

# Glyphs

Presented by Bertrand Low

# Presentation Overview

- A Taxonomy of Glyph Placement Strategies for Multidimensional Data Visualization Matthew O. Ward, Information Visualization Journal, Palmgrave, Volume 1, Number 3-4, December 2002, pp 194-210.
- Managing software with new visual representations, Mei C. Chuah, Stephen G. Eick, Proc. InfoVis 1997
- Interactive Data Exploration with Customized Glyphs, Martin Kraus, Thomas Ertl, Proc. of WSCG '01, P20-P23.

# What is a Glyph!?

- Problem: Analyzing large, complex, multivariate data sets
- Solution: Draw a picture!
- Visualization provides a qualitative tool to facilitate analysis, identification of patterns, clusters, and outliers.

# What is a Glyph!? Cont.

- Problem: What to draw?
- Want interactivity for exploration (“Overview first, zoom and filter, then details on demand”, Shneiderman)
- Solution: Glyphs (aka icons) to convey information visually.
- Glyphs are graphical entities which convey one or more data values via attributes such as shape, size, color, and position

# Goal of Paper

- Problem: Where do you put the glyph?
- Recall: Spatial Position best for all data types (be it quantitative, ordinal, or nominal). Effective in communicating data attributes. Good for detection of similarities, differences, clustering, outliers, or relations.
- Comprehensive taxonomy of glyph placement strategies to support the design of effective visualizations

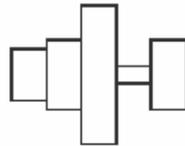
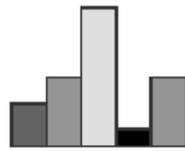
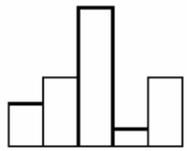
# Glyph Fundamentals

- *Multivariate data*:  $m$  number of points, each point defined by an  $n$ -vector of values
- Observation: nominal or ordinal, (may have a distance metric, ordering relation, or absolute zero)
- Each variable/dimension may be independent or dependent.

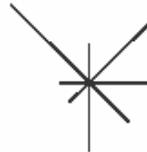
# Glyph Fundamentals Cont.

- A *glyph* consists of a graphical entity with  $p$  components, each of which may have  $r$  geometric attributes and  $s$  appearance attributes.
- **geometric attributes:** shape, size, orientation, position, direction/magnitude of motion
- **appearance attributes:** color, texture, and transparency

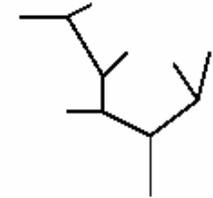
# Examples



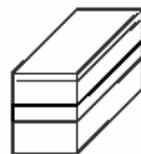
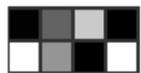
Variations on Profile glyphs



Stars and Anderson/metroglyphs



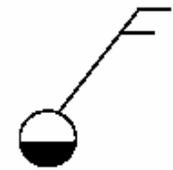
Sticks and Trees



Autoglyph and box glyph



Face glyphs



Arrows and Weathervanes

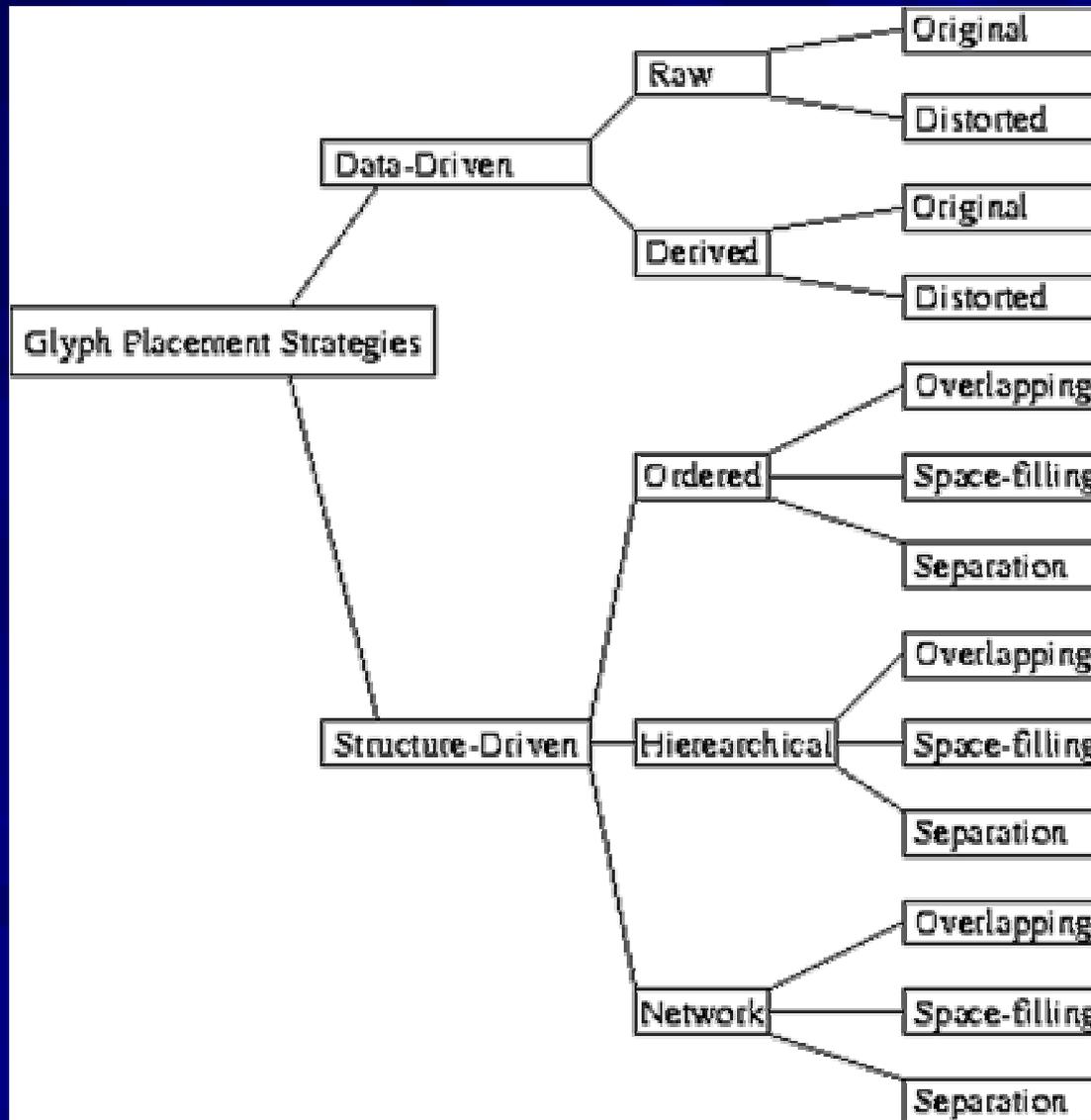
# Glyph Limitations

- 1) Mappings introduce biases in the process of interpreting relationships between dimensions.
- 2) Some relations are easier to perceive (e.g., data dimensions mapped to adjacent components) than others.
- 3) Accuracy with which humans perceive different graphical attributes varies tremendously.
- 4) Accuracy varies between individuals and for a single observer in different contexts.
- 5) Color perception is extremely sensitive to context.
- 6) Screen space and resolution is limited; too many glyphs = overlaps or very small glyphs;
- 7) Too many data dimensions can make it hard to discriminate individual dimensions.

# Glyph Placement Issues

- 1) *data-driven* (e.g., based on two data dimensions) vs. *structure-driven* (e.g., based on an order (explicit or implicit) or other relationship between data points)
- 2) Overlaps vs. non-overlaps
- 3) optimized screen utilization (e.g., space-filling algorithms) vs. use of white space to reinforce distances
- 4) Distortion vs. precision

# Glyph Placement Strategies

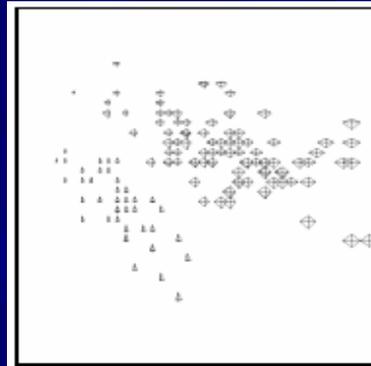


# Data-Driven Glyph Placement

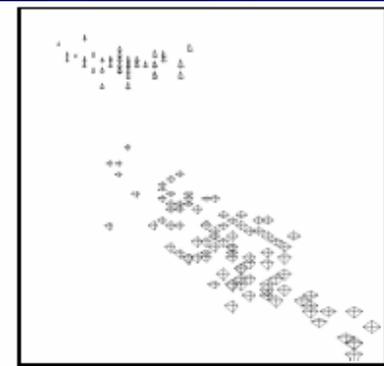
- Data used to compute or specify the location parameters for the glyph
- Two categories: *raw* and *derived*

