carry out tasks more effectively.”

— T. Munzner
Data

• What is this data?

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R011</td>
<td>42ND STREET &amp; 8TH AVENUE</td>
<td>00228985</td>
<td>00008471</td>
<td>00000441</td>
<td>00001455</td>
<td>00000134</td>
<td>00033341</td>
<td>00071255</td>
<td></td>
</tr>
<tr>
<td>R170</td>
<td>14TH STREET-UNION SQUARE</td>
<td>00224603</td>
<td>00011051</td>
<td>00000827</td>
<td>00003026</td>
<td>00000660</td>
<td>00089367</td>
<td>00199841</td>
<td></td>
</tr>
<tr>
<td>R046</td>
<td>42ND STREET &amp; GRAND CENTRAL</td>
<td>00207758</td>
<td>00007908</td>
<td>00000323</td>
<td>00001183</td>
<td>00003001</td>
<td>00040759</td>
<td>00096613</td>
<td></td>
</tr>
</tbody>
</table>

• **Semantics**: real-world meaning of the data

• **Type**: structural or mathematical interpretation

• Both often require **metadata**
  - Sometimes we can infer some of this information
  - Line between data and metadata isn’t always clear
Semantics

• The meaning of the data
• Example: 94023, 90210, 52790, 02747
Semantics

• The meaning of the data
• Example: 94023, 90210, 52790, 02747
  - Attendance at college football games?
Semantics

• The meaning of the data
• Example: 94023, 90210, 52790, 02747
  - Attendance at college football games?
  - Salaries?
Semantics

• The meaning of the data
• Example: 94023, 90210, 52790, 02747
  - Attendance at college football games?
  - Salaries?
  - Zip codes?
• Cannot always infer based on what the data looks like
• Often require semantics to better understand data
• Column names help with semantics
• May also include rules about data: a zip code is part of an address that uniquely identifies a residence
• Useful for asking good questions about the data
## Data

<table>
<thead>
<tr>
<th>REMOTE</th>
<th>STATION</th>
<th>FF</th>
<th>SEN/DIS</th>
<th>7-D AFAS UNL</th>
<th>D AFAS/RMF</th>
<th>JOINT RR TKT</th>
<th>7-D UNL</th>
<th>30-D UNL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R011 42ND STREET &amp; 8TH AVENUE</td>
<td>00228985</td>
<td>00008471</td>
<td>00000441</td>
<td>00001455</td>
<td>00000134</td>
<td>00033341</td>
<td>00071255</td>
</tr>
<tr>
<td>2</td>
<td>R170 14TH STREET-UNION SQUARE</td>
<td>00224603</td>
<td>00011051</td>
<td>00000827</td>
<td>00003026</td>
<td>00000660</td>
<td>00089367</td>
<td>00199841</td>
</tr>
<tr>
<td>3</td>
<td>R046 42ND STREET &amp; GRAND CENTRAL</td>
<td>00207758</td>
<td>00007908</td>
<td>00000323</td>
<td>00001183</td>
<td>00003001</td>
<td>00040759</td>
<td>00096613</td>
</tr>
<tr>
<td>4</td>
<td>R012 34TH STREET &amp; 8TH AVENUE</td>
<td>00188311</td>
<td>00006490</td>
<td>00000498</td>
<td>00001279</td>
<td>00003622</td>
<td>00035527</td>
<td>00067483</td>
</tr>
<tr>
<td>5</td>
<td>R293 34TH STREET - PENN STATION</td>
<td>00168768</td>
<td>00006155</td>
<td>00000523</td>
<td>00001065</td>
<td>00005031</td>
<td>00030645</td>
<td>00054376</td>
</tr>
<tr>
<td>6</td>
<td>R033 42ND STREET/TIMES SQUARE</td>
<td>00159382</td>
<td>00005945</td>
<td>00000378</td>
<td>00001205</td>
<td>00000690</td>
<td>00058931</td>
<td>00078644</td>
</tr>
<tr>
<td>7</td>
<td>R022 34TH STREET &amp; 6TH AVENUE</td>
<td>00156008</td>
<td>00006276</td>
<td>00000487</td>
<td>00001543</td>
<td>00000712</td>
<td>00058910</td>
<td>00110466</td>
</tr>
<tr>
<td>8</td>
<td>R084 59TH STREET/COLUMBUS CIRCLE</td>
<td>00155262</td>
<td>00009484</td>
<td>00000589</td>
<td>00002071</td>
<td>00000542</td>
<td>00053397</td>
<td>00113966</td>
</tr>
<tr>
<td>9</td>
<td>R020 47-50 STREETS/ROCKEFELLER</td>
<td>00143500</td>
<td>00006402</td>
<td>00000384</td>
<td>00001159</td>
<td>00000723</td>
<td>00037978</td>
<td>00090745</td>
</tr>
<tr>
<td>10</td>
<td>R179 86TH STREET-LEXINGTON AVE</td>
<td>00142169</td>
<td>00010367</td>
<td>00000470</td>
<td>00001839</td>
<td>00000271</td>
<td>00050328</td>
<td>00125250</td>
</tr>
<tr>
<td>11</td>
<td>R023 34TH STREET &amp; 6TH AVENUE</td>
<td>00134052</td>
<td>00005005</td>
<td>00000348</td>
<td>00001112</td>
<td>00000649</td>
<td>00031531</td>
<td>00075040</td>
</tr>
<tr>
<td>12</td>
<td>R029 PARK PLACE</td>
<td>00121614</td>
<td>00004311</td>
<td>00000287</td>
<td>00000931</td>
<td>00000792</td>
<td>00025404</td>
<td>00065362</td>
</tr>
<tr>
<td>13</td>
<td>R047 42ND STREET &amp; GRAND CENTRAL</td>
<td>00100742</td>
<td>00004273</td>
<td>00000185</td>
<td>00000704</td>
<td>00001241</td>
<td>00022808</td>
<td>00068216</td>
</tr>
</tbody>
</table>
Data Terminology

