Deconstructing Visualizations

Maneesh Agrawala

CS 448B: Visualization
Fall 2017

NYT Election 2016 (based on 2012)
Last Time: Spatial Layout

Problem

**Input:** Set of graphic elements (scene description)

**Goal:** Select visual attributes for elements

- Position
- Orientation
- Size
- Color
- ...

![Diagram of spatial layout example]
Direct Rule-Based Methods

Rule-based timeline labeling

- Alternate above/below line
- Center labels with respect to point on line
Pros and cons

Pros
- Designed to run extremely quickly
- Simple layout algorithms are easy to code

Cons
- Complex layouts require large rule bases with lots of special cases

Linear Constraint Satisfaction
Pros and cons

Pros

- Often run fast (at least one-way constraints)
- Constraint solving systems are available online
- Can be easier to specify relative layout constraints than to code direct layout algorithm

Cons

- Easy to over-constrain the problem
- Constraint solving systems can only solve some types of layout problems
- Difficult to encode desired layout in terms of mathematical constraints
Optimization

Demo
Layout as optimization

Scene description
- **Geometry**: polygons, bounding boxes, lines, points, etc.
- **Layout parameters**: position, orientation, scale, color, etc.

Large design space of possible layouts

To use optimization we will specify …
- Initialize/Perturb functions: Form a layout
- Penalty function: Evaluate quality of layout
- .. and find layout that minimizes penalty

Optimization algorithms

There are lots of them:
- line search, Newton’s method, A*, tabu, gradient descent, conjugate gradient, linear programming, quadratic programming, simulated annealing, …

Differences
- Speed
- Memory
- Properties of the solution
- Requirements
Simulated annealing

currL ← Initialize()
while(! termination condition)
newL ← Perturb(currL)
currE ← Penalty(currL)
newE ← Penalty(newL)
if((newE < currE) or (rand[0,1) < \exp(-\Delta E/T)))
    then currL ← newL
Decrease(T)

Perturb: Efficiently cover layout design space
Penalty: Describes desirable/undesirable layout features

Scene description

Geometry
- Pie slices
  anchors for labels
- Labels
  bounding boxes
Layout parameters

- Position (x, y)
- Leader line
- Word wrap
- Color
- Alignment
- Orientation
- Scale
Many dimensions → large space

- Position (x, y)
- Leader line
- Word wrap
- Color
- Alignment
- Orientation
- Scale

2D x 50 labels → 100D space
Penalties

Overlap & Distance
- Label – anchor slice
- Label – other slices
- Label – label

Leader lines
- Length
- Intersections

Word Wrap

Annealing minimizes sum of all penalties

Overlap: Label – Anchor Slice

Avoid partial overlap: No penalty if fully inside /outside
Overlap: Label – Anchor Slice

Penalize partial overlap by overlap amount

Distance: Label – Anchor Slice

Ensure label near center of edge of anchor slice
Distance: Label – Anchor Slice

Minimize distance $d$

Penalties

- Overlap & Distance
  - Label – anchor slice
  - Label – other slices
  - Label – label

- Leader lines
  - Length
  - Intersections

- Word Wrap

Annealing minimizes sum of all penalties
Demo

Pros and cons

Pros

- Much more flexible than linear constraint solving systems

Cons

- Can be relatively slow to converge
- Need to set penalty function parameters (weights)
- Difficult to encode desired layout in terms of mathematical penalty functions
Design principles

Sometimes specified in design books
- Tufte, Few, photography manuals, cartography books …
- Often specified at a high level
- Challenge is to transform principles into constraints or penalties

Cartographer Eduard Imhof’s labeling heuristics transformed into penalty functions for an optimization based point labeling system [Edmondson 97]

Example-Based Methods
Preference elicitation [Gajos and Weld 05]

Learn characteristics of good designs
- Generate designs based on a parameterized design space
- Ask designers if they are good or bad
- Learn good parameters values based on responses

Nonlinear Inverse Opt. [Vollick et al. 07]

Learn label layout style from single example

Horizontal/Vertical
Nonlinear Inverse Opt. [Vollick et al. 07]

Learn label layout style from single example

Parallel Leader Lines

Pros and cons

Pros

- Often much easier to specify desired layout via examples

Cons

- Usually requires underlying model
- Model will constrain types of layouts possible
- Large design spaces likely to require lots of examples to learn parameters well
Announcements

Final project

Design new visualization method (e.g. software)
- Pose problem, Implement creative solution
- Design studies/evaluations less common but also possible (talk to us)

Deliverables
- Implementation of solution
- 6-8 page paper in format of conference paper submission
- Project progress presentations

Schedule
- Project proposal: Mon 11/6
- Project progress presentation: 11/13 and 11/15 in class (3-4 min)
- Final poster presentation: 12/6 Location: Lathrop 282
- Final paper: 12/10 11:59pm

