Identifying Design Principles

Maneesh Agrawala

CS 294-10: Visualization
Fall 2013

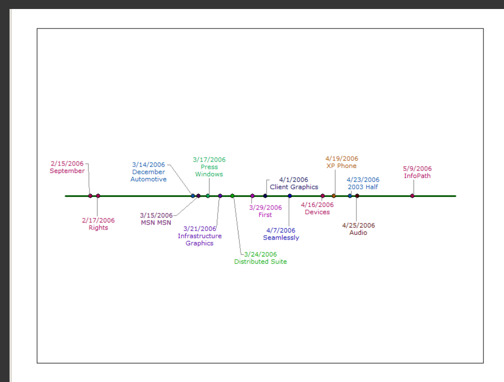
Last Time: Spatial Layout

Problem

Input: Set of graphic elements (scene description)

Goal: Select visual attributes for elements

- **Position**
- Orientation
- Size
- Color
- …



Approaches

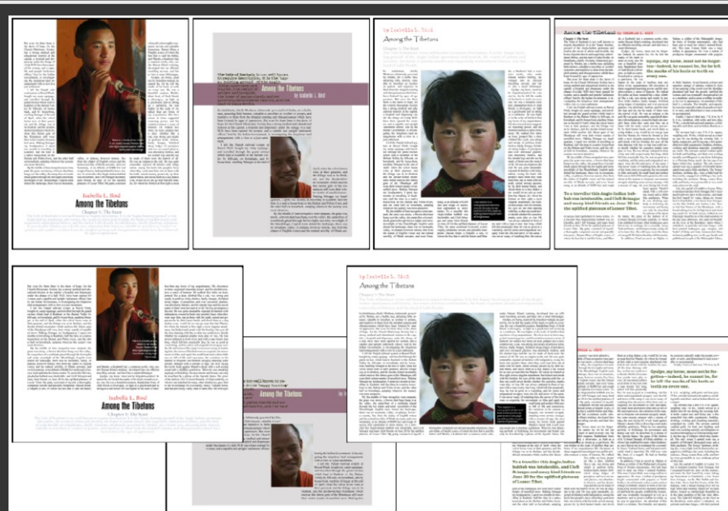
Direct rule-based methods

Constraint satisfaction

Optimization

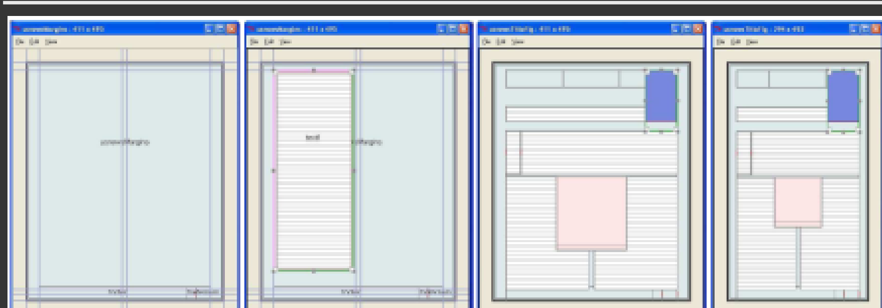
Example-based methods

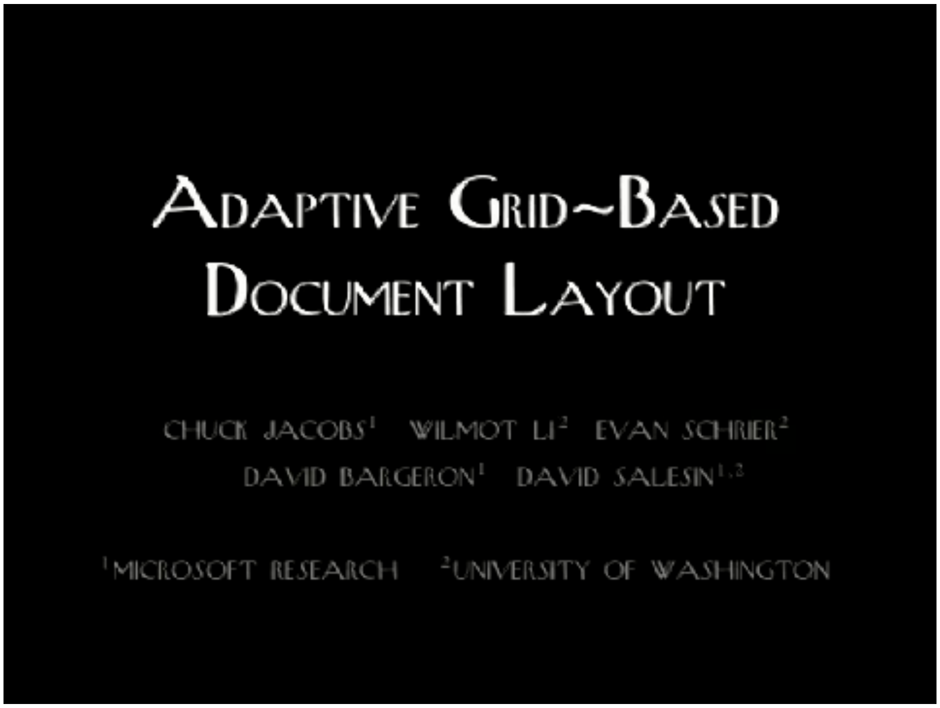
Adaptive document layout [Jacobs 03]



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

ADL template authoring [Jacobs 03]





Pros and cons

Pros

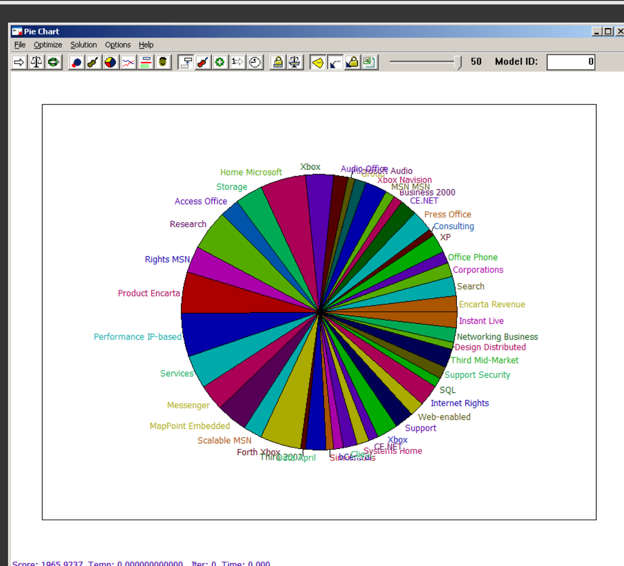
- Often run fast (at least one-way constraints)
- Constraint solving systems are available online
- Can be easier to specify relative layout constraints than to code direct layout algorithm

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

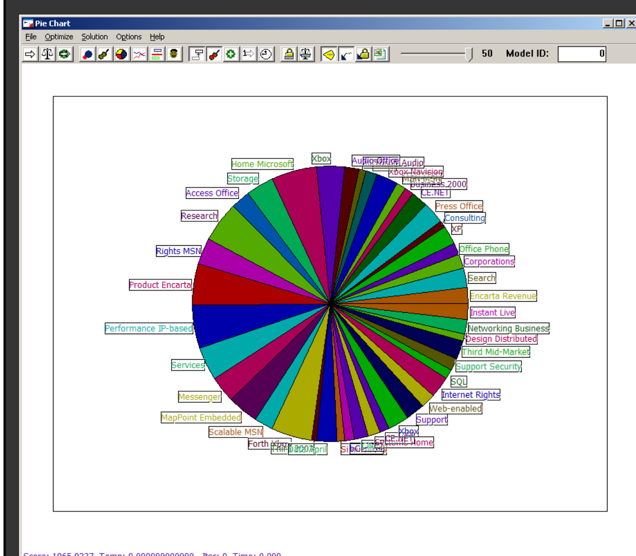
Simulated annealing

currL ← Initialize() → Form initial layout
while(**! termination condition)**
  newL ← Perturb(***currL***) → Perturb to form new layout
  currE ← Penalty(***currL***) → Evaluate quality of layouts
  newE ← Penalty(***newL***)
  if(**(*newE* < *currE*) or** → Always accept lower penalty
   (rand[0,1) < $e^{-\Delta E/T}$**))** → Small probability of accepting higher penalty
   then ***currL*** ← ***newL***
  Decrease(*T*)

