Spatial Layout

Maneesh Agrawala
Jessica Hullman

CS 294-10: Visualization
Fall 2014

Last Time: Using Space Effectively: 2D

Phases of the Tides

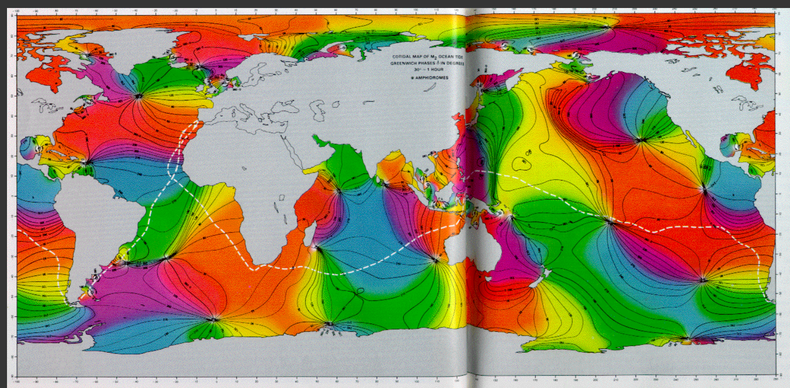
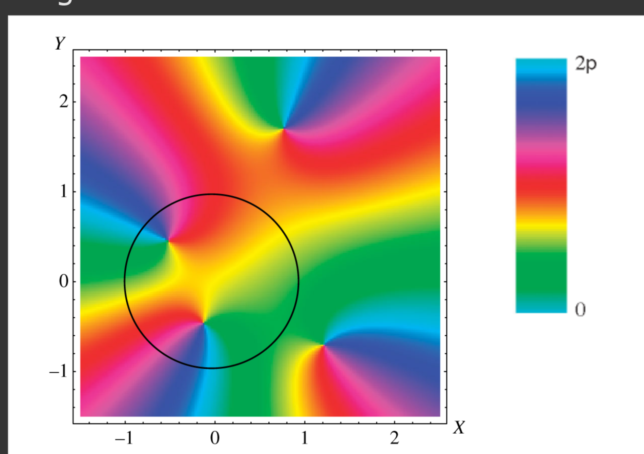


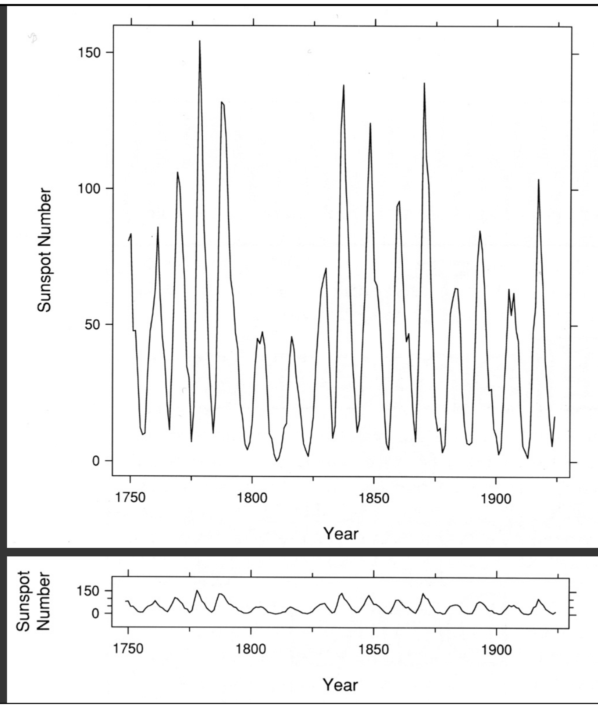
Figure 1.9. Cotidal chart. Tide phases relative to Greenwich are plotted for all the world's oceans. Phase progresses from red to orange to yellow to green to blue to purple. The lines converge on anphidromic points, singularities on the earth's surface where there is no defined tide. [Winfree, 1987 #1195 , p. 17].

Phase Diagrams (hue scale)

Singularities occur where all colors meet

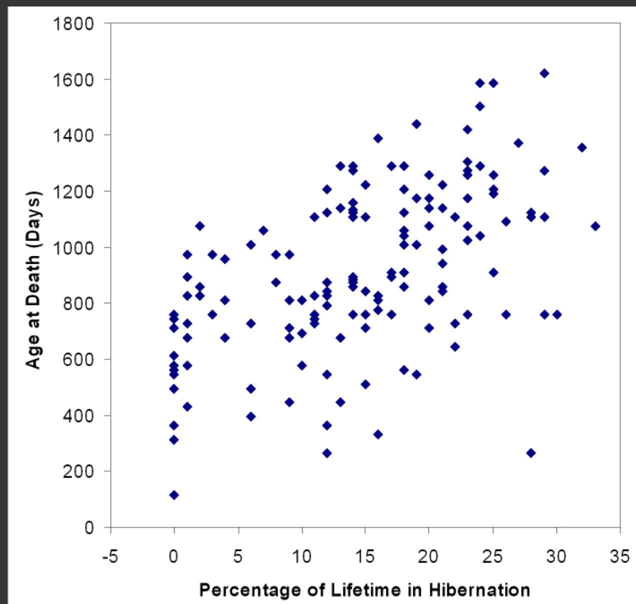


The optical singularities of bianisotropic crystals, by M. V. Berry

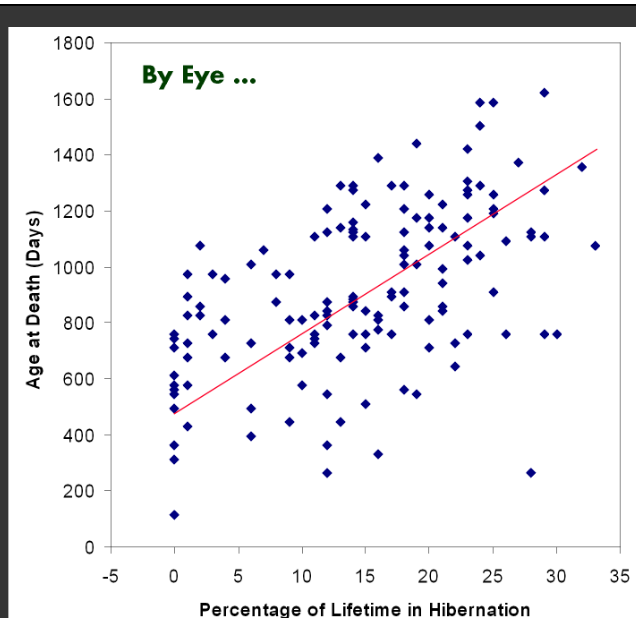


William S. Cleveland
The Elements of Graphing Data

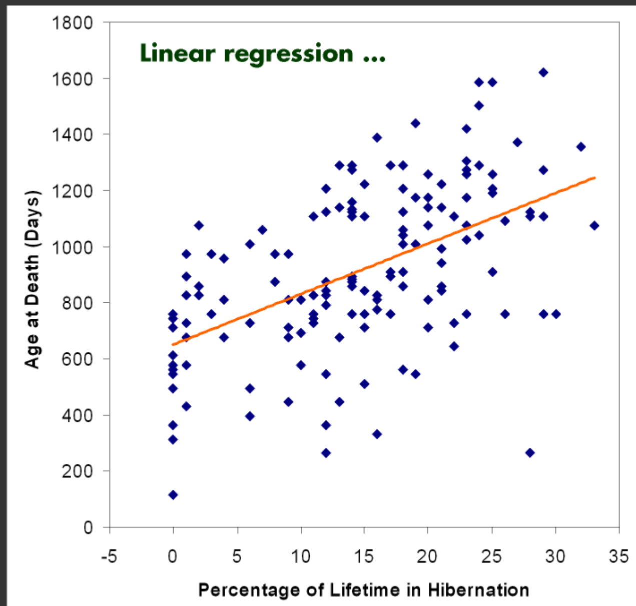
Fitting the Data



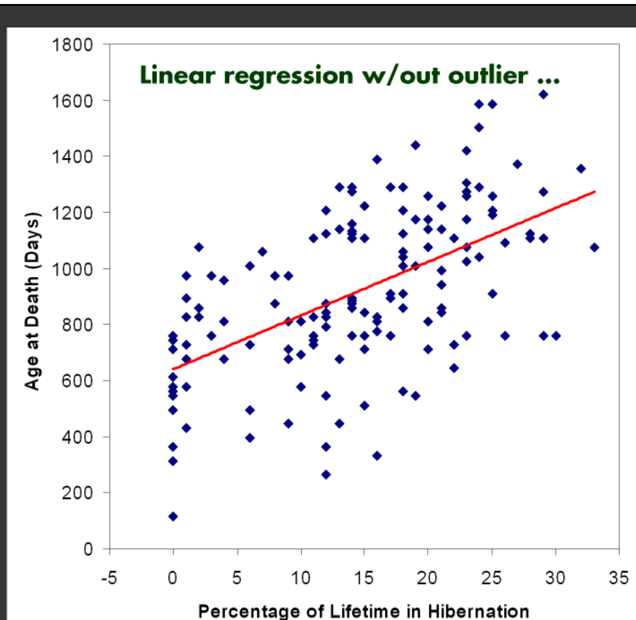
[The Elements of Graphing Data. Cleveland 94]



[The Elements of Graphing Data. Cleveland 94]



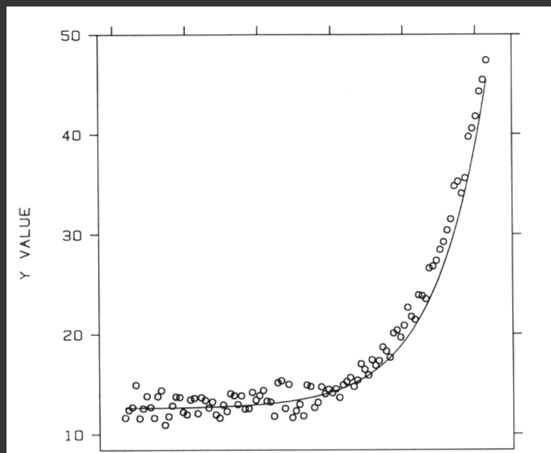
[The Elements of Graphing Data. Cleveland 94]



[The Elements of Graphing Data. Cleveland 94]

Transforming data

How well does curve fit data?

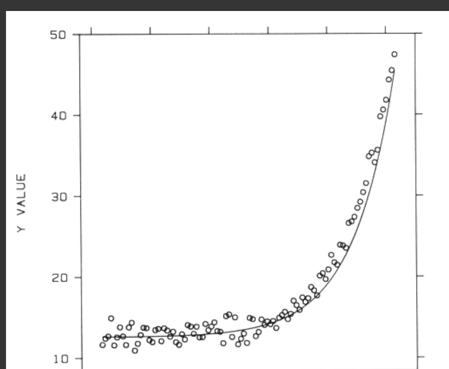


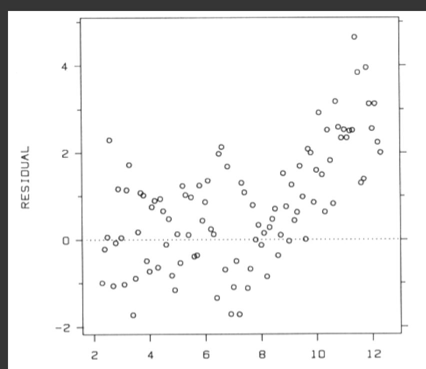
[Cleveland 85]

Transforming data

Residual graph

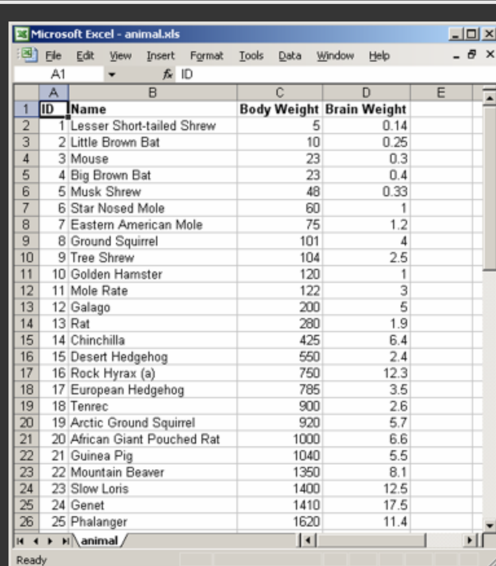
- Plot vertical distance from best fit curve
- Residual graph shows accuracy of fit





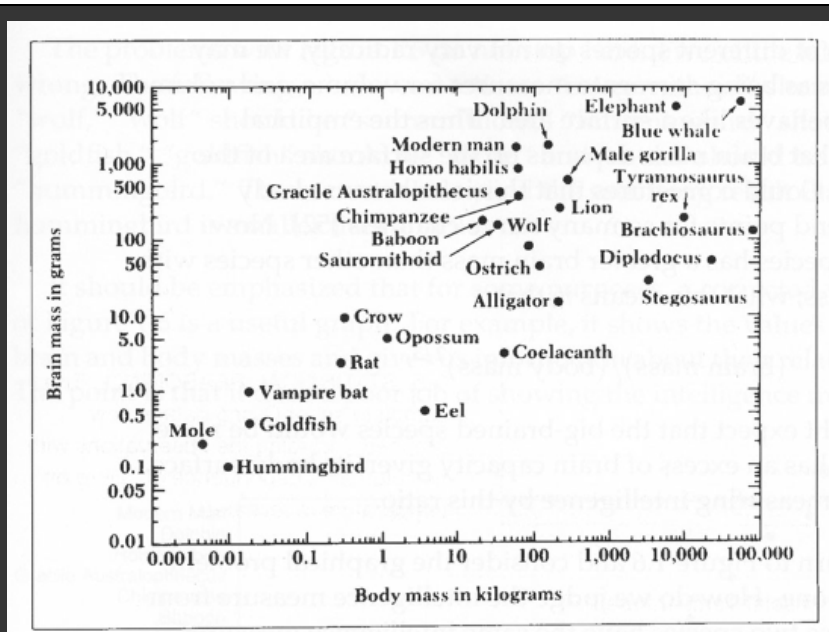
[Cleveland 85]

Most powerful brain?