# Raw DDGP

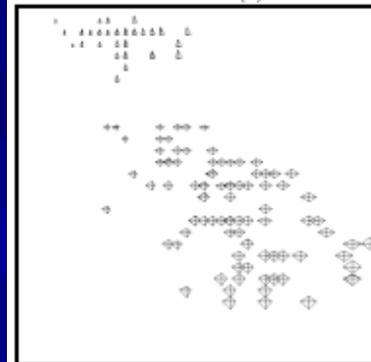
- One, two or three of the data dimensions are used as positional components



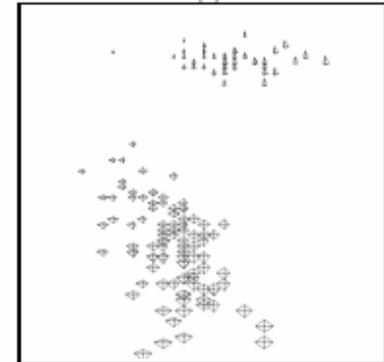
(a)



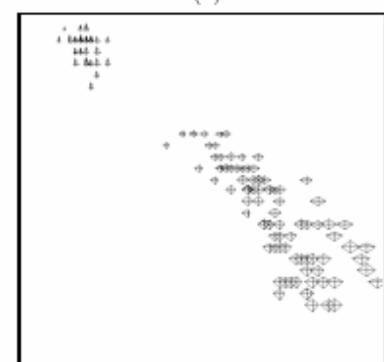
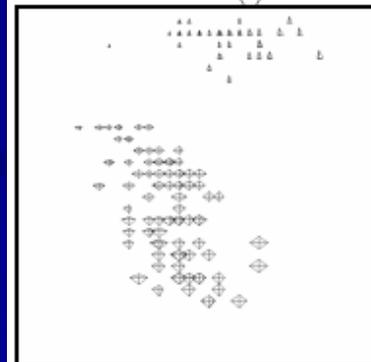
(b)



(c)



(d)



# Raw DDGP Cont.

- + Conveys detailed relationships between dimensions selected
- Ineffective mapping => substantial cluttering and poor screen utilization.
- Some mappings may be more meaningful than others (But, which one?).
- Bias given to dimensions involved in mapping. Thus, conveys only pairwise (or three-way, for 3-D) relations between the selected dimensions.
- Most useful when two or more of the data dimensions are spatial in nature.

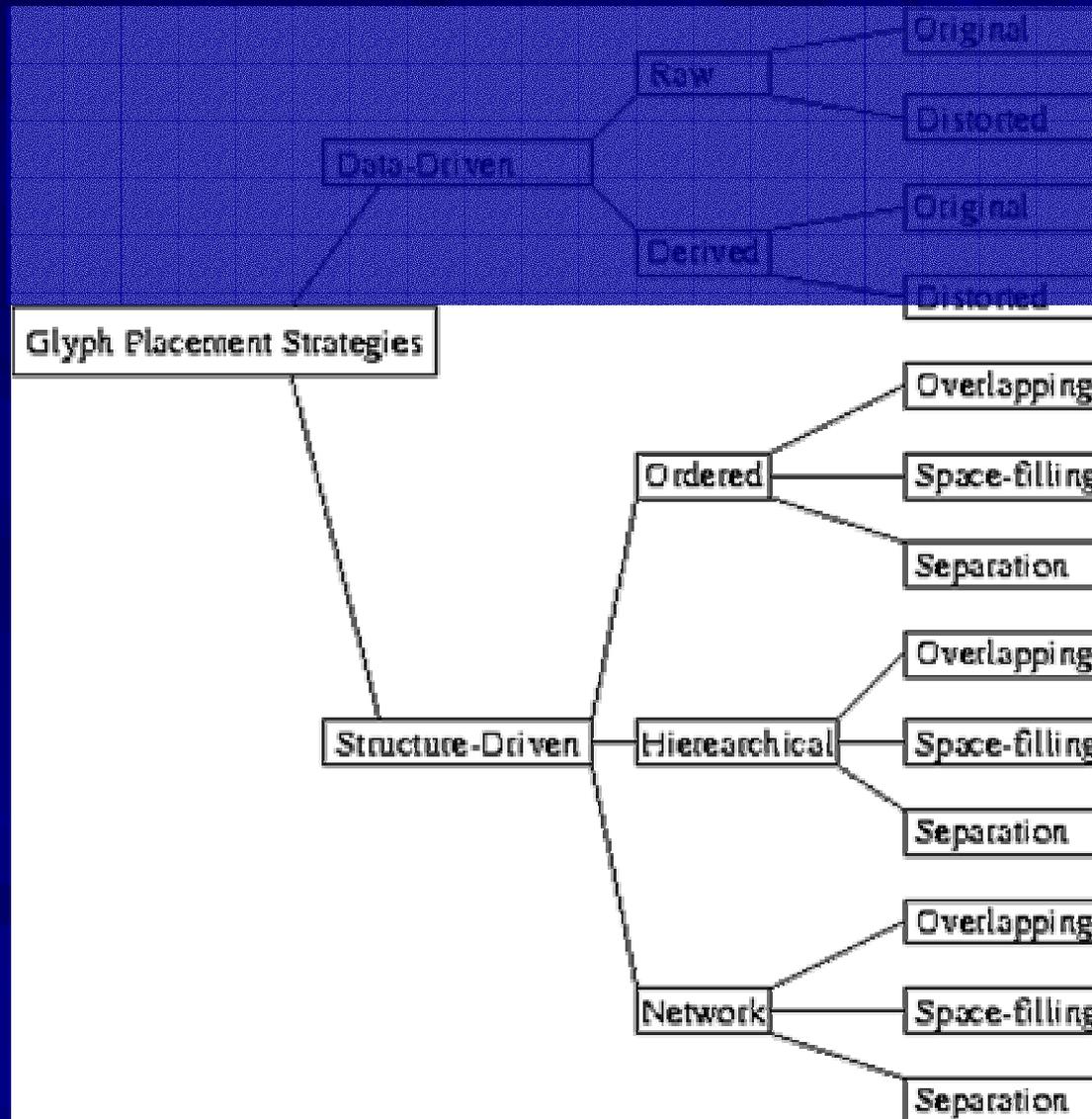
# Derived DDGP

- Dimension Reduction
  - Techniques include Principal Component Analysis (PCA), Multidimensional Scaling (MDS), and Self-Organizing Maps (SOMs).
- Resulting display coordinates have no semantic meaning

# Data-Driven Placement Cont.

- Issues: reduce clutter and overlap
  
- Solution: Distortion
  - 1) Random Jitter
  - 2) Shift positions to minimize or avoid overlaps.
- But, how much distortion allowed?
- Selectively vary the level of detail shown in the visualization

# Glyph Placement Strategies



# Structure-Driven Glyph Placement

- Structure implies relationships or connectivity
- Explicit structure (one or more data dimensions drive structure)      **v.s.**
- Implicit structure (structure derived from analyzing data)
- Common structures: ordered, hierarchical, network/graph

