A Overview of Information Visualization

Michael McGuffin, Ph.D.

Associate Professor
Department of Software and IT Engineering
École de technologie supérieure (ETS)
Montreal, Canada
http://profs.etsmtl.ca/mmcguffin/
SECTION 1: Visualization of multidimensional multivariate (mdmv) data, also known as tabular data or relational data
Some elementary math

• Given two sets, a \textit{domain} (example: \(\mathbb{R}\)) and a \textit{codomain} (example: \(\mathbb{R}\)), we can form the \textit{cartesian produit} (\(\mathbb{R} \times \mathbb{R} = \mathbb{R}^2\)) that is the set of all possible pairs \((x,y)\)
  – Other examples of cartesian products:: \(A \times B = \{(a,b) \mid a \in A \text{ and } b \in B\}\);
    \(A \times B \times C \times D = \{(a,b,c,d) \mid a \in A \text{ and } b \in B \text{ and } c \in C \text{ and } d \in D\}\)

• A \textit{relation} is a subset of a cartesian product
  – Example: the equation \(x = y^2\) corresponds to a subset of \(\mathbb{R}^2\); the inequation \(x < y\) corresponds to another subset of \(\mathbb{R}^2\)

• A relation is a \textit{fonction} if each member \(x\) of the domain has at most one corresponding member \(y\) in the codomain
  – \(x = y^2\) is not a function because \((4,2)\) and \((4,-2)\) are both members of the relation defined by the equation

• A simple way to represent a relation (or a function) is to simply enumerate the pairs of the relation in a table
The function $y = x^{0.5}$:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
</tr>
</tbody>
</table>

Examples of mathematical relations (i.e., multidimensional multivariate data). A relation is a subset of a cartesian product of two or more sets (example: a subset of $\mathbb{R} \times \mathbb{R}$). In the examples here, each row is a tuple (member of the relation), and each column is a set within the cartesian product.

The relation in a table in a relational database:

<table>
<thead>
<tr>
<th>Client_name</th>
<th>Product_bought</th>
<th>Price</th>
<th>Date</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robert G.</td>
<td>Trombone</td>
<td>500.00</td>
<td>2008 March 7</td>
<td>.</td>
</tr>
<tr>
<td>Robert G.</td>
<td>Partitions vol. 1</td>
<td>45.00</td>
<td>2008 March 7</td>
<td>.</td>
</tr>
<tr>
<td>Lucie M.</td>
<td>Flute</td>
<td>180.00</td>
<td>2007 Nov 11</td>
<td>.</td>
</tr>
<tr>
<td>Cynthia S.</td>
<td>Partitions vol. 2</td>
<td>40.00</td>
<td>2008 June 16</td>
<td>.</td>
</tr>
<tr>
<td>Jules T.</td>
<td>Piano</td>
<td>6000.00</td>
<td>2008 Jan 10</td>
<td>.</td>
</tr>
<tr>
<td>Jules T.</td>
<td>Partitions vol. 1</td>
<td>45.00</td>
<td>2008 Jan 13</td>
<td>.</td>
</tr>
</tbody>
</table>

A video (for example, an .avi file):

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>time</th>
<th>red</th>
<th>green</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>200</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>255</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.1</td>
<td>255</td>
<td>200</td>
<td>190</td>
</tr>
</tbody>
</table>
Terminology

A video:

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>time</th>
<th>red</th>
<th>green</th>
<th>blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>255</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>200</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>255</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.1</td>
<td>255</td>
<td>200</td>
<td>190</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Domains**
  - Independent variables
  - Dimensions

- **Co-domains**
  - Dependant variables
  - Variables (hence the term “mdmv”)
  - Measures (database terminology)

- **Columns, Variables**

- **Tuple, Multidimensional point, Vector, Row**

I’ll use the terms in **bold**
To visualize data, we need to choose a mapping

Input data: an arbitrary number of dimensions and measures

Output graphical representation: at most 3 spatial dimensions (often just 2), at most 1 temporal dimension (animation)
1 dimension + 1 measure: barchart (or linechart)

<table>
<thead>
<tr>
<th>Region (dimension)</th>
<th>Sales (measure)</th>
</tr>
</thead>
<tbody>
<tr>
<td>North</td>
<td>90</td>
</tr>
<tr>
<td>South</td>
<td>49</td>
</tr>
<tr>
<td>East</td>
<td>26</td>
</tr>
<tr>
<td>West</td>
<td>67</td>
</tr>
</tbody>
</table>
2 measures: scatterplot

<table>
<thead>
<tr>
<th>Year (dimension)</th>
<th>Sales (measure)</th>
<th>Expenses (measure)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>2010</td>
<td>27</td>
<td>7</td>
</tr>
<tr>
<td>2011</td>
<td>62</td>
<td>20</td>
</tr>
<tr>
<td>2012</td>
<td>80</td>
<td>32</td>
</tr>
<tr>
<td>2013</td>
<td>30</td>
<td>45</td>
</tr>
</tbody>
</table>
2 dimensions + 1 measure: heatmap