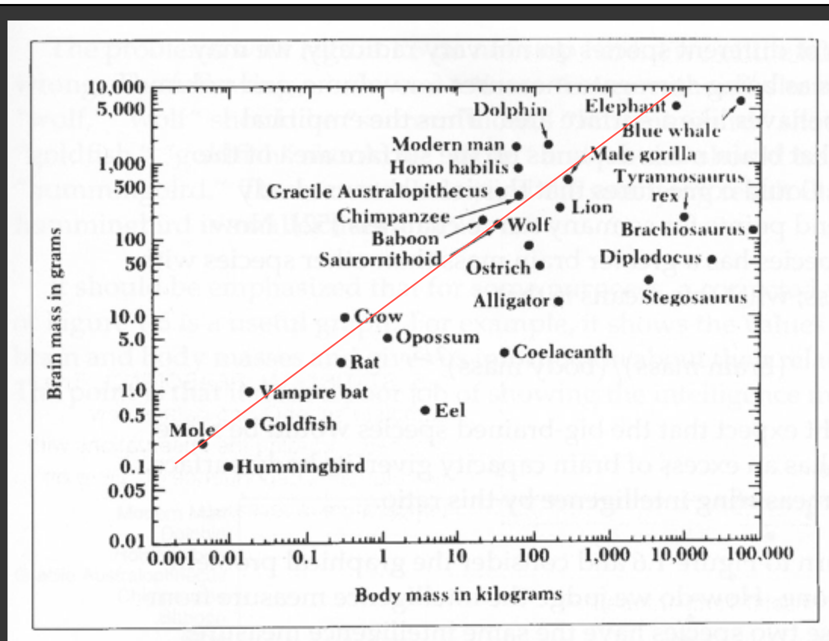


Microsoft Excel - animal.xls

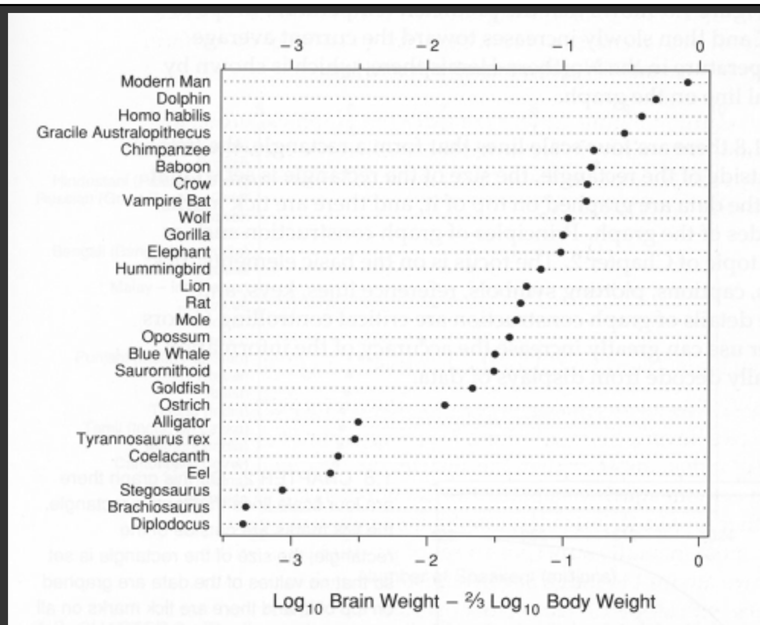
ID	Name	Body Weight	Brain Weight
1	Lesser Short-tailed Shrew	5	0.14
2	Little Brown Bat	10	0.25
3	Mouse	23	0.3
4	Big Brown Bat	23	0.4
5	Musk Shrew	48	0.33
6	Star Nosed Mole	60	1
7	Eastern American Mole	75	1.2
8	Ground Squirrel	101	4
9	Tree Shrew	104	2.5
10	Golden Hamster	120	1
11	Mole Rate	122	3
12	Galago	200	5
13	Rat	280	1.9
14	Chinchilla	425	6.4
15	Desert Hedgehog	550	2.4
16	Rock Hyrax (a)	750	12.3
17	European Hedgehog	785	3.5
18	Tenrec	900	2.6
19	Arctic Ground Squirrel	920	5.7
20	African Giant Pouched Rat	1000	6.6
21	Guinea Pig	1040	5.5
22	Mountain Beaver	1350	8.1
23	Slow Loris	1400	12.5
24	Genet	1410	17.5
25	Phalanger	1620	11.4



The Dragons of Eden [Carl Sagan]



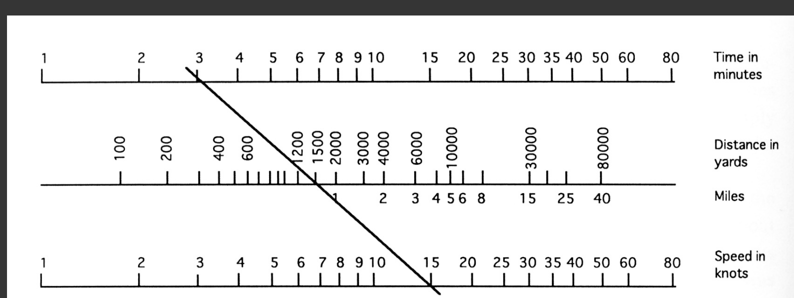
The Dragons of Eden [Carl Sagan]



The Elements of Graphing Data [Cleveland]

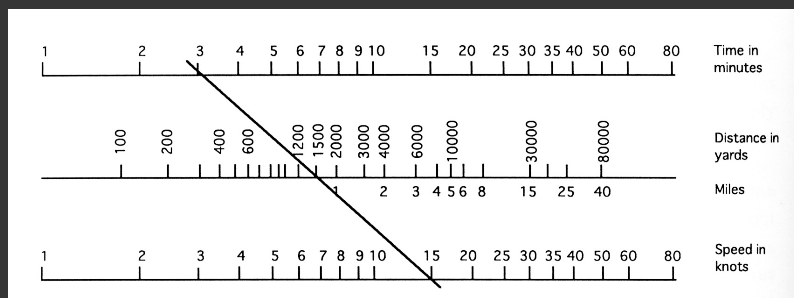
Graphical Calculations

Nomograms



Sailing: The Rule of Three

Nomograms

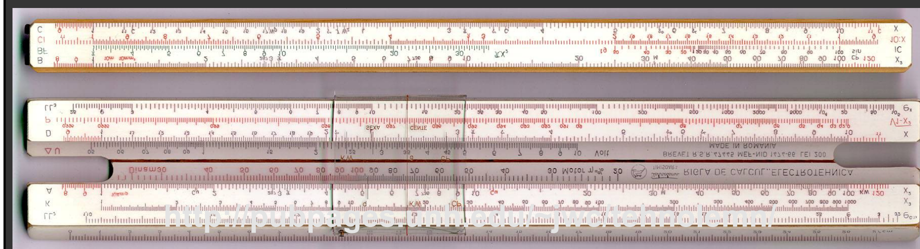


1. Compute in any direction; fix n-1 params and read nth param
2. Illustrate sensitivity to perturbation of inputs
3. Clearly show domain of validity of computation

Theory

$$\begin{vmatrix} x_1(u) & y_1(u) & w_1(u) \\ x_2(v) & y_2(v) & w_2(v) \\ x_3(s,t) & y_3(s,t) & w_3(s,t) \end{vmatrix} = 0$$