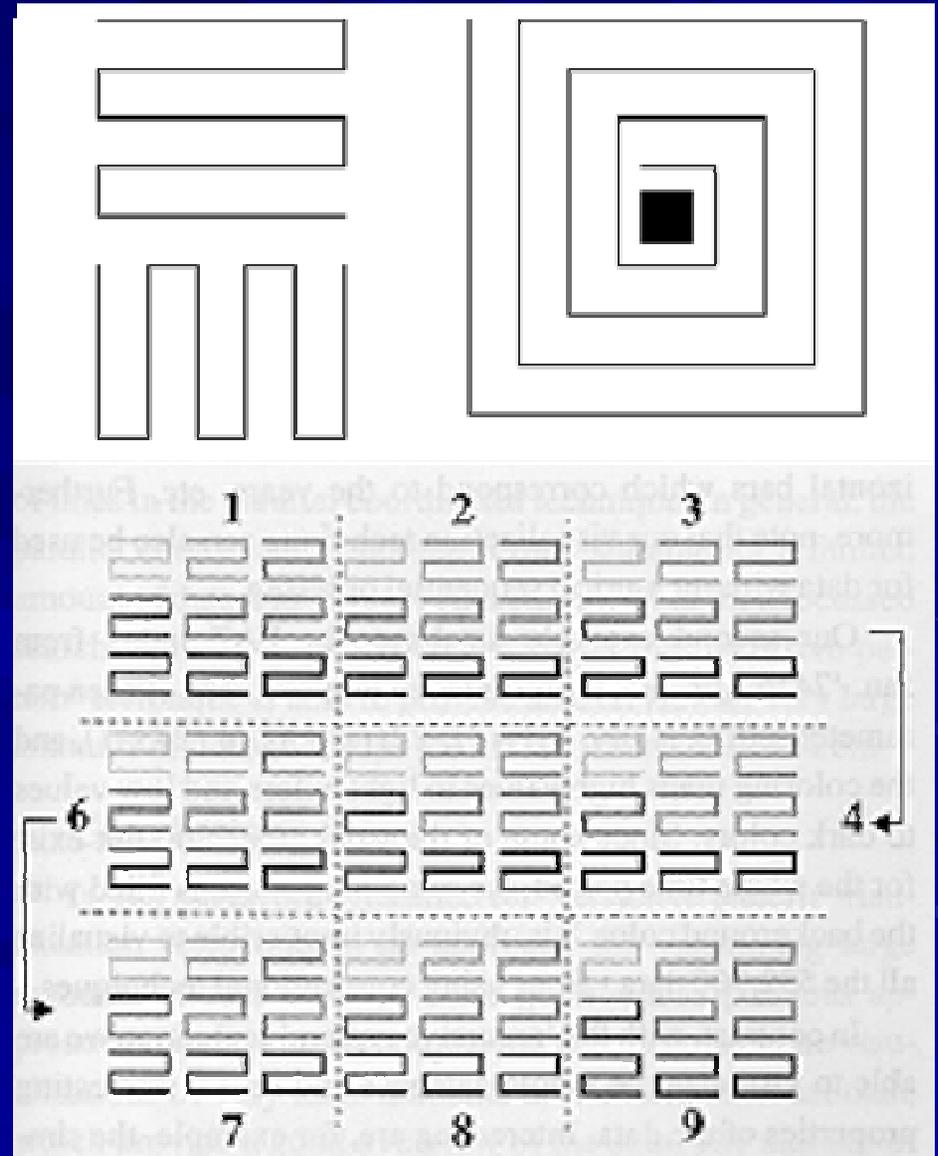
# SDGP – Ordered Structure

- May be linear (1-D) or grid-based ( $N$ -D)

- + Good for detection of changes in the dimensions used in the sorting

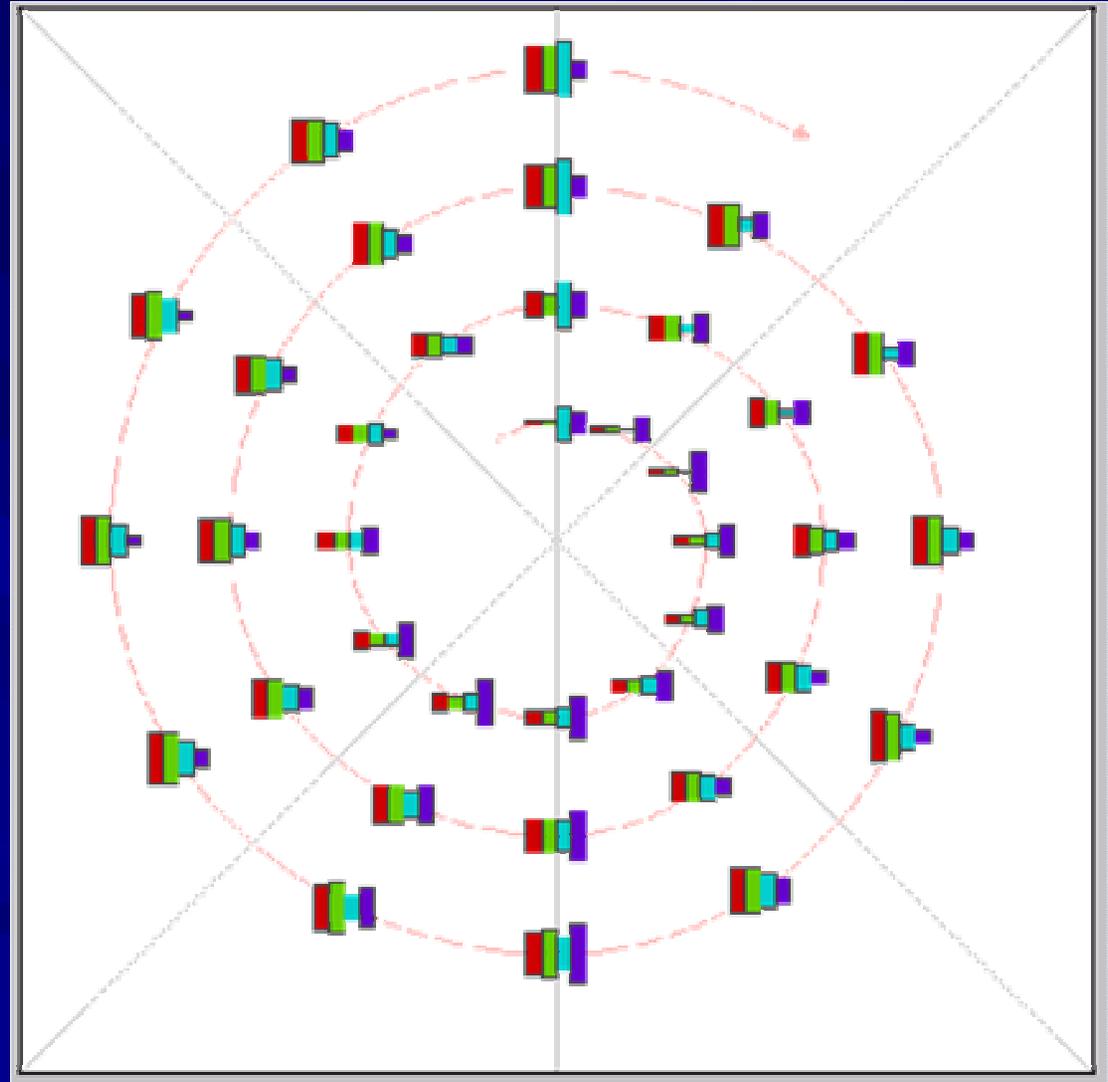
# SDGP – Ordered Structure Cont.

- Common linear ordering include raster scan, circular, and recursive space-filling patterns



# SDGP – Ordered Structure Cont.

- Dimensions (from left to right): Dow Jones average, Standard and Poors 500 index, retail sales, and unemployment.
- Data for December radiate straight up (the 12 o'clock orientation). Low unemployment, High Sales.



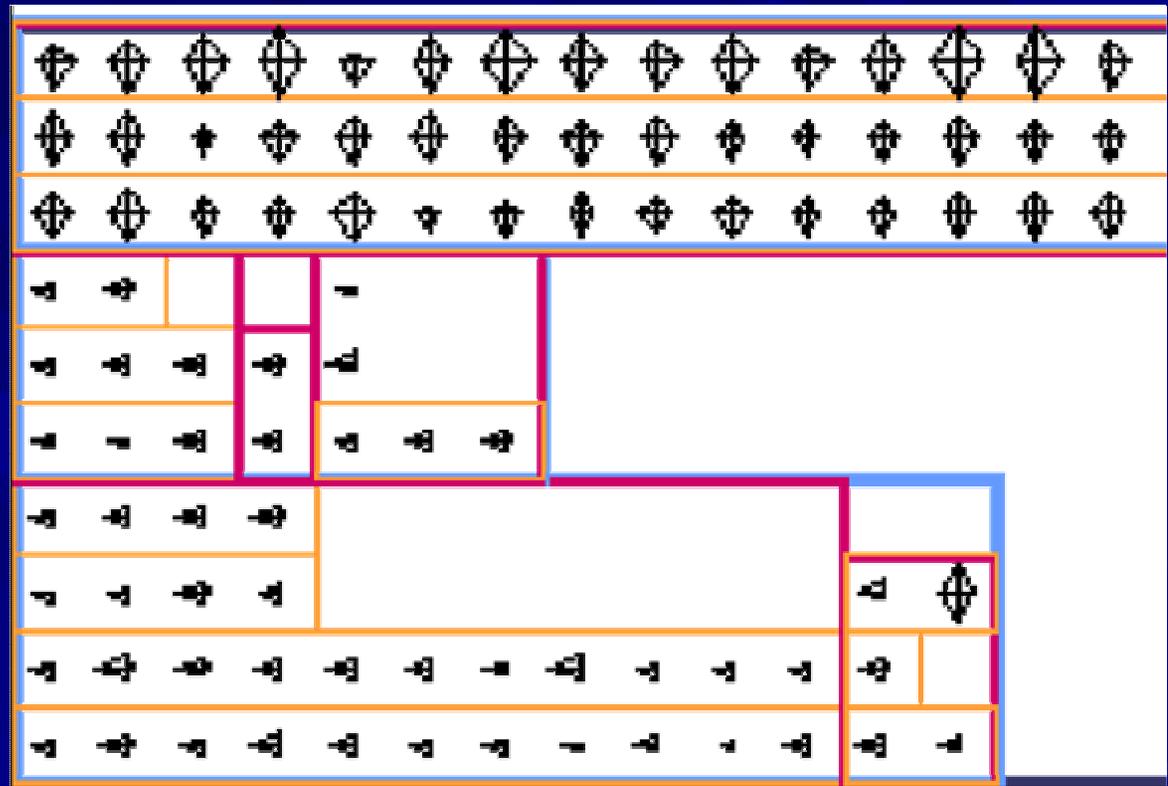
# SDGP – Hierarchical Structure

Common structures: ordered, hierarchical, network/graph

- Either **Explicit** (use partitions of a single dimension to define level in the hierarchy) or
- **Implicit** (use clustering algorithms to define a level in the hierarchy)
  
