Advanced Data Visualization

CS 6965

Fall 2019

Prof. Bei Wang Phillips

University of Utah

Lecture 22

Graph Layout Edge Bundling

NV

Matrix Reordering

EuroVis 2016
R. Maciejewski, T. Ropinski, and A. Vilanova
(Guest Editors)

Volume 35 (2016), Number 3
STAR – State of The Art Report

Matrix Reordering Methods for Table and Network Visualization

Michael Behrisch[1], Benjamin Bach[2], Nathalie Henry Riche[3], Tobias Schreck[4], Jean-Daniel Fekete[5]

[1]Universität Konstanz, Germany
[2]Microsoft Research-Inria Joint Centre, France
[3]Microsoft Research, USA
[4]University of Technology Graz, Austria
[5]Inria, France

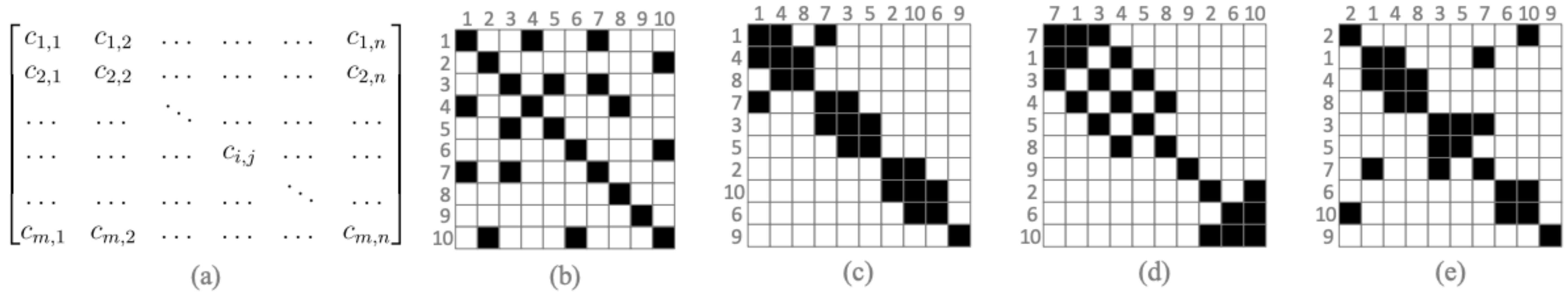


Figure 1: Visual matrix of numerical data (a) ordered randomly (b) and with three algorithms (c-e) revealing different patterns.

Abstract

This survey provides a description of algorithms to reorder visual matrices of tabular data and adjacency matrix of networks. The goal of this survey is to provide a comprehensive list of reordering algorithms published in different fields such as statistics, bioinformatics, or graph theory. While several of these algorithms are described in publications and others are available in software libraries and programs, there is little awareness of what is done across all fields. Our survey aims at describing these reordering algorithms in a unified manner to enable a wide audience to understand their differences and subtleties. We organize this corpus in a consistent manner, independently of the application or research field. We also provide practical guidance on how to select appropriate algorithms depending on the structure and size of the matrix to reorder, and point to implementations when available.

Reorder.js

Reorder.js

Reorder.js is a JavaScript library for reordering matrices, i.e. either tables, graphs vertices, or parallel coordinates axes.

Want to learn more? See the wiki.

https://github.com/jdfekete/reorder.js/

Node and edge clustering

Edge Bundling

Survey: ZhouXuYuan2013

Survey: http://www.chaofz.me/asset/file/Edge%20Bundling%20Survey.pdf [Zhou2017]

Edge Bundling Classification

- Hierarchical Edge Bundling
- Flow Map
- Geometry-based
- Force-directed
- Image-based
- Skeleton-based

[Zhou2017]

