Design and Perception in Information Visualisation

Matthew Chalmers
Computing Science
University of Glasgow
Overview

• Origins and basic principles
  – generalisations, taxonomies, rules of thumb
  – ‘perception is stronger than mathematics’
• Techniques for specific data types
  – but... types are interchangeable
• General interaction techniques
  – filtering, categorisation, multiple views...
Overview

• What is InfoVis about?
  – absolute and relative views
• Shneiderman’s taxonomy
  – dimensional reduction
• Interaction techniques
What is InfoVis About?

- Information overload
- Three types of navigation
  - spatial, semantic and social
- Metaphor and design
- Good and bad design
- 2D versus 3D visualisation
- Examples of perceptual effects
A Response to ‘Information Overload’

• Too much information to handle
  – can be large data sets, or complex data
  – too much for simple textual displays
  – can 2D and 3D graphics help?

• Opening up the database ‘black box’
  – data hidden from view, queries reach inside

• The tiresome term ‘data mining’
InfoVis as ‘Unrestricted I/O’

• Direct Manipulation in the extreme
  – "visual representation of the world of action"

• Just throwing resources at the problem?
  – CPU, disc space, bandwidth, screen resolution, colours, sound, shape, texture...
Central Themes

• Interaction with (large) data sets
• Abstract, often multiD data
  – cf. scientific visualisation, GIS
• Fit with perceptual skills
  – visual perception, spatial memory
• Understanding relative properties
  – patterns, trends, structure
A Small Example: the ‘Wolfsberg Demo’

- Dual view display
  - change in one affects the other
- Sliders form dynamic queries
  - lightweight query construction
- 24 dimensional data, 2D display
  - map made using spring model
- Perceptually linear colour scales
Three Types of Navigation

• **Spatial**: inherently 2D or 3D data
  – geography, physics

• **Semantic**: metaphorical use of space
  – close ‘is’ similar, high ‘is’ important

• **Social**: using others’ past activity
  – Edit Wear and Read Wear
  – collaborative filtering and paths
Metaphor and Design

• More than a mathematical mapping
• Info. design has to make ‘sense’
  – perceptually: skyscraper example
  – culturally: hot=blue, cold=red?
• Interaction has been proven to be non-formalisable (Wegner)
• Information design: not science but craft
A Skyscraper of Information?

Relevance 0.71

Relevance 0.76

Relevance 0.84

Relevance 0.88

Relevance 0.91
How much of this picturesque scene is *informative*?
Detail is set amongst a lot of polished clutter
Bad Visualisation Design

• ‘Clean’ and ‘natural’, but uninformative
  – the skyscraper example

• Cluttered, overloaded, difficult to pick out what is useful
  – ‘point clouds’ e.g. early Bead, VR-VIBE
  – Xerox PARC’s seminal work (ACM CHI91)
    • ConeTrees, Perspective Wall
Good Visualisation Design

• Technologically efficient and robust
  – science and engineering
• In accord with perception and use
  – psychology, sociology, aesthetics
• Shows useful detail and its context
  – SeeSoft [B1], Spotfire [B4,B5], late Bead?, Wolfsberg demo?
2D v. 3D display

• 3D has another degree of freedom?
  – Can express more relationships,
  – but can you perceive them?

• "The world is 3D"
  – Not in physics -- relativity, etc.

• Not perceptually either
  – height is a privileged property
  – vision: surfaces not volumes
2D v. 3D

• A debate that 2D is slowly winning
• 3D better for spatial navigation?
  – perhaps, if the space/model is familiar
• 2D better for semantic & social nav.
  – occlusion, clutter and orientation
  – burn CPU cycles on data, not pixels
  – text is not designed for 3D but is ubiquitous
Example: Trees

• Two Xerox PARC systems
  – Cone Trees [Plate 15.12]
  – Hyperbolic trees [Plate 15.14]
    • www.inxight.com

• Also many others e.g. Stretch
  – www.elastictech.com
Cone Tree (Robertson et al.)
Cone Tree
Mathematics v. Perception

• What is recorded? Orthogonal? Linear?
  – context and history of use
  – people who made it, use it, want it...
  – position, colour, shape, lightness…