- Examples: file systems, organizational charts
  
- **GOAL: position glyphs in manner which best conveys hierarchical structure**

# SDGP – Hierarchical Structure Cont.

e.g. Tree-Maps



# SDGP – Hierarchical Structure Cont.

- Node-link graphs also fall into this category – Parent / Child nodes, graphical representation of links not required
- Connectivity implied via positioning

# SDGP – Network/Graph Structure

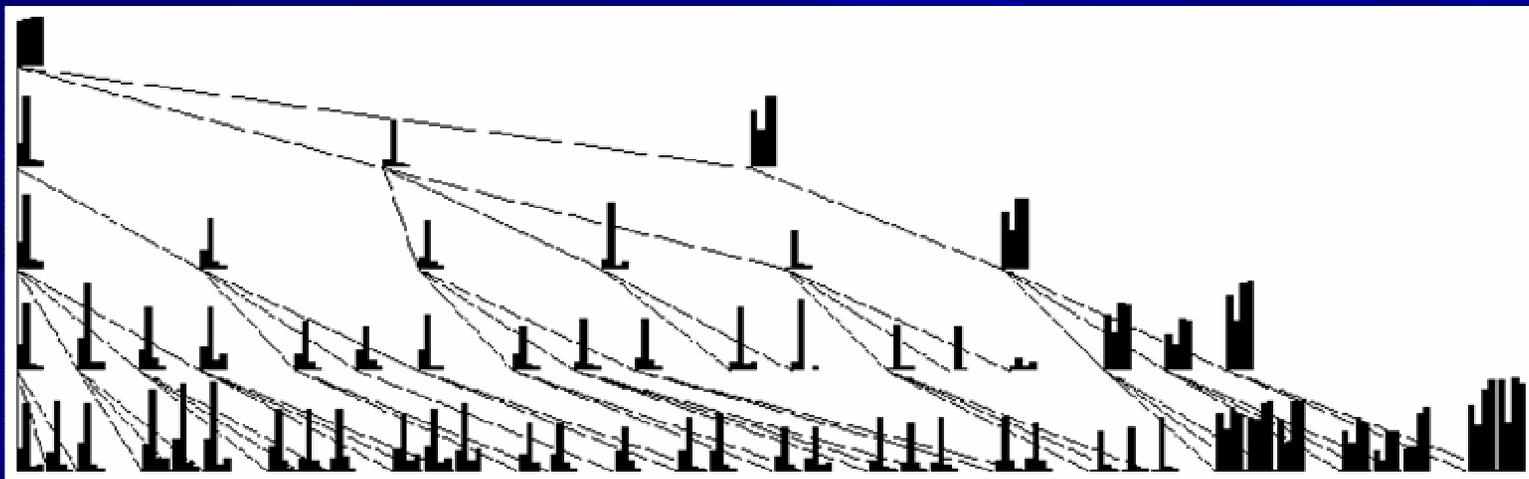
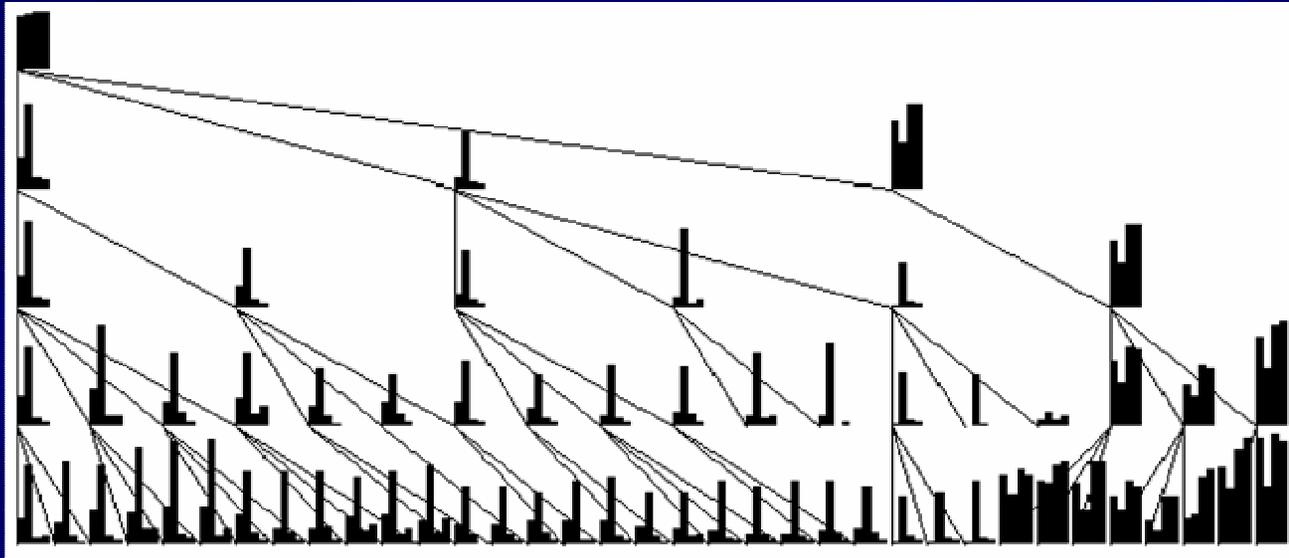
Common structures: ordered, hierarchical, network/graph

- Generalization of Hierarchical Structure (which was simply set of nodes and relations)
- Harder to imply relation with just positioning - need explicit links
  
- Many factors to consider:
  - 1) minimizing crossings
  - 2) uniform node distribution
  - 3) drawing conventions for links (i.e. straight line or 90° bend)
  - 4) centering, clustering subgraphs
  - 5) \* Greatest concern: Scalability (as with Hierarchical Structure)  
esp. since Links may convey info other than connectivity (e.g. traffic volume)

# Distortion Techniques for Structure-Driven Layouts:

- 1) Emphasize subsets while maintaining context (e.g., lens techniques)
  - 2) Shape distortion to convey area or other scalar value
  - 3) Random jitter, shifting to reduce overlap
  - 4) Add space to emphasize differences
- Trade off between screen utilization, clarity, and amount of information conveyed
  - Some overlap acceptable for some applications

# Distortion Techniques for Structure-Driven Layouts Cont.:



# Critique of Paper 1

- + Offers list of factors to consider when selecting a placement algorithm
- + Offers suggestions for future work
- + Motivates author's stated future work
- Figures not labelled, and all located at the end
- Overview paper – details missing, and assumes familiarity with terms

Data Characteristics	User Task	Recommended Strategy
Small to moderate size	univariate analysis	sorted structure-driven
Small to moderate size	bivariate analysis	raw data-driven
Small to large size	outlier detection	raw or derived data-driven
Moderate to large size	cluster analysis	derived data-driven, e.g., MDS
Small to moderate size	cluster analysis	hierarchical structure-driven after imposing a hierarchy
Small to moderate size, relational	link analysis	network structure-driven

# Presentation Overview

- A Taxonomy of Glyph Placement Strategies for Multidimensional Data Visualization Matthew O. Ward, Information Visualization Journal, Palmgrave, Volume 1, Number 3-4, December 2002, pp 194-210.
- Managing software with new visual representations, Mei C. Chuah, Stephen G. Eick, Proc. InfoVis 1997
- Interactive Data Exploration with Customized Glyphs, Martin Kraus, Thomas Ertl, Proc. of WSCG '01, P20-P23.

# Project Management Issues:

- 1) Time (meeting deadlines) – track milestones, monitor resource usage patterns, anticipate delays
- 2) Large Data Volumes – multi-million line software
- 3) Diversity/Variety – different types of resources, attributes
- 4) Correspondence to “real world” concepts – maintain “objectness” (properties of data element – e.g. user 123 - grouped together visually)