• Items
  - An item is an individual discrete entity
  - e.g. row in a table, node in a network

• Attributes
  - An attribute is some specific property that can be measured, observed, or logged
  - a.k.a. variable, (data) dimension
  - e.g. a column in a table
## Items & Attributes

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>S</th>
<th>T</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order ID</td>
<td>Order Date</td>
<td>Order Priority</td>
<td>Product Container</td>
<td>Product Base Margin</td>
<td>Ship Date</td>
</tr>
<tr>
<td>3</td>
<td>10/14/06</td>
<td>5-Low</td>
<td>Large Box</td>
<td>0.8</td>
<td>10/21/06</td>
</tr>
<tr>
<td>6</td>
<td>2/21/08</td>
<td>4-Not Specified</td>
<td>Small Box</td>
<td>0.55</td>
<td>2/22/08</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Small Pack</td>
<td>0.79</td>
<td>7/17/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Jumbo Box</td>
<td>7/17/07</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>7/18/07</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>7/18/07</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>Wrap Bag</td>
<td>0.52</td>
<td>10/24/07</td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>Small Box</td>
<td>0.58</td>
<td>10/25/07</td>
</tr>
<tr>
<td>36</td>
<td>11/3/07</td>
<td>1-Urgent</td>
<td>Small Box</td>
<td>0.55</td>
<td>11/3/07</td>
</tr>
<tr>
<td>65</td>
<td>3/18/07</td>
<td>1-Urgent</td>
<td>Small Pack</td>
<td>0.49</td>
<td>3/19/07</td>
</tr>
<tr>
<td>66</td>
<td>3/20/05</td>
<td>5-Low</td>
<td>Wrap Bag</td>
<td>0.56</td>
<td>1/20/05</td>
</tr>
<tr>
<td>69</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.59</td>
<td>12/23/06</td>
</tr>
<tr>
<td>69</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Wrap Bag</td>
<td>0.82</td>
<td>12/23/06</td>
</tr>
<tr>
<td>96</td>
<td>4/17/05</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.55</td>
<td>4/19/05</td>
</tr>
<tr>
<td>97</td>
<td>1/29/06</td>
<td>3-Medium</td>
<td>Small Box</td>
<td>0.38</td>
<td>1/30/06</td>
</tr>
<tr>
<td>129</td>
<td>11/19/08</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.37</td>
<td>11/28/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.37</td>
<td>5/9/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.38</td>
<td>5/10/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.6</td>
<td>5/11/08</td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Medium Box</td>
<td>0.6</td>
<td>6/12/06</td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Jumbo Box</td>
<td>0.69</td>
<td>6/14/06</td>
</tr>
<tr>
<td>134</td>
<td>5/1/08</td>
<td>4-Not Specified</td>
<td>Large Box</td>
<td>0.82</td>
<td>5/3/08</td>
</tr>
<tr>
<td>135</td>
<td>10/21/07</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.64</td>
<td>10/23/07</td>
</tr>
<tr>
<td>166</td>
<td>9/12/07</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.55</td>
<td>9/14/07</td>
</tr>
<tr>
<td>193</td>
<td>8/8/06</td>
<td>1-Urgent</td>
<td>Medium Box</td>
<td>0.57</td>
<td>8/10/06</td>
</tr>
<tr>
<td>194</td>
<td>4/5/08</td>
<td>3-Medium</td>
<td>Wrap Bag</td>
<td>0.42</td>
<td>4/7/08</td>
</tr>
</tbody>
</table>
Data Types

• Nodes
  - Synonym for item but in the context of networks (graphs)

• Links
  - A **link** is a relation between two items
  - e.g. social network friends, computer network links
Data Types