Perturb: Efficiently cover layout design space

Penalty: Describes desirable/undesirable layout features

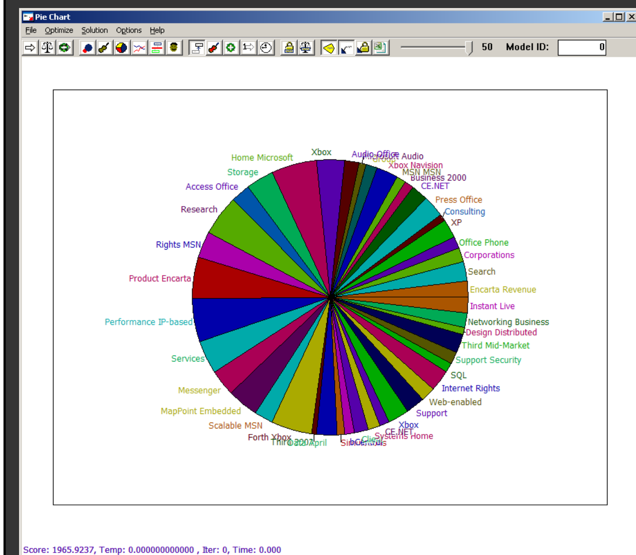
Scene description



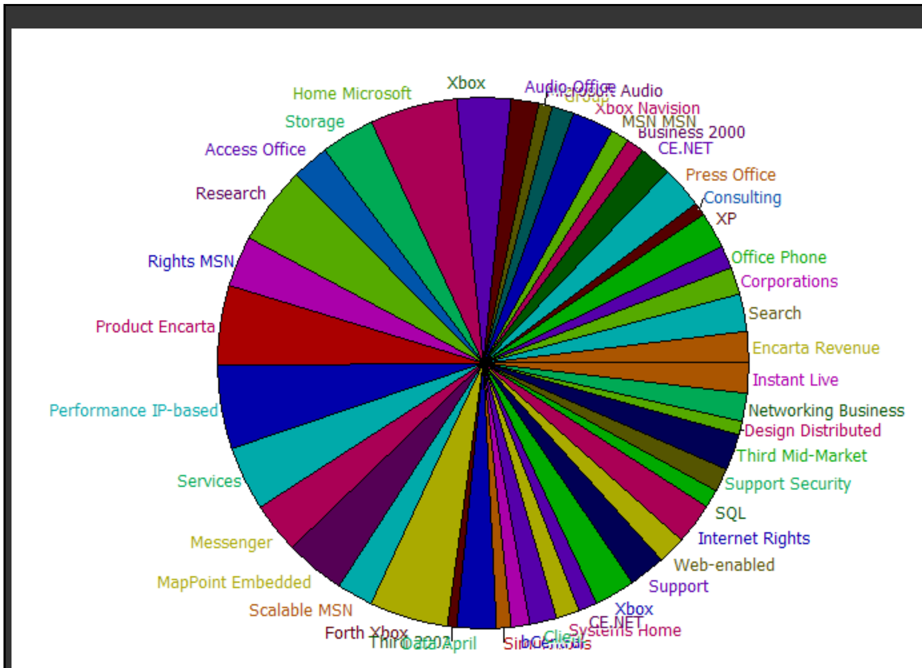
Geometry

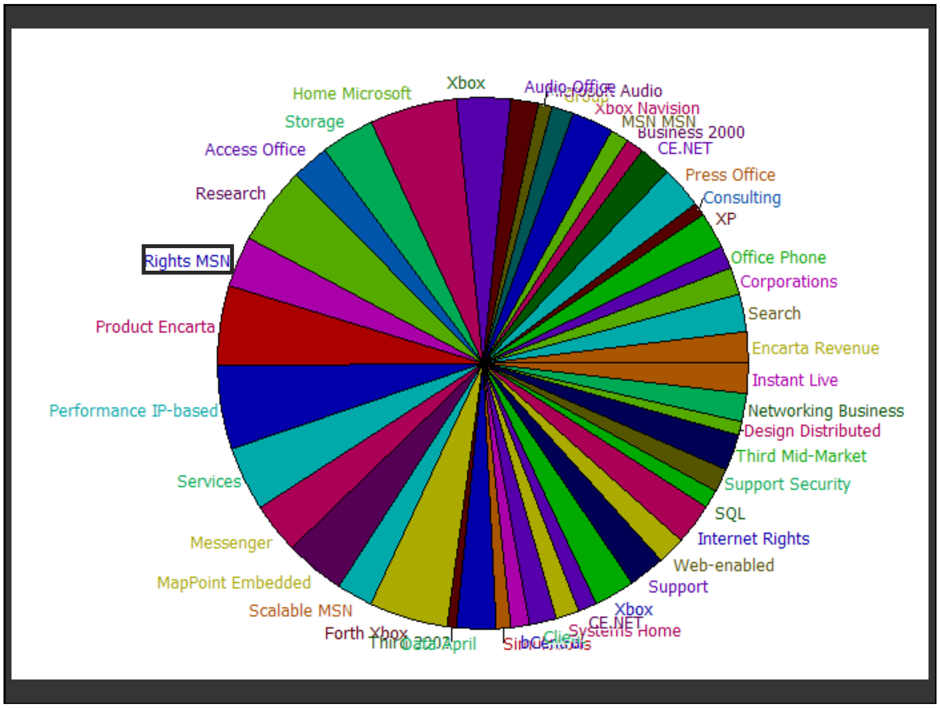
- **Pie slices**
  anchors for labels
- **Labels**
  bounding boxes

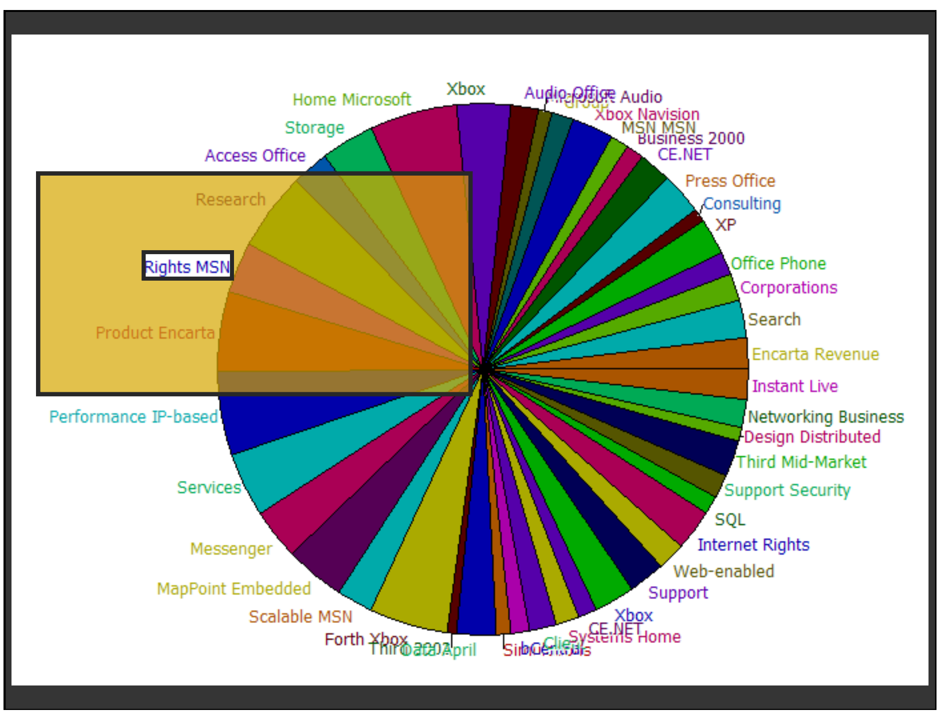
Layout parameters

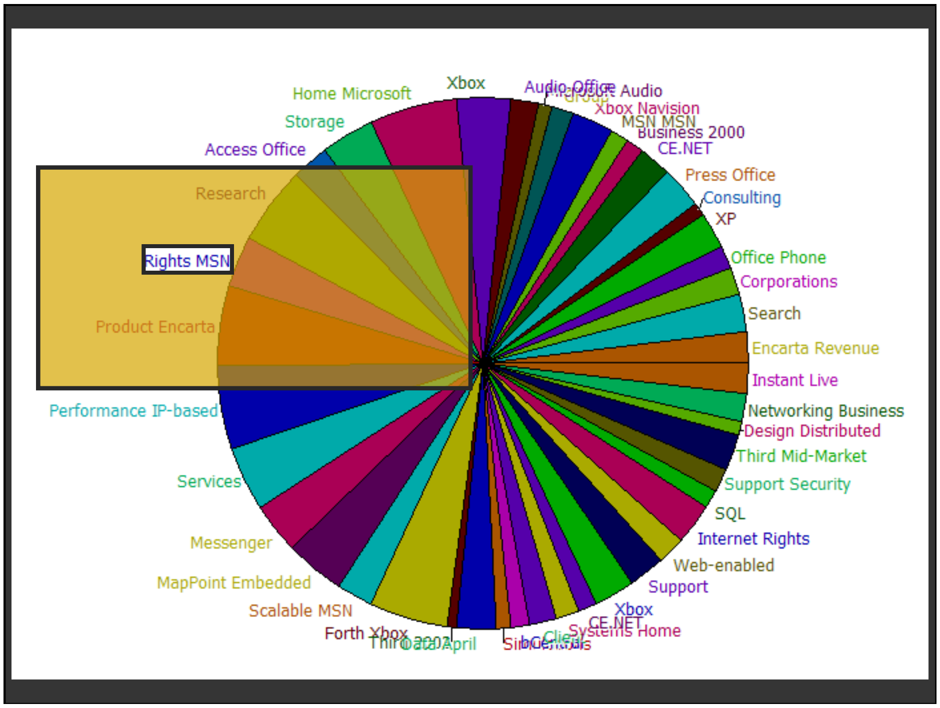


- **Position *(x, y)***
- **Leader line**
- **Word wrap**
- Color
- Alignment
- Orientation
- Scale