<table>
<thead>
<tr>
<th>Year (dimension)</th>
<th>Region (dimension)</th>
<th>Sales (measure)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>North</td>
<td>0</td>
</tr>
<tr>
<td>2009</td>
<td>South</td>
<td>10</td>
</tr>
<tr>
<td>2009</td>
<td>East</td>
<td>70</td>
</tr>
<tr>
<td>2009</td>
<td>West</td>
<td>2</td>
</tr>
<tr>
<td>2010</td>
<td>North</td>
<td>4</td>
</tr>
<tr>
<td>2010</td>
<td>South</td>
<td>22</td>
</tr>
<tr>
<td>2010</td>
<td>East</td>
<td>71</td>
</tr>
<tr>
<td>2010</td>
<td>West</td>
<td>6</td>
</tr>
<tr>
<td>2011</td>
<td>North</td>
<td>15</td>
</tr>
<tr>
<td>2011</td>
<td>South</td>
<td>68</td>
</tr>
<tr>
<td>2011</td>
<td>East</td>
<td>79</td>
</tr>
<tr>
<td>2011</td>
<td>West</td>
<td>11</td>
</tr>
<tr>
<td>2012</td>
<td>North</td>
<td>27</td>
</tr>
<tr>
<td>2012</td>
<td>South</td>
<td>80</td>
</tr>
<tr>
<td>2012</td>
<td>East</td>
<td>90</td>
</tr>
<tr>
<td>2012</td>
<td>West</td>
<td>50</td>
</tr>
</tbody>
</table>
A video

[Image: Three frames of a video and a 3D diagram with axes labeled Blue, Green, and Red.]

[Gareth Daniel and Min Chen, 2003]
Fluid visualization

Which dimensions and measures would be involved in such data?
Chernoff faces (1973)  
(an example of a "glyph")

Advantage: better than text for getting an overall impression of the data, and for identifying interesting elements.

Disadvantage: the mapping from variables to faces has an effect on the salience of each variable.

Disadvantage (?): redundancy of symmetry of faces.
Other examples of glyphs

Other examples of glyphs

<table>
<thead>
<tr>
<th></th>
<th>$d\theta$</th>
<th>$dm$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Fig. 5. Variety of glyphs considered for magnitude and angular uncertainty.

Fig. 14. Uncertainty glyphs, area scaled to magnitude, Codar vector calculation Method II.

Interactive Presentation from the UN
(United Nations Development Programme, Human Development Report)

Notice that the “points” here are glyphs, each with a radius and a color.

See also Hans Rosling’s presentations on http://www.ted.com
Tableau: software for visualizing databases
(Mackinlay et al. 2007, tableausoftware.com)
This is called *dimensional stacking*, and has been used in previous work before Tableau, but Tableau is a really nice implementation of it, and a successful commercial product.
Tableau

• For more information:
  http://www.tableausoftware.com/products/tour
  http://www.tableausoftware.com/products/desktop/demo
Kinds of variables

• Quantitative (or continuous or metric)
  – Examples: x, y, time, temperature, money

• Ordinal
  – We can put the values in order, but we cannot say that one value is N times larger than some other value
  – Example: level of education (high school diploma, bachelor’s degree, master’s degree, …)

• Categorical (or nominal)
  – There is no natural ordering (except maybe alphabetical, which is arbitrary and language-dependent)
  – Example: food groups (meat, dairy, fruits and vegetables, cereals)
  – Example: bachelor’s in mechanical engineering, bachelor’s in civil engineering, etc.
  – Example: Honda, Toyota, GM, Chrysler, etc.
  – Example: names of countries
Visualization is a mapping

Input data:
each variable can be
{independent, dependent}
i.e., a dimension or measure)
and
{quantitative, ordinal, categorical}

Output graphical representation:
at most 3 spatial dimensions (often just 2),
at most 1 temporal dimension (animation)

... and also many **graphical variables**
Hierarchy of graphical variables (Mackinlay, 1986)

Fig. 14. Accuracy ranking of quantitative perceptual tasks. Higher tasks are accomplished more accurately than lower tasks. Cleveland and McGill empirically verified the basic properties of this ranking.
Hierarchy of graphical variables (Mackinlay, 1986)

Fig. 15. Ranking of perceptual tasks. The tasks shown in the gray boxes are not relevant to these types of data.
A study to confirm the hierarchy
(Jeffrey Heer et Michael Bostock, "Crowdsourcing Graphical Perception: Using Mechanical Turk to Assess Visualization Design", CHI 2010)

Figure 4: Proportional judgment results (Exp. 1A & B). Top: Cleveland & McGill’s [7] lab study. Bottom: MTurk studies. Error bars indicate 95% confidence intervals.
Tableau applies heuristics like these to choose graphics according to the type of data.