Hierarchical Edge Bundling

Hierarchical Edge Bundling

- Starting from tree visualization…

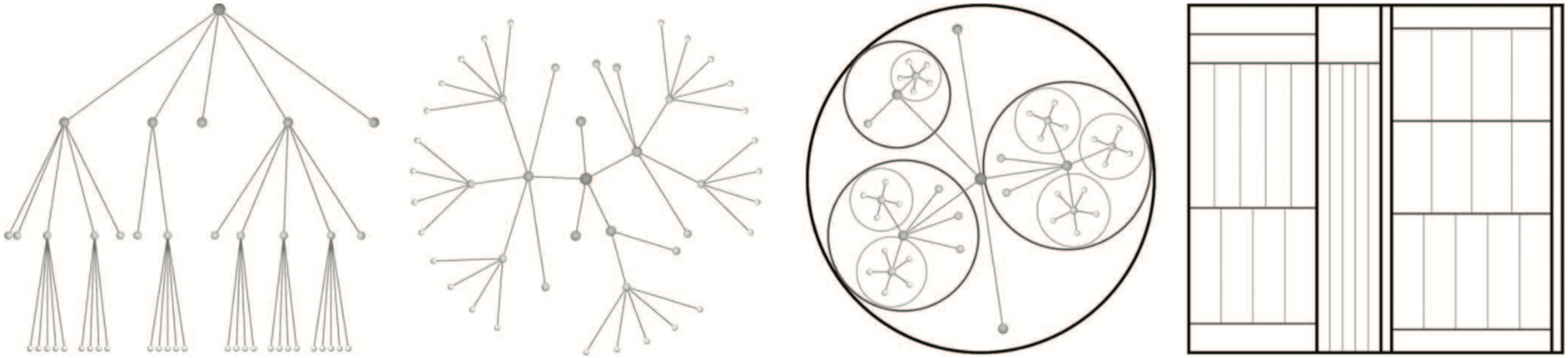


Fig. 1. Common tree visualization techniques. From left-to-right: rooted tree, radial tree, balloon tree, and treemap layout.