• All perceptually non-orthogonal and non-linear
  – even Cartesian coordinates
  – and even similarity: it’s asymmetric
Linearised Colour Scales

- RGB is perceptually non-linear
  - e.g. (25,50,75) + (25,50,75) \neq (50,100,150)
- Even grey scales are non-linear
Linearised Colour Scales

non-linear

perceptually linear
Linearised Colour Scales

• Applets and Java sources
  – From U. Massachusetts at Lowell
  – examples from book by Haim Levkowitz
  – http://web.cs.uml.edu/~haim/ColorCenter
Absolute and Relative

• Roots in scientific visualisation
• Absolute and relative questions
  – IR/DB for absolute, InfoVis for relative?
• Searching and browsing
• Visualisations as workspaces
  – as places people spend time working
Roots in Scientific Visualisation

- n-Vision: Feiner and Beshers
- Financial Options pricing
- 6D displayed: ‘worlds within worlds’
  - 3D graph at cursor, within 3D graph
- Comparing points?
- Making relative assessments?
- Getting an overview?
Worlds Within Worlds
Relative and Absolute

• "You don’t have to be good to be the best."

• Absolute properties
  – maximum, average, exact query match
  – need can often can be formalised
  – suited for a command language
Relative and Absolute

• Relative properties
  – overview, trends, patterns
  – distributions and outliers, ‘sense’
  – often difficult to formalise need
  – suited for direct manipulation [p.71]
The Importance of Relative Properties

- Items of data do not supply the information necessary for decision making. What must be seen are the relationships which emerge from consideration of the entire set of data. In decision making the useful information is drawn from the overall relationships of the entire set. (*Jacques Bertin*)
DB and IR

• I’m lumping the two together here
• Obviously a powerful approach, but:
  – what if a query returns too many results?
  – what if it is hard to express what you need?
  – how to use it to make sense of a db?
• Too often, IR and DB systems:
  – focus on the query in isolation
  – lump together everything else: ‘browsing’
Searching v. Browsing

• Search: query sent into the black box
  – automatic but ‘distant’
  – work done for you, but you can’t guide it
  – query has to represent your need
  – involves the database design or information representation
  – as well as a formal query language
Searching v. Browsing

• Browsing: the box is opened up
  – manual but ‘close’
  – *you* do the work (or whatever you’d call it)
  – you don’t have to formally express what you need,
  – but you still rely on the data’s representation
Query Languages

• Example: Marchionini’s 1989 study
  – encyclopaedia in print v. electronic form
  – only half the subjects used AND
  – none used OR or NOT
  – ‘electronic’ group took twice as long

• Formal query languages are hard
  – SQL requires 2 to 20 hours of training,
  – even then users make frequent errors
Complementarity with IR/DB

• We need searching *and* browsing, IR *and* InfoVis

• Visualisation inside retrieval
  – vis of search results e.g. Hearst’s TileBars

• Retrieval inside visualisation
  – highlight all map elements matching a query [Plate B4]
The electronic cadaver: computerized anatomy lessons and digital dissection.

Army tests prototype battlefield information system.

Better ADP could cut VA delays 40%, officials say.

VA automation means faster admissions. (US Department of Veterans Affairs)

"Hospital bridges breathe of data: includes related articles on growth of a me.

LabVIEW 2.0 still tops the horizon. (National Instruments LabVIEW 2.0)

"Computer graphics in medicine: from pictures to analysis/extension. (Industry Progr.

DOD health system costs break $1.1 billion barrier. (Composite Health Care

Paragon ports its 2-D imaging software to DEC's A/V. (Paragon Imaging in
Visualisation as Workspace

• Place/setting for fine-grained activities
  – seeing how search results fit
    • Bell Labs’ SeeSoft [Plate B1]
    • Bead ‘worlds in miniature’
  – spreadsheet-like sorts, classifications...
    • Xerox/InXight’s TableLens
  – seeing where others have been or are now
    • e.g. VRs, ‘wear’ [15.2, 15.3]
WIMs: Small Multiples

Miniatures of main image, showing most recent searches

Main image
new Graphical reasoning models for understanding

Graphical tools lead approach to user interaction description

Graphical techniques in a spreadsheet

Virtual
Edit Wear and Read Wear

• Metaphor, involving familiar idea
  – wear on well-thumbed books
  – shows accumulated history
• Can extend well-known repns
  – or be part of newer ones
• No big conceptual jump here
  – but very effective technique
‘Wear marks’ show pattern of where file has been most used.