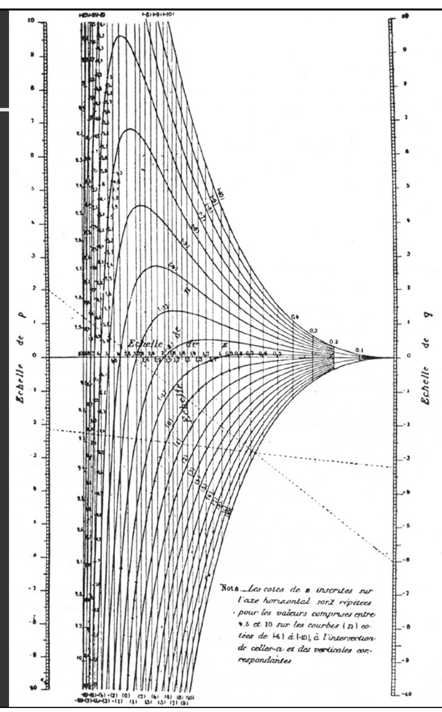
Slide rule

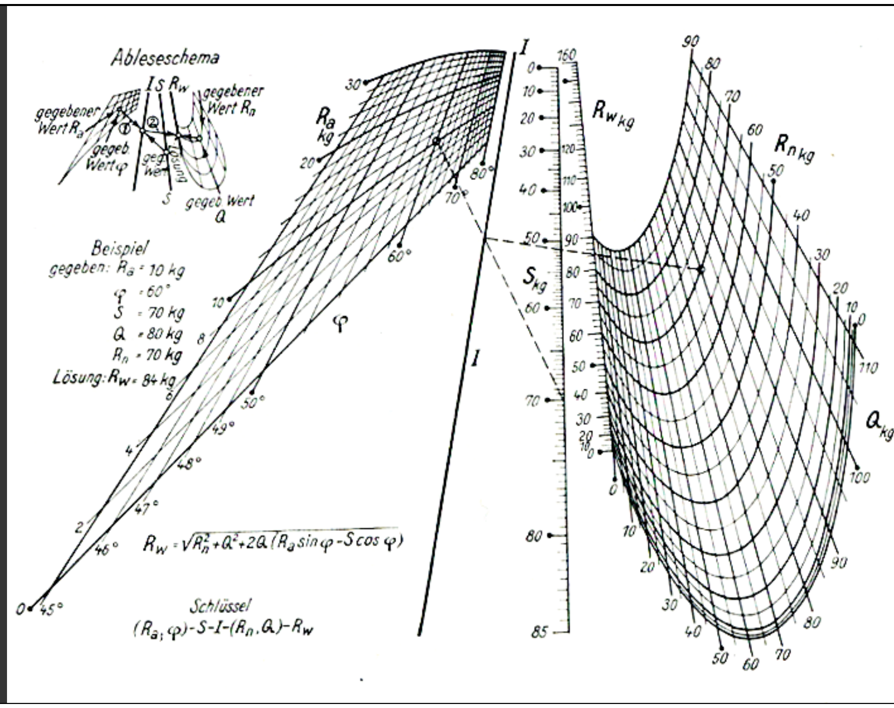


Model 1474-66 Electrotechnica 18 Scales

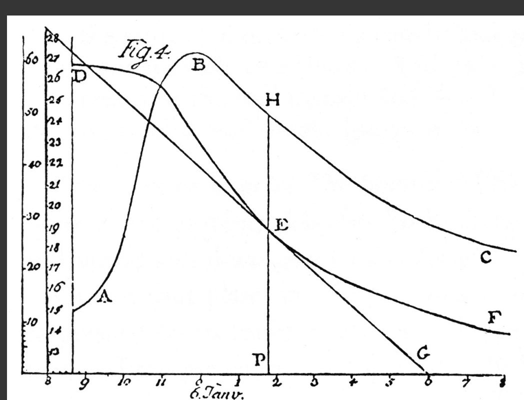
Tehnolemn Timisoara Slide Rule Archive

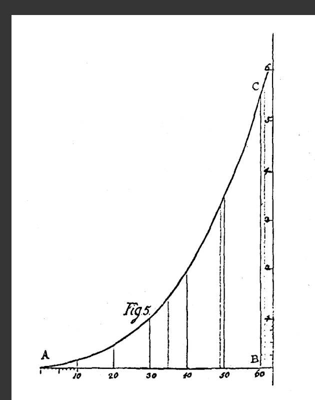
http://pubpages.unh.edu/~jwc/tehnolemn/



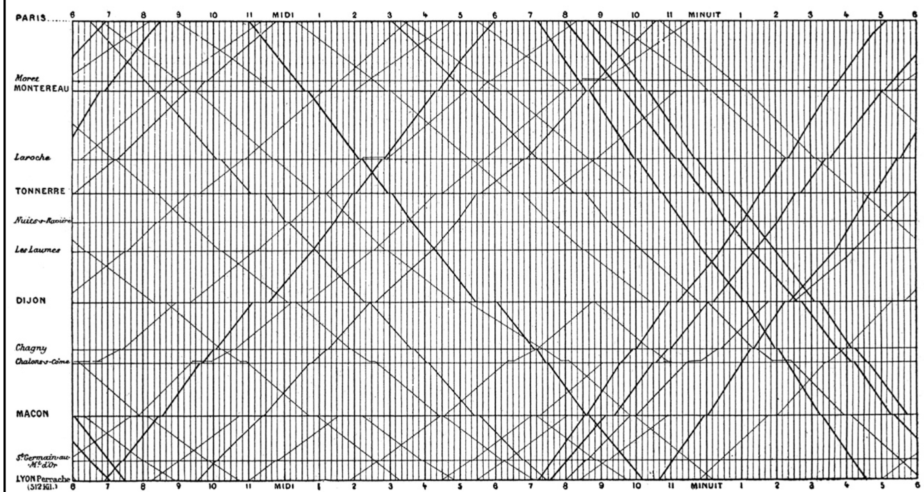


Lambert's graphical construction





Johannes Lambert used graphs to study the rate of water evaporation as function of temperature [from Tufte 83]

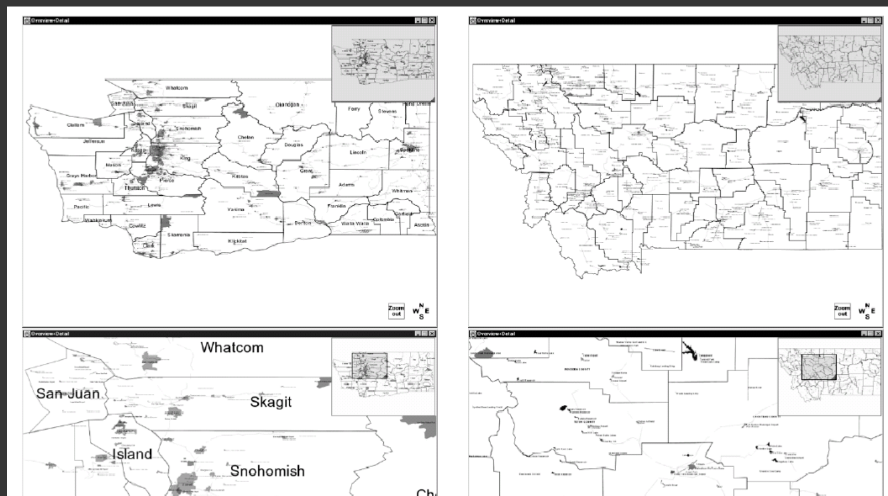


E. J. Marey, *La Méthode Graphique* (Paris, 1885), p. 20. The method is attributed to the French engineer, Ibry.

Zooming

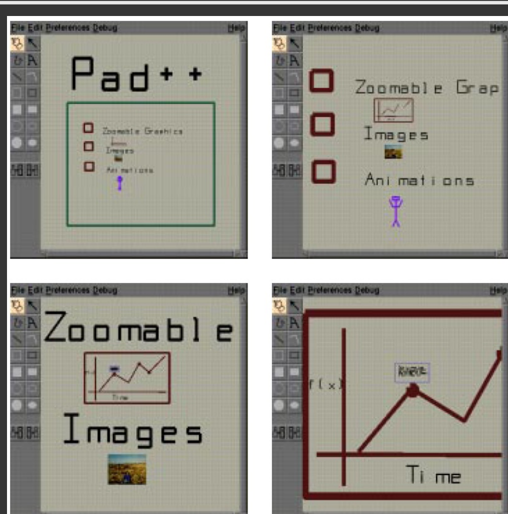
Eames' Powers of Ten [http://www.powersof10.com/film]

Overview + details

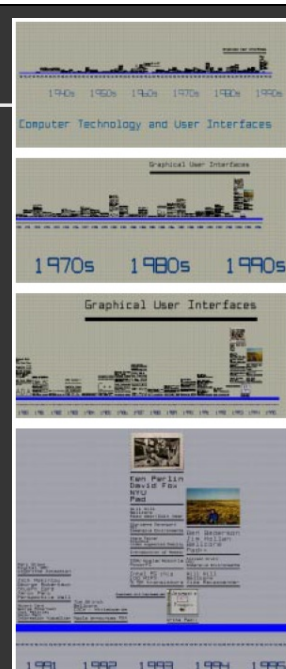


[Hornbaek et al. 2002]

Interactive zooming



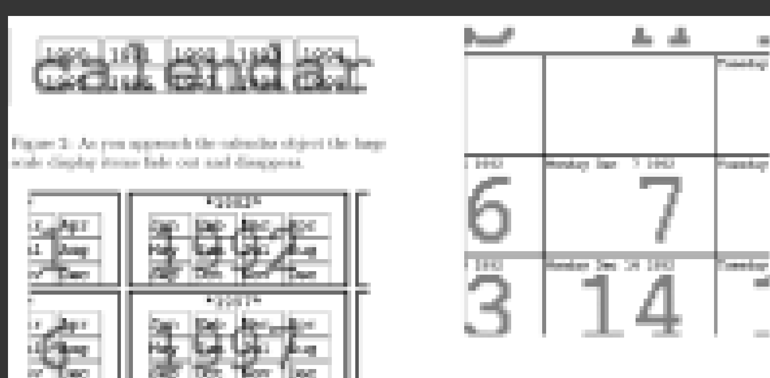
Pad++ [Bederson and Hollan 94]



Pad++

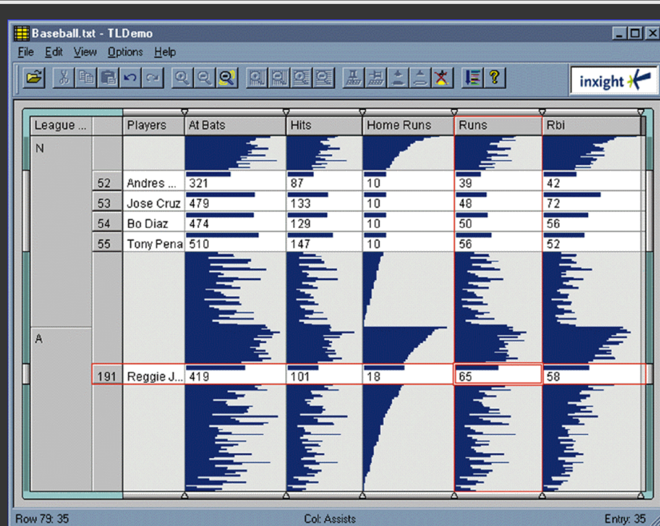
Semantic zooming

Change visual representations as zoom level changes



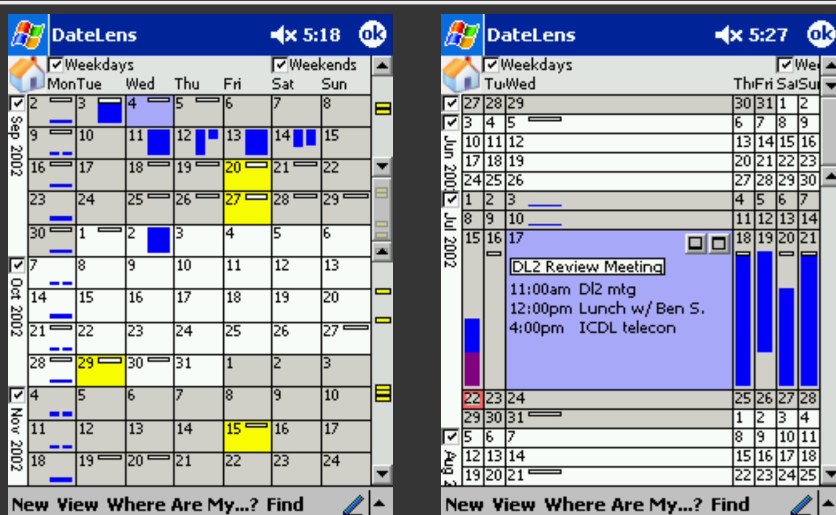
PAD [Perlin and Fox 93]

TableLens [Rao & Card 94]



http://www.youtube.com/watch?v=qWqTrRAC52U

Datelens

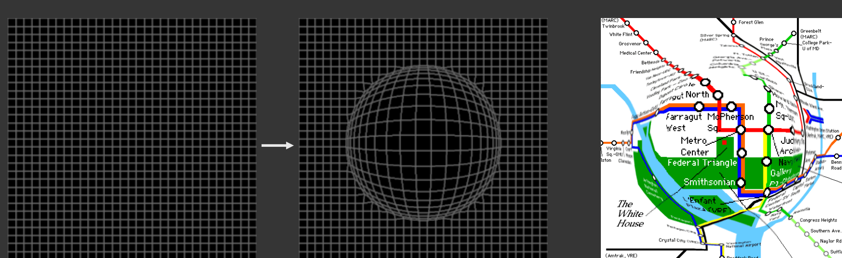


[Bederson et al. 04]

Distortion

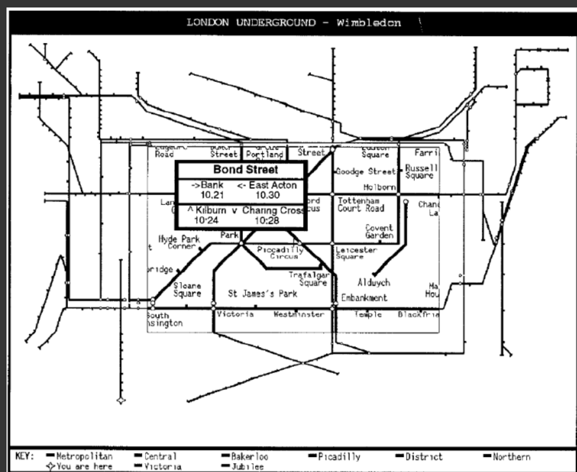
Single view detail + context

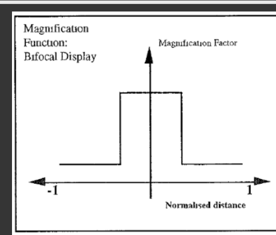
- Focus area – local details
- De-magnified area – surrounding context
- Like a rubber sheet with borders tacked down

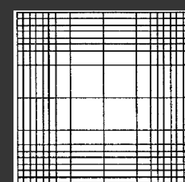


Nonlinear Magnification Infocenter [http://www.cs.indiana.edu/%7Etkeahey/research/nlm/nlm.html]

Bifocal display [Leung and Apperley 94]