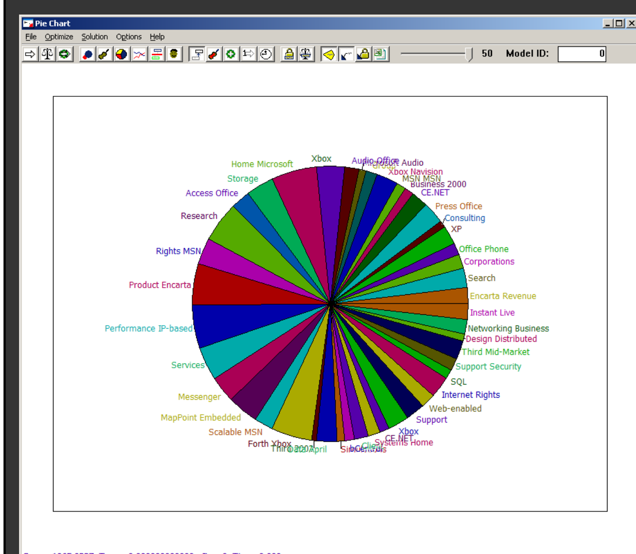



Home Microsoft
Xbox
Audio
Xbox Navision
MSN MSN
Business 2000
CE.NET
Storage
Access Office
Press Office
Consulting
XP
Research
Rights MSN
Office Phone
Corporations
Search
Product Encarta
Encarta Revenue
Instant Live
Performance IP-based
Networking Business
Design Distributed
Third Mid-Market
Services
Support Security
SQL
Messenger
Internet Rights
Web-enabled
MapPoint Embedded
Support
Scalable MSN
Xbox
CE.NET
Home
Forth Xbox
April


Home Microsoft
Xbox
Audio
Xbox Navision
MSN MSN
Business 2000
CE.NET
Storage
Access Office
Press Office
Consulting
XP
Research
Rights MSN
Office Phone
Corporations
Search
Product Encarta
Encarta Revenue
Instant Live
Performance IP-based
Networking Business
Design Distributed
Third Mid-Market
Services
Support Security
SQL
Messenger
Internet Rights
Web-enabled
MapPoint Embedded
Support
Scalable MSN
Xbox
CE.NET
Home
Forth Xbox
April



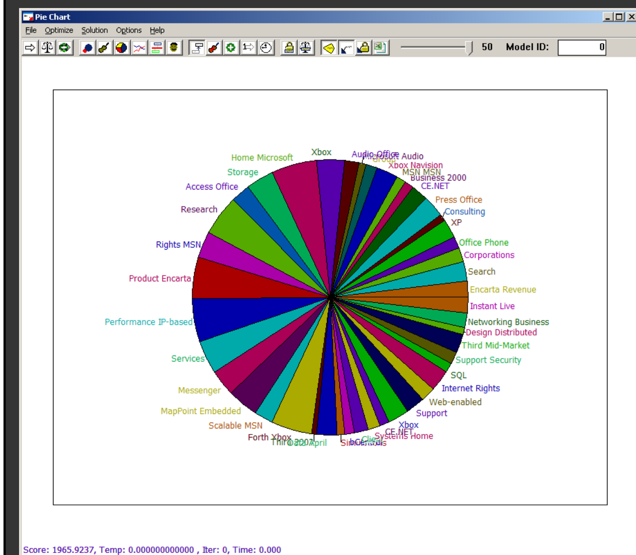
Many dimensions → large space



- **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

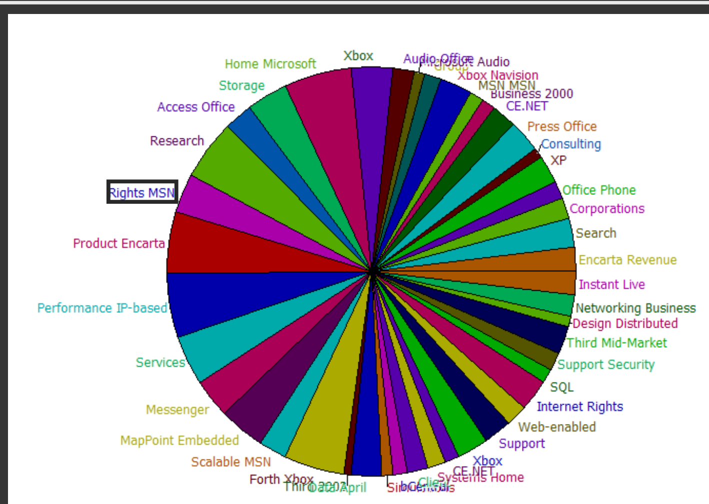
Leader lines

- Length
- Intersections

Word Wrap

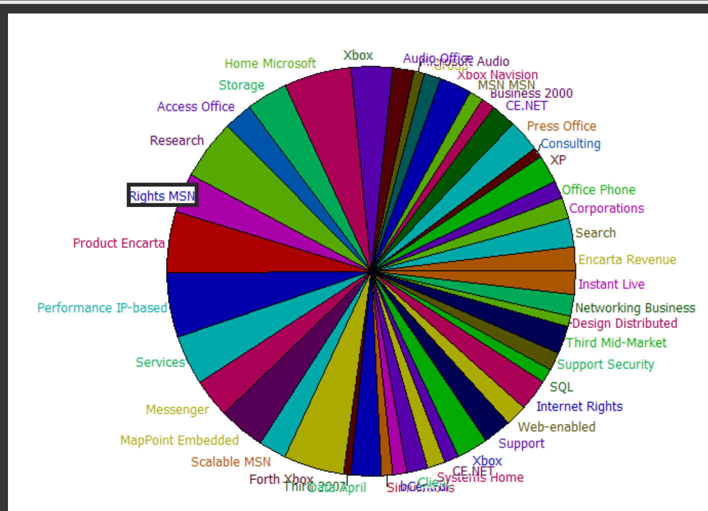
Annealing minimizes sum of all penalties

Overlap: Label – Anchor Slice



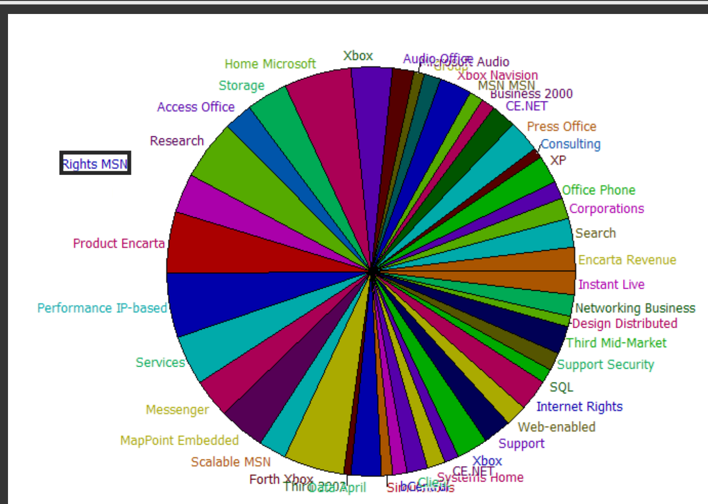
Avoid partial overlap: No penalty if fully inside /outside

Overlap: Label – Anchor Slice



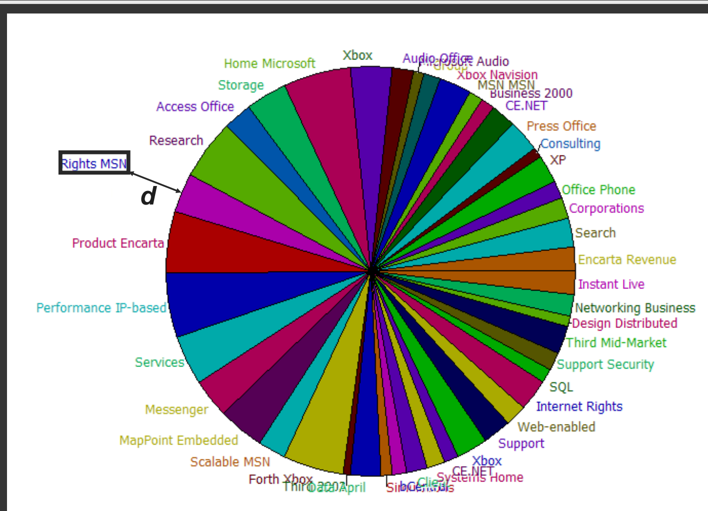
Penalize partial overlap by overlap amount