Rectangles can show where each person is currently reading.

Can represent more than one feature by using multiple bands, such as reading, edits, Y2K ‘sensitivity’ etc.
Looking Back

• A first (naïve?) characterisation
  – absolute ↔ DB/IR ↔ searching
  – relative ↔ InfoVis ↔ browsing

• Perceptual issues vitally important
  – 2D as superior to 3D
  – over time, link fine-grained activities
  – show history to help guide future activity
After Tea

• Shneiderman’s taxonomy of data types
  – dimensional reduction
  – ‘heterogeneous’ data

• Interaction techniques
  – linking, distorting, lenses...
Shneiderman’s Taxonomy of Data Types

• 1D and temporal data
• 2D: maps, scatterplots
• 3D e.g. the world
• hierarchical e.g. trees
• graph data e.g. web pages and links
• multidimensional data: everything else?
1D Data

• Could just use a scrolling list
• Different scale views e.g. scrollbars [15.3]
  – edit wear and read wear
• Dual, shared view
  – allows focus on detail as usual
  – adds extra context unobtrusively
Focus+Context

• A powerful interaction approach
  – magnify the centre or focus
  – compress the periphery or context

• Assumption: focus is more important
  – so distort or split to give it more pixels

• Best known approaches
  – scrollbar: edit wear and read wear
  – Perspective Wall, 2D: ‘fisheye’ lens [13.15]
Temporal Data

• Shneiderman distinguishes this from 1D
  – very common: records, logs...
• ‘Parallel’ features, sharing time axis
• Can use 1D techniques
• Can set 1D in 3D e.g. Perspective Wall
  – early InfoVis exemplar from Xerox PARC
  – jazzy use of 3D, but is it really useful?
Not the best picture, but anyway...
Perspective Wall

• Effectively a distortion technique
  – better image: [15.8]
  – central *focus* region has detail
  – gradually compressed *context*

• Appeal to natural perspective view
  – closer things take up more of field of view
  – further things take up progressively less
  – as distance $\to \infty$, resolution $\to$ zero
1D Distortion (Furnas, CHI97)

(a) apple
   banana
   carrot
dill
endive
fennel
ginger
herbs
indigo
jicama
kale
lemon
mango
nutmeg
orange
plum
quince

(b) apple
   banana
   carrot
dill
endive
fennel

(c) apple
   banana
   carrot
dill
endive
fennel

(d) ginger
   herbs
   indigo
   jicama
   kale
   lemon
   mango
   nutmeg
   orange
   plum
   quince

(e)
1D Distortion
2D Data

• 2D structure in 2D view
  – scatterplots and maps: Spotfire

• 2D structure in 3D view
  – landscapes, increasingly common in VRs
  – 3D features placed on ground plane
  – but: high values block low ones
  – can fail due to occlusion
3D Data

• Appeal to the ‘natural’ world
  – we often think of the world as a 3D shape
  – but we don’t treat it that way
  – e.g. how wide is a city? How high?
  – e.g. can only see surfaces of most objects

• Often invites occlusion problems
  – nearby objects block view of distant ones
  – e.g. can only see half of a sphere
Global Network Traffic
Hierarchical Data

• Trees:
  – Cone trees, cam trees, ... (3D)
  – Hyperbolic trees, Stretch... (2D)

• Still difficult to handle
  – some people use focus+context
  – show detail on central focus area
  – use distortion to retain rough context
Graph Data

• Important e.g. Web, but difficult
• Sometimes reduce to simpler type
  – e.g. convert to a tree: choose a root, lift up, cut off excess links
• Sometimes use optimisation algorithms
  – attempt to minimise edge crossings
  – connectivity $\rightarrow$ ideal proximity $\rightarrow$ error
• Or let the user(s) handle the problem
WebBook (Xerox PARC)
Multidimensional Data

• Dimensional reduction
  – transform to one of the other types
  – e.g. force-based layouts to make a 2D map

• Sometimes data inherently too complex
  – even apparently simple cases e.g.
  – 4 objects, all 1m from each other → 2D?