<table>
<thead>
<tr>
<th>Continuous variable as a function of a categorical variable</th>
<th>Bar chart (diagramme à barres)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous variable as a function of a continuous variable</td>
<td>Line graph (diagramme à ligne brisée)</td>
</tr>
<tr>
<td>Continuous variable as a function of (ordinal) time</td>
<td></td>
</tr>
<tr>
<td>Two dependent continuous variables</td>
<td>Scatter plot (nuage de points)</td>
</tr>
<tr>
<td>Categorical variable as a function of a continuous variable</td>
<td>Gantt chart</td>
</tr>
<tr>
<td>Categorical independent variable + continuous independent variable</td>
<td></td>
</tr>
<tr>
<td>Two independent categorical variables</td>
<td>Cross tabulation (“cross tab”)</td>
</tr>
</tbody>
</table>
Question 1: Which graph makes it easier to determine whether Mid-Cap U.S. Stock or Small-Cap U.S. Stock has the greater share? (Click one of the two buttons below the graphs.)
Question 4: Which graph makes it easier to focus on the pattern of change through time, instead of the individual values.
Bar chart versus line graph

Which makes it easier to see changes in slope?
Length vs area

Tufte (1983)
Question 2: Which of these line graphs is easier to read?
Question 7: Which graph makes it easier to determine R&D’s travel expense?

2006 Expenses by Department

- 3-D Bar Graph (left)
- 2-D Bar Graphs (below)

2006 Expenses by Department in Millions of USD
Figure 2 - Likelihood of experiencing problems as a function of key risk areas and key process areas
Examples from a course by Marilyn Ostergren at U Washington

( http://courses.washington.edu/info424/Week3Practice_ExcelGraphs.html )

<table>
<thead>
<tr>
<th>Country</th>
<th>beef</th>
<th>horse</th>
<th>lamb</th>
<th>pork</th>
<th>veal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Belgium</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Germany</td>
<td>32</td>
<td>6</td>
<td>5</td>
<td>45</td>
<td>15</td>
</tr>
<tr>
<td>France</td>
<td>37</td>
<td>61</td>
<td>69</td>
<td>29</td>
<td>59</td>
</tr>
<tr>
<td>Italy</td>
<td>16</td>
<td>23</td>
<td>21</td>
<td>9</td>
<td>14</td>
</tr>
<tr>
<td>Netherlands</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>12</td>
<td>9</td>
</tr>
</tbody>
</table>
Question 6: Which map makes it easier to find all of the counties with positive growth rates?

Map A

Map B
Rogowitz and Treinish, “Why Should Engineers and Scientists Be Worried About Color?”
1 Perceptual ordering. (a) We can easily place the gray paint chips in order based on perception, (b) but cannot do this with the colored chips.

2 Spatial contrast sensitivity function. Frequency increases to the right and contrast increases toward the bottom of both images in the figure. We can see detail at much lower contrast in the (a) luminance-varying gray-scale image than with the (b) rainbow color map.

Class exercise:
Design one or more graphics to visualize a data set with the following variables:

• Car model: \{Accord, AMC Pacer, Audi 5000, BMW 320i, Champ, Chev Nova, ...\} (19 models in total, one model for each tuple; i.e. 19 tuples)

• Car price: \[$0, $13500\]

• Car mileage: [0,40]

• Repair record: \{Great, Good, OK, Bad, Terrible\}

• Car weight: [0,5500]
- Car model: \{Accord, AMC Pacer, Audi 5000, BMW 320i, Champ, Chev Nova, \ldots\} (19 models in total, one model for each tuple; i.e. 19 tuples)
- Car price: \[$0, $13500\]
- Car mileage: [0,40]
- Repair record: \{Great, Good, OK, Bad, Terrible\}
- Car weight: [0,5500]
mdmv data

Marks of students taking 4 courses:
• Physics: 90%
• Math: 95%
• French literature: 65%
• History: 70%

Each student is like a tuple:
• (90%, 95%, 65%, 70%)
• Etc.
Parallel Coordinates
(Alfred Inselberg)

Physics  Math  French Literature  History

(90%, 95%, 65%, 70%)
Parallel Coordinates
(Alfred Inselberg)

(30%, 20%, 90%, 90%)
(90%, 95%, 65%, 70%)
Scatterplot Matrix (SPLOM)

Math

French Literature

History

Physics  Math  French Literature

(90%, 95%, 65%, 70%)
Scatterplot Matrix (SPLOM)
Scatterplot Matrix or SPLOM

Within each scatterplot, we could be interested in seeing outliers, correlations, etc.