Distance: Label – Anchor Slice



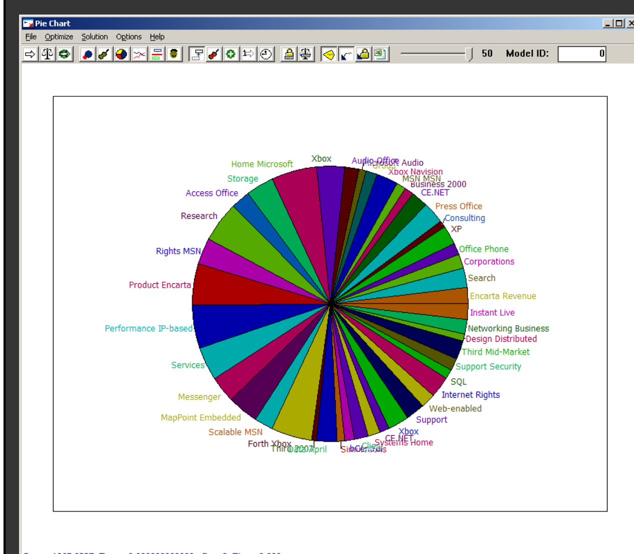
Ensure label near center of edge of anchor slice

Distance: Label – Anchor Slice



Minimize distance *d*

Penalties



Overlap & Distance

- Label – anchor slice
- Label – other slices
- Label – label

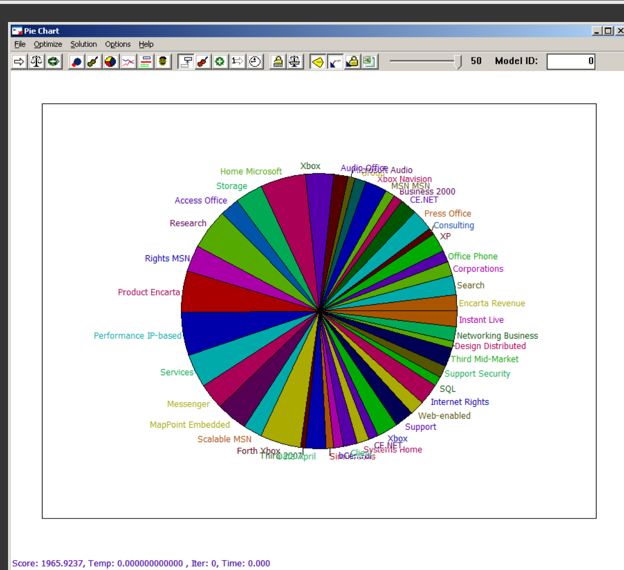
Leader lines

- Length
- Intersections

Word Wrap

Annealing minimizes sum of all penalties

Demo



Pros and cons

Pros

- Much more flexible than linear constraint solving systems

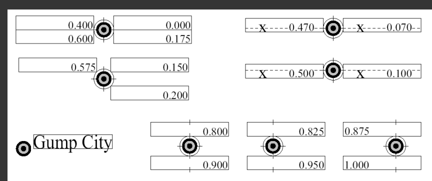
Cons

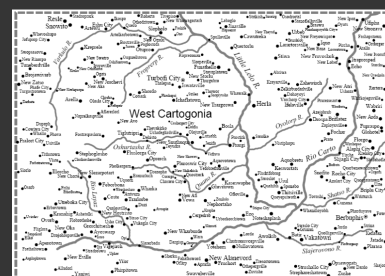
- Can be relatively slow to converge
- Need to set penalty function parameters (weights)
- Difficult to encode desired layout in terms of mathematical penalty functions

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





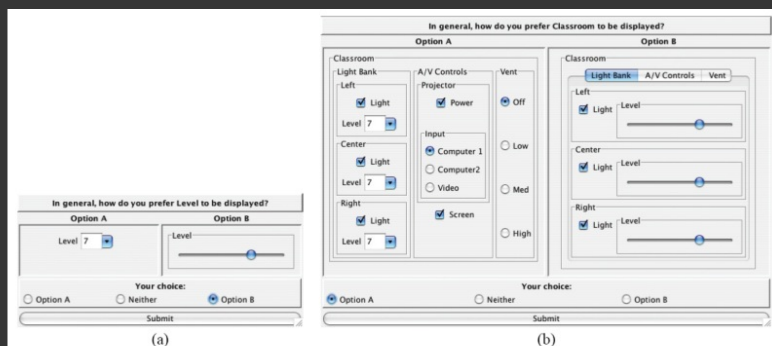
Cartographer Eduard Imhof's labeling heurists transformed into penalty functions for an optimization based point labeling system [Edmondson 97]

Example-Based Methods

Preference elicitation [Gajos and Weld 05]

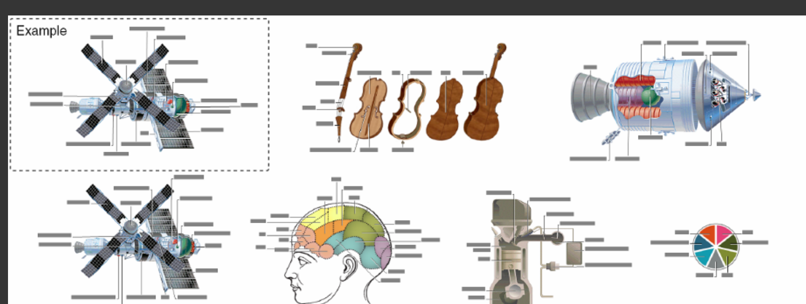
Learn characteristics of good designs

- 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]

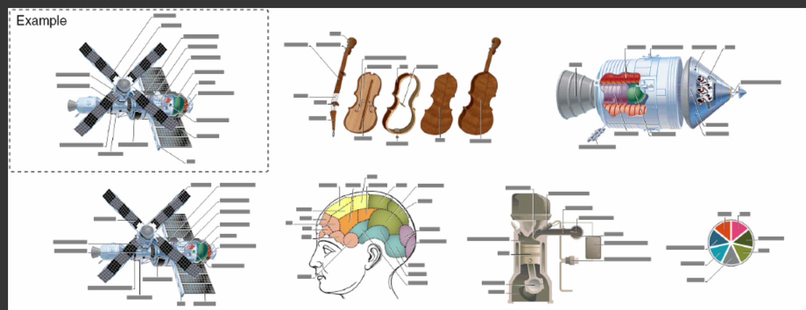
Learn label layout style from single example



Horizontal/Vertical

Nonlinear Inverse Opt. [Vollick et al. 07]

Learn label layout style from single example



Parallel Leader Lines

Artistic Resizing

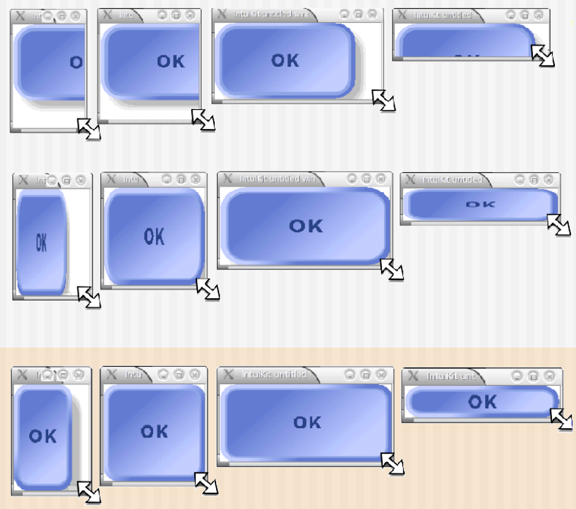
A Technique for Rich
Scale-Sensitive Vector Graphics