Holten2006

Hierarchical Edge Bundling

- Bundling adjacent edges using hierarchical relations

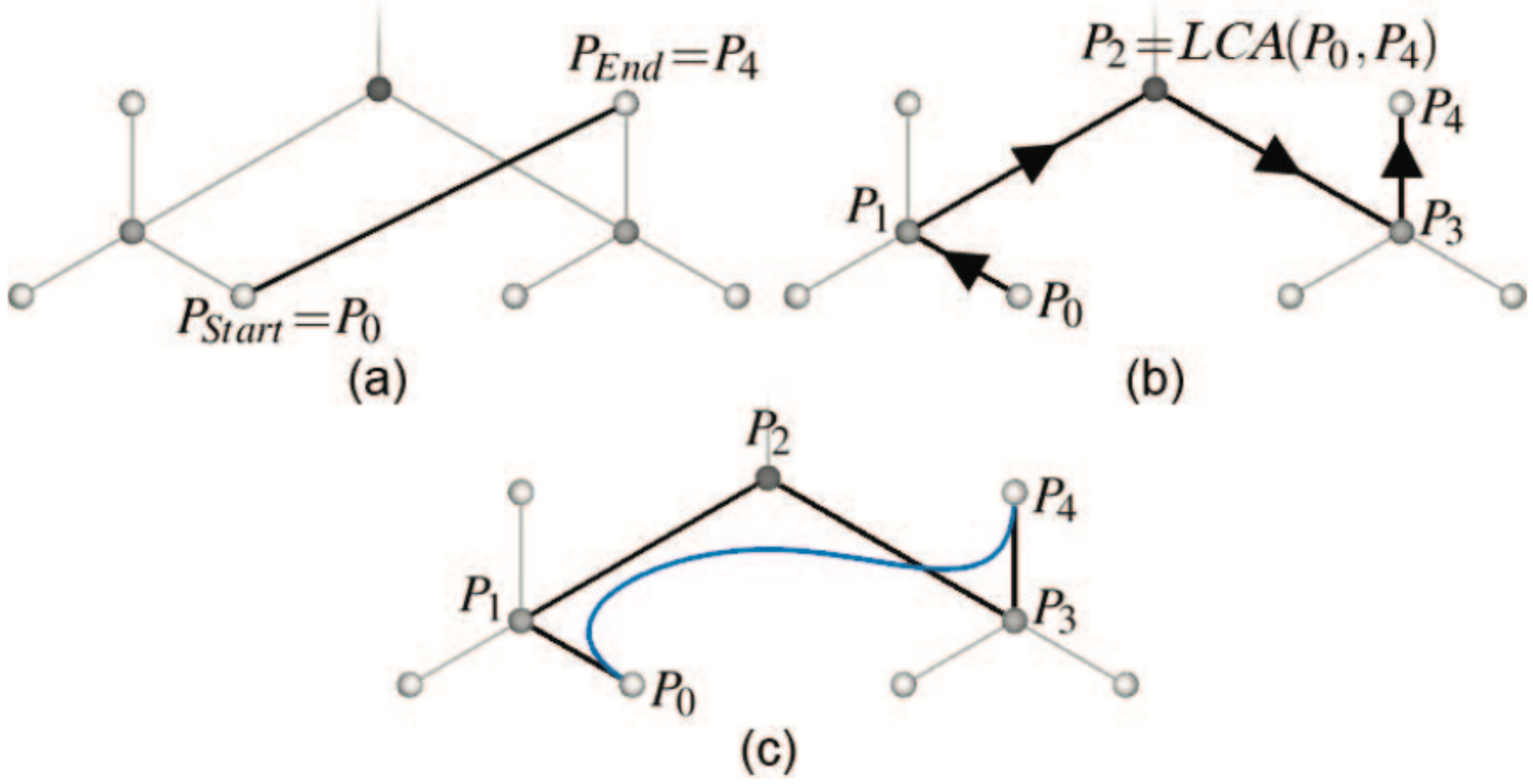
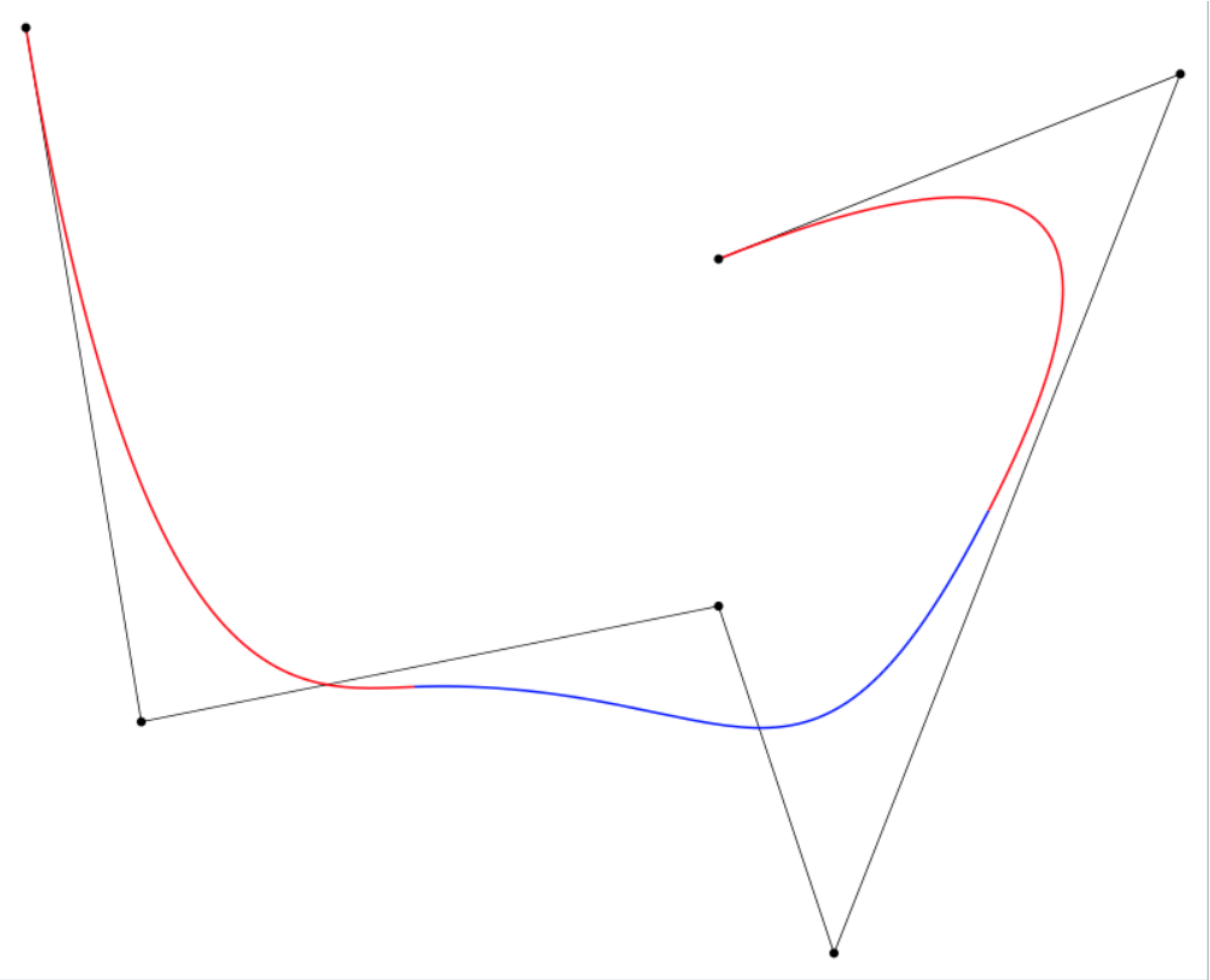