Notice: the upper triangular half is the same as the lower triangular half, and the diagonal is not very interesting.
Scatterplot Matrix or SPLOM

Note: the diagonal is used to show the names of the variables

Matrix of correlation coefficients

When we have many dimensions, we can summarize each scatterplot by computing its correlation coefficient and displaying only that, instead of displaying all the individual data points. The below interface also allows the user to select one scatterplot and see a zoomed-in view for details.

ScatterDice (Elmqvist et al. 2008)

Fig. 1: Scatterplot matrix navigation for a digital camera dataset. The user is building queries for maximum camera resolution against price ranges and then studies them in relation to release year. The transition is performed using an animated 3D rotation.
Example from Matlab: “carbig.mat”

Summary of techniques for visualizing mdmv data

- 1 dimension + 1 measure:

- 0 dimensions + 2 measures:

- 2 dimensions + 1 measure:

- Several dimensions:

- Many measures:
"Nuts and Bolts" dataset (72 rows):

<table>
<thead>
<tr>
<th>Region</th>
<th>Month</th>
<th>Product</th>
<th>Sales</th>
<th>Equipment_costs</th>
<th>Labor_costs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.76</td>
<td>0.92</td>
<td>4.3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4.92</td>
<td>1.64</td>
<td>4.3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4.2</td>
<td>1</td>
<td>4.3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>8.4</td>
<td>2</td>
<td>4.3</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0</td>
<td>5.28</td>
<td>9.6</td>
<td>4.3</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>1</td>
<td>14.52</td>
<td>26.4</td>
<td>4.3</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
<td>5.016</td>
<td>0.88</td>
<td>4.3</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>1</td>
<td>8.436</td>
<td>1.48</td>
<td>4.3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>0</td>
<td>8.82</td>
<td>2.1</td>
<td>4.9</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>1</td>
<td>9.156</td>
<td>2.18</td>
<td>4.9</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>0</td>
<td>5.655</td>
<td>1.45</td>
<td>5.2</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>1</td>
<td>9.4965</td>
<td>2.435</td>
<td>5.2</td>
</tr>
</tbody>
</table>
“Nuts and Bolts” dataset
“Nuts and Bolts” dataset

Not very useful

The SPLOM works well with measures, but not with dimensions
“Nuts and Bolts” dataset
“Nuts and Bolts” dataset

Parallel coordinates work well with measures, but not with dimensions

Not very useful
“Nuts and Bolts” dataset

Example view that would be possible with Tableau:

Each of the above examples only show 4 of the 6 variables. Showing all 6 variables (3 dimensions and 3 measures) would require much space.
“Nuts and Bolts” dataset

Example view that would be possible with Tableau:

The above example only shows 4 of the 6 variables. One of these is “Month”, which has 12 levels, greatly increasing space requirements.
Generalized PLOt Matrix (GPLOM) of the “Nuts and Bolts” dataset

Scales better than Tableau’s dimensional stacking to a large number of dimensions (and can also show many measures)
Generalized PLOt Matrix (GPLOM)
[Im, McGuffin, Leung, IEEE InfoVis 2013]
SECTION 2: Visualization of networks, also called graphs
Structure of the UN

New Internationalist
issue 375
2005 Jan/Feb
http://www.newint.org
Ishkur’s Guide to Electronic Music
Metabolic pathways

http://www.genome.jp/kegg/pathway/map/map01100.png
How to compute the layout of a graph?

• Many algorithms are available
  – “Graph Drawing” is a research area that grew out of computational geometry
  – See “Graph Drawing” annual conference proceedings
  – See books “Graph Drawing: Algorithms for the Visualization of Graphs” by Di Battista et al. (1999), and “Drawing Graphs: Methods and Models” by Kaufmann and Wagner (2001)
  – Example algorithms: Reingold-Tilford (1981) for binary trees, Sugiyama et al. (1981) for directed acyclic graphs (DAGs)

• Most of these algorithms are not easy to implement
How to compute the layout of a graph? (2)

- One class of algorithms that are easy to implement (in their naïve form) and that are applicable to any graph: **force-directed layout**
Force-directed layout

- Pseudo-physical simulation of masses and springs that converges toward a final layout
- The nodes are masses that are mutually repelled by an electric force
- The edges are springs

http://profs.etsmtl.ca/mmcguffin/code/#SimpleNetworkVisualizer
Force-directed layout

• Simulation of spring forces and repulsive forces, converging to a minimal energy state

• Example:
  – Spring force $F_S = k(x-x_0)$ between each pair of adjacent nodes, where $k$ is a constant, $x$ is the distance between the nodes, and $x_0$ is the rest length of the spring
  – Repulsive force $F_R = \frac{\text{alpha}}{x^2}$ between each pair of nodes, where alpha is a constant, and $x$ is the distance between the nodes
• simulateOneStepOfForceDirectedLayout() {
  // initialization
  for ( int i = 0; i < nodes.size(); ++i ) {
    Node n = nodes.get(i);
    // These will store the net force
    n.forceX = 0;
    n.forceY = 0;
  }
}