Pierre Dragicevic
Stéphane Chatty
David Thevenin
intuilab

Jean-Luc Vinot
Direction Générale de l'Aviation Civile
dgac

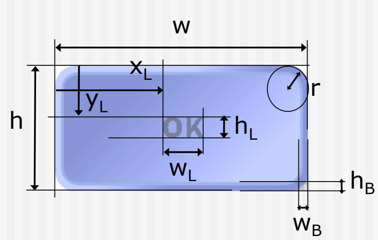
The Resizing Problem

- Fixed size
- Naive scaling
- Artistic resizing



Expressing Artistic Resizing

- Commonly described using formulae



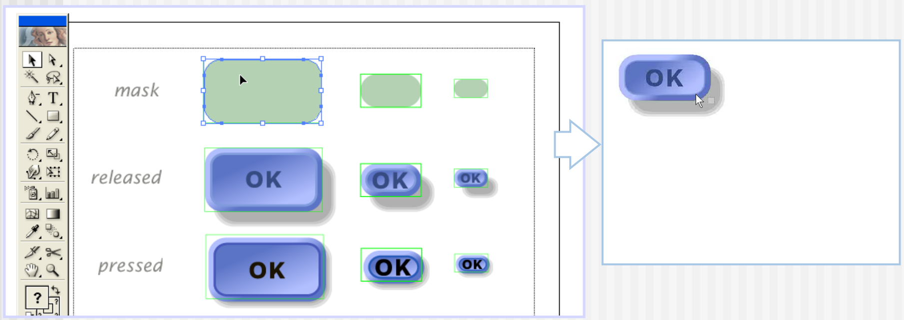
- $x_L = (w - w_L) / 2$
- $y_L = (h - h_L) / 2$
- $w_L = 20$
- $h_L = 10$
- $w_B = 5$
- $h_B = 5$
- $r = 20$

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

Example-Based Approach

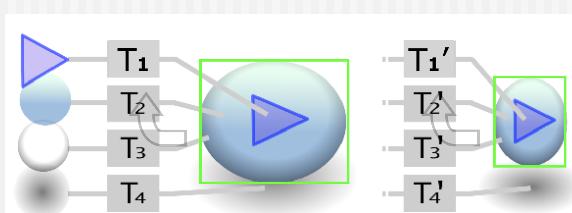
1. Designers produce variants using their authoring tool

2. System interprets the example set



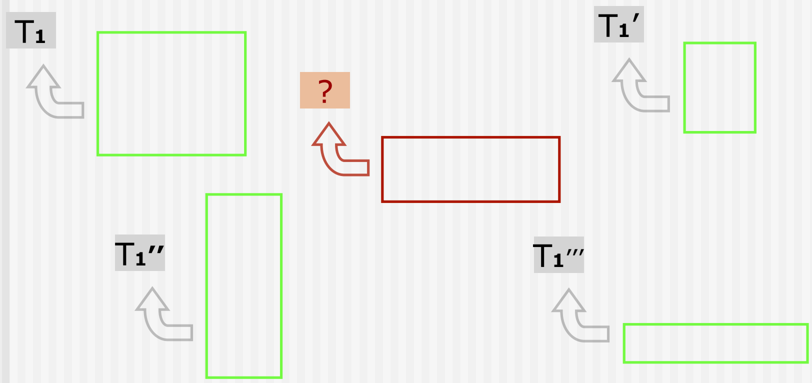
Artistic Resizing
How does it work?

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



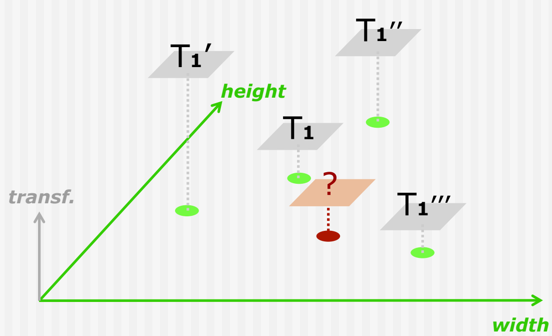
Artistic Resizing
How does it work?

- 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

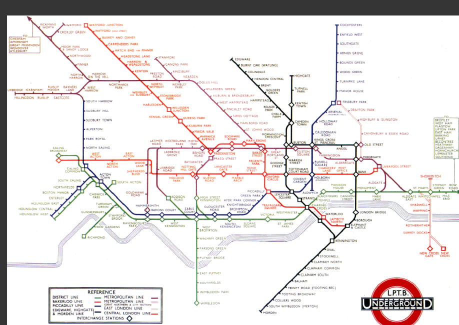
- Project proposal: 10/28
- Project presentation: 11/11-11/13
- Final paper and presentation: 12/2-12/6

Grading

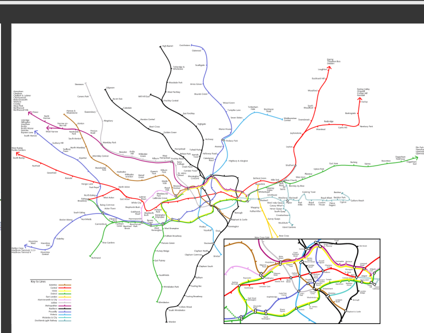
- Groups of up to 3 people, graded individually
- Clearly report responsibilities of each member

Identifying Design Principles

Good Design Improves Effectiveness

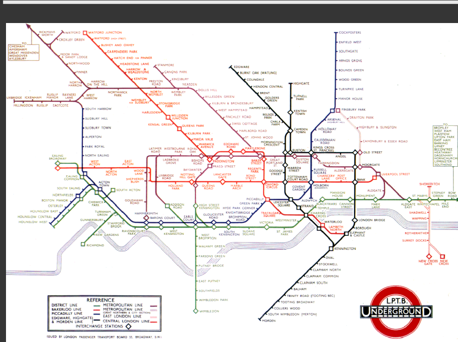


London Underground [Beck 33]

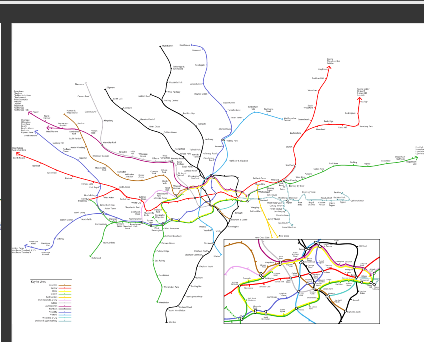


Geographic version of map

Good Design Improves Effectiveness



London Underground [Beck 33]



Geographic version of map

Design principle:

- Straighten lines to emphasize sequence of stops

Technique used to emphasize/de-emphasize information

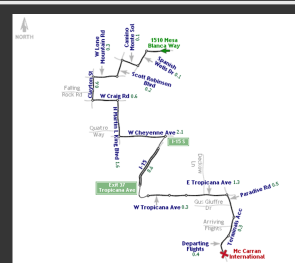
Approach

Identify design principles

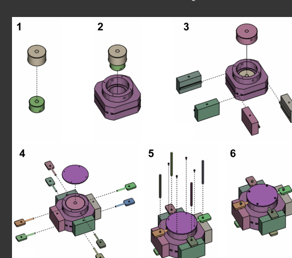
- Cognition and perception

Instantiate design principles

- Principles become constraints that guide an optimization process



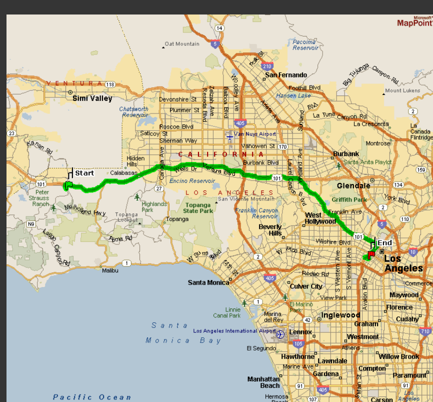
Route maps



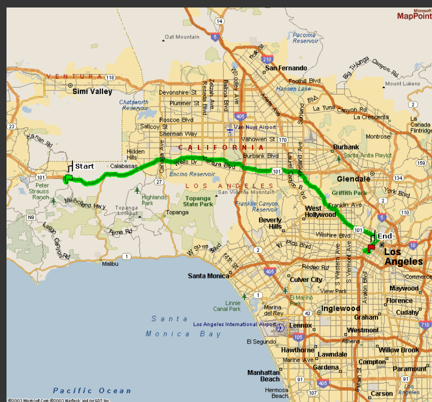
Assembly instructions

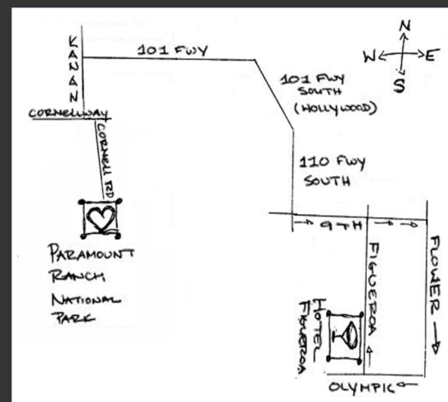
Route Maps

Visualizing Routes

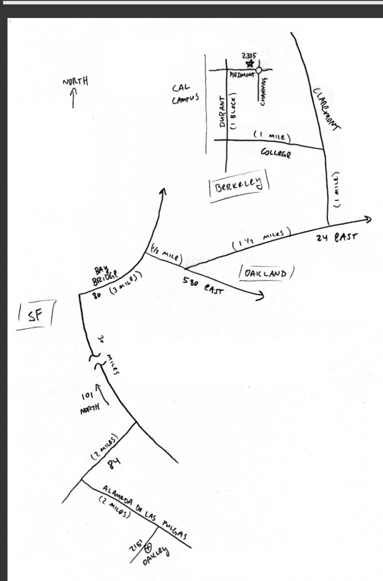


A Better Visualization





Cognition of Route Maps



Essential information

- Turning points
- Route topology

Secondary context information

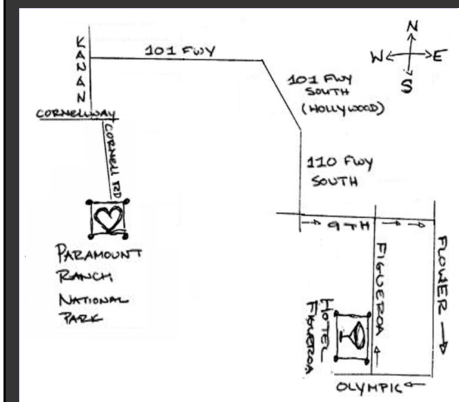
- Local landmarks, cross streets, etc.
- Overview area landmarks, global shape