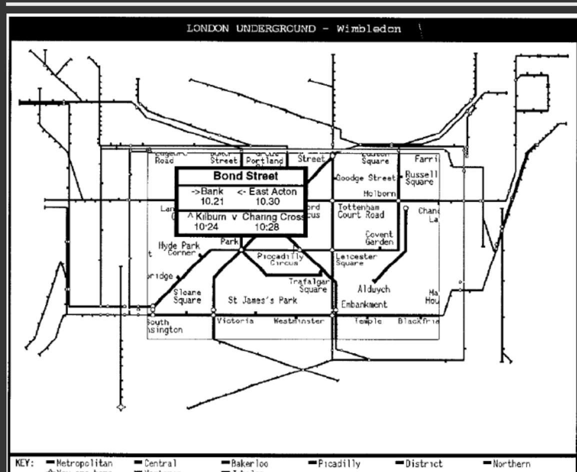


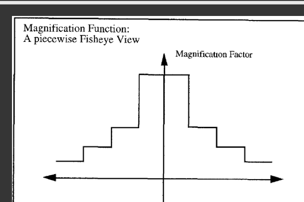




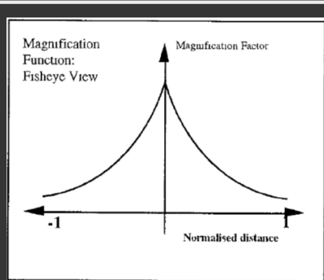
2D distortion

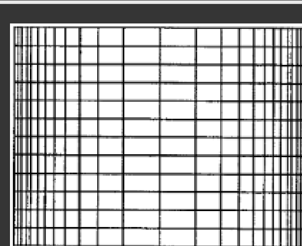
Multifocal display [Leung and Apperley 94]

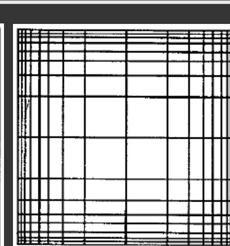




Fisheye [Leung and Apperley 94]

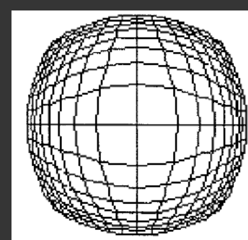






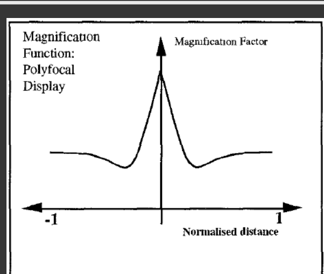
1D

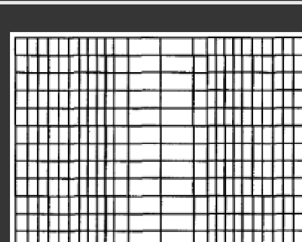
2D

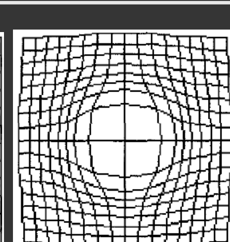


Polar

Nonlinear magnification [Leung and Apperley 94]

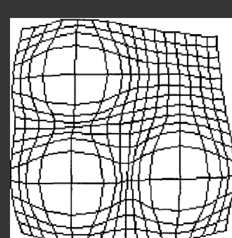


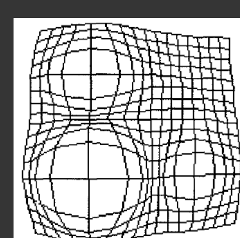




1D

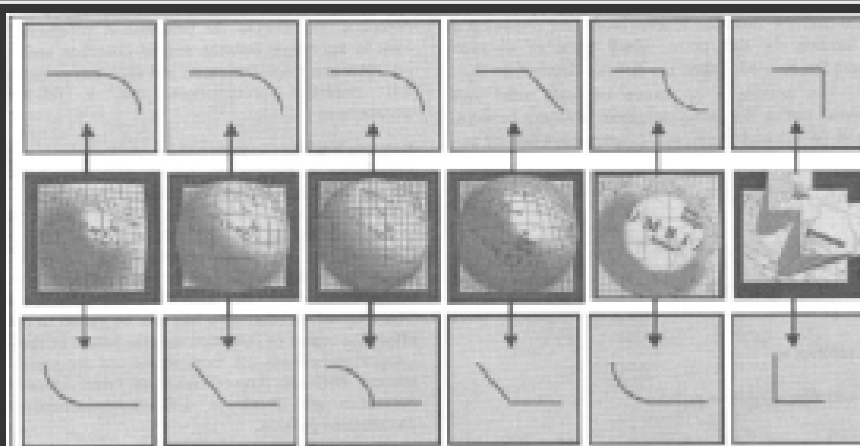
2D





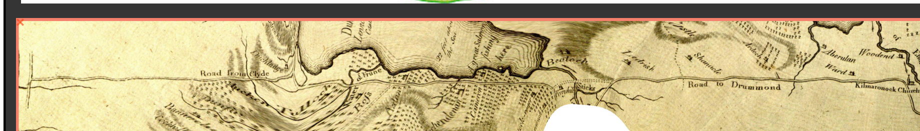
Multifocal

6 types of distortions [Carpendale & Montagnese 01]



Gaussian, Cosine, Hemisphere, Linear, Inverse Cosine and Manhattan. Top row shows transition from focus to distortion, bottom row from distortion to context.

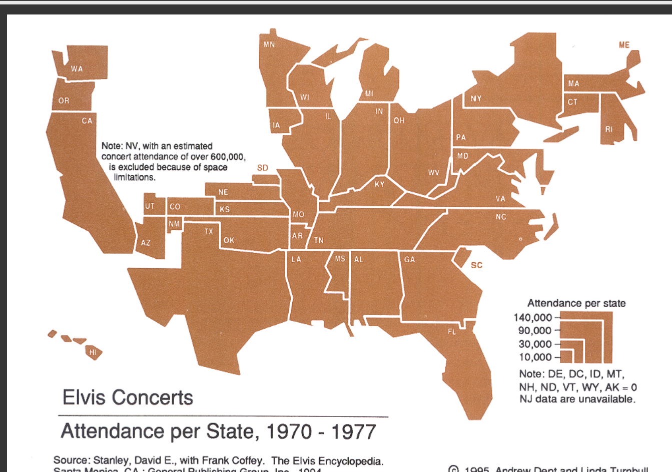
Transmogrifiiers [Brosz et al. 13]



http://www.transmogrifiers.org/

Cartographic Distortions

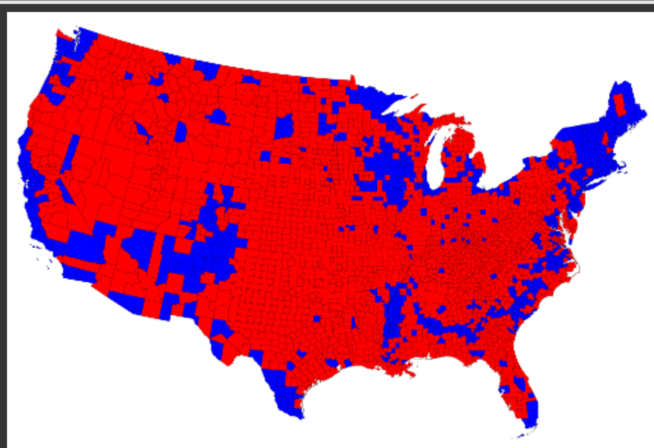
Cartograms: Distort areas



Scale area by data

[From *Cartography*, Dent]

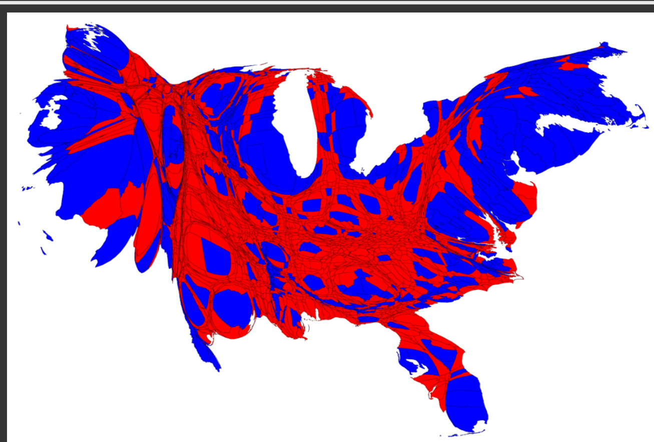
Election 2012 map



% voted democrat
% voted republican

http://www-personal.umich.edu/~mejn/election/

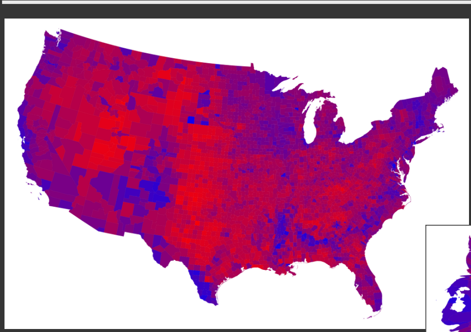
Election 2012 map



% voted democrat
% voted republican

http://www-personal.umich.edu/~mejn/election/

Election 2012 map



http://www-personal.umich.edu/~mejn/election/

Statistical map with shading

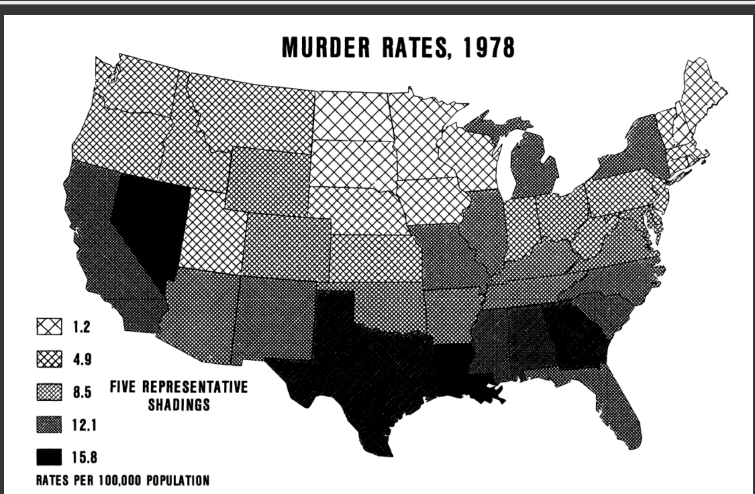
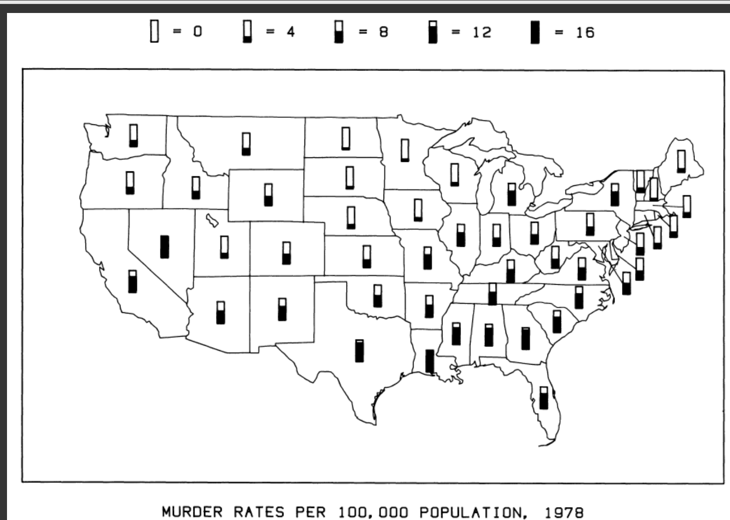


Figure 5. Statistical map with shading.