...
// repulsive force between all pairs of nodes
for ( int i = 0; i < nodes.size(); ++i ) {
    Node n1 = nodes.get(i);
    for ( int j = i+1; j < nodes.size(); ++j ) {
        Node n2 = nodes.get(j);
        float dx = n2.x - n1.x;
        float dy = n2.y - n1.y;
        if ( dx == 0 && dy == 0 ) {
            dx = (float)Math.random() - 0.5f;
            dy = (float)Math.random() - 0.5f;
        }
        float distanceSquared = dx*dx + dy*dy;
        float distance = (float)Math.sqrt( distanceSquared );
        float force = alpha / distanceSquared;
        dx *= force / distance;
        dy *= force / distance;
        n1.forceX -= dx;
        n1.forceY -= dy;
        n2.forceX += dx;
        n2.forceY += dy;
    }
}

Why j=i+1?
// how to traverse pairs of nodes
for ( int i = 0; i < nodes.size(); ++i ) {
    Node n1 = nodes.get(i);
    // i is the global index of n1
    for ( int j = i+1; j < nodes.size(); ++j ) {
        Node n2 = nodes.get(j);
        // j is the global index of n2
        ...
    }
}

(don’t do this!)
If we did for ( j=0; ...

<table>
<thead>
<tr>
<th>i=0</th>
<th>j=0</th>
<th>j=1</th>
<th>j=2</th>
<th>j=3</th>
<th>j=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>i=1</td>
<td>j=0</td>
<td>j=1</td>
<td>j=2</td>
<td>j=3</td>
<td>j=4</td>
</tr>
<tr>
<td>i=2</td>
<td>j=0</td>
<td>j=1</td>
<td>j=2</td>
<td>j=3</td>
<td>j=4</td>
</tr>
<tr>
<td>i=3</td>
<td>j=0</td>
<td>j=1</td>
<td>j=2</td>
<td>j=3</td>
<td>j=4</td>
</tr>
<tr>
<td>i=4</td>
<td>j=0</td>
<td>j=1</td>
<td>j=2</td>
<td>j=3</td>
<td>j=4</td>
</tr>
</tbody>
</table>

The gray cells are redundant pairs of nodes.

This cell, for example, corresponds to the pair \((n_1, n_3)\)
// spring force
for ( int i = 0; i < nodes.size(); ++i ) {
    Node n1 = nodes.get(i);
    for ( int k = 0; k < n1.neighbours.size(); ++k ) {
        Node n2 = n1.neighbours.get(k);
        int j = network.getIndexOfNode(n2); // get index in the global array
        if ( j < i )
            continue;
        float dx = n2.x - n1.x;
        float dy = n2.y - n1.y;
        float distance = (float)Math.sqrt(dx*dx + dy*dy);
        if ( distance > 0 ) {
            float distanceFromRestLength = distance - SPRING_REST_LENGTH;
            float force = k * distanceFromRestLength;
            dx *= force / distance;
            dy *= force / distance;

            n1.forceX += dx;
            n1.forceY += dy;
            n2.forceX -= dx;
            n2.forceY -= dy;
        }
    }
}
// how to traverse the edges
for ( int i = 0; i < nodes.size(); ++i ) {
    Node n1 = nodes.get(i);
    // i is the global index of n1
    for ( int k = 0; k < n1.neighbours.size(); ++k ) {
        Node n2 = n1.neighbours.get( k );
        // k is the local index of n2 in n1’s list of neighbors
        int j = network.getIndexOfNode( n2 );
        // j is the global index of n2
        if ( j < i )
            continue;
    }
}

The gray cells are skipped.
Notice: 5 edges, 5 white cells, and 5 gray cells.
Force-directed layout
Step 4/4

// update positions
for ( int i = 0; i < nodes.size(); ++i ) {
    Node n = nodes.get(i);

    float dx = timeStep * n.forceX;
    float dy = timeStep * n.forceY;
    float displacementSquared = dx*dx + dy*dy;
    final float MAX_DISPLACEMENT = 10;
    final float MAX_DISPLACEMENT_SQUARED = MAX_DISPLACEMENT * MAX_DISPLACEMENT;
    if ( displacementSquared > MAX_DISPLACEMENT_SQUARED ) {
        float s = MAX_DISPLACEMENT / (float)Math.sqrt( displacementSquared );
        dx *= s;
        dy *= s;
    }
    n.x += dx;
    n.y += dy;
}
} // simulateOneStepOfForceDirectedLayout()
Layout by “stress majorization”