• Positions:
  - A position is a location in space (usually 2D or 3D)
  - May be subject to projections
  - e.g. cities on a map, a sampled region in an CT scan

• Grids:
  - A grid specifies how data is sampled both geometrically and topologically
  - e.g. how CT scan data is stored
Positions and Grids
Dataset Types

- **Tables**
  - Attributes (columns)
  - Items (rows)
  - Cell containing value

- **Networks**
  - Link
  - Node (item)

- **Fields (Continuous)**
  - Grid of positions
  - Cell
  - Attributes (columns)
  - Value in cell

- **Geometry (Spatial)**
  - Position

- **Multidimensional Table**
  - Key 1
  - Key 2
  - Attributes
  - Value in cell

- **Trees**

[Munzner (ill. Maguire), 2014]
# Tables

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>S</th>
<th>T</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Order Date</td>
<td>Order Priority</td>
<td>Product Container</td>
<td>Product Base Margin</td>
<td>Ship Date</td>
</tr>
<tr>
<td>3</td>
<td>10/14/06</td>
<td>S-Low</td>
<td>Large Box</td>
<td>0.8</td>
<td>10/21/06</td>
</tr>
<tr>
<td>6</td>
<td>2/21/08</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.55</td>
<td>2/22/08</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Small Pack</td>
<td>0.79</td>
<td>7/17/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Jumbo Box</td>
<td>0.98</td>
<td>7/17/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.93</td>
<td>7/18/07</td>
</tr>
<tr>
<td>32</td>
<td>7/16/07</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.93</td>
<td>7/18/07</td>
</tr>
<tr>
<td>35</td>
<td>10/23/07</td>
<td>4-Not Specified</td>
<td>Wrap Bag</td>
<td>0.52</td>
<td>10/24/07</td>
</tr>
<tr>
<td>35</td>
<td>12/18/07</td>
<td>4-Not Specified</td>
<td>Small Box</td>
<td>0.58</td>
<td>10/25/07</td>
</tr>
<tr>
<td>36</td>
<td>11/3/07</td>
<td>1-Urgent</td>
<td>Small Box</td>
<td>0.55</td>
<td>11/3/07</td>
</tr>
<tr>
<td>65</td>
<td>3/18/07</td>
<td>1-Urgent</td>
<td>Small Pack</td>
<td>0.49</td>
<td>3/19/07</td>
</tr>
<tr>
<td>66</td>
<td>1/20/05</td>
<td>S-Low</td>
<td>Wrap Bag</td>
<td>0.56</td>
<td>1/20/05</td>
</tr>
<tr>
<td>69</td>
<td>12/18/06</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.44</td>
<td>6/6/05</td>
</tr>
<tr>
<td>69</td>
<td>12/18/06</td>
<td>S-Low</td>
<td>Wrap Bag</td>
<td>0.6</td>
<td>6/6/05</td>
</tr>
<tr>
<td>70</td>
<td>12/18/06</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.59</td>
<td>12/23/06</td>
</tr>
<tr>
<td>96</td>
<td>4/17/05</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.55</td>
<td>4/19/05</td>
</tr>
<tr>
<td>97</td>
<td>1/29/06</td>
<td>3-Medium</td>
<td>Small Box</td>
<td>0.38</td>
<td>1/30/06</td>
</tr>
<tr>
<td>129</td>
<td>11/19/08</td>
<td>5-Low</td>
<td>Small Box</td>
<td>0.37</td>
<td>11/28/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.37</td>
<td>5/9/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Medium Box</td>
<td>0.38</td>
<td>5/10/08</td>
</tr>
<tr>
<td>130</td>
<td>5/8/08</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.6</td>
<td>5/11/08</td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Medium Box</td>
<td>0.6</td>
<td>6/12/06</td>
</tr>
<tr>
<td>132</td>
<td>6/11/06</td>
<td>3-Medium</td>
<td>Jumbo Box</td>
<td>0.69</td>
<td>6/14/06</td>
</tr>
<tr>
<td>134</td>
<td>5/1/08</td>
<td>4-Not Specified</td>
<td>Large Box</td>
<td>0.82</td>
<td>5/3/08</td>
</tr>
<tr>
<td>135</td>
<td>10/21/07</td>
<td>4-Not Specified</td>
<td>Small Pack</td>
<td>0.64</td>
<td>10/23/07</td>
</tr>
<tr>
<td>166</td>
<td>9/12/07</td>
<td>2-High</td>
<td>Small Box</td>
<td>0.55</td>
<td>9/14/07</td>
</tr>
<tr>
<td>193</td>
<td>8/8/06</td>
<td>1-Urgent</td>
<td>Medium Box</td>
<td>0.57</td>
<td>8/10/06</td>
</tr>
<tr>
<td>194</td>
<td>4/5/08</td>
<td>3-Medium</td>
<td>Wrap Bag</td>
<td>0.42</td>
<td>4/7/08</td>
</tr>
</tbody>
</table>
Tables