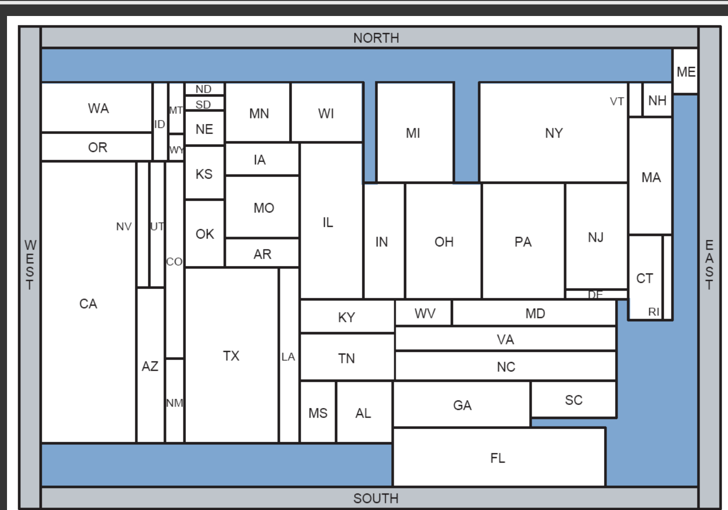
[Cleveland and McGill 84]

Framed rectangle chart



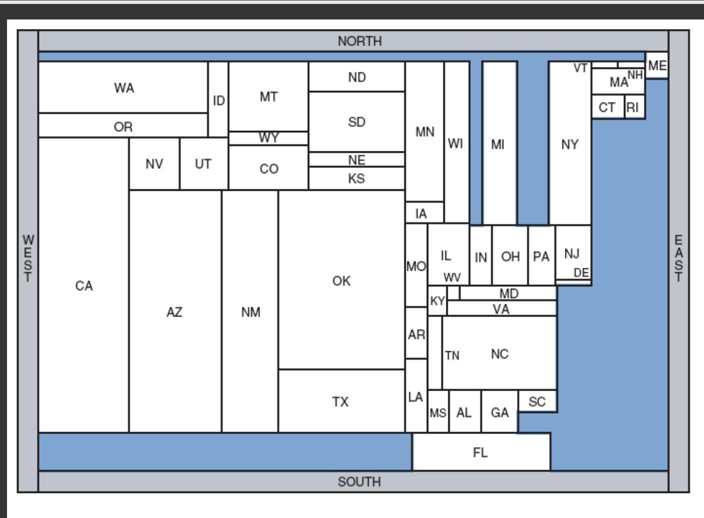
[Cleveland and McGill 84]

Rectangular cartogram



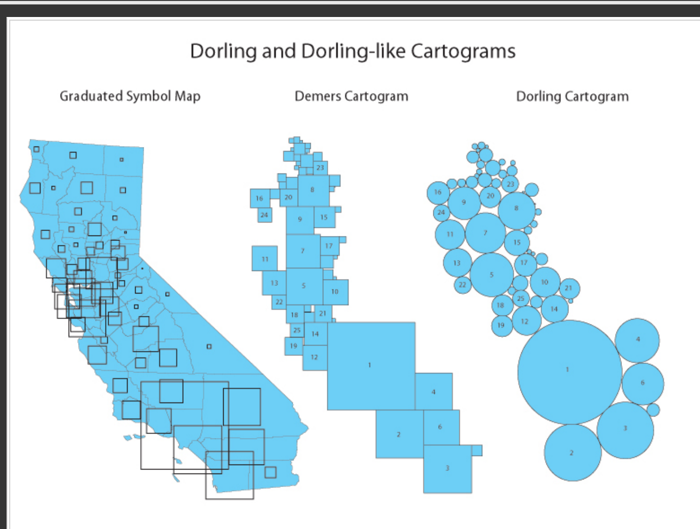
American population [van Kreveld and Speckmann 04]

Rectangular cartogram



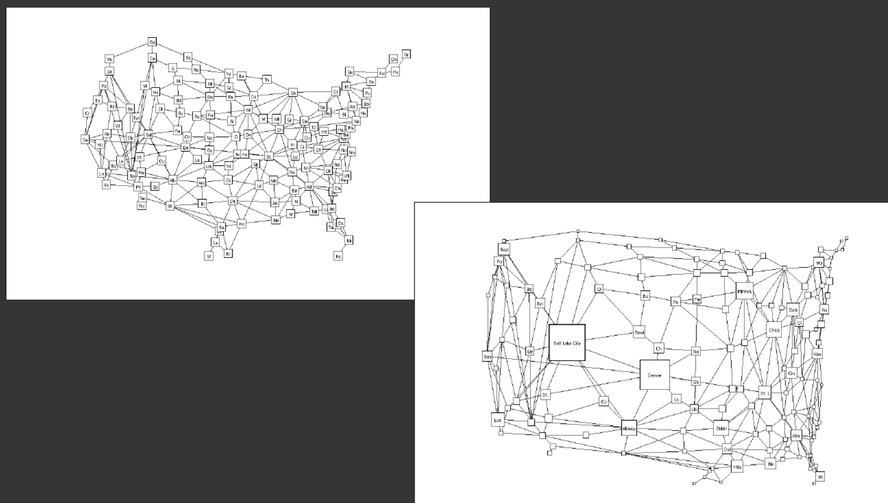
Native American population [van Kreveld and Speckmann 04]

Dorling cartogram



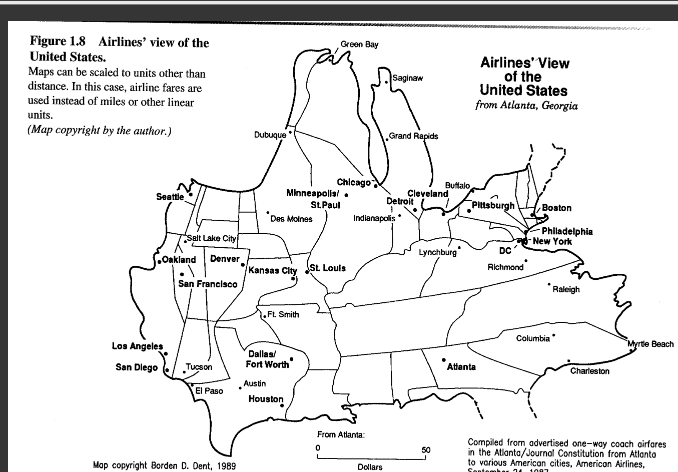
http://www.ncgia.ucsb.edu/projects/Cartogram_Central/types.html

States as nodes in a graph



Graphical fisheye views of graphs [Sarkar & Brown 92]

Distorting distances



Scale distance by data

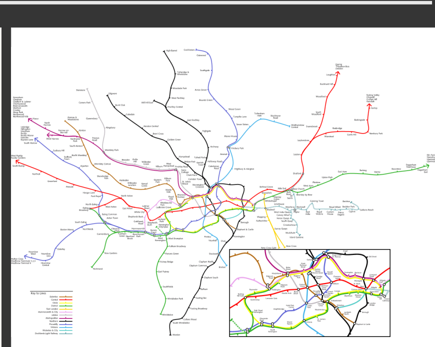
[From *Cartography*, Dent]

London underground

http://www.thetube.com/content/history/map.asp

Comparison to geographic map

Distorted



Undistorted

Summary

- Spatial layout is the most important visual encoding
- Geometric properties of spatial transforms support geometric reasoning
- Show data with as much resolution as possible
- Use distortions to emphasize important information

Announcements

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

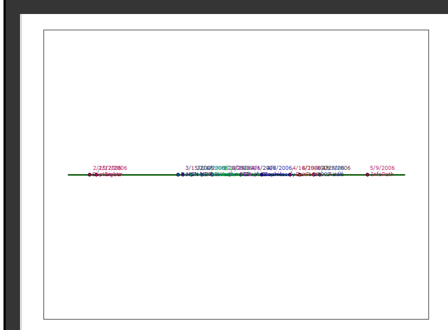
Schedule

- Project proposal: 10/27
- Project presentation: 11/10, 11/12
- Final paper and presentation: TBD, likely 12/1-12/5

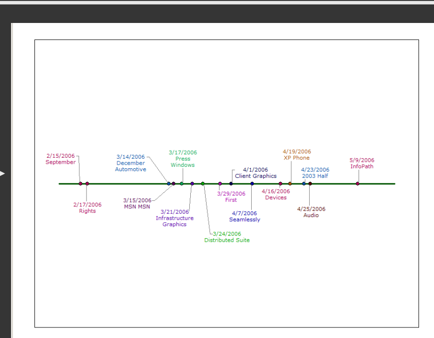
Grading

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

Example: Timeline label layout



→

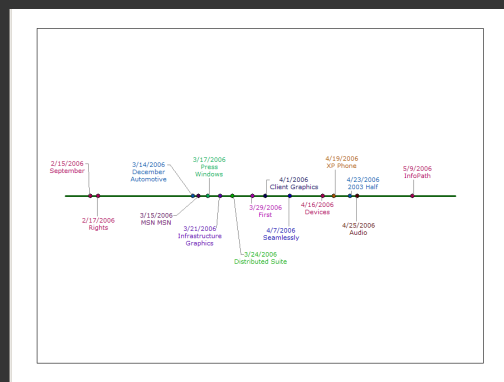


Problem

Input: Set of graphic elements (scene description)

Goal: Select visual attributes for elements

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



Approaches

Direct rule-based methods

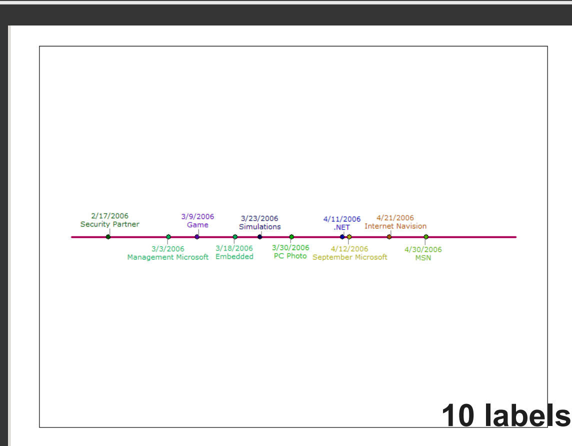
Constraint satisfaction

Optimization

Example-based methods

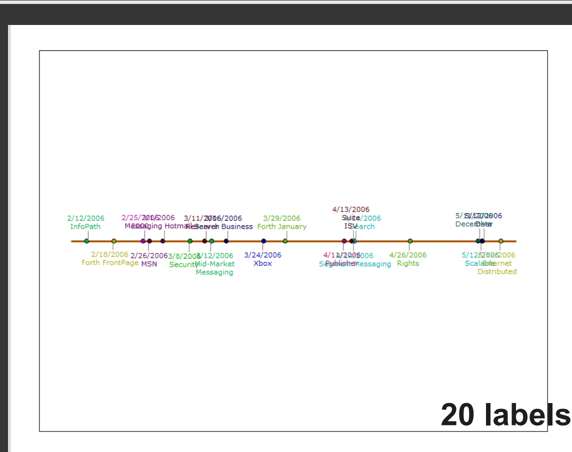
Direct Rule-Based Methods

Rule-based timeline labeling



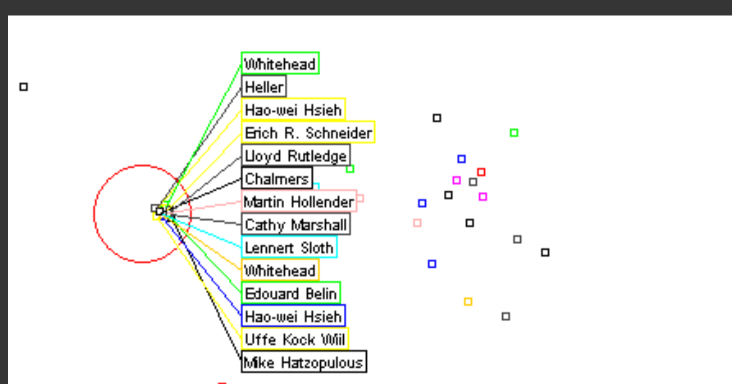
- Alternate above/below line
- Center labels with respect to point on line

Rule-based timeline labeling



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

Excentric labeling [Fekete & Plaisant 99]



http://www.cs.umd.edu/hcil/excentric/

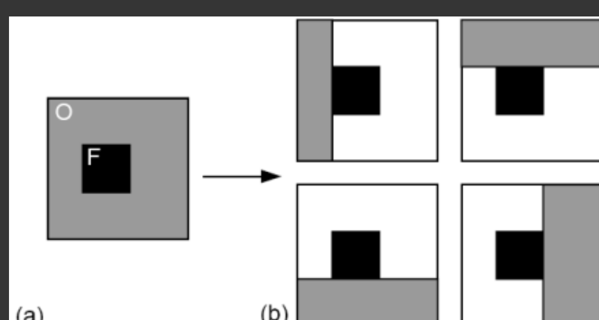
Dynamic space management [Bell 00]

Manage *free space* on desktop to prevent window overlap

Dynamic space management [Bell 00]

Goal: Place new elements to avoid overlap

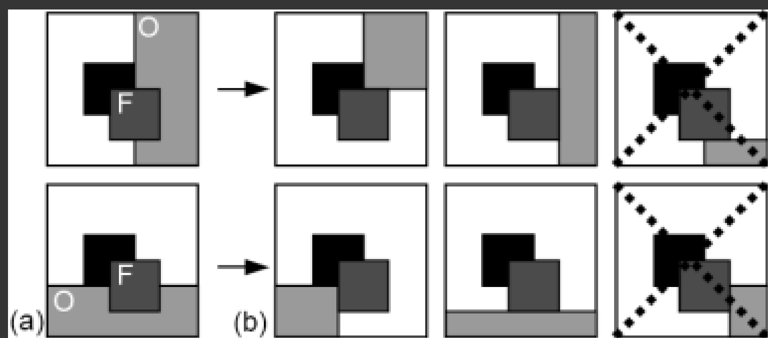
- Elements are axis-aligned rectangles
- Keep track of largest empty space rectangles