**Paper presents 3 novel glyphs**

# Viewing Time-Oriented Information

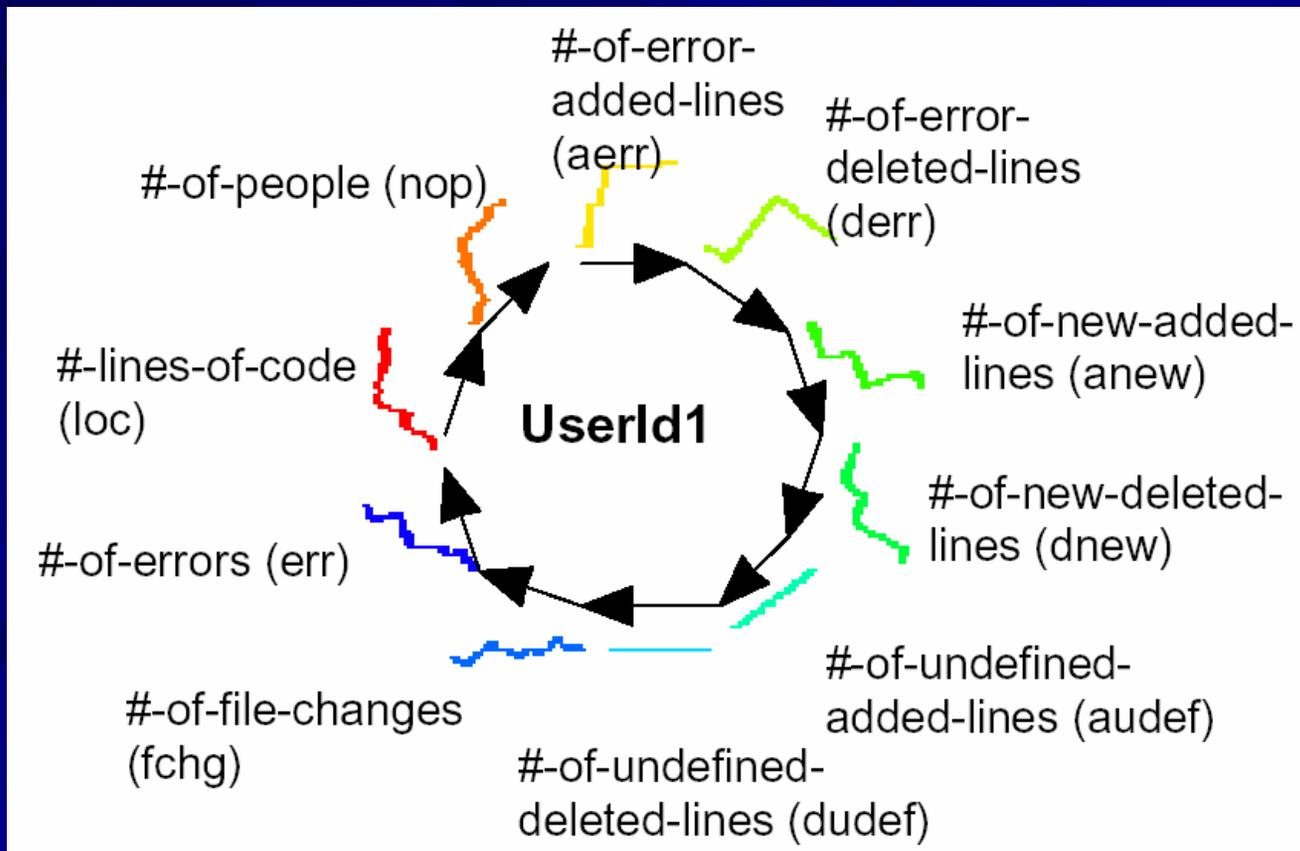
- Animation effective for identifying outliers
  - but less effective than traditional time-series plots for determining overall time patterns

## Glyphs

1. TimeWheel
2. 3D-Wheel

# 1. TimeWheel

- **GOAL:** Quickly (possibly preattentively) pick out objects based on time trends



# 1. TimeWheel Cont.

- Displays 2 major trends:

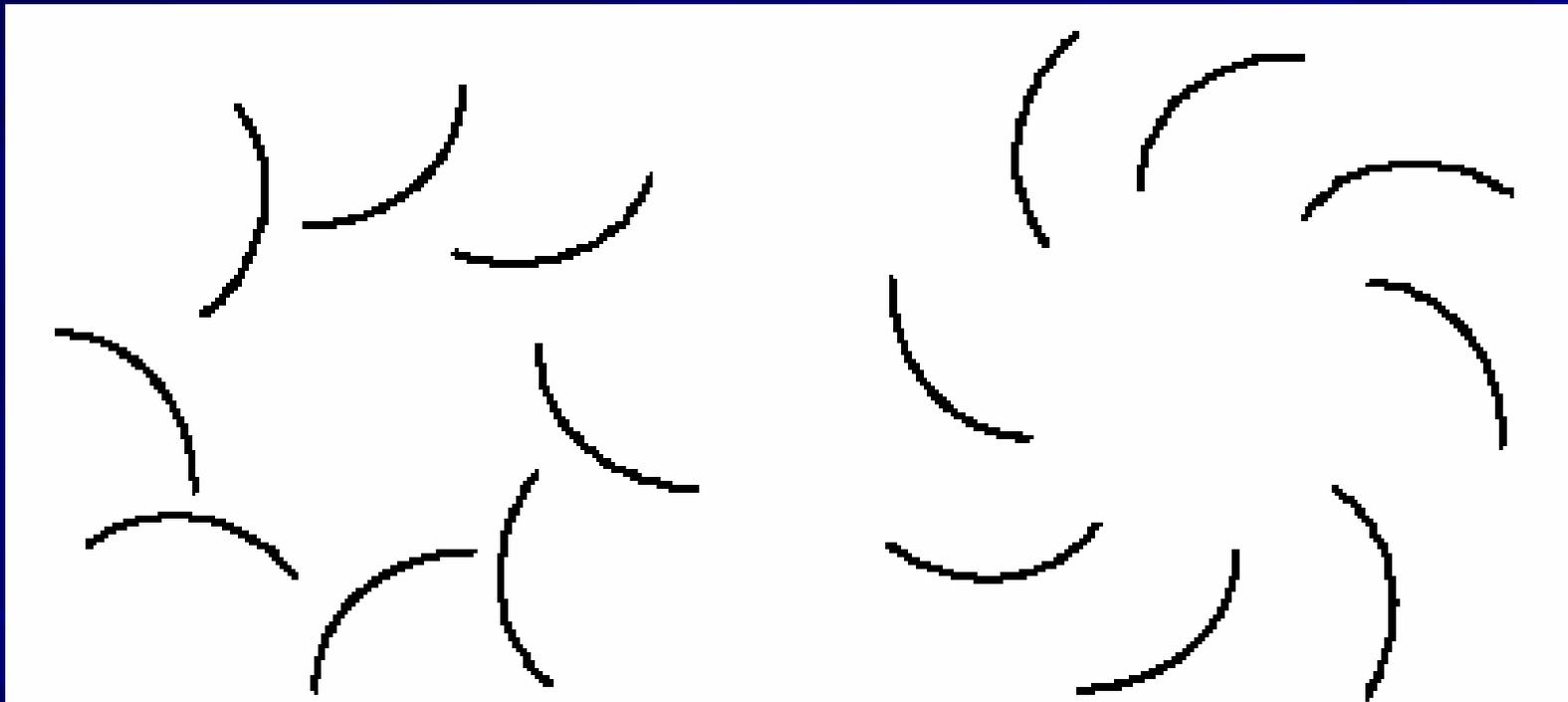


Figure 2: Left - increasing trend timeWheel (prickly fruit); Right- decreasing trend timeWheel (hairy fruit)

# 1. TimeWheel Cont.



# 1. TimeWheel Cont.

Linear

V.S.

Circular



Figure 4: Linear layout of *object-1* and *object-2*

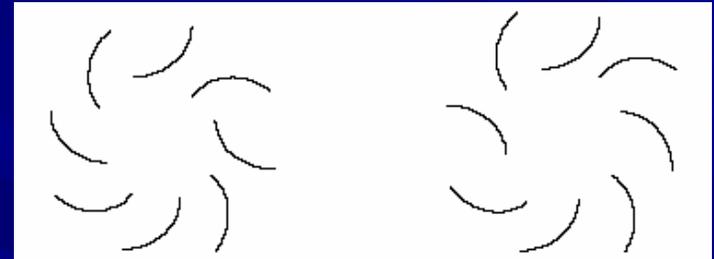


Figure 5: Circular layout of *object-1* and *object-2*

**+ Reduces number of Eye Movements per object**

- Limit to number of object attributes in timeWheel for it to fit within area of an eye fixation.

# 1. TimeWheel Cont.

Linear

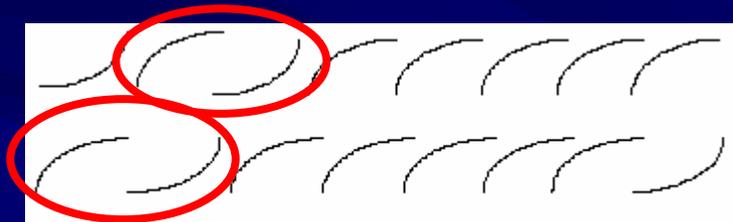


Figure 4: Linear layout of *object-1* and *object-2*

V.S.