• Garanteed to approach optimal layout (with minimal stress) with each iteration; oscillations are impossible; converges in many fewer iterations than typical force-directed algorithms

• Unfortunately, requires $\Theta(N^2)$ memory, and each iteration requires $\Theta(N^2)$ time (though approximate shortcuts are probably possible for sparse graphs)

• Details: Emden R. Gansner, Yehuda Koren, Stephen North, "Graph Drawing by Stress Majorization", Graph Drawing (GD) 2004; http://scholar.google.ca/scholar?q=Gansner+Koren+North+%22Graph+Drawing+by+Stress+Majorization%22; equation 12
Layout by “stress majorization”

- Takes as input the desired distance $d_{i,j}$ between each pair of nodes $\{n_i, n_j\}$ and the weight $w_{i,j}$ (or importance) of each distance.
- Easy way to compute these inputs: define $d_{i,j} = 1$ if $n_i, n_j$ are adjacent, then compute $d_{i,j}$ for other pairs of nodes with the Floyd–Warshall algorithm for finding all shortest paths (http://en.wikipedia.org/wiki/Floyd–Warshall_algorithm) which requires cubic time, then define $w_{i,j} = 1/(d_{i,j})^2$
- Once all the $d_{i,j}$ and $w_{i,j}$ are known, iterate equation 12 from the paper. The stress majorization algorithm will position nodes to try to respect as closely as possible the desired distances. Iterate until the nodes move very little (e.g., less than 1 pixel).
- An easy optimization (to save memory and accelerate the Floyd–Warshall algorithm) can be performed by noticing that the matrix of $d_{i,j}$ and $w_{i,j}$ values is symmetric
2.3 Localized Optimization

Following the idea of Kamada and Kawai [15], we can fix the positions of all nodes, except some node \( i \). Then, by the same argument given above for the full majorization process, it can be shown that the stress function is decreased by setting the position of \( i \) as follows

\[
X_i^{(a)} \leftarrow \frac{\sum_{j \neq i} w_{ij} \left( X_j^{(a)} + d_{ij} (X_i^{(a)} - X_j^{(a)}) \text{inv}(\|X_i - X_j\|) \right)}{\sum_{j \neq i} w_{ij}}, \quad a = 1, \ldots, d.
\]

(12)

This way we can iterate through all nodes, and in each iteration relocate all the \( d \) coordinates of node \( i \) according to (12). Each iteration is guaranteed to strictly decrease the stress until convergence. Hence, oscillations and non-convergence are impossible.

In practice, we have only used the more involved global process (9) and have no experience yet with the local version. We provide this local version here mainly because it is simple and easy to implement, requiring no equation solver.\(^1\)
Force-Directed Layouts

43 nodes, 80 edges

50 nodes, 200 edges

[McGuffin, 2012]
Arc Diagrams

43 nodes, 80 edges, random ordering, 180 degree arcs

barycenter heuristic ordering

100 degree arcs

[McGuffin, 2012]
Other ways of using arcs: Visualization of music (Martin Wattenberg, 2001)

Beethoven, *Clair de lune*  
Talking Heads, *As She Was*
Other ways of using arcs:
Takanori Makitani

“Flight density during one week between international airports”
http://www.visualcomplexity.com/vc/project.cfm?id=67
Idea Magazine #324 (?), diagram originally from 1968 (?)
Graphs (or networks) can be represented with node-link diagrams, or with adjacency matrices.
A graph displayed as a node-link diagram, and as an adjacency matrix

```
    A
   /|
  B /|
 /  |
D---E

A    B    C    D    E
A 1 1
B 1 1 1
C 1 1
D 1 1
E 1
```

Notice: the matrix is symmetric, because the graph is undirected. A directed graph (digraph) would have an asymmetric matrix. A graph with weighted edges would have a matrix of weights.
43 nodes, 80 edges

[McGuffin, 2012]
Node-link diagram vs adjacency matrix

43 nodes, 80 edges

[McGuffin, 2012]
Lattix
43 nodes, 80 edges, random ordering

barycenter heuristic ordering, 180 degree arcs added à la MatLink [Henry+Fekete 2007]

[McGuffin, 2012]
Three ways to draw a graph

Node-link diagram, force-directed layout
Node-link diagram with circular layout
Adjacency matrix

Images from http://www.nature.com/nmeth/journal/v9/n2/full/nmeth.1862.html
50 nodes, 200 edges

For a tutorial on simple algorithms for these:
http://profs.etsmtl.ca/mmcguffin/research/#mcguffin_2012

[McGuffin, 2012]
Four ways to draw a graph

- Node-link Diagram
- Arc Diagram [Jacques Bertin 1967]
- Adjacency Matrix
- MatLink [Nathalie Henry and Jean-Daniel Fekete 2007]
SECTION 3: Visualization of trees
## Basic Tiramisu (serves 8)