• More on this later...
Limits of the Taxonomy?

• Heterogenous data?
• He treats the web *only* as a graph
  – assumes use is driven only by local links
  – counterexample: search engines
• He focuses on the authored content…
  – …but not on how we read and use it
• He focuses on visual perception…
  – …but not on how use affects perception
Let the Users do the Work

• Xerox PARC WebBook
  – people place web objects in 3D
  – similar: Pad++ web viewer

• Problems
  – manual, scaling to large numbers

• Still, it’s good to see multiple pages
Heterogeneous Data

• Handling a mix of types e.g. on the Web
  – text, images, sound, programs

• Often very complex data
  – music, films, novels… web pages

• Content analysis doesn’t work (yet?)
  – how to define ‘similarity’?
  – attempts to work via abstract metadata fail
The Path Model

• Extension of collaborative filtering (CF)
  – Resnick/Varian CACM special issue

• *Path*: a user’s history of choices
  – order over time is represented

• Can use co-occurrence statistics
  – co-occurrence treated as similarity
  – use dimensional reduction to make maps
Similarity Metric

• A matrix, \( f(\text{URL}_i, \text{URL}_j) \rightarrow \text{real} \)
• Use statistics of co-occurrence
  – for each window of length \( k \) URLs
    • get each pair of URLs \( i \) and \( j \) in the window
    • add 1 to \( f(\text{URL}_i, \text{URL}_j) \) for each \( i,j \)
  – step window one along path by one
  – when end reached, normalise
CF and Paths

- Complement traditional IR/DB work
  - focus on individuals’ choices/activity
  - IR focuses on data independent of activity
  - CF and paths need IR to feed them
- Adaptive, subjective, minimise metadata
- Handle heterogeneous data
  - which IR can not do
Shneiderman’s Taxonomy

• OK for a first cut, but too simple
• Data is transformable between types
  – graphs to trees, multiD to 2D, tree to 1D
  – …and what about heterogeneous data?
• No ‘perfect’ way to handle data
  – subjective, depends on situation
  – tasks are interwoven and interdependent
  – …so use multiple views and transforms
Dimensional Reduction

• What is it?
• Similarity metrics
  – functions to say how alike two things are
• Two non-linear approaches
  – (Kohonen maps (or ‘SOMs’))
  – Force-based layouts
• Traditional approaches too constrained?
  – PCA, SVD...
What is it?

• n-dimensional data reduced to 2D
  – preserve relationships of n-D
  – metaphor: close ≈ similar, far ≈ dissimilar

• What does ‘close’ mean?
  – Euclidean distance on 2D map?

• What does ‘similar’ mean?
  – Euclidean distance in n-D space?
  – used in similar ways by similar people?
Similarity Metrics

- Textual documents: word frequencies
- Doc represented as vector of pairs \((\text{word}_i, f_i)\)
- Similarity: use scalar product
  - Sum over each word of \(f_i \cdot f_j\)
Similarity Metrics

• Spreadsheets: matrix of data $M_{ij}$
• Need metric for similarity of two rows
  – build from metrics for simpler data types
  – numerical types: reals, integers
  – ordinal types: name, currency
• Sum effect of each column in each row
  – usually do some normalisation first
Similarity Metrics

• Numerical columns: sum over columns
  \[ s = 0 \]
  \[ s += (M_{ik} - M_{jk}) \quad \text{for each column } k \]

• Ordinal columns: product over columns
  \[ p = 1 \]
  \[ \text{if } (M_{ik} = M_{jk}) \quad \text{then } p \quad \text{else } 0.75*p \]

• Similarity = \[(1 - s) * p\]
A Sweeping Generalisation

• For any data type, some similarity metric is feasible
  – a function $f(a,b) \rightarrow \text{real}$

• Problem is whether it makes any sense
  – normalisation and outliers
  – limits of applicability of mathematics

• So, make some initial rough cut
  – be aware that it is likely to be rough
Normalisation

- Spreadsheets: summing differences
  - what if one column has large numbers...
  - ...and another has small numbers?
  - Quick normalisation: scale by $max-min$
  - Or: scale by 1 or 2 standard deviations

![Graph showing number of objects vs. data value in column]
Applicability of Mathematics

• Word frequency
  – Twice as frequent ⇒ twice as important?
  – Zero means word not relevant to doc?