Fig. 3. Bundling adjacency edges by using the available hierarchy. (a) Straight line connection between P_0 and P_4; (b) path along the hierarchy between P_0 and P_4; (c) spline curve depicting the connection between P_0 and P_4 by using the path from (b) as the control polygon.

Holten2006

Spline Models



B-spline with control points/control polygon, and marked component curves

Piecewise cubic B-spline

https://en.wikipedia.org/wiki/B-spline

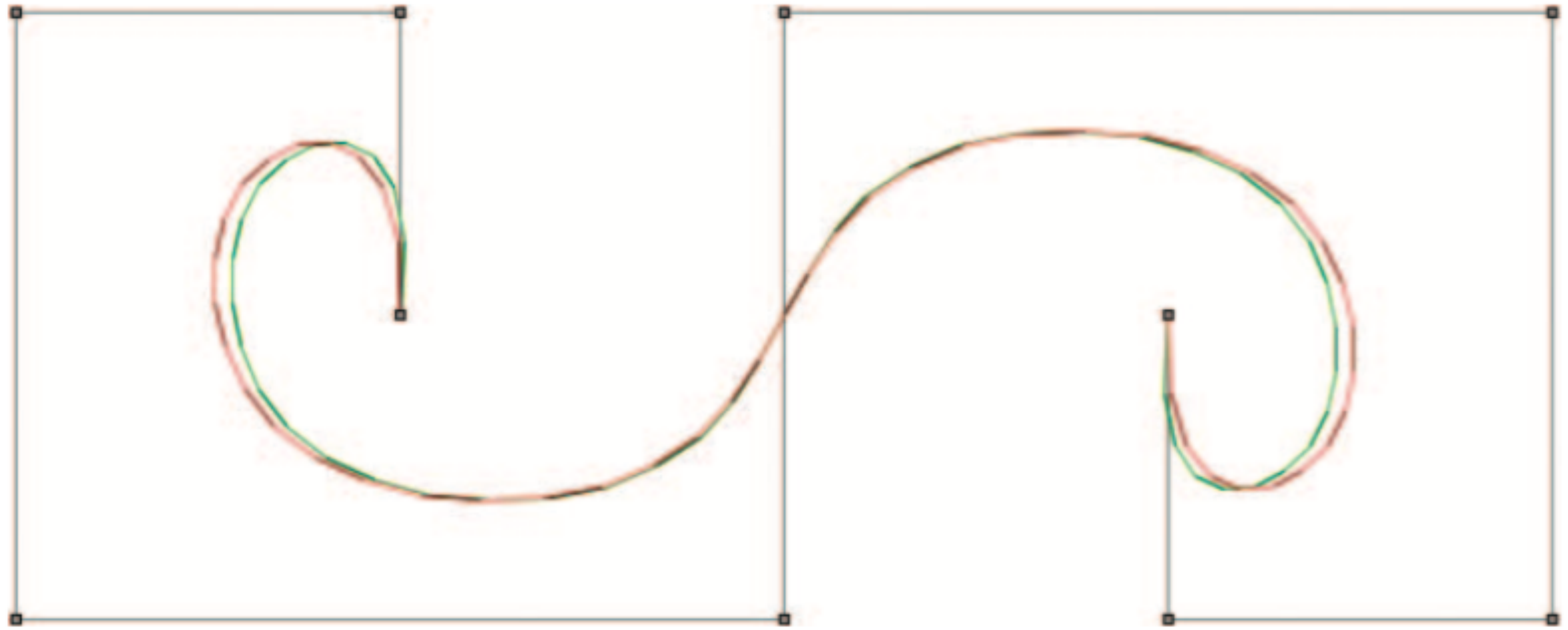
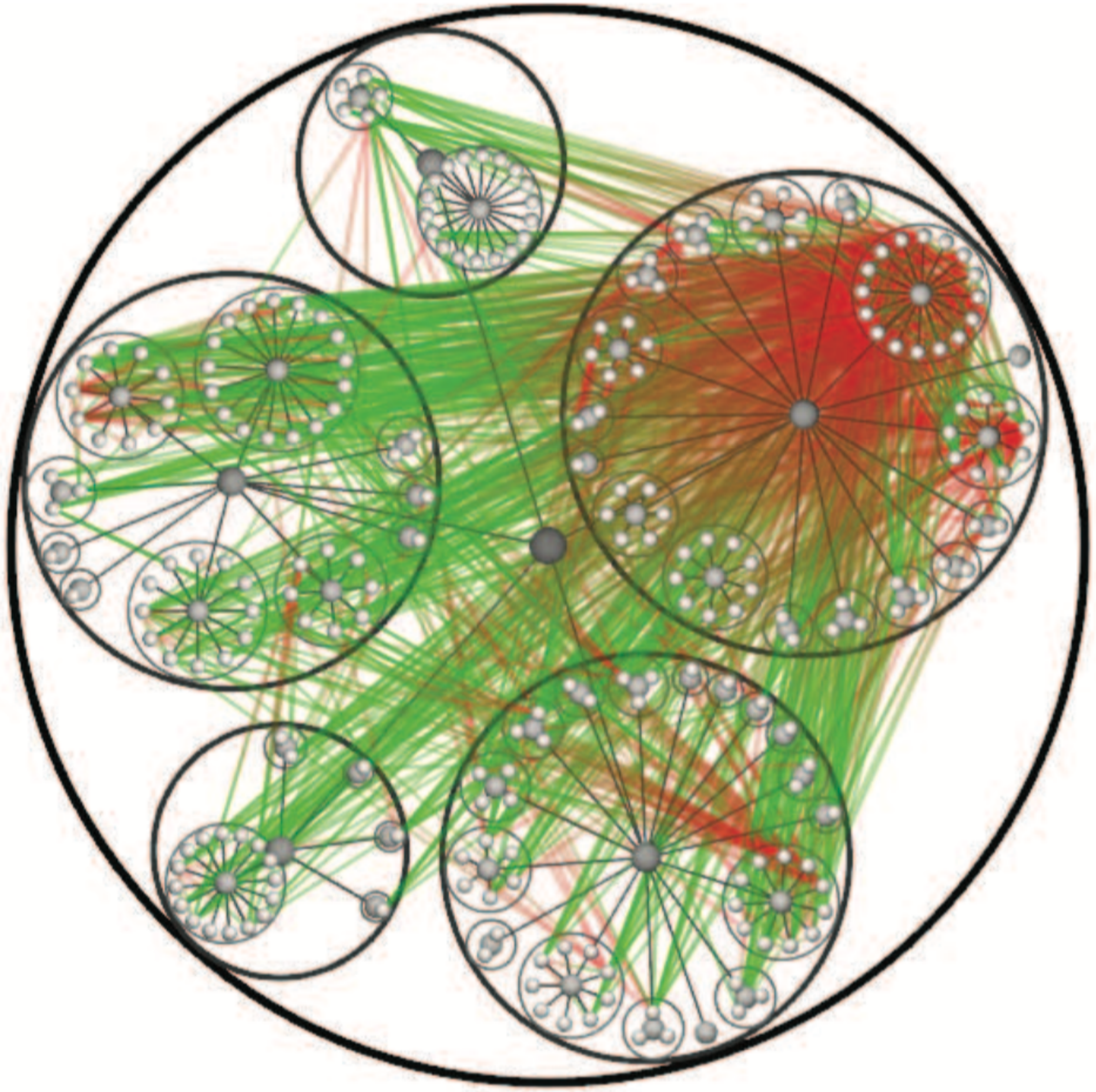