Grading
- Groups of up to 3 people, graded individually
- Clearly report responsibilities of each member
Deconstructing Visualizations
Pixels are poor representation
Hard for machines to retrieve data

2005 NIH Research Budget per Death
Cardiovascular (CVD = Heart & Stroke)
Diabetes
Hepatitis C
Hepatitis B
Prostate
Alzheimer's
Parkinson's
AIDS

© Copyright FAIR Foundation 2004
Pixels are poor representation
Hard for machines to retrieve data
Hard for people to manipulate
Pixels are a poor representation of charts and graphs
Cannot index, search, manipulate or interact with the data

Goal: Reconstruct higher-level representation of charts and graphs that lets machines and people redesign, reuse and revitalize them

What is a good representation?
<table>
<thead>
<tr>
<th>Year</th>
<th>Exports</th>
<th>Imports</th>
</tr>
</thead>
<tbody>
<tr>
<td>1700</td>
<td>170,000</td>
<td>300,000</td>
</tr>
<tr>
<td>1701</td>
<td>171,000</td>
<td>302,000</td>
</tr>
<tr>
<td>1702</td>
<td>176,000</td>
<td>303,000</td>
</tr>
<tr>
<td>1703</td>
<td>180,000</td>
<td>312,000</td>
</tr>
<tr>
<td>1704</td>
<td>187,000</td>
<td>319,000</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

**2005 NIH Research Budget per Death**

<table>
<thead>
<tr>
<th>Disease</th>
<th>Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aids</td>
<td>70.0%</td>
</tr>
<tr>
<td>Alzheimer's</td>
<td>5.0%</td>
</tr>
<tr>
<td>Cardiovascular</td>
<td>5.0%</td>
</tr>
<tr>
<td>Diabetes</td>
<td>4.8%</td>
</tr>
<tr>
<td>Hepatitis B</td>
<td>4.1%</td>
</tr>
<tr>
<td>Hepatitis C</td>
<td>3.8%</td>
</tr>
<tr>
<td>Parkinson'</td>
<td>6.0%</td>
</tr>
<tr>
<td>Prostate</td>
<td>5.2%</td>
</tr>
</tbody>
</table>

© Copyright FAIR Foundation 2004
### Approach

**Classification:** Determine chart type  
**Mark extraction:** Retrieve graphical marks  
**Data extraction:** Retrieve underlying data table
Classification

Training the Classifier
Training the Classifier

Bar Charts

Pie Charts

Scatter Plots
Classifying an Input Image

Asset allocation by type
- Platinum
- Silver
- Gold
- Bonds
- Stocks
- Cash
Classifying an Input Image
Classifying an Input Image

- SVM Classifier
- Pie Chart

### Corpus: 667 charts, 5 chart types [Prasad 2007]

<table>
<thead>
<tr>
<th>Classifier Type</th>
<th>Average Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Prasad 2007] Multi-class SVM</td>
<td>84%</td>
</tr>
<tr>
<td>ReVision: Multi-class SVM</td>
<td>88%</td>
</tr>
<tr>
<td>ReVision: Binary SVM (yes/no for each chart type)</td>
<td>96%</td>
</tr>
</tbody>
</table>

ReVision binary SVMs give 96% classification accuracy

http://vis.berkeley.edu/papers/revision

Our Corpus

Over 2500 labeled images and 10 chart types

- Area Graphs (90+17)
- Bar Graphs (363+81)
- Curve Plots (318)
- Maps (249)
- Pareto Charts (168)
- Pie Charts (230+21)
- Radar Plots (137)
- Scatter Plots (372)
- Tables (263)
- Venn Diagrams (263)
Mark and Data Extraction

Assumptions

Bar charts and pie charts only
No shading or texture, 3D, stacked bars, or exploded pies
Bar Charts

marks: lines

<table>
<thead>
<tr>
<th>y-value</th>
<th>x-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>A</td>
</tr>
<tr>
<td>25</td>
<td>B</td>
</tr>
<tr>
<td>4</td>
<td>C</td>
</tr>
<tr>
<td>75</td>
<td>D</td>
</tr>
</tbody>
</table>

Extract Marks
- Find Foreground Rectangles
- Identify Orientation and Baseline

Extract Data
- Recover Bar Values
- Associate Labels with Bars

Scale: 2 pixels/unit
Pie Charts

Extract Marks
- Fit Ellipse Using RNASAC
- Unroll Pie and Find Transitions

Extract Data
- Compute Area Percentages
- Associate Labels with Areas

marks: areas

<table>
<thead>
<tr>
<th>percentage</th>
<th>category</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.3</td>
<td>A</td>
</tr>
<tr>
<td>22.4</td>
<td>B</td>
</tr>
<tr>
<td>10.8</td>
<td>C</td>
</tr>
<tr>
<td>5.6</td>
<td>D</td>
</tr>
<tr>
<td>5.6</td>
<td>E</td>
</tr>
<tr>
<td>33.3</td>
<td>F</td>
</tr>
</tbody>
</table>