• Spreadsheets
  – why \((M_{ik} - M_{jk})\) and not \((M_{ik} - M_{jk})^2\)
  – why 0.75p and not 0.751p^{23}

• Usually people ‘just try stuff’
  – can’t model subjectivity and contextuality
Self-Organising Maps (SOMs)

- 2D grid of points $p_i$ with vectors $v_i$
- Make grid ‘fit’ dist$^n$ of input n-D data
  - not a map of input data, but a map of statistical patterns/distribution of input
- Similar grid points are close together
- Relies on metric:
  - similarity $(v_i, \text{input}) \rightarrow \text{real}$
- http://websom.hut.fi
Kohonen SOM Algorithm

• create grid: $p_i$
• give random n-D vector to each: $v_i$
• for each n-D training input $t$
  – find the node most similar to $t$: $v_{\text{winner}}$
  – adjust: $v_{\text{winner}} = av_{\text{winner}} + (1-a) t$
    • $a$ determines how quickly the map is adjusted
      i.e. the ‘learning rate’
  – for each neighbour node of $v_{\text{winner}}$
    • use higher $a$ (slower learning) and adjust it
After the Map has been Made

• Create a set for each grid point, to store a number of the inputs. Initially null.
• For each one of the training inputs
  – find most similar node, add input to set
• Then, a mouse click on a node →
  – pop up its related set of inputs
  – Or use in searches, fisheyes, zooms…
• Show set size as ‘density’ → colour
WebSOM example
Strengths and Weaknesses

• Cost
  – \( O(\text{trainingSetSize} \cdot \#\text{dimensions} \cdot \text{gridSize}) \)

• Distance \((p_i, p_j) \rightarrow ???\)

• Distance between two inputs \( \rightarrow ??? \)

• All dimensions combined
  – not for exploring individual dimensions
Force-Based Models

• Spring model, force-directed placement
• A simulated physical system
  – Positions, velocities and forces
  – Far, similar objects pull each other close
  – Close, dissimilar objects push apart
• Relies on metric:
  – similarity (obj, obj) → real
• Force (obj, obj) \( \propto \) similarity - closeness
Simplified Algorithm

• for each iteration
  • for each object i
    • for each other object j
      totalForce += force(i,j)
    • adjust velocity(i) using totalForce
    • adjust position(i) using velocity(i)
Energy over Time

- Sum of squared errors for each iteration
A Bead Map
Strengths and Weaknesses

• Cost
  – basic algorithm: $O(n^2)$ per iteration
  – approximate algorithm: $O(n)$ per iteration

• Position shows global relationships
  – neighbours in lowD but not in highD?

• All dimensions combined in position
  – not for exploring individual dimensions

• Limited scale: a thousand or two?
Dimensional Reduction

• Widely applicable technique, but
  – can be computationally expensive
  – question of how good the metric is
  – putting parameters in users’ hands

• Force-based models
  – show more of data set structure

• Kohonen maps (SOMs)
  – scale better to large numbers
Interaction Techniques

• Varied views, abstractions and interpretations

• User control and adaptive techniques
  – filtering
  – categorisation and clustering
  – zooming and semantic zooming
  – getting hidden detail
  – multiple views [Chapter 13]
Brushing and Linking

Linking together multiple views, so that ‘brushing’ a selection in one view colours matching objects in other views.
Filtering

• Traditional query languages
  – often too static, difficult to understand
  – give absolute more than relative info
• ‘Dynamic queries’ and sliders
  – Ahlberg’s *Spotfire*: (www.ivee.com) [B4,B5]
  – Lisa Tweedie’s histograms & related tools
• Magic Lenses (Bier et al.)
In (a), six wedge-shaped objects are positioned, and in (b) a palette of six ‘click-through’ fill colours is used to make one segment green. In (c), another wedge has its outline coloured blue by using an outline colour lens.