Circular

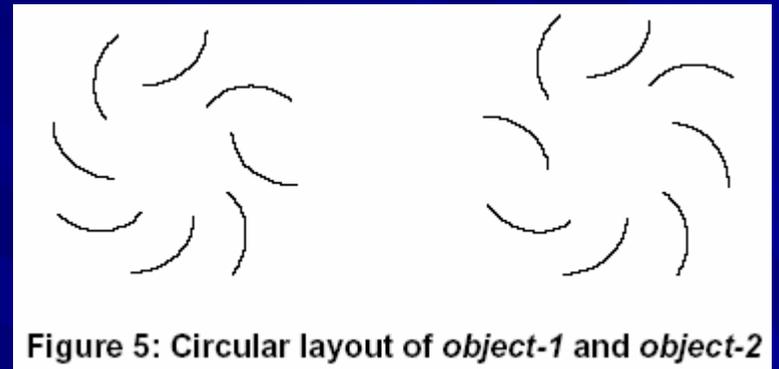


Figure 5: Circular layout of *object-1* and *object-2*



Figure 6: Cyclic local pattern

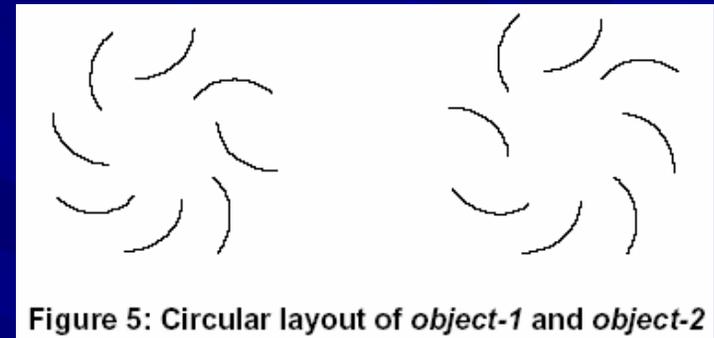
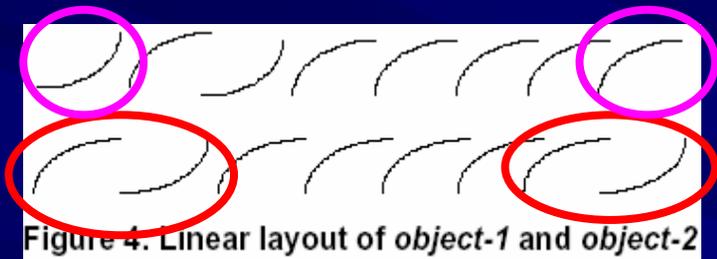
**+ Does not highlight local patterns (see above example on gestalt closure principle)**

# 1. TimeWheel Cont.

Linear

V.S.

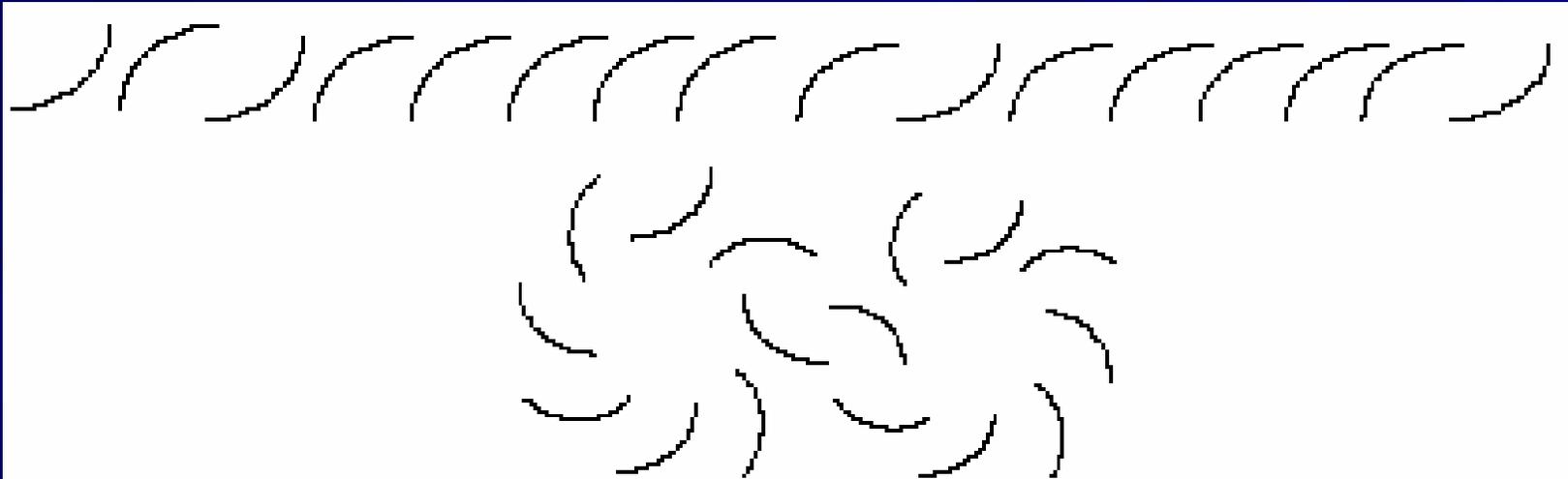
Circular



Encourages Left to Right reading (But attribute types unordered => false impressions!)

