Image Based Information Visualization

or How to Unify Scivis and Infovis

prof. dr. Alexandru (Alex) Telea

Institute of Mathematics and Computer Science
University of Groningen, the Netherlands
Introduction

Who am I?

• PhD in scientific visualization (TU Eindhoven, 2000)
• assistant professor in visualization (TU Eindhoven, 2000-2007)
• professor in multiscale visual analytics (RuG, since 2007)
• 15 PhD students, 70+ MSc students
• 200+ international publications in data visualization
• co-founder SolidSource BV
Outline

1. A bit of (Personal) History
2. Modeling Visualization
3. Image-Based Information Visualization
4. Lessons learned & Where to go next
A Bit of (Personal) History

Before 1980

Around 2000
<2000: Scientific Visualization

A. Telea (2000) Visualisation and Simulation with Object-Oriented Networks; PhD thesis
>2000: Information Visualization

Scivis vs Infovis

• what are the differences?
• what are the similarities?
• how to **unify** them to better
  • Understand
  • Explain
  • Reuse
  • Progress
None of these (fully) clarifies how/why Scivis and Infovis are different…
The Visualization Pipeline: A Technical View

Direct vs Inverse Mapping Principles

A. Telea, Data Visualization – Principles and Practice, 2nd ed., CRC Press, 2014
The Visualization Pipeline: A Perceptual View

Interpretation challenges

- low-level vision: must know how the **eye** sees colors, contrasts, textures, …
- pattern recognition: must know how the **brain** assigns meaning to shapes
- high-level sensemaking: must know how the user **decides** based on semantics

How to *design* a visualization so it’s *interpreted* the way we want?
Rules for Visual Design: Visual Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Ground</th>
<th>Figure</th>
<th>Associative</th>
<th>Selective</th>
<th>Nominal (Non-Ordered)</th>
<th>Ordinal (Ordered)</th>
<th>Numerical (Quantitative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>location</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>G</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td>size</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>G</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td>shape</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>G</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>orientation</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>G</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>color hue</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>G</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>color value</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>P</td>
<td>G</td>
<td>M</td>
</tr>
<tr>
<td>texture</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>G</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>color saturation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>arrangement</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>crispness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>resolution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>transparency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A New Look at Data Mapping

**Data Variables**

\[ f: D \rightarrow C \]

- **Codomain** \( C \)
  - categorical (\( =, \neq \))
  - ordinal (\( =, \neq, < \))
  - integral (\( =, \neq, <, +, - \))
  - quantitative (\( =, \neq, <, +, -, \ast \))

- **Domain** \( D \)
  - anything really (!)

**Much like SciVis + Infovis**

**Visual Variables**

\[ f: D \rightarrow C \]

- **Codomain** \( C \)
  - brightness, hue, contrasts, edges, textures, ...
  - continuous variables (!)

- **Domain** \( D \)
  - 2D Euclidean space

**Much like Scivis**

Much like Scivis
SciVis vs InfoVis, revisited

SciVis

Hybrids

InfoVis

What are the differences you see between the three types in terms of visualization but also displayed data?
SciVis vs InfoVis, revisited: Focus on SciVis

SciVis
- visual variables: 2D and 3D
- quantitative data (temperature, pressure, velocity, density, etc)
- data is *numerical* and *continuous*
- data is defined over a 2D or 3D spatial domain (location is *given*)
- every point in this domain carries a data value (data is *dense*)
InfoVis

- visual variables: 2D (mostly)
- any data (quantitative, text, categories, relations)
- data is not necessarily *numerical* and is usually *discontinuous* (e.g. relations)
- data has no spatial association (location is *chosen* by the visualization design)
- not every point in the visualization has a data value (data is *discrete*)
Hybrids

- visual variables: 2D or 2.5D
- any data (like in InfoVis)
- at least one attribute is numerical and continuous (e.g. space in a map, time in a stock chart) and at least one is not (e.g. population measured per county)
- examples: geovisualization, timeline charts
Extra complication: Big Data

Little Data

- hundreds..thousands of items
- 1..3 dimensions
- focus on details

Big Data

- (tens of) millions of items
- tens..hundreds of dimensions
- focus on the big picture
Big Data Solution: Multiscale nature of images!
SciVis vs InfoVis data

SciVis

Continuous, numerical, spatial data

- bone dataset, 80K points
- subsample ✓
- bone detail, 88 polygons
- subsample ✓

- we throw away 75% of the data
- the semantics stays the same

- interpolation: simple
- resampling: Cauchy-continuous 😊

InfoVis

Discrete, non-numerical, non-spatial data

- bone dataset, 20K points
- subsample ✗
- bone detail, 87 polygons
- subsample ✗

- we throw away one single character
- the semantics becomes fully different!