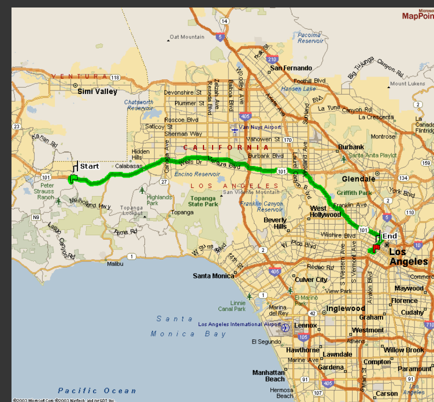
Exact geometry less important

- *Not* apprehended accurately
- *Not* drawn accurately

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

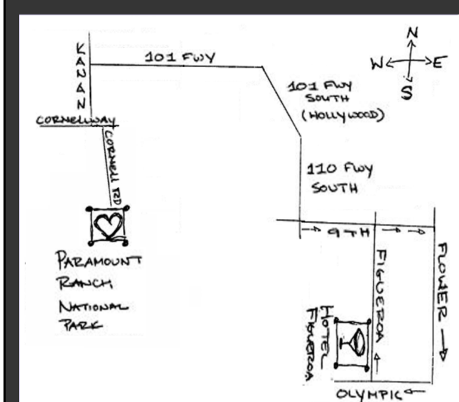
Design Principles



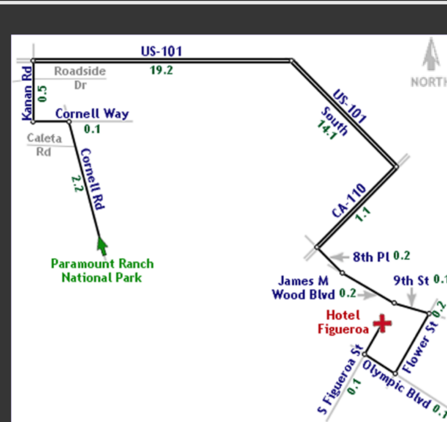


- Exaggerate road length
- Regularize turning angles
- Simplify road shape

LineDrive



Hand-drawn route map



LineDrive route map

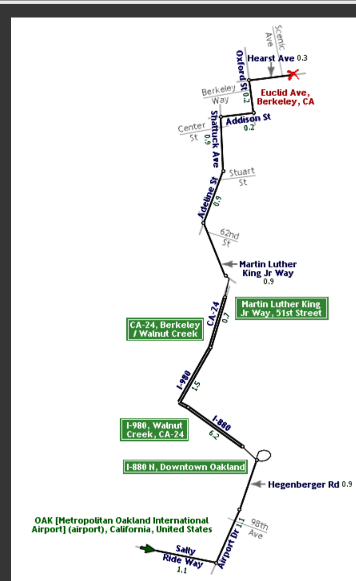
Map Design via Optimization

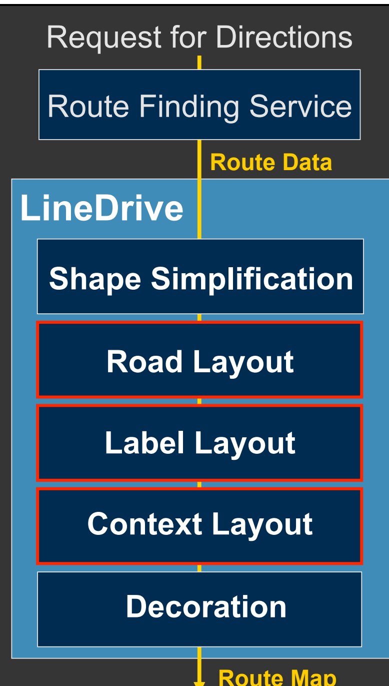
Set of graphic elements
- Roads, labels, cross-streets, …

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

Develop constraints based on design principles

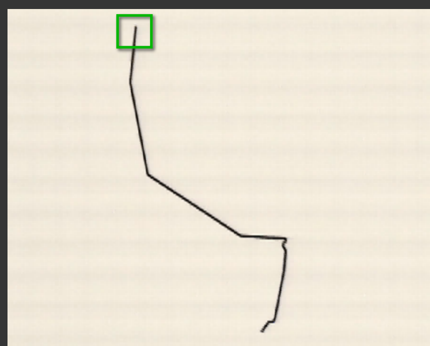
Simulated annealing
- Perturb: Form a layout
- Score: Evaluate quality
- Minimize score



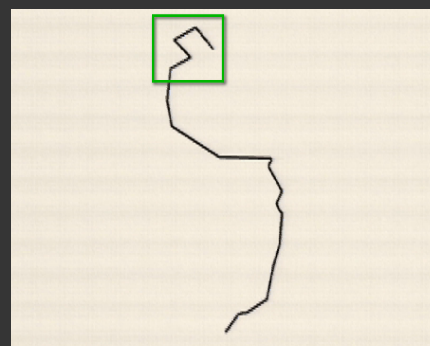


Road Layout

Choose road lengths and orientations



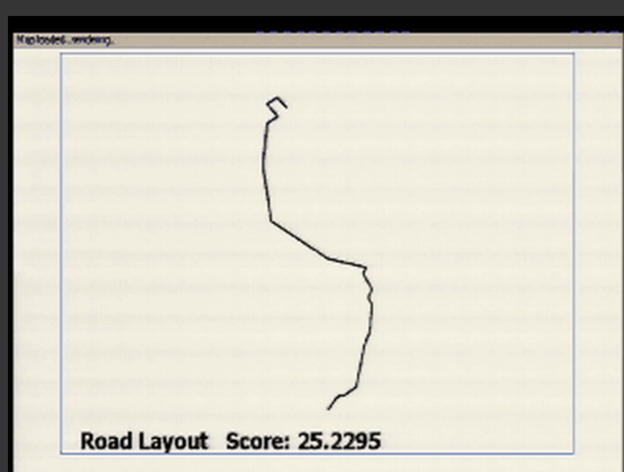
Before road layout



After road layout

Road Layout

Choose road lengths and orientations



Road Layout Constraints

Length

- Ensure all roads visible — $((L_{min} - l(r_i))/ L_{min})^2 * W_{small}$
- Maintain ordering by length — $W_{shuffle}$

Orientation

- Maintain original orientation — $|\alpha_{curr}(r_i) - \alpha_{orig}(r_i)| * W_{orient}$

Topological errors

- Prevent false — $\min(d_{origin}, d_{dest}) * W_{false}$
- Prevent missing — $d * W_{missing}$
- Ensure separation — $\min(d_{ext}, E) * W_{ext}$

Overall route shape

- Maintain endpoint direction — $|\alpha_{curr}(v) - \alpha_{orig}(v)| * W_{enddir}$
- Maintain endpoint distance — $|d_{curr}(v) – d_{orig}(v)| * W_{enddist}$

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

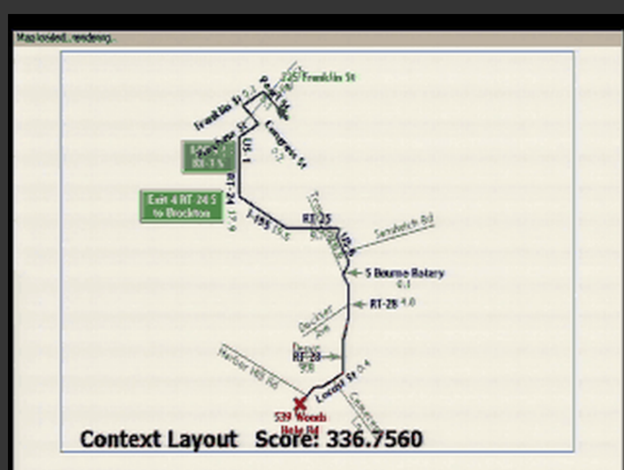
- Users given maps containing errors
- Rated which errors most confusing