**+ Time series position has much weaker ordering implication**

# 1. TimeWheel Cont.

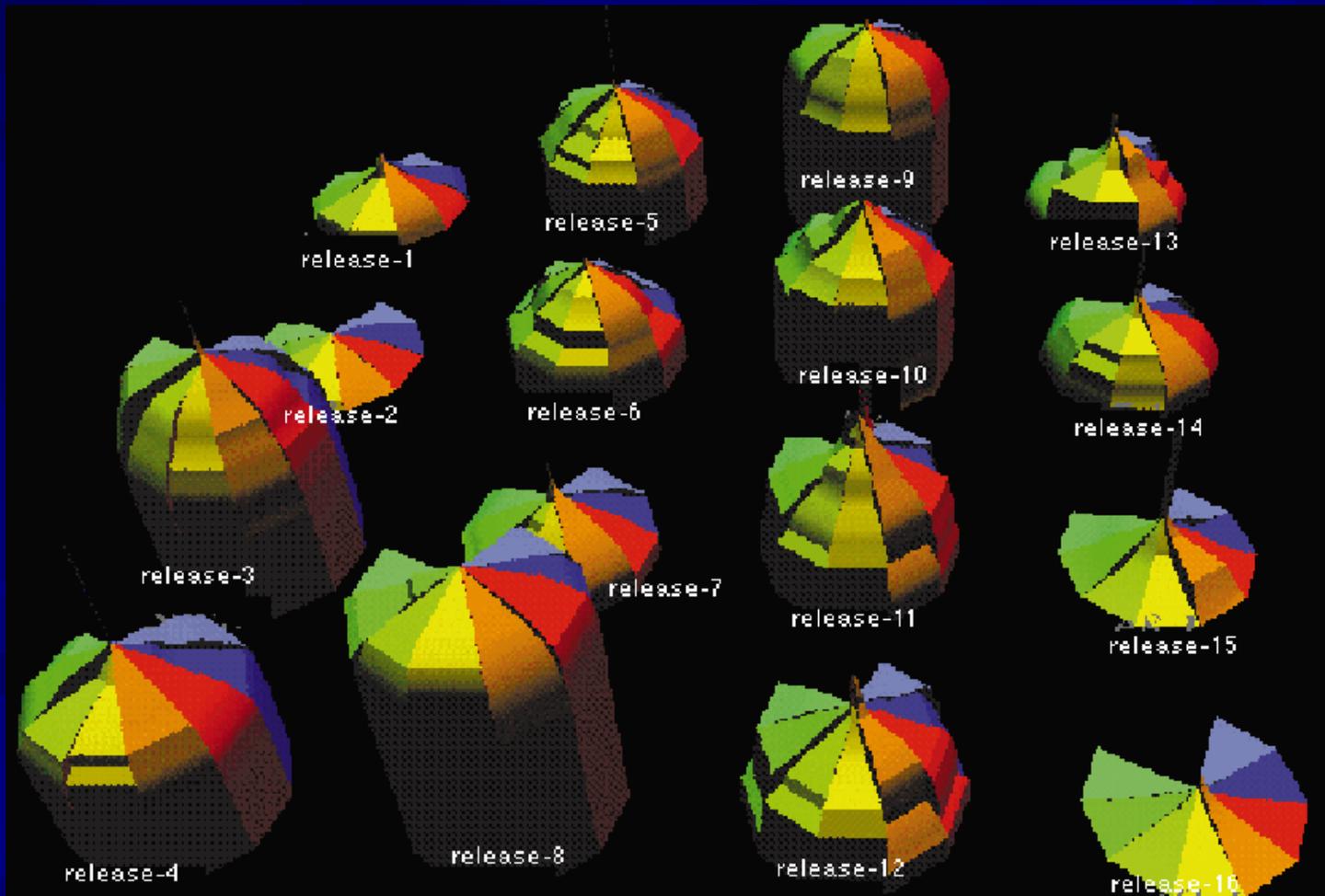


**Figure 7: The objects in Figure 4 and Figure 5 placed close to each other**

**+ Strong gestalt pattern – circular pattern is common shape => we see two separate objects**

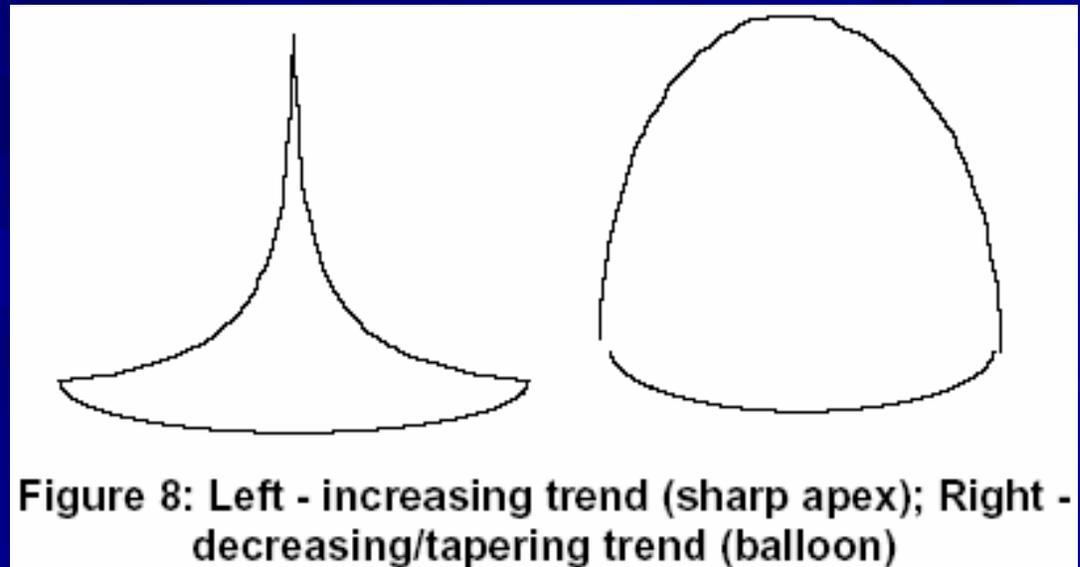
## 2. 3D-Wheel

- Encodes same data attributes as timeWheel but uses height dimension to encode time



## 2. 3D-Wheel Cont.

- Each variable = slice of base circle
- Radius of slice = size of variable



**+ Perceive dominant time trend through shape**

# Viewing Summaries

- **InfoBUG** represents 4 important classes of software data

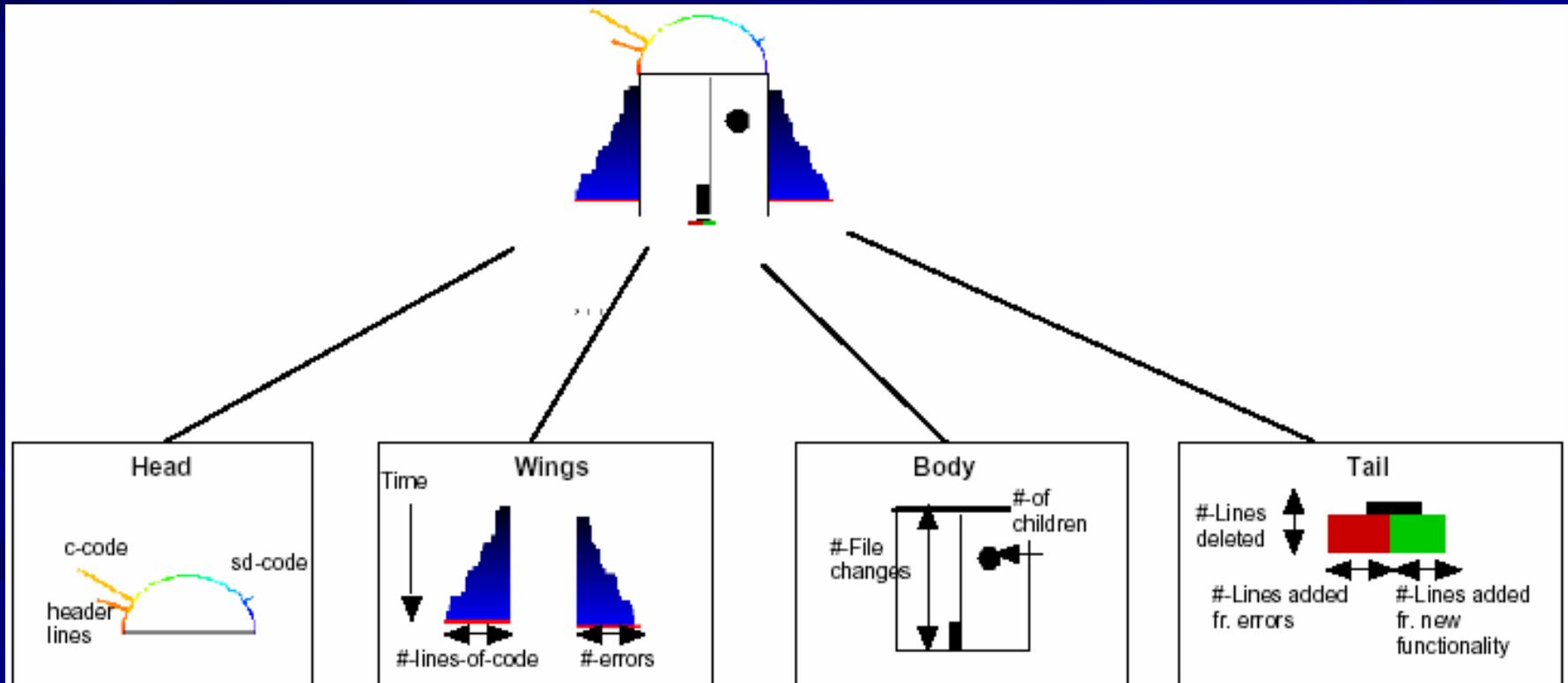
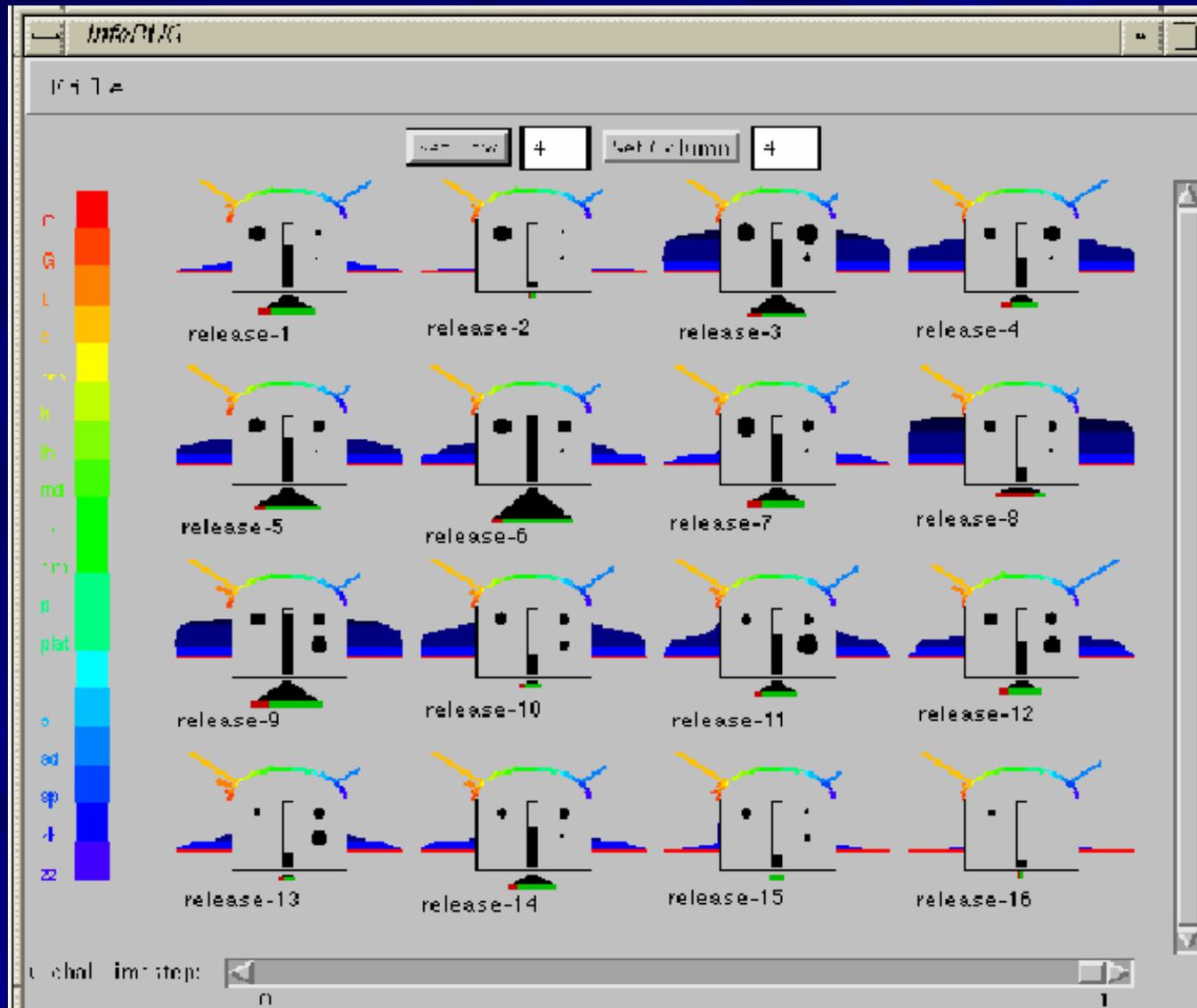


Figure 12: InfoBUG glyph

# 3. InfoBUG



# Critique of Paper 2

- + Concepts well explained, useful figures
- + Well motivated
- + Issues stated at outset, solutions carefully explain how issues solved (good example scenarios)
- + Convincing arguments to effectiveness of glyphs
- No user tests
- Glyph overlapping issues (3D-Wheel)
- Scalability (how many such glyphs on screen at a time?)
- Learning curve to familiarize with glyph?