- Data organized by rows & columns
  - row ~ item (usually)
  - column ~ attribute
  - label ~ attribute name
- Key: identifies each item (row)
  - Usually **unique**
  - Allows **join** of data from 2+ tables
  - Compound key: key split among multiple columns, e.g. (state, year) for population
- Multidimensional:
  - Split compound key: data cube with (state, year)

[Munzner (ill. Maguire), 2014]
Networks

- Why networks instead of graphs?
- Tables can represent networks
  - Many-many relationships
  - Also can be stored as specific graph databases or files
Networks

Figure 7: US airlines graph (235 nodes, 2101 edges) (a) not bundled and bundled using (b) FDEB with inverse-linear model, (c) GBEB, and (d) FDEB with inverse-quadratic model.

Figure 8: US migration graph (1715 nodes, 9780 edges) (a) not bundled and bundled using (b) FDEB with inverse-linear model, (c) GBEB, and (d) FDEB with inverse-quadratic model. The same migration flow is highlighted in each graph.

Figure 9: A low amount of straightening provides an indication of the number of edges comprising a bundle by widening the bundle. (a) $s = 0$, (b) $s = 10$, and (c) $s = 40$. If $s$ is 0, color more clearly indicates the number of edges comprising a bundle.

We generated our data using the rendering technique described in Section 4.1. To facilitate the comparison of migration flow in Figure 8, we use a similar rendering technique as the one that Cui et al. [CZQ108] used to generate Figure 8c.

The airlines graph is comprised of 235 nodes and 2101 edges. It took 19 seconds to calculate the bundled airlines graphs (Figures 7b and 7d) using the calculation scheme presented in Section 3.3. The migration graph is comprised of 1715 nodes and 9780 edges. It took 80 seconds to calculate the bundled migration graphs (Figures 8b and 8d) using the same calculation scheme. All measurements were performed on an Intel Core 2 Duo 2.66GHz PC running Windows XP with 2GB of RAM and a GeForce 8800GT graphics card.

Our prototype was implemented in Borland Delphi 7.

[Holten & van Wijk, 2009]
Networks

Figure 7: US airlines graph (235 nodes, 2101 edges) (a) not bundled and bundled using (b) FDEB with inverse-linear model, (c) GBEB, and (d) FDEB with inverse-quadratic model.

Figure 8: US migration graph (1715 nodes, 9780 edges) (a) not bundled and bundled using (b) FDEB with inverse-linear model, (c) GBEB, and (d) FDEB with inverse-quadratic model. The same migration flow is highlighted in each graph.

Figure 9: A low amount of straightening provides an indication of the number of edges comprising a bundle by widening the bundle. (a) $s = 0$, (b) $s = 10$, and (c) $s = 40$. If $s$ is 0, color more clearly indicates the number of edges comprising a bundle.

We generated the bundled US airlines graph (Figures 7b and 7d) using the calculation scheme presented in Section 3.3. The airlines graph is comprised of 235 nodes and 2101 edges. It took 19 seconds to calculate the bundled airlines graphs using the same calculation scheme. All measurements were performed on an Intel Core 2 Duo 2.66GHz PC running Windows XP with 2GB of RAM and a GeForce 8800GT graphics card. Our prototype was implemented in Borland Delphi 7.

[Holten & van Wijk, 2009]
Fields

Scalar Fields
(Order-0 Tensor Fields)

Vector Fields
(Order-1 Tensor Fields)

Tensor Fields
(Order-2+)

Each point in space has an associated...

\[ s_0 \]

Scalar

\[ \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix} \]

Vector

\[
\begin{bmatrix}
\sigma_{00} & \sigma_{01} & \sigma_{02} \\
\sigma_{10} & \sigma_{11} & \sigma_{12} \\
\sigma_{20} & \sigma_{21} & \sigma_{22}
\end{bmatrix}
\]

Tensor
Fields

- Difference between **continuous** and **discrete** values
- Examples: temperature, pressure, density
- **Grids** necessary to sample continuous data:

  ![Diagram of grids](image)

  - uniform
  - rectilinear
  - structured
  - unstructured

- **Interpolation**: “how to show values between the sampled points in ways that do not mislead”
Spatial Data Example: MRI

[via Levine, 2014]
Scivis and Infovis

- Two subfields of visualization
- **Scivis** deals with data where the spatial position is given with data
  - Usually continuous data
  - Often displaying physical phenomena
  - Techniques like isosurfacing, volume rendering, vector field vis
- In **Infovis**, the data has no set spatial representation, designer chooses how to visually represent data
SciVis

[Google Image Search for “scientific visualization”, 2017]
InfoVis

[Google Image Search for "information visualization", 2017]