Dynamic space management [Bell 00]

Goal: Place new elements to avoid overlap

- Elements are axis-aligned rectangles
- Keep track of largest empty space rectangles



Pros and cons

Pros

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

Cons

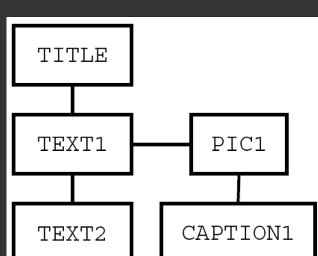
- **Complex layouts require large rule bases with lots of special cases**

Linear Constraint Satisfaction

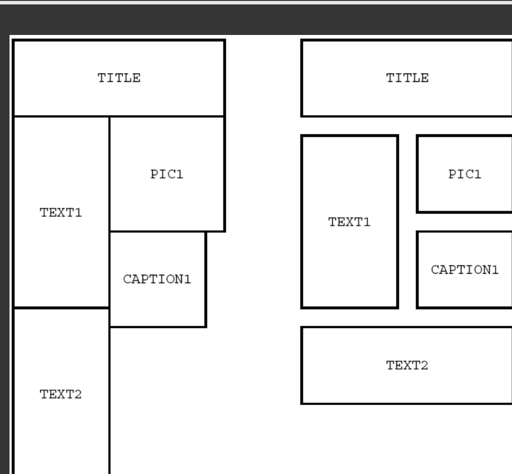
Network of layout constraints

TITLE ABOVE TEXT1
TITLE FULL PAGE WIDTH
TEXT1 LEFT OF PIC1
CAPTION1 BELOW PIC1
TEXT2 BELOW TEXT1

Constraints



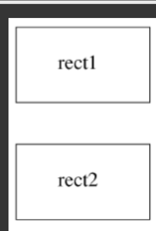
Network



Two possible layouts

[from Lok and Feiner 01]

Constraints as linear equations



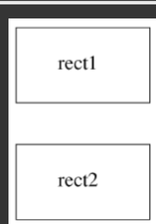
C1: rect2.top = rect1.top + rect1.height + 10
C2: rect2.height = rect1.height
C3: rect2.bottom = rect2.top + rect2.height

Local propagation
- Set any variable
- Update other variables to maintain constraints

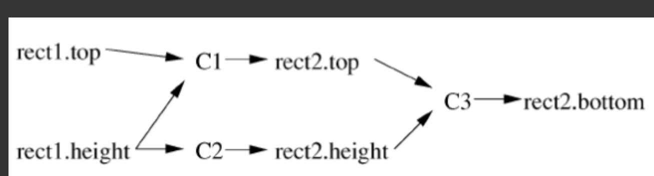
One-way
- Each constraint has 1 output variable
- Update output when any input changes

Multi-way
- Each constraint can be written so that any variable is output
- More complicated to maintain

One-way constraints



C1: rect2.top = rect1.top + rect1.height + 10
C2: rect2.height = rect1.height
C3: rect2.bottom = rect2.top + rect2.height

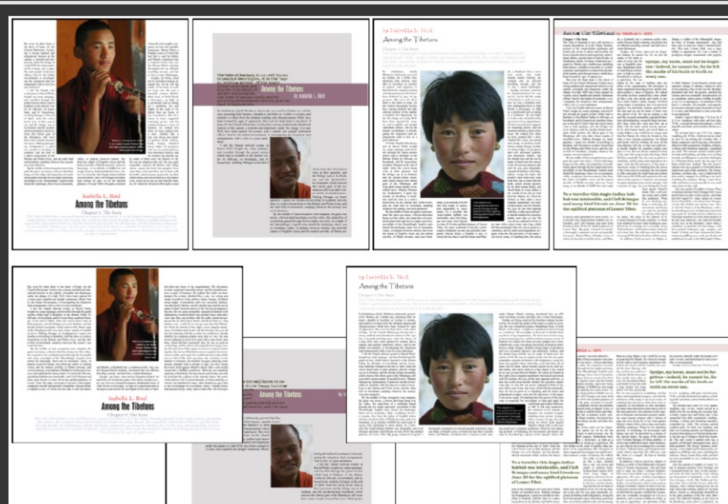


One-way constraints form a directed acyclic graph (DAG). Given the value for any variable we propagate it's value locally through the graph updating the other variable.

Page layout example [Weitzman and Wittenburg 94]

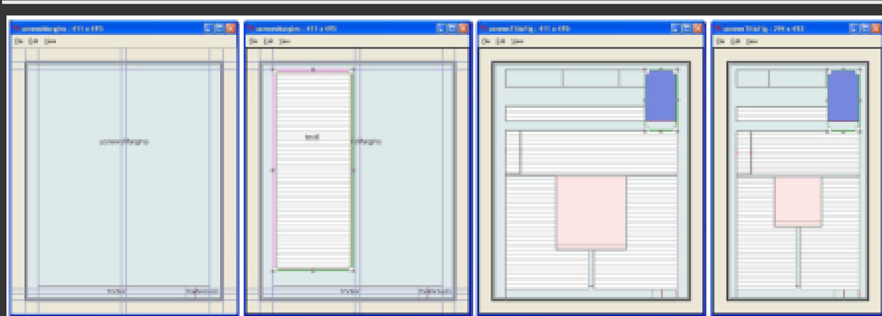
```
(Defrule (Make-Article The-Grammar)
  Article -> Text Text Text Number Image
     0        1    2    3     4      5
  (Author-Of 2 1)
  (Description-Of 4 1)
  (Page-Of 4 1)
  (Image-Of 5 1)
  (article-name 0) = r
  (article-image 0) = 5
  :OUT
  (right-of 1 5)
  (top-aligned 1 5)
  (top-aligned 5 4)
  (spaced-below 2 1)
  (spaced-below 3 2)
)
```

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]



Adaptive Grid~Based Document Layout

Chuck Jacobs[1] Wilmot Li[2] Evan Schrier[2]
David Bargeron[1] David Salesin[1,2]

[1]Microsoft Research [2]University of Washington

Pros and cons

Pros

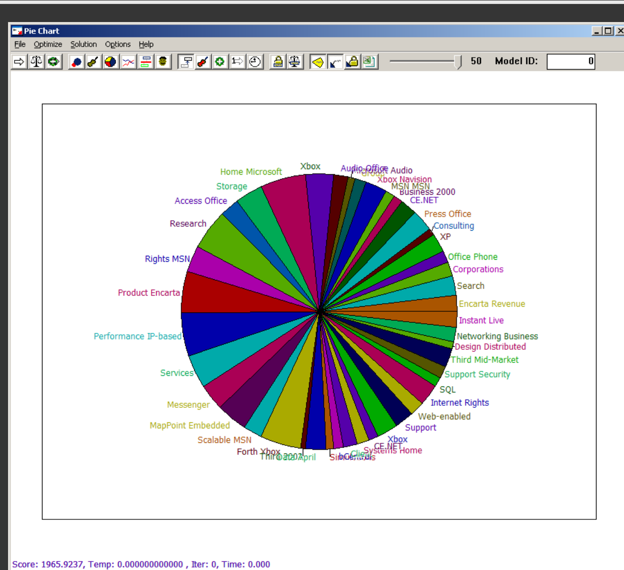
- Often run fast (at least one-way constraints)
- Constraint solving systems are available online
- Can be easier to specify relative layout constraints than to code direct layout algorithm

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

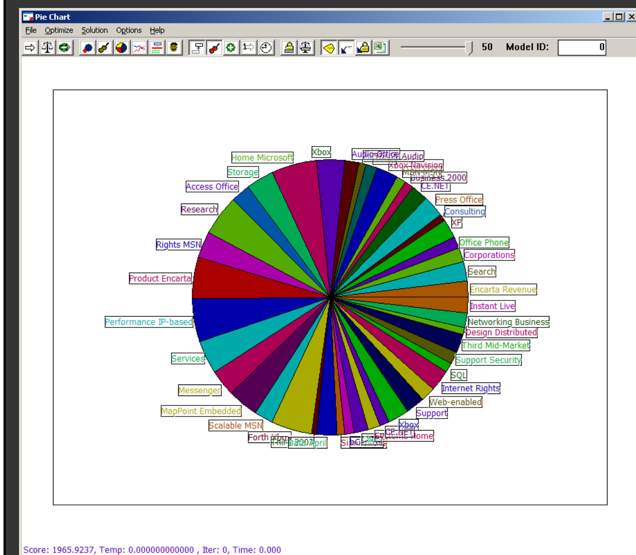
Simulated annealing

```
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^(-ΔE/T)))          Small probability of accepting
    then currL ← newL                   higher penalty
  Decrease(T)
```

Perturb: Efficiently cover layout design space

Penalty: Describes desirable/undesirable layout features

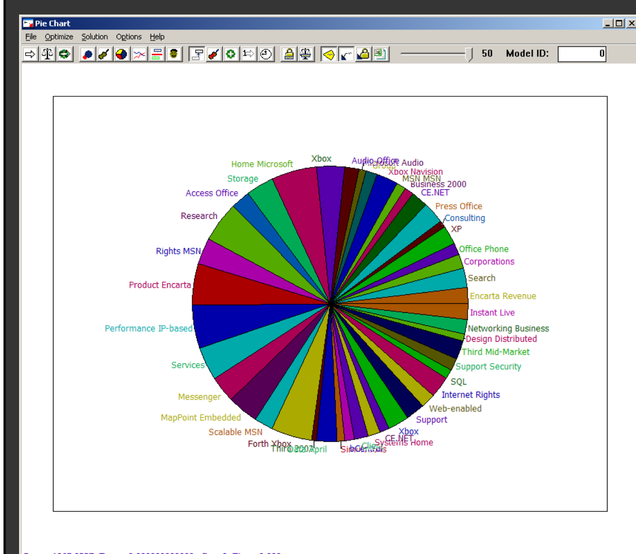
Scene description



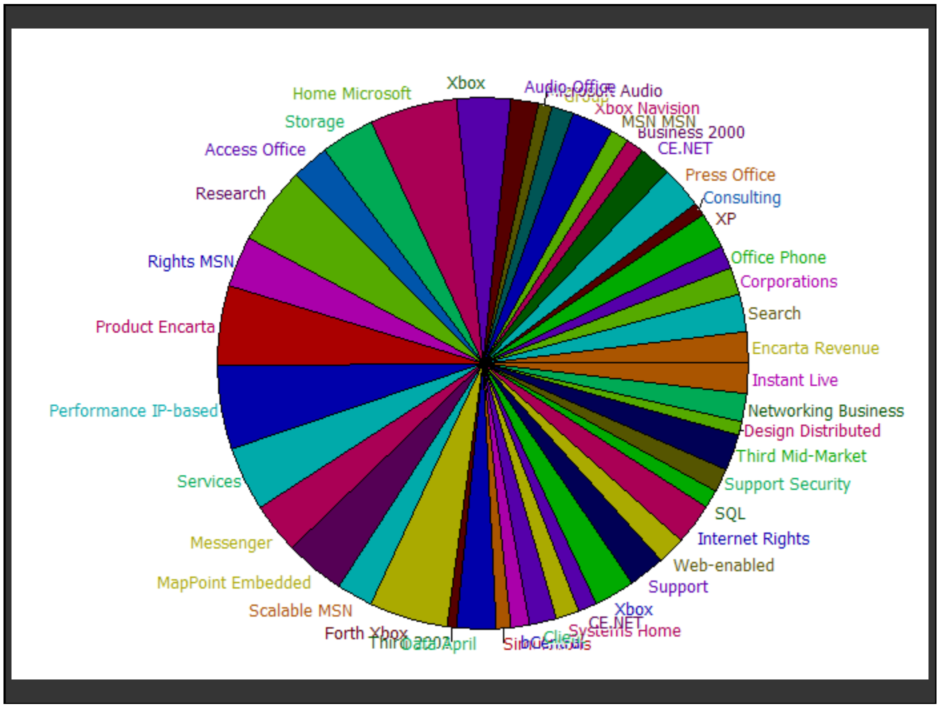
Geometry

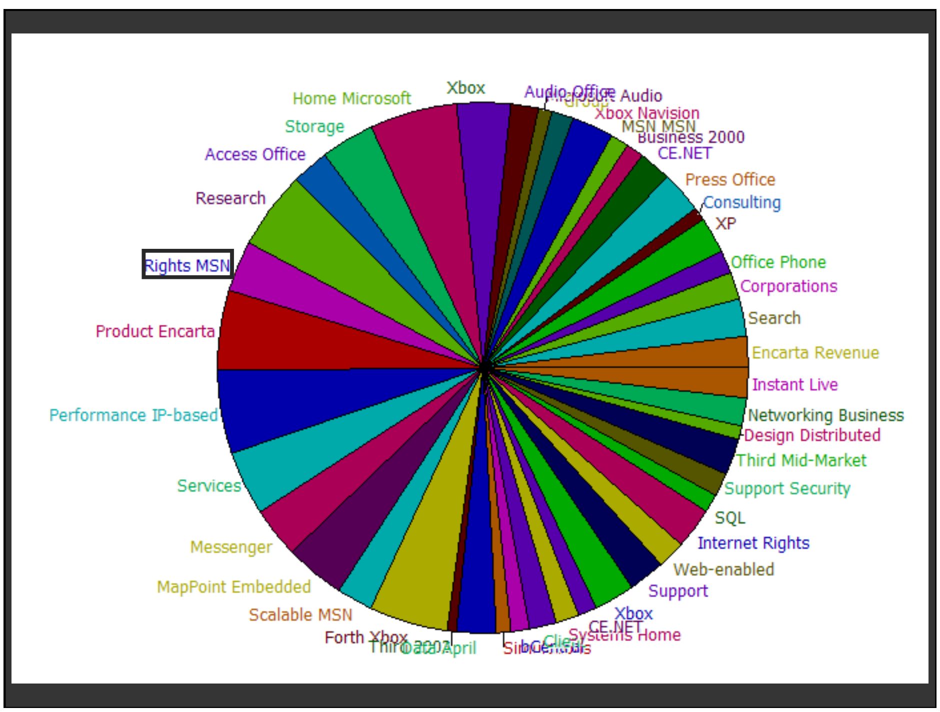
- **Pie slices**
  - anchors for labels
- **Labels**
  - bounding boxes

