Identifying Design Principles

Maneesh Agrawala

CS 294-10: Visualization
Fall 2013

Last Time: Spatial Layout
Problem

**Input:** Set of graphic elements (scene description)
**Goal:** Select visual attributes for elements
- Position
- Orientation
- Size
- Color
- ...

Approaches

- Direct rule-based methods
- Constraint satisfaction
- Optimization
- Example-based methods
Adaptive document layout [Jacobs 03]

Users authors templates which use one-way constraints to adapt to changes in page size

ADL template authoring [Jacobs 03]
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

\[
\text{currL} \leftarrow \text{Initialize()}
\]
while(! termination condition)
\[
\text{newL} \leftarrow \text{Perturb(currL)}
\]
\[
\text{currE} \leftarrow \text{Penalty(currL)}
\]
\[
\text{newE} \leftarrow \text{Penalty(newL)}
\]
\[
\text{if} ((\text{newE} < \text{currE}) \text{ or } (\text{rand}(0,1) < e^{-\Delta E/T}))
\]
\[
\text{then } \text{currL} \leftarrow \text{newL}
\]
\[
\text{Decrease(T)}
\]

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 $\rightarrow$ large space

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

2D x 50 labels $\rightarrow$ 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

Artistic Resizing

A Technique for Rich Scale-Sensitive Vector Graphics

Pierre Dragicevic
Stéphane Chatty
David Thevenin
Jean-Luc Vinot
The Resizing Problem

- Fixed size
- Naive scaling
- Artistic resizing

Expressing Artistic Resizing

- Commonly described using formulae
  \n  \begin{align*}
  x_L &= \frac{w-w_L}{2} \\
  y_L &= \frac{h-h_L}{2} \\
  w_L &= 20 \\
  h_L &= 10 \\
  w_B &= 5 \\
  h_B &= 5 \\
  r &= 20
  \end{align*}

- These formulae are:
  - Translated into code by the programmer
  - Or used as an input to constraint-solving systems
Example-Based Approach

1. Designers produce variants using their authoring tool

2. System interprets the example set

Artistic Resizing
How does it work?

- Assumes the exclusive use of:
  - Copy & paste for adding new examples
  - Affine transformation tools (move, scale, rotate, shear)

- Based on local interpolation of transformations
Artistic Resizing
How does it work?

- Each variant of T1 is associated with the example’s bounding box

Artistic Resizing
How does it work?

- Problem of multivariate interpolation
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
Assignments 2 and 3

Grades have been posted to bspace

If you used real estate data for A2 please let Jennifer Baires know what you did
bairesjen@gmail.com

Final project

Design new visualization method
- Pose problem, Implement creative solution

Deliverables
- Implementation of solution
- 8-12 page paper in format of conference paper submission
- 1 or 2 design discussion presentations

Schedule
- Project proposal: 10/28
- Project presentation: 11/11-11/13
- Final paper and presentation: 12/2-12/6

Grading
- Groups of up to 3 people, graded individually
- Clearly report responsibilities of each member
Identifying Design Principles

Good Design Improves Effectiveness

London Underground [Beck 33] Geographic version of map
Good Design Improves Effectiveness

Design principle:
- Straighten lines to emphasize sequence of stops

Technique used to emphasize/de-emphasize information

Approach

Identify design principles
- Cognition and perception

Instantiate design principles
- Principles become constraints that guide an optimization process
Route Maps

Visualizing Routes
A Better Visualization

Cognition of Route Maps

Essential information
- Turning points
- Route topology

Secondary context information
- Local landmarks, cross streets, etc.
- Overview area landmarks, global shape

Exact geometry less important
- Not apprehended accurately
- Not drawn accurately

[Tversky 81] [Tufte 90] [Tversky 92]  
[MacEachren 95] [Denis 97] [Tversky 99]
Design Principles

- Exaggerate road length
- Regularize turning angles
- Simplify road shape

LineDrive

Hand-drawn route map  LineDrive route map
Map Design via Optimization

Set of graphic elements
- Roads, labels, cross-streets, …

Choose visual attributes
- Position, orientation, size, …
- Distortions increase flexibility

Develop constraints based on design principles

Simulated annealing
- Perturb: Form a layout
- Score: Evaluate quality
- Minimize score
Road Layout

Choose road lengths and orientations

Before road layout

After road layout
Road Layout Constraints

Length
- Ensure all roads visible
- Maintain ordering by length

\[ \frac{(L_{\text{min}} - l(r_i)) / L_{\text{min}}}{W_{\text{shuffle}}} \]

Orientation
- Maintain original orientation

\[ |\alpha_{\text{curr}}(r_i) - \alpha_{\text{orig}}(r_i)| \times W_{\text{orient}} \]

Topological errors
- Prevent false
- Prevent missing
- Ensure separation

\[ \min(d_{\text{orig}}, d_{\text{dest}}) \times W_{\text{false}} \]
\[ d \times W_{\text{missing}} \]
\[ \min(d_{\text{ext}}, E) \times W_{\text{ext}} \]

Overall route shape
- Maintain endpoint direction
- Maintain endpoint distance

\[ |\alpha_{\text{curr}}(v) - \alpha_{\text{orig}}(v)| \times W_{\text{enddir}} \]
\[ |d_{\text{curr}}(v) - d_{\text{orig}}(v)| \times W_{\text{enddist}} \]

Balancing the Constraints

Prioritize scores by importance
1. Prevent topological errors
2. Ensure all roads visible
3. Maintain original orientation
4. Maintain ordering by length
5. Maintain overall route shape

Priorities set based on usability tests
- Users given maps containing errors
- Rated which errors most confusing
Label Layout
Find overlap-free position for each label

Context Layout
Place cross-streets and exit signs if possible
System Performance

7727 routes  (sampled over 1 day at MapBlast!)
- Median distance 52.5 miles
- Median number turning points 13
- Median computation time 0.7 sec
- Short roads 5.4 %
- False intersections 0.3 %
- Missing intersections 0.2 %
- Label-label overlap 0.5 %
- Label-road overlap 11.7 %

Results

Beta version 6 months
- 150,000 maps served

2242 responses
- Replace standard 55.6 %
- Use with standard 43.5 %
- Prefer standard 0.9 %

At peak
- Deployed at: mappoint.com
- Served 750,000 maps/day
- Taken offline in fall 2011
Original Design

Layout

- Map and text close together
- Overview and destination maps for more content

Limited Resolution PDA
Next Steps: Wedding Maps

Hand-designed Wedding Map www.WeddingMaps.CC

Input map drawn to scale

Our result

1st Ave. and 19th Ave. NW, Seattle WA

http://www.bing.com/maps/explore/#/c7pvw1whdkp6qqwv (Requires Windows, IE, Silverlight)
1st Ave. and 19th Ave. NW, Seattle WA

http://www.bing.com/maps/explore/#/c7pvw1whdkp6gqvw (Requires Windows, IE, Silverlight)

Evergreen Ave., Boston MA

http://www.bing.com/maps/explore/#/c7pvw1whdkp6gqvw (Requires Windows, IE, Silverlight)