- interpolation: often not possible
- resampling: not Cauchy continuous 😞

How to handle this challenge for Infovis data?
Solution Idea: Image-Based Visualizations

- Spot noise (1991)
- IBFV (2002)
- Multiscale IBFV (2006)

- LIC for 3D flow (2008)
- LIC for tensor fields (2009)
How to build image-based visualizations for Infovis big data?
Idea 1: Dense Pixel Displays

a) every pixel shows information (little..no whitespace, output=dense field)
b) close pixels = similar/related data items (again, related to field notion)

Idea 2: Use Shading

a) **shading creates shapes**

b) **shapes show data** (patterns, groups, relations)
Idea 2: Use Texture

Texture encodes (multiple) attribute values

two-corner treemaps (2007)  
ratio-contrast treemaps (2007)  
multiattribute contrast treemaps (2007)

extended table lenses (2007)  
importance-based antialiasing (2008)  
data encoding in texture-frequency (2006)
Idea 3: Simplify Data in Image Space

If \textit{data} is suitably mapped to a (dense) image space then we can simplify it much as we do with \textit{images}!

\begin{align*}
\text{Map (Simplify (data))} &= \text{Simplify (Map (Data))}
\end{align*}

W. De Leeuw, R. van Liere (2003) GraphSplatting: visualizing graphs as continuous fields; IEEE TVCG 9(2)
Applications 1: Multivariate/Dynamic Networks

- one of most complex Infovis data types
- relations, attributes, multiple data types, time-dependent data
- datasets of millions of nodes/links, tens of attributes/item

S. Diehl, A. Telea (2014) Multivariate Networks in Software Engineering; Springer
Multiscale Solution: Bundling

Graph Bundling
- straight lines
- edge bundles

Trail Bundling
- directed bundling
- Bundling by direction and time
A bit of history: (1) The early phase

1864: Flow map of French wine exports (Minard)

1898: Sankey diagrams

1989: Edge concentration (Newbery)

2003: Confluent drawings (Dickerson et al.)
A bit of history: (2) The advent of bundling

2005: Flow map layouts (Phan et al.)

2006: Progressive edge clustering (Qu et al.)

2005: Improved circular layouts (Gansner et al.)

2006: Hierarchically bundled edges (Holten)
A bit of history: (3) The consolidation

2008: Bundling general graphs (Holten et al.)

2010: Image-based techniques (Ersoy et al.)

2012: Bundling dynamic graphs (Nguyen et al.)

2016: Bundling huge graphs (v/d Zwan et al.)
Many application domains...

- software engineering
- network flow analysis
- medical sciences
- bioinformatics
- multidimensional data
- air traffic control

A. Lhuillier, C. Hurter, A. Telea (2017) State-of-the-art in graph and trail bundling; CGF (STAR EuroVis)
Many methods...
Definitions

Dataset
- graph
- trail-set

Drawing
\( D \)

Bundling
\( B \)

\[ P = \{ p_i \} \]

\[ D(P) = \{ D(p_i) \} \]

\[ B(D(P)) = \{ B(D(p_i)) \} \]

\[ \forall (p_i, p_j) \in P \times P | p_i \neq p_j \land \kappa(p_i, p_j) < \kappa_{\text{max}} \rightarrow \]

\[ \delta(B(D(p_i)), B(D(p_j))) \ll \delta(D(p_i), D(p_j)) \]

\( \delta : \) distance between two curves in drawing space
\( \kappa : \) dissimilarity between two paths in data and drawing spaces

A. Lhuillier, C. Hurter, A. Telea (2017) State-of-the-art in graph and trail bundling; CGF (EuroVis STAR)
1. Static graphs - Hierarchical compound

Graph $G = (V, E)$

Tree $T = (V_T, E_T)$, $V \subset V_T$

Internal: rooted tree layout

External: circular icicle plot

Spline routing via $D(T)$
1. Static graphs - Hierarchical compound

How to show the simplified structure of a bundled graph (including bundle directions)?
- use image-based edge bundles (IBEB)

A. Telea and O. Ersoy (2010) Image-based edge bundles: Simplified visualization of large graphs; CGF 29(3)
1. Static graphs - Hierarchical compound variations

- bubble tree
- treemap (2D)
- treemap (3D)
- hierarchy comparison
- DAG
- hierarchy comparison (image-based)
2. Static graphs - General undirected graphs

Force-directed methods: FDEB

**graph drawing** $D(G)$

**edge compatibility** $\kappa$

**bundling** $B(D(G))$

Basic idea

- like force-directed graph layouts, but done for
  - sampling points along edges in $D(G)$
  - point-point interactions determined dynamically via spatial proximity (in graph layouts, forces act on nodes of $G$)
- works for general graphs (unlike HEB)
- basic idea is very slow ($O(N^2)$ for $N$ edge-sampling points)