<table>
<thead>
<tr>
<th>Ingredient</th>
<th>Action</th>
<th>Serving</th>
</tr>
</thead>
<tbody>
<tr>
<td>about 20 lady's fingers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 shots espresso</td>
<td>mix &amp; chill</td>
<td>dip</td>
</tr>
<tr>
<td>1/2 cup coffee</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 cup heavy whipping cream</td>
<td>whisk to stiff peaks</td>
<td>layer &amp; spread twice</td>
</tr>
<tr>
<td>1 lb. mascarpone cheese</td>
<td>mix</td>
<td></td>
</tr>
<tr>
<td>1/2 cup granulated sugar</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 tablespoons rum (or brandy)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cocoa powder</td>
<td></td>
<td></td>
</tr>
<tr>
<td>shavings of unsweetened dark chocolate</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Filelight
http://www.methylblue.com/filelight/
A naïve and easy-to-program layout: each subtree has an interval in x that is not overlapped by the neighboring subtrees. A postorder depth-first-traversal combines the intervals of subtrees to yield the interval of a parent node.

A "Reingold Tilford" style layout: saves space in x by moving subtrees together as much as possible. (For details, see section 3 of Christoph Buchheim, Michael Jünger and Sebastian Leipert, "Improving Walker's Algorithm to Run in Linear Time", Proceedings of Symposium on Graph Drawing (GD) 2002, pages 344-353.) The is more complicated to program.
A naïve and easy-to-program layout: each subtree has an interval in x that is not overlapped by the neighboring subtrees. A postorder depth-first-traversal combines the intervals of subtrees to yield the interval of a parent node.

A "Reingold Tilford" style layout: saves space in x by moving subtrees together as much as possible. (For details, see section 3 of Christoph Buchheim, Michael Jünger and Sebastian Leipert, "Improving Walker's Algorithm to Run in Linear Time", Proceedings of Symposium on Graph Drawing (GD) 2002, pages 344-353.) The is more complicated to program.
A naïve and easy-to-program layout: each subtree has an interval in $x$ that is not overlapped by the neighboring subtrees. A postorder depth-first-traversal combines the intervals of subtrees to yield the interval of a parent node.

A "Reingold Tilford" style layout: saves space in $x$ by moving subtrees together as much as possible. (For details, see section 3 of Christoph Buchheim, Michael Jünger and Sebastian Leipert, "Improving Walker's Algorithm to Run in Linear Time", Proceedings of Symposium on Graph Drawing (GD) 2002, pages 344-353.) The is more complicated to program.

Another easy-to-program layout: a preorder depth-first-traversal encounters nodes in order of their $y$ coordinates, and the $x$ coordinate of each node is proportional to its depth.
@article{wetherell1979,
  author = {Charles Wetherell and Alfred Shannon},
  title = {Tidy Drawings of Trees},
  journal = {IEEE Transactions on Software Engineering},
  year = 1979,
  month = {September},
  volume = {SE-5},
  number = 5,
  pages = {514--520}
}

@article{reingold1981,
  author = {Edward M. Reingold and John S. Tilford},
  title = {Tidier Drawings of Trees},
  journal = {IEEE Transactions on Software Engineering},
  year = 1981,
  month = {March},
  volume = {SE-7},
  number = 2,
  pages = {223--228}
}

@article{walker1990,
  author = {{Walker II}, John Q.},
  title = {A Node-Positioning Algorithm for General Trees},
  journal = {Software---Practice and Experience},
  year = 1990,
  month = {July},
  volume = 20,
  number = 7,
  pages = {685--705}
}

@inproceedings{buchheim2002,
  author = {Christoph Buchheim and Michael J"unger and Sebastian Leipert},
  title = {Improving {Walker's} Algorithm to Run in Linear Time},
  booktitle = {conf_gd},
  year = 2002,
  pages = {344--353}
}
Treemaps
(Ben Shneiderman and others)
http://www.cs.umd.edu/hcil/treemap-history/

Marc Smith and Andrew Fiore, 2001

Martin Wattenberg, 1998
and
http://www.smartmoney.com/map-of-the-market/
Trees

Figure 2: common tree representations, each showing the same tree in a different way. A: node-link. B: nested containment, or nested enclosure. C: use of alignment and adjacency. D: indented outline style.