Label Layout

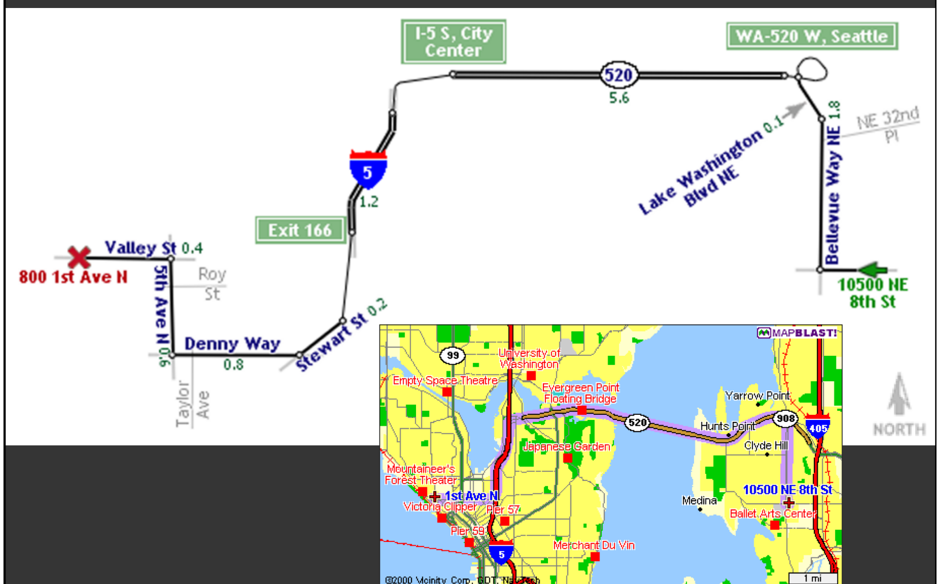
Find overlap-free position for each label

Context Layout

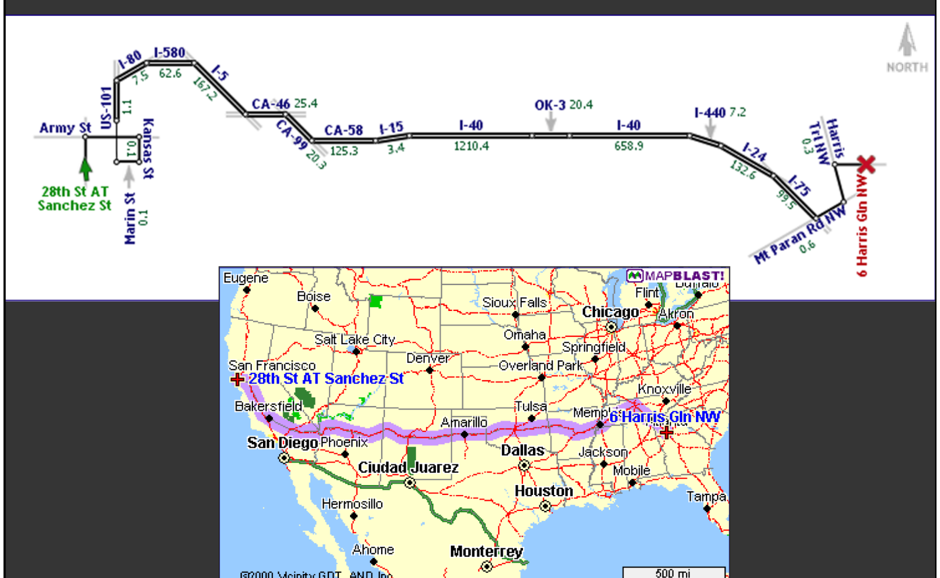
Place cross-streets and exit signs if possible



Bellevue to Seattle



Cross-Country Route



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

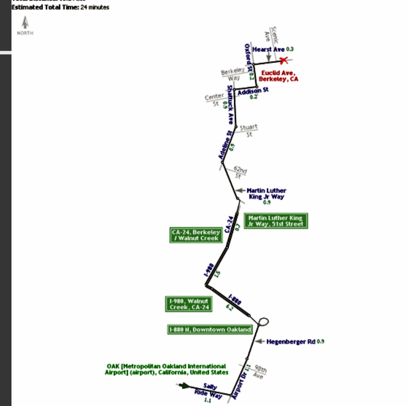
2242 responses

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

Start: OAK [Metropolitan Oakland International Airport] (airport), California, United States
End: Euclid Ave, Berkeley, CA 94709
Total Distance: 15.2 Miles
Estimated Total Time: 24 minutes

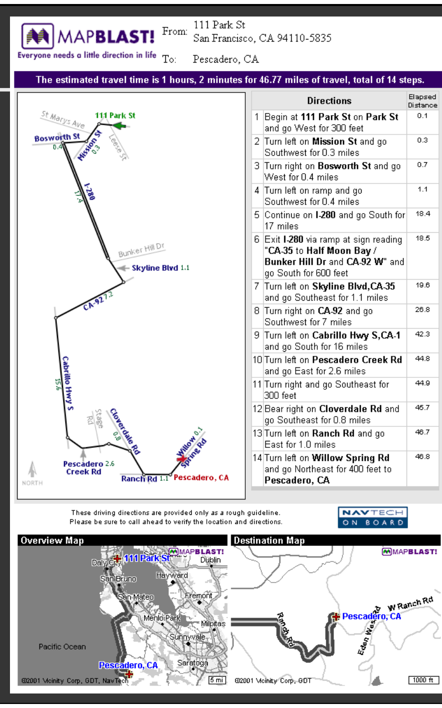


Directions	Miles
Start: Depart OAK [Metropolitan Oakland International Airport] (airport), California, United States on Sally Ride Way (East)	1.0
1: Turn LEFT (North) onto Airport Dr	1.0
2: Turn RIGHT (North) onto Hegenberger Rd	0.8
3: Bear RIGHT (North-West) onto ramp	0.2
4: Continue (North-West) on I-880,Nimitz Fwy	6.1
5: Continue (North) on I-980	1.5
6: Continue (North) on CA-24,Grove Shafter Fwy	0.7
7: Continue (North) on ramp	0.7
8: Bear RIGHT (North) onto Martin Luther King Jr Way	0.9
9: Turn RIGHT (North) onto Adeline St	0.9
10: Bear RIGHT (North) onto Shattuck Ave	0.8
11: Turn RIGHT (East) onto Addison St	0.1
12: Turn LEFT (North) onto Oxford St	0.2
13: Turn RIGHT (East) onto Hearst Ave	0.3
End: Arrive Euclid Ave, Berkeley, CA 94709	

Original Design

Layout

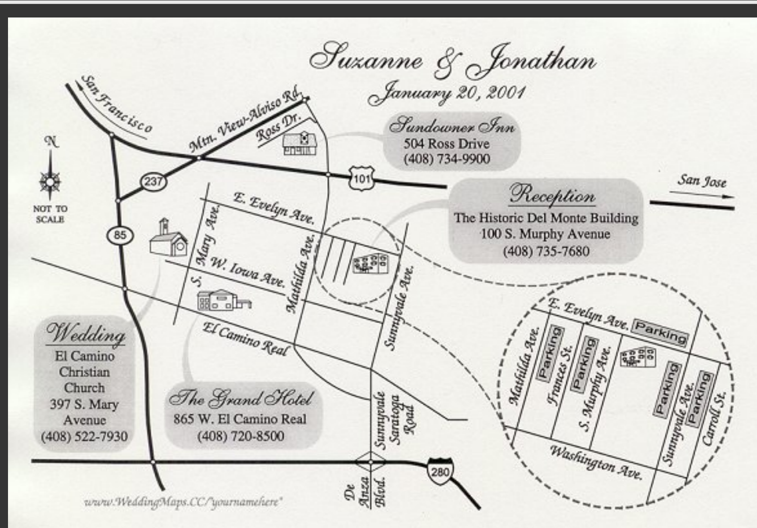
- Map and text close together
- Overview and destination maps for more content