A sheet of magic lens widgets: fill colours, shape drawing, clipboard, snap grid, delete button, and a button to navigate to further widgets.
Magic Lenses

An outline colour palette, over a magnifying lens, over a ‘snowflake’ shape. This is an example of stacking lenses so as to compose their functions. Here, the magnifier makes it easy to work on individual outlines within a complex shape.

In (a), a ‘symmetry clipboard’ lens is placed over a yellow ‘tear’ shape. The lens replicates the shape in a symmetrical pattern (b), and this pattern can be moved and pasted elsewhere (c).
Zooming — Pad++

• ‘Infinite’ detail potentially available
• Focus in on detail but retain overview
  – can see detail, and remember context
• Semantic Zoom
  – things don’t just get bigger, but adapt
• Recently added in a web browser
• Free from www.cs.umd.edu/hcil/pad++
Categorisation

• Another kind of view or abstraction
  – e.g. subsets that share a property
  – or subsets that are relatively alike
• Can be done statically *a priori*
  – traditional cluster analysis, etc.
• Can then map to colour, shape, etc.
  – or make it an extra dimension
Clusters in Bead
Cluster Labels and Popups

• Cluster labelling
  – Find words that are common in a cluster?
  – Find words that are particularly local
    • higher local density than density over all set

• Dynamic pop-up detail
  – choose a few objects to be shown in detail
  – weight by importance e.g. search hits
  – keep sampling, biased by these weights
Cluster Labels and Popups
Excentric Labelling

Labels pop up in the neighbourhood of the cursor, and are spread to avoid overlapping. When objects are too numerous, the total number is shown along with a subset of the labels.

Fekete and Plaisant, Proc. CHI 99, pp. 512-519
http://www.cs.umd.edu/hcil/excentric
Subjective Categorisation

• Categories are usually objective
  – e.g. clusters made independent of use
• Labels in Bead also used history
  – words weighted by frequency of use
• Usage discs
  – ‘edit wear and read wear’ idea again
  – radius proportional to use
  – easy to do this in 2D also
Subjective Categorisation

• Try to find what people *really* do
  – variation between people, jobs, etc.

• Maybe can’t do this beforehand
  – so use ongoing activity and adapt to fit
  – hot topic nowadays: *adaptive systems*

• Privacy issues: you are being watched
  – telephones, credit cards, ATMs...
Multiple Views

• Linking and brushing just one example
• Getting different views of the same data
  – no one view is going to suit every use
  – can combine filters, 2D views, 3D views, categorisations, distortions...
  – what happens in one view affects the others
Masui’s file hierarchy viewer
Software Architecture Tip

• Build one central model e.g. MVC
  – same model controls many views
  – any change to model? All views notified
  – use same protocol for all views

• Interaction via each view
  – translate view-specific interaction into model’s protocol
  – easy to add or remove views
Interaction as Interpretation

• Emphasise a useful subset
  – zooming, filtering
• Extract useful abstraction or metadata
  – clustering and categorisation
• But who determines what is useful?
  – user control, adaptation with use
• Offer multiple views/interpretations
Review

• Perceptual issues vitally important
  – 2D better for abstract data
  – avoid naïve use of colour
  – beware of mathematical ‘convenience’
  – over time, link fine-grained activities
• Link related interpretations
  – over space: multiple views
  – over time: wear, usage, paths
Review

• MultiD data and dimensional reduction
  – algorithms: computation v. richness
  – display: show reduced version in 2D
    • but offer access via each original dimension too

• Interaction
  – filtering, zooming, categorisation
  – dynamism, adaptation, linked views
Future Trends

• Integration into everyday programs
  – small InfoVis widgets inside other tools

• Multiple views and tools

• Multiple data types interwoven
  – different choice as to what ‘similar’ means

• Adaptation and dynamism
  – with individual’s use, with individuals’ use
Further Reading

- *How Maps Work*, Alan MacEachren
  - ideas are general to InfoVis & cartography
- *Readings in Information Visualization*, Card, MacKinlay & Shneiderman
- *Envisioning Information*, Edward Tufte
  - ‘information design’ for graphs, charts, etc.
- Proc. IEEE InfoVis, ACM CHI and UIST
- http://www.otal.umd.edu/Olive/
Contact Information

http://www.dcs.gla.ac.uk/~matthew

matthew@dcs.gla.ac.uk

...or see you at the banquet.