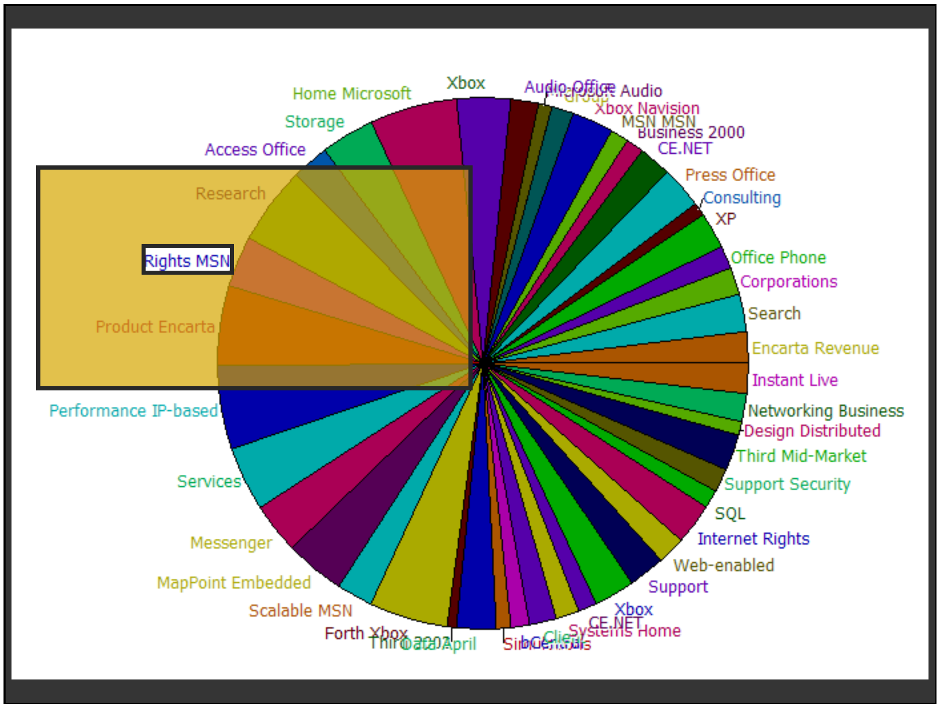
Layout parameters

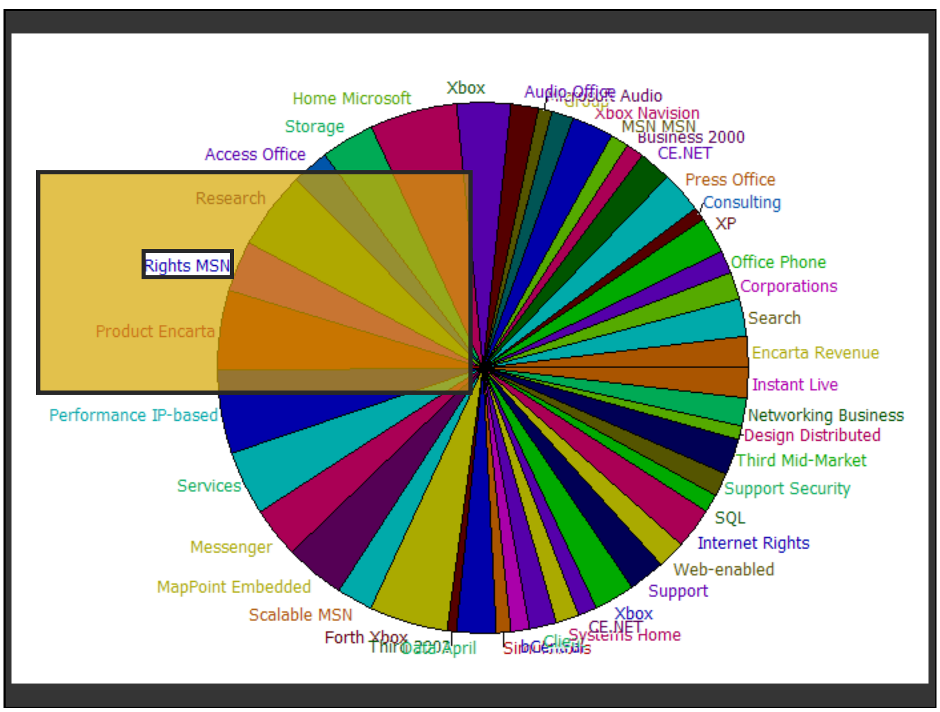


- **Position *(x, y)***
- **Leader line**
- **Word wrap**
- Color
- Alignment
- Orientation
- Scale

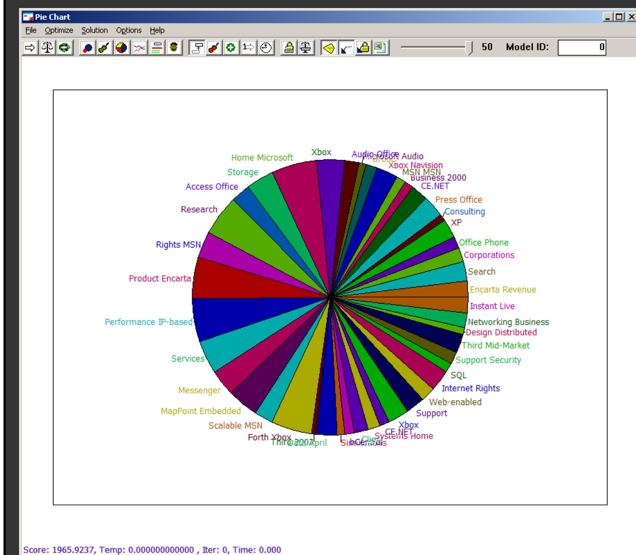

Xbox
Home Microsoft
Storage
Access Office
Research
Rights MSN
Product Encarta
Performance IP-based
Services
Messenger
MapPoint Embedded
Scalable MSN
Forth Xbox
Xbox Navision
MSN MSN
Business 2000
CE.NET
Press Office
Consulting
XP
Office Phone
Corporations
Search
Encarta Revenue
Instant Live
Networking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


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


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


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

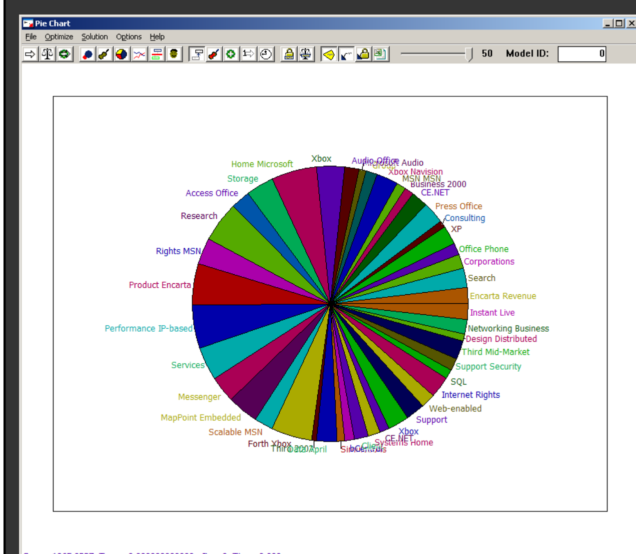
Many dimensions → large space



- **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

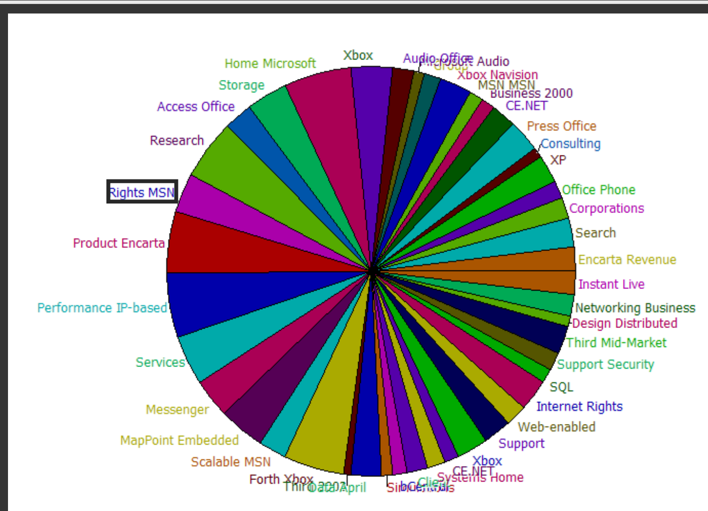
Leader lines

- Length
- Intersections

Word Wrap

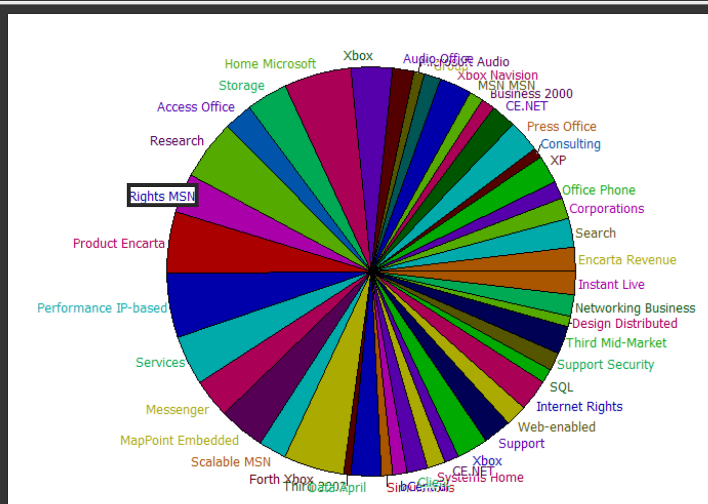
Annealing minimizes sum of all penalties

Overlap: Label – Anchor Slice



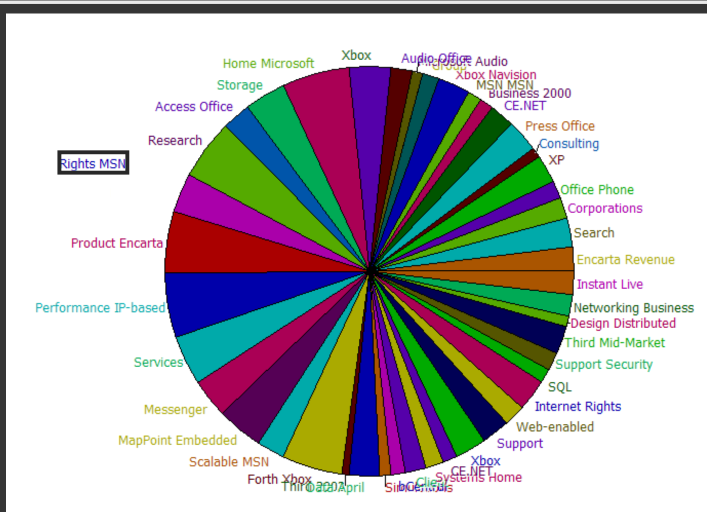
Avoid partial overlap: No penalty if fully inside /outside