Zhao, McGuffin, and Chignell 2005
Trees

Michael McGuffin and Jean-Marc Robert, 2010
Asymptotic Analysis of the Space-Efficiency of Tree Representations

Key ideas:
- Impose a 1×1 bounding square on all representations
- Evaluate size of smallest nodes, not just total area
- Evaluate size of labels as a function of their aspect ratio $L$
- Examine limits of these sizes as depth $D \to \infty$

<table>
<thead>
<tr>
<th>Figure</th>
<th>Nested enclosure of nodes</th>
<th>Leaf nodes have an unbounded aspect ratio</th>
<th>Leaf nodes have a fixed aspect ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>treemap with margins and labels</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>nested circles</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>concentric squares</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>concentric circles</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>13</td>
<td>icicle</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>14</td>
<td>classical variant</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>indented outline</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>radial</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>icicle variant</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>18</td>
<td>classical (laycroft)</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leaf label area $a_l(a_{in})$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total label area $A_l$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total label area $A_{in}$ as $D \to \infty$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rank if $L \geq 5$ (“skinny” labels)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Rank if $L = 1$ (“squarish” labels)</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>Allows all leaf labels to be oriented the same way, for easier reading</td>
<td>YES</td>
<td>YES</td>
<td>no</td>
<td>no</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>Allows balanced partitioning without reducing total area $A$</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>YES</td>
<td>YES</td>
<td></td>
</tr>
</tbody>
</table>
End of section on visualization of trees
Blogs and Websites

- http://infosthetics.com
- http://flowingdata.com
- http://eagereyes.org
- http://www.visualcomplexity.com/vc/blog/
- http://manyeyes.alphaworks.ibm.com/blog/
- http://www.gapminder.org/
- Business Intelligence
  - http://www.perceptualedge.com/blog/
  - http://timoelliott.com/blog/
Libraries and Software

- Many Eyes (website for sharing visualizations of data) [http://manyeyes.alphaworks.ibm.com/](http://manyeyes.alphaworks.ibm.com/)
- For visualizing graphs
  - **Tulip** (software application) [http://www.tulip-software.org/](http://www.tulip-software.org/)
  - **Gephi** (software application) [http://gephi.org/](http://gephi.org/)
  - **Pajek** (software application) [http://pajek.imfm.si/](http://pajek.imfm.si/)
  - **NetworkX** (Python library) [http://networkx.lanl.gov/](http://networkx.lanl.gov/)
  - **OGDF** (C++ library) [http://www.ogdf.net](http://www.ogdf.net)
  - **Graphviz** (applications and libraries) [http://www.graphviz.org/](http://www.graphviz.org/)
  - **WebCoLa** / cola.js (JavaScript library) [http://marvl.infotech.monash.edu/webcola/](http://marvl.infotech.monash.edu/webcola/)
  - **JUNG** (Java library) [http://jung.sourceforge.net/](http://jung.sourceforge.net/)
  - **JGraph** (Java library) [http://www.jgraph.com/](http://www.jgraph.com/)
  - **yFiles** (Java library) [http://www.yworks.com](http://www.yworks.com)
- For visualizing multidimensional data
  - **XmdvTool** (software application) [http://davis.wpi.edu/~xmdv/](http://davis.wpi.edu/~xmdv/)
  - **Spotfire** (software application) [http://www.cs.umd.edu/hcil/spotfire/](http://www.cs.umd.edu/hcil/spotfire/)
  - **Polaris** (software application) [http://window.stanford.edu/projects/polaris/](http://window.stanford.edu/projects/polaris/)
  - **Tableau** (software application) [http://www.tableausoftware.com](http://www.tableausoftware.com)
  - **Google Charts** (JavaScript library) [https://developers.google.com/chart/](https://developers.google.com/chart/)
  - **Mondrian** (software application) [http://rosuda.org/Mondrian](http://rosuda.org/Mondrian)
Visual Surveys of Work

- [http://treevis.net](http://treevis.net) (trees)
- [http://survey.timeviz.net](http://survey.timeviz.net) (time)
- [http://spacetimecubevis.com](http://spacetimecubevis.com) (space-time “cube”)
- [http://dynamicgraphs.fbeck.com](http://dynamicgraphs.fbeck.com) (dynamic graphs)
- [http://www.setviz.net](http://www.setviz.net) (sets)
- [http://financevis.net](http://financevis.net) (financial data – from ÉTS)
- [http://textvis.lnu.se](http://textvis.lnu.se) (text)
- [http://aviz.fr/physvis](http://aviz.fr/physvis) (physical visualizations)
- [http://multivis.net](http://multivis.net) (“multifaceted data”)
Books

- Jacques Bertin (1977), La graphique et le traitement graphique de l'information.
- John Wilder Tukey (1977), Exploratory Data Analysis.
- Edward R. Tufte (1990), Envisioning Information.
- Di Battista, Giuseppe and Peter Eades and Roberto Tamassia and Ioannis G. Tollis (1999), Graph Drawing: Algorithms for the Visualization of Graphs.
- Colin Ware (2000), Information Visualization: Perception for Design.
- Tamara Munzner (2014), Visualization Analysis and Design.
A bibliography of books and papers

http://profs.logti.etsmtl.ca/mmcguffin/bib/vis.txt
“Meet up” group in Montreal

• http://www.meetup.com/vismtl/