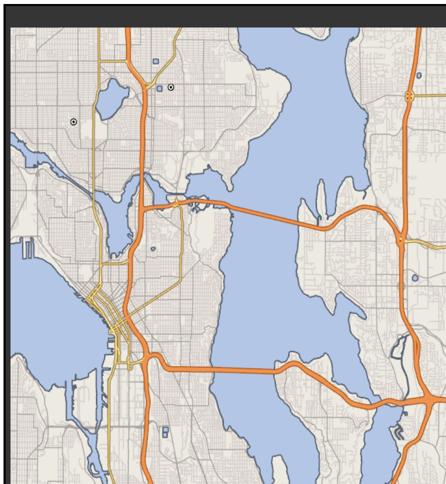
	Directions	Elapsed Distance
1	Begin at **111 Park St** on **Park St** and go West for 300 feet	0.1
2	Turn left on **Mission St** and go Southwest for 0.3 miles	0.3
3	Turn right on **Bosworth St** and go West for 0.4 miles	0.7
4	Turn left on ramp and go Southwest for 0.4 miles	1.1
5	Continue on **I-280** and go South for 17 miles	18.4
6	Exit **I-280** via ramp at sign reading "**CA-35** to **Half Moon Bay / Bunker Hill Dr** and **CA-92 W**" and go South for 600 feet	18.5
7	Turn left on **Skyline Blvd,CA-35** and go Southeast for 1.1 miles	19.6
8	Turn right on **CA-92** and go Southwest for 7 miles	26.8
9	Turn left on **Cabrillo Hwy S,CA-1** and go South for 16 miles	42.3
10	Turn left on **Pescadero Creek Rd** and go East for 2.6 miles	44.8
11	Turn right and go Southeast for 300 feet	44.9
12	Bear right on **Cloverdale Rd** and go Southeast for 0.8 miles	45.7
13	Turn left on **Ranch Rd** and go East for 1.0 miles	46.7
14	Turn left on **Willow Spring Rd** and go Northeast for 400 feet to **Pescadero, CA**	46.8

Limited Resolution PDA

Next Steps: Wedding Maps



Hand-designed Wedding Map www.WeddingMaps.CC

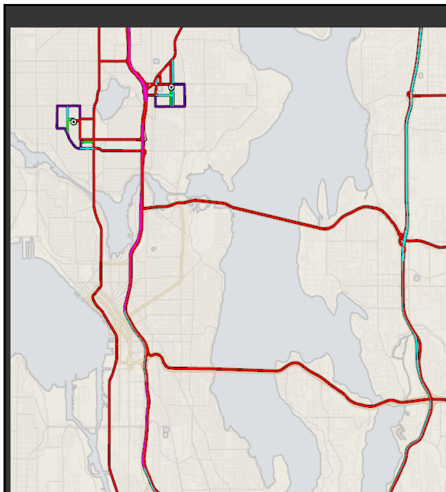


Input map drawn to scale

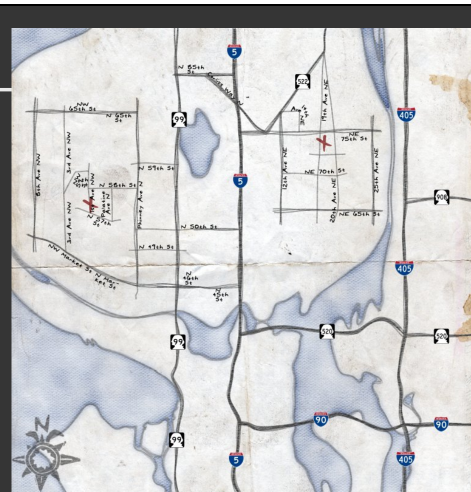
Our result

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

http://www.bing.com/maps/explore/#/c7pvw1whdkp6ggvw (Requires Windows, IE, Silverlight)



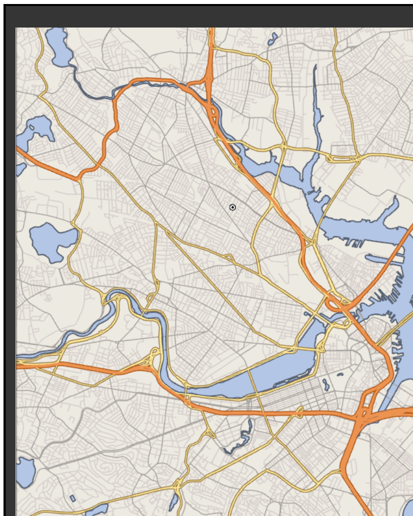
Roads selected from input



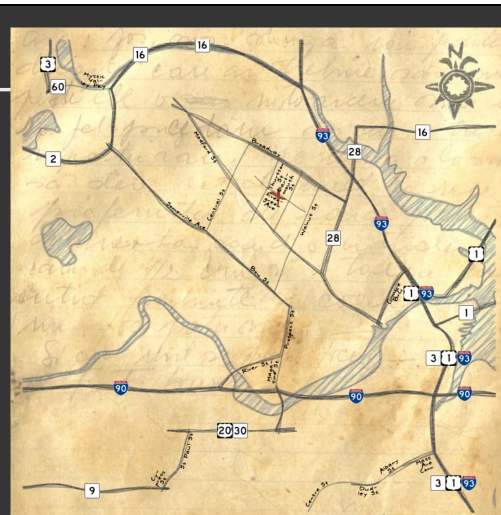
Our result

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

http://www.bing.com/maps/explore/#/c7pvw1whdkp6ggvw (Requires Windows, IE, Silverlight)



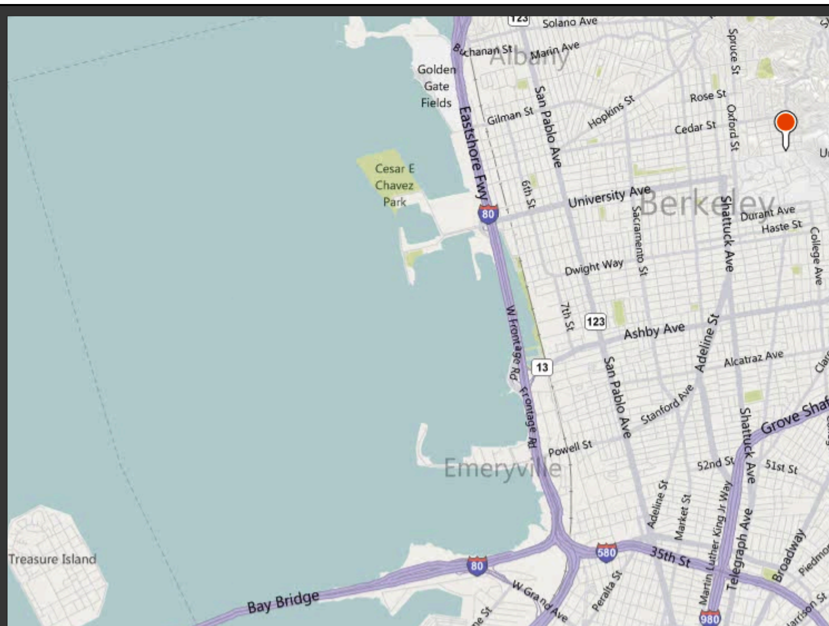
Input map drawn to scale



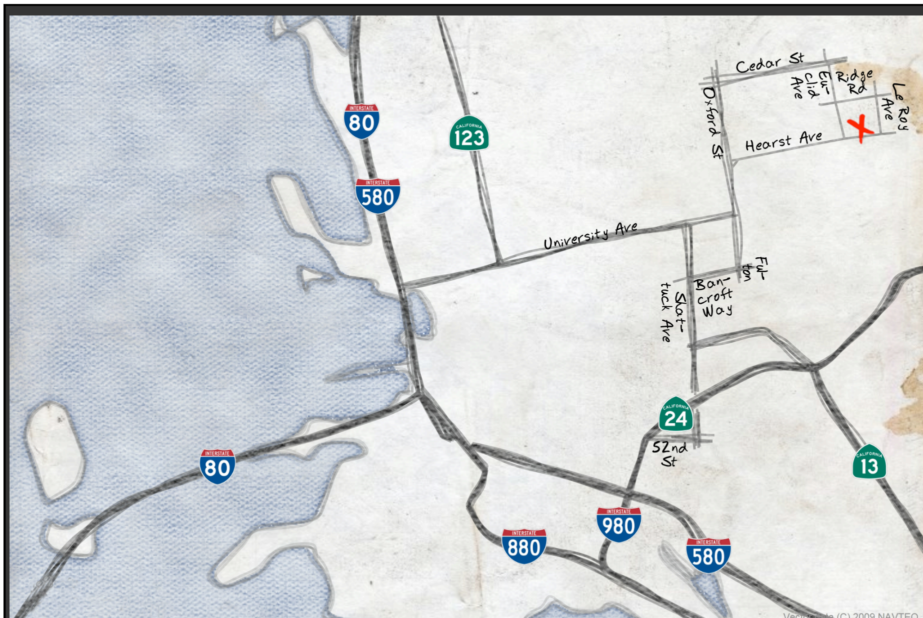
Our result

Evergreen Ave., Boston MA

http://www.bing.com/maps/explore/#/c7pvw1whdkp6ggvw (Requires Windows, IE, Silverlight)



635 Soda Hall, Berkeley CA



635 Soda Hall, Berkeley CA