The keyhole problem
The keyhole problem
The keyhole problem
Schneiderman’s Mantra

Overview first, zoom and filter, details on demand
Data scalability

There is always more data
Solution 1: pixel space

Keep squishing those representations
Solution 1: pixel space
if (IS_GET_DAT (pcb1a_ptr))
{
    /* Get DSL Information */
    /* the macro call IS_GET_DAT has just
    * the DPB for us, how lucky we are!! */
    dslinfo.dsl = dpb_ptr->port;
    /* Use RTretval for RTERRASSRT/RTT */
    RTretval = DBfrdtpup(RLDSLEQUIP, &i);
    switch( RTretval )
        case GLSUCCEED:
            break;
        case DPNO_MATCH:
            /* Do Stuff */
}
```javascript
function makeScatterplot()
{
    var margin = {top:20, bottom:20, left:60, right: 20};
    var width = 500, height = 500;
    var xValue = function(d){return d[0];};
    var yValue = function(d){return d[1];};
    var xScale = d3.scale.linear();
    var yScale = d3.scale.linear();
    var xAxis = d3.svg.axis().scale(xScale).orient("bottom");
    var yAxis = d3.svg.axis().scale(yScale).orient("left");

    function chart(selection){
        selection.each(function(data){
            xScale.range([0,width - margin.left - margin.right])
                .nice()
                .domain(d3.extent(data, xValue));

            yScale.range([height - margin.top - margin.bottom, 0])
                .nice()
                .domain(d3.extent(data, yValue));

            var svg = d3.select(this).append("svg")
                .attr({width:width, height:height});

            var canvas = svg.append("g")
                .attr("transform","translate("+margin.left+","+margin.top+")");

            // create the dots
            var dots = canvas.selectAll("circle")
                .data(data)
                .enter()
                .append("circle");
```

(Additional code continues below the visible part of the snippet.)
Solution 1: pixel space

Get bigger screens!
Solution 2: data space / attribute space

Reduce # of attributes

Reduce # of items

Reduce range of items

<table>
<thead>
<tr>
<th>doctor</th>
<th>name</th>
<th>companions</th>
<th>start</th>
<th>end</th>
<th>episodes</th>
<th>duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>William Hartnell</td>
<td>10</td>
<td>1963</td>
<td>1966</td>
<td>135</td>
<td>3288</td>
</tr>
<tr>
<td>2</td>
<td>Patrick Troughton</td>
<td>5</td>
<td>1966</td>
<td>1970</td>
<td>127</td>
<td>3183</td>
</tr>
<tr>
<td>3</td>
<td>Jon Pertwee</td>
<td>3</td>
<td>1970</td>
<td>1974</td>
<td>129</td>
<td>3206</td>
</tr>
<tr>
<td>4</td>
<td>Tom Baker</td>
<td>8</td>
<td>1974</td>
<td>1982</td>
<td>174</td>
<td>4248</td>
</tr>
<tr>
<td>5</td>
<td>Peter Davidson</td>
<td>6</td>
<td>1982</td>
<td>1984</td>
<td>69</td>
<td>1800</td>
</tr>
<tr>
<td>6</td>
<td>Colin Baker</td>
<td>2</td>
<td>1984</td>
<td>1987</td>
<td>31</td>
<td>1029</td>
</tr>
<tr>
<td>7</td>
<td>Sylvester McCoy</td>
<td>2</td>
<td>1987</td>
<td>1989</td>
<td>42</td>
<td>1025</td>
</tr>
<tr>
<td>8</td>
<td>Paul McGann</td>
<td>1</td>
<td>1996</td>
<td>1996</td>
<td>1</td>
<td>84</td>
</tr>
<tr>
<td>9</td>
<td>Christopher Eccleston</td>
<td>3</td>
<td>2005</td>
<td>2005</td>
<td>13</td>
<td>568</td>
</tr>
<tr>
<td>10</td>
<td>David Tennant</td>
<td>5</td>
<td>2005</td>
<td>2010</td>
<td>48</td>
<td>2368</td>
</tr>
<tr>
<td>11</td>
<td>Matt Smith</td>
<td>4</td>
<td>2010</td>
<td>2013</td>
<td>44</td>
<td>2083</td>
</tr>
</tbody>
</table>
Elimination

Eliminate items

Eliminate attributes
Aggregation

CS101 Midterm Grades Histogram
Aggregation
What to group by?
- categorical data or shared data values
- spatial position
- algorithmic (i.e., clustering based on attributes)
- user defined

How to group?
- math function on attributes (e.g., min, max, mean, mode, sum, count, etc...)
- semantics or shared abstraction
Pixel-level binning
Pixel-level binning
Aggregation
Aggregation

[Graph with labeled axes: MPG, Cylinders, Horsepower, Weight, Acceleration, Year, Origin. Specific values are listed along each axis.]
Aggregation
Aggregation
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.

To facilitate the comparison of migration flow in Figure 8, we use a similar rendering technique as the one that Cui et al. [CZQ 08] 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.

Holden and van Wijk, “Force Directed Edge Bundling for Graph Visualization”, 2009
Clustering

Clustering

InSpire, PNNL