# Presentation Overview

- A Taxonomy of Glyph Placement Strategies for Multidimensional Data Visualization Matthew O. Ward, Information Visualization Journal, Palmgrave, Volume 1, Number 3-4, December 2002, pp 194-210.
- Managing software with new visual representations, Mei C. Chuah, Stephen G. Eick, Proc. InfoVis 1997
- Interactive Data Exploration with Customized Glyphs, Martin Kraus, Thomas Ertl, Proc. of WSCG '01, P20-P23.

# Customized Glyphs for Data Exploration

- System for non-programmers to explore multivariate data
- Motivation: To visualize multivariate data with glyphs, the specification of the glyphs' geometric and appearance attributes (incl. the dependencies on the data) is required. However, for many data sets, the best mapping from input data to glyph attributes is unknown.
- Moreover, single best mapping may not exist
- Claim: Interactive switching between different geometric and appearance attributes is desirable.

# Goals

- Minimize interaction required to perform following tasks:
  - 1) Switching to another data set with different variables, different number of data points, and/or unrelated data ranges
  - 2) Mapping any variable to a previously defined glyph attribute
  - 3) Filtering data points via imposing constraints on certain variables

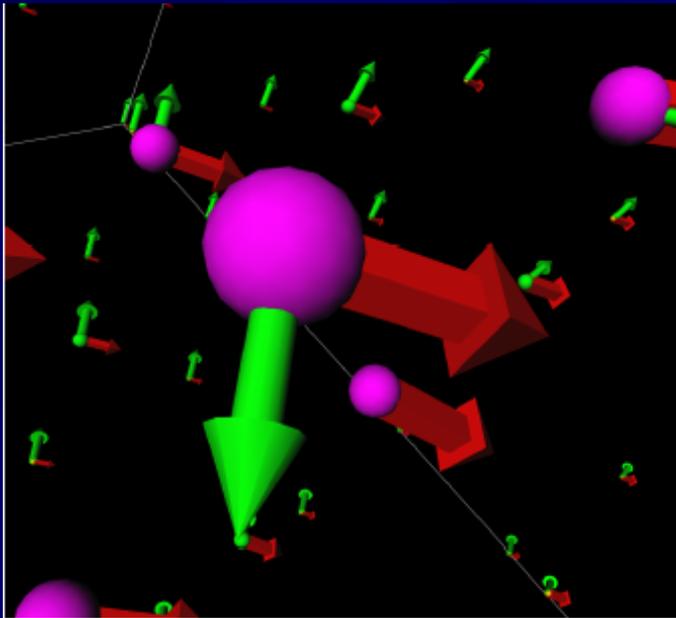
# System Overview

- Use GUI to allow user to define complex, composite glyphs (thus “programmerless”)
- Employs Data-Driven Placement
  - Allows user to quantitatively analyze up to 3 variables (3D graphics)
- Implemented as an IRIS Explorer module

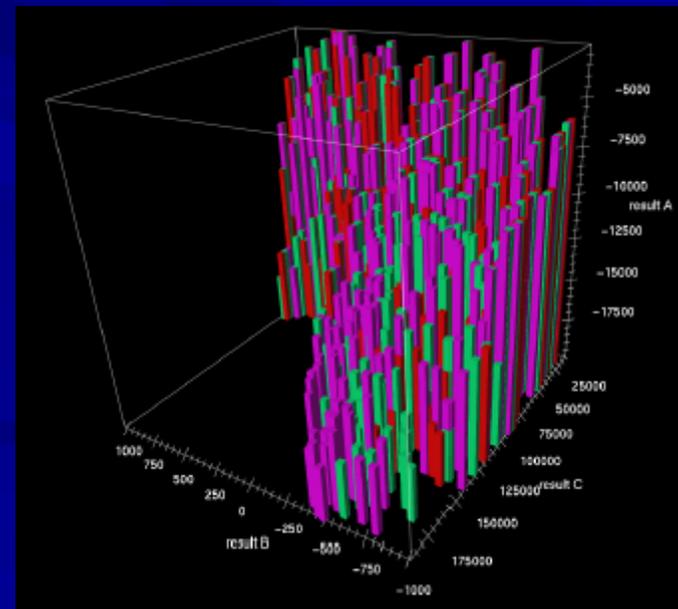


# Example of Composite Glyphs

## Vector Field Visualization

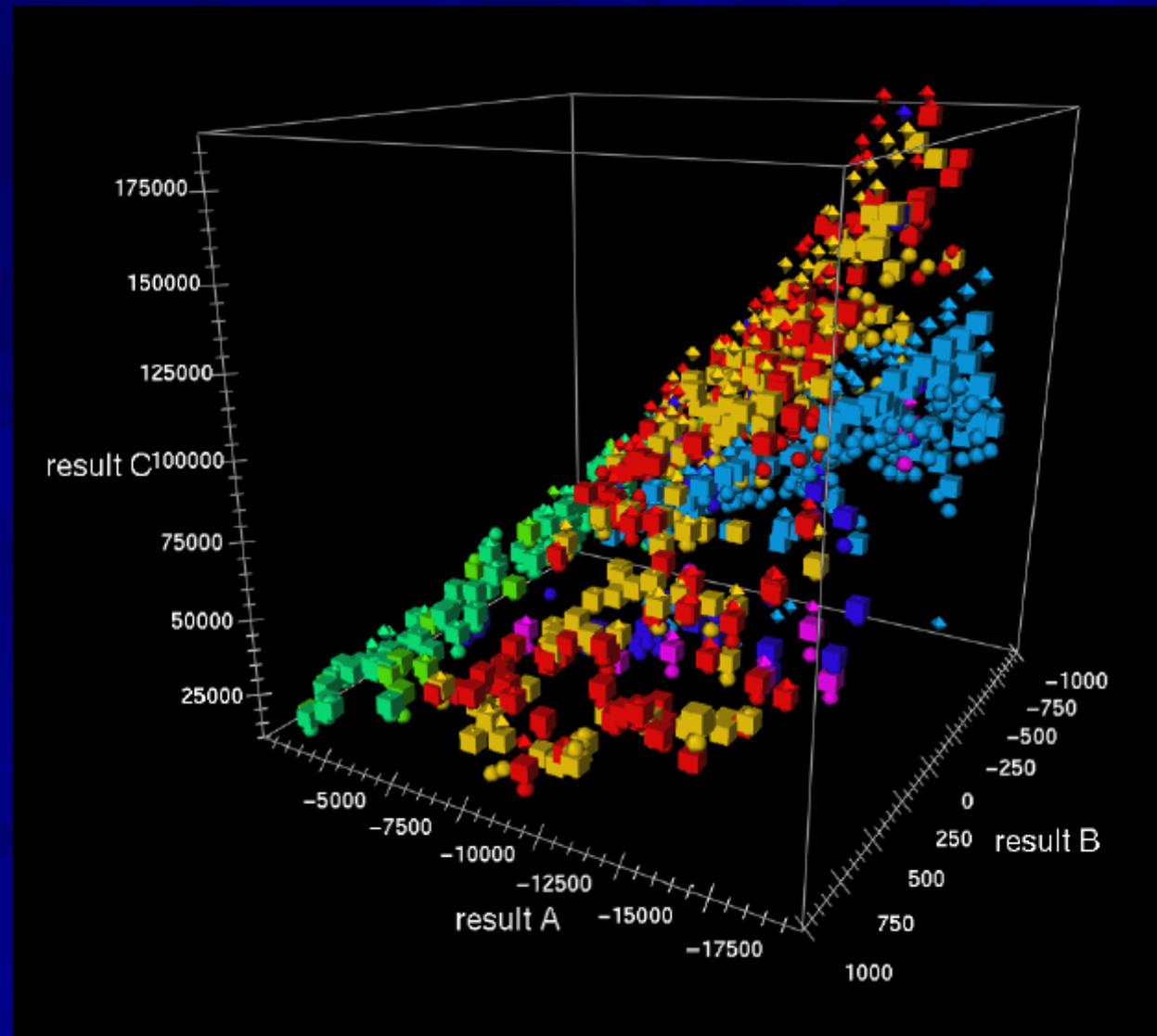


## Scatterplot with bar glyphs



# Example Application

- Scatterplot:
  - 3 Variables mapped to coordinates,
  - 1 mapped to Shape (Cube, Octahedron, or Sphere),
  - 1 mapped to Colour



# Critique of Paper 3

- + Good Motivation / Potential
- + Design choices well explained
- + Goals clearly stated
  
- Lacking implementation detail
- Lack of demo
- Use of distortion in placement strategy
- Scalability details?
- No user feedback/evaluation

Questions?