D. Holten and J. J. van Wijk (2008) Force Directed Edge Bundling for Graph Visualization; CGF/EuroVis
2. Static graphs - General undirected graphs (cont’d)

Geometric/image methods: SBEB

- Input shape
- Skeletonization
- Output skeleton

- Graph drawing
- Edge clusters
- For each cluster...
- Blurred drawing

- Binary shape
- Skeleton
- Edge bundles

O. Ersoy et al. (2011) Skeleton-based Edge Bundling for Graph Visualization; TVCG 17(12)
2. Static graphs - General undirected graphs (cont’d)

Geometric/image methods: SBEB

US migrations (~10K edges)  
software calls (~5K edges)
2. Static graphs - General undirected graphs (cont’d)

Image-based methods: KDEEB

If bundling sharpens the edge density, then sharpening the edge density should bundle.

C. Hurter, O. Ersoy, A. Telea (2010) Graph bundling by kernel density estimation; CGF 31(2)
2. Static graphs - General undirected graphs (cont’d)

Image-based methods: KDEEB

Move edges towards local density maximum.
2. Static graphs - General undirected graphs (cont’d)

Image-based methods: CUBu, FFTEB

MINGLE (2012): several minutes on a standard PC

CUBu (2015): **0.15 seconds**
400x400 pixels
19M sample points

FFTEB (2017): **0.09 seconds**
1000x1000 pixels
24M sample points

M. van der Zwan, V. Codreanu, A. Telea (2016) CUBu: Universal real-time bundling for large graphs; IEEE TVCG 22(12)
A. Lhuillier, C. Hurter, A. Telea (2017) FFTEB: Edge bundling of huge graphs by the Fast Fourier transform (PacificVis)
2. Static graphs - Directed graphs, comparison
3. Dynamic streaming graphs

How to show changes in a network?
• use KDEEB on the dynamic graph (simple!)

US flights (Aug 2008)
(~20K flights)

World flights (June 2013)
(~1M flights)

[Ersoy et al. ‘14]
3. Dynamic streaming graphs: Eye-tracking data

How to analyze how people see scenes?

- evaluate/optimize user-interface design for highly-critical devices (e.g. aircraft, surgery)
- bundle the eye-gaze tracks (recorded by an eye tracker)

V. Peysakhovich et al. (2014) Attribute-Driven Edge Bundling for General Graphs with Applications in Trail Analysis; IEEE PacificVis
4. Dynamic sequence graphs

How to show changes between a graph and the previous/next one?

Changes of code duplication (clones) in the evolution of a software system

[Ersoy et al. ‘14]
4. Dynamic sequence graphs: Execution traces

Given several executions of a program, how to spot differences?
• used for finding performance/quality problems in software

J. Trumper, J. Dollner, A. Telea (2013) Multiscale visual comparison of execution traces; IEEE ICPC
5. Simplified visualization of general graphs

Generalize image-based edge bundles (IBEB)

SBEB
[Ersoy et al. 2011]

CUBu
[v/d Zwan et al. 2016]
6. Multidimensional data

Visualize errors in multidimensional projections: Replace scatterplots by continuous fields!
6. Multidimensional data

Explain projections by most-relevant attributes: Replace scatterplots by continuous fields!

Data: 2400 wine samples, 12 attributes/sample
Goal: see why wine sorts resemble each other
What we have seen

Image-based information visualization

- **synergy** of graphics, data analysis, information visualization, imaging
- data filtering, mapping, rendering get merged in the image space
- compared to Scivis: all is the same, but Infovis data is
  - defined on non-Euclidean domains and potentially not continuous
    …thus not easily interpolable!
- **continuous, natural-like** images solve the above problems
  - pack lots of information (every pixel shows something)
  - have a multiscale nature (overview & details easy to produce)
  - are intuitive to interpret (resemble familiar shapes)
  - …and are nice (attract attention)
Where to from here?

Open challenges

• explore links of bundling, clustering, segmentation, skeletonization (towards an **unified image-based theory** of data simplification?)

• **teaching** Scivis and Infovis in an unified setting

• image-based visualization for **high-dimensional** data / machine learning

P. Rauber *et al.* (2016) Visualizing the hidden activity of artificial neural networks; IEEE TVCG 23(1)
To finish: My favorite example 😊

19-dimensional dataset (images), visualized with mix of image-based techniques
- points: 2D projection of 19-dimensional data, shaded by one attribute
- bundles: point-to-point projection errors
- cushions: clusters of similar points

Cover image for Data Visualization: Principles and Practice, CRC Press, 2014
Thank you for your interest!

Alex Telea

a.c.telea@rug.nl

www.cs.rug.nl/svcg

• examples, applications
• code
• datasets
• papers