Scale: 50 pixels/percent

Extraction Results

<table>
<thead>
<tr>
<th>Number of Charts</th>
<th>Total charts</th>
<th>Mark extractions</th>
<th>Data extractions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bar</td>
<td>52</td>
<td>41</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>79%</td>
<td>56%</td>
<td></td>
</tr>
<tr>
<td>Pie</td>
<td>53</td>
<td>33</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>62%</td>
<td>40%</td>
<td></td>
</tr>
</tbody>
</table>
Redesign
Redesign #1

Original

Redesign #2

Limitations

Additional Chart Types

Handling Legends

Legend:
- TPM(160p)
- 8/12L
- RH60
- Dm12N
- RH15N
- 4H24N
- RH36N
Graphical Overlays

Visual elements that are layered onto a chart to facilitate the perceptual and cognitive processes involved in chart reading
Graphical overlay gallery

This gallery contains examples of graphical overlays, described in our paper. We have extracted marks and data from the charts using Revision (for bar and pie charts) and Dataset (for line charts), but all of the overlays are generated in-browser. Try out some of the parameters, or click on an image thumbnail below the gallery to view some example overlays.

European Union budgets since 2000

<table>
<thead>
<tr>
<th>Year</th>
<th>Money spent</th>
<th>Approved budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>50</td>
<td>100</td>
</tr>
<tr>
<td>2001</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>2002</td>
<td>200</td>
<td>250</td>
</tr>
<tr>
<td>2003</td>
<td>300</td>
<td>350</td>
</tr>
<tr>
<td>2004</td>
<td>400</td>
<td>450</td>
</tr>
<tr>
<td>2005</td>
<td>500</td>
<td>550</td>
</tr>
<tr>
<td>2006</td>
<td>600</td>
<td>650</td>
</tr>
<tr>
<td>2007</td>
<td>700</td>
<td>750</td>
</tr>
<tr>
<td>2008</td>
<td>800</td>
<td>850</td>
</tr>
<tr>
<td>2009</td>
<td>900</td>
<td>950</td>
</tr>
<tr>
<td>2010</td>
<td>1000</td>
<td>1100</td>
</tr>
</tbody>
</table>

Chart type: Bar
Chart: [Bar chart]
Overlay type: [Reference structures]
- Regular gridlines
- Lines emanating from marks

Parameters:
- Overlay
- Underlay
- Static
- Interactive
- Divisions
- Line thickness

Places regular gridlines at user defined intervals.

Refence Structures

- Help by breaking marks into regular segments and aid reading axis values
Draws viewers’ attention to specific marks

Emphasize data values or trends
Summary Statistics

Enables comparison with statistics based on the data

Annotation

Provide context and support collaboration
Most overlays only require access to marks
- Reference structures (marks)
- Highlights (marks)
- Redundant encodings (marks and data)
- Summary statistics (marks)
- Annotations (marks)

Interactive Documents

How can we facilitate reading text and charts together?
Goal: Extract references between text and chart

Problem: Diversity of writing styles
Skepticism for capitalism is lowest in Brazil (22%), China (19%), Germany (29%) (although East Germans are less supportive than West Germans) and the U.S. (24%). Skepticism for free markets is highest in Mexico (60%) and Japan (60%).
Top earners have attracted more opprobrium as their salaries and the performance of the economy have headed in opposite directions. Europeans and Latin Americans tend to have similar attitudes to the rich; the Anglo-Saxon world is a bit more forgiving.
Preprocessing
- Document segmentation
  - Mark and data extraction

Crowdsourcing
- Reference extraction

Clustering and Merging
- Merge
  - Split
    - Cluster
      - Select representative

Demo
Evaluation

Avg. $F_1$ distance: expert specified references vs. crowd specified references

Ongoing and Future Work
Deconstructing D3 Charts

User Interface Software Technology (UIST) 2014.

Automatic Redesign

Can we automatically redesign charts to improve
Perceptual effectiveness?
Visual aesthetics?
Accessibility for vision impaired users?
Document Collections

Many specialized collections
- Scientific: PLOS, JSTOR, ACM DL, …
- Web visualizations: D3, Processing, …

How can deconstruction aid search?
- Search by chart type, data type, marks, data, …
- Similarity search with inexact matching
- Query expansion

Takeaways

A chart is a collection of mappings between data and marks

We can reconstruct this representation from chart bitmaps

Such reconstruction enables redesign, reuse and revitalization