Overlap: Label – Anchor Slice



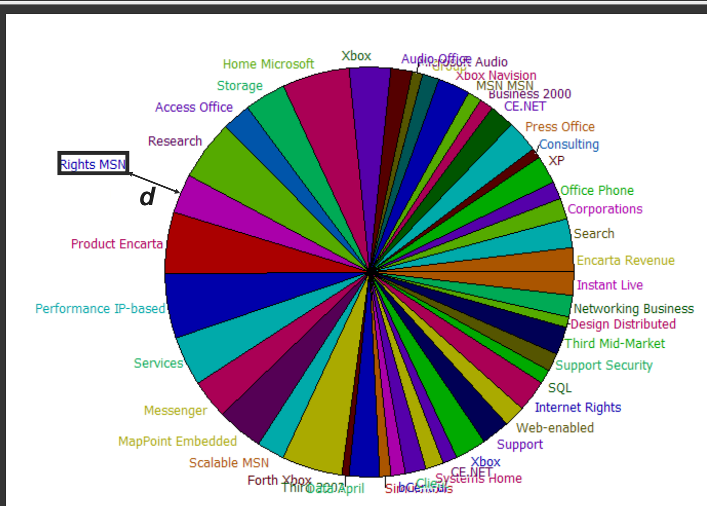
Penalize partial overlap by overlap amount

Distance: Label – Anchor Slice



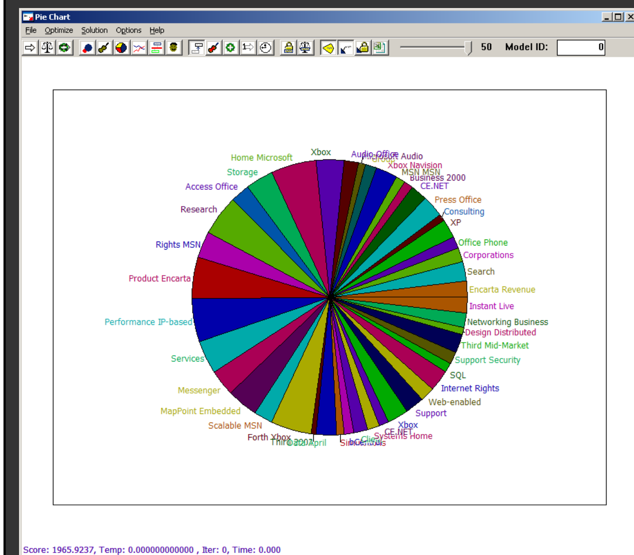
Ensure label near center of edge of anchor slice

Distance: Label – Anchor Slice



Minimize distance *d*

Penalties



Overlap & Distance

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

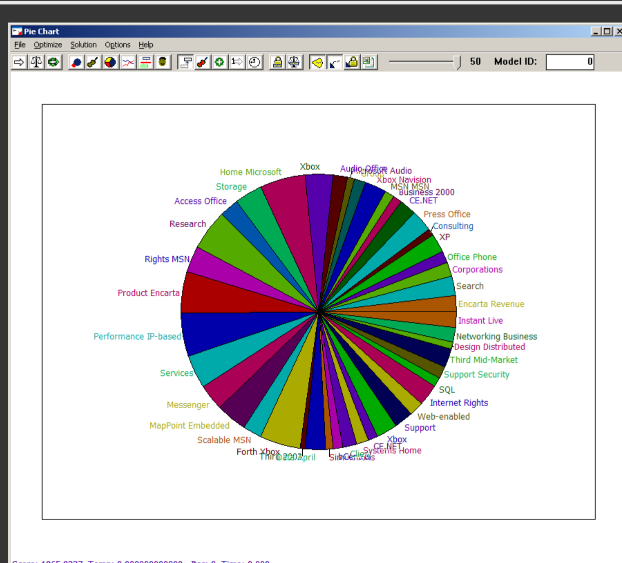
Leader lines

- Length
- Intersections

Word Wrap

Annealing minimizes sum of all penalties

Demo



Pros and cons

Pros

- Much more flexible than linear constraint solving systems

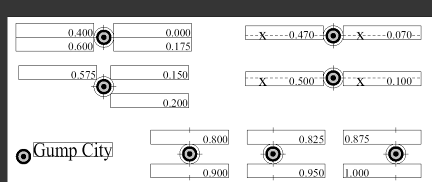
Cons

- 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



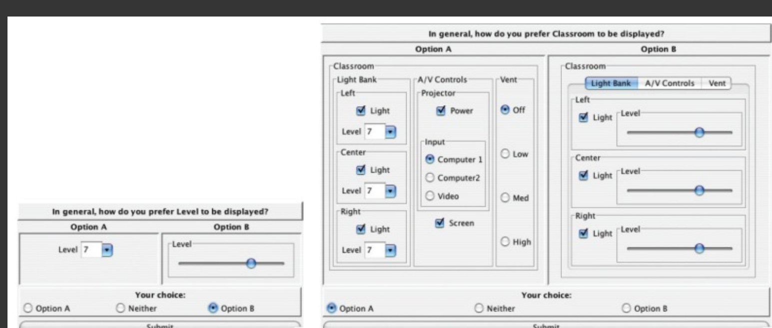
Cartographer Eduard Imhof's labeling heurists transformed into penalty functions for an optimization based point labeling system [Edmondson 97]

Example-Based Methods

Preference elicitation [Gajos and Weld 05]

Learn characteristics of good designs

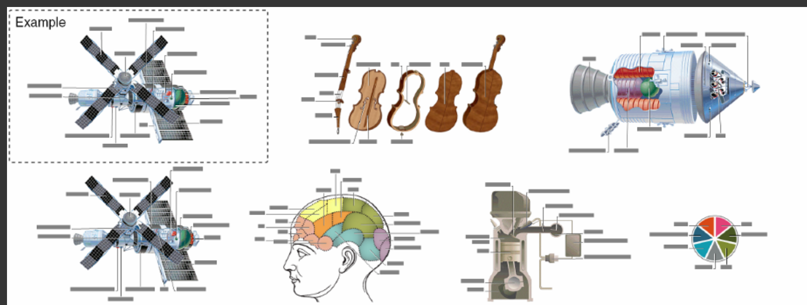
- Generate designs based on a parameterized design space
- Ask designers if they are good or bad
- Learn good parameters values based on responses



(a) (b)

Nonlinear Inverse Opt. [Vollick et al. 07]

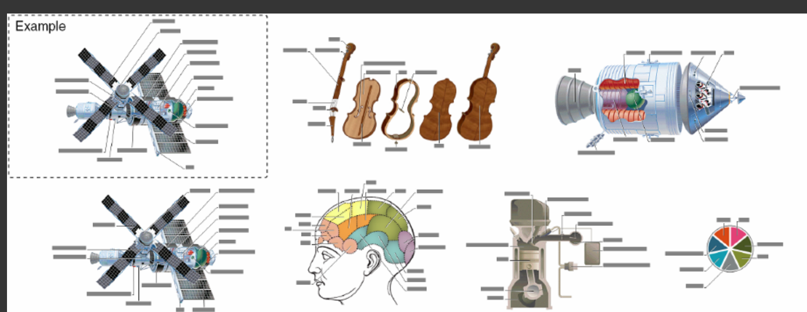
Learn label layout style from single example



Horizontal/Vertical

Nonlinear Inverse Opt. [Vollick et al. 07]

Learn label layout style from single example



Parallel Leader Lines

Artistic Resizing

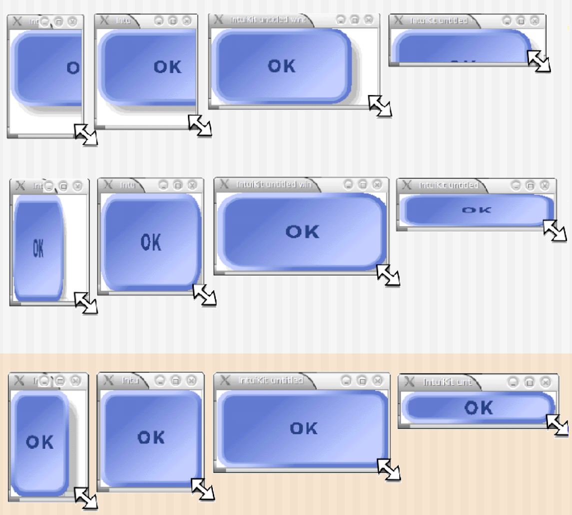
A Technique for Rich
Scale-Sensitive Vector Graphics

Pierre Dragicevic
Stéphane Chatty
David Thevenin
intuilab

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

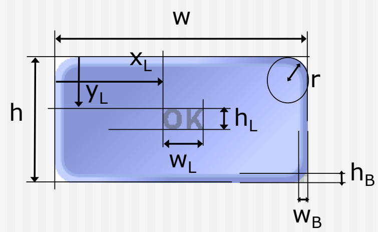
The Resizing Problem

- Fixed size
- Naive scaling
- Artistic resizing



Expressing Artistic Resizing

- Commonly described using formulae



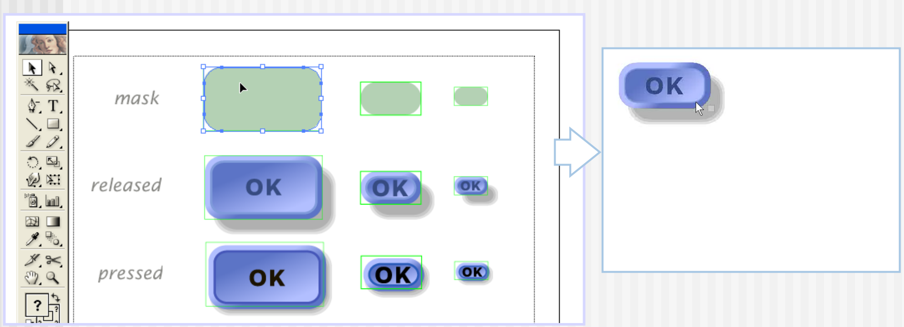
- $x_L = (w-w_L) / 2$
- $y_L = (h-h_L) / 2$
- $w_L = 20$
- $h_L = 10$

- $w_B = 5$
- $h_B = 5$
- $r = 20$

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

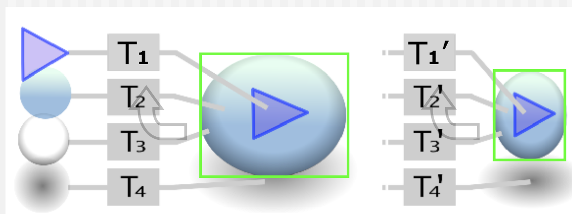
Example-Based Approach

1. Designers produce variants using their authoring tool
2. System interprets the example set



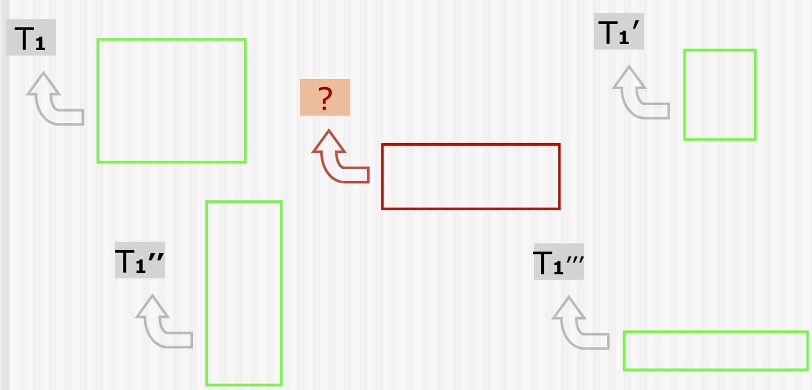
Artistic Resizing
How does it work?

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



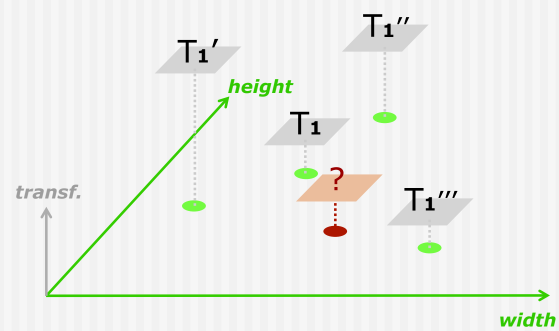
Artistic Resizing
How does it work?

- 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