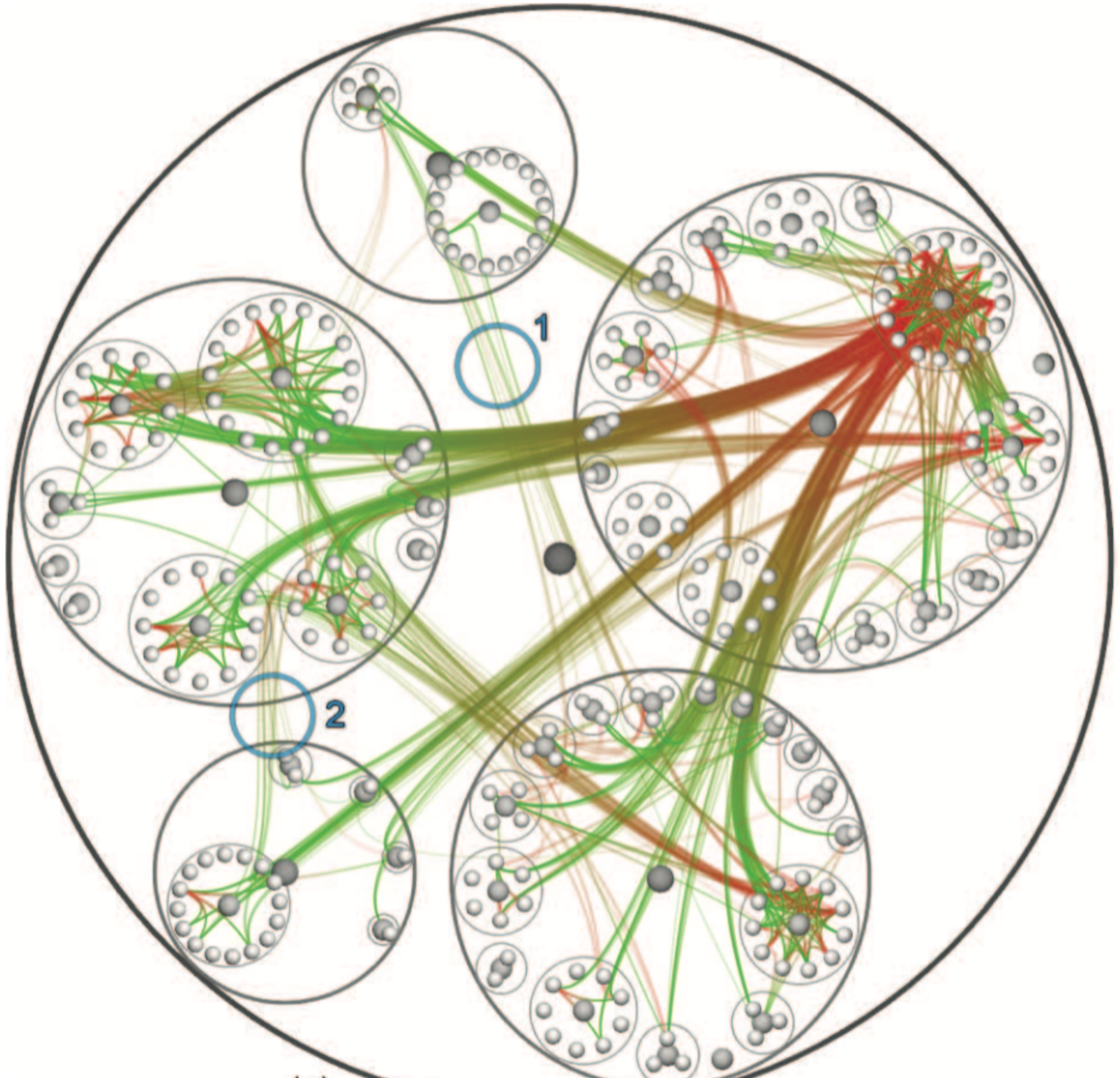
Fig. 5. Spline curve straightening by means of control polygon straightening (green) and spline point straightening (red) yield somewhat different results, but these differences are minimal from a visual point of view.

Holten2006

Hierarchical Edge Bundling



(a)



(a)

Holten2006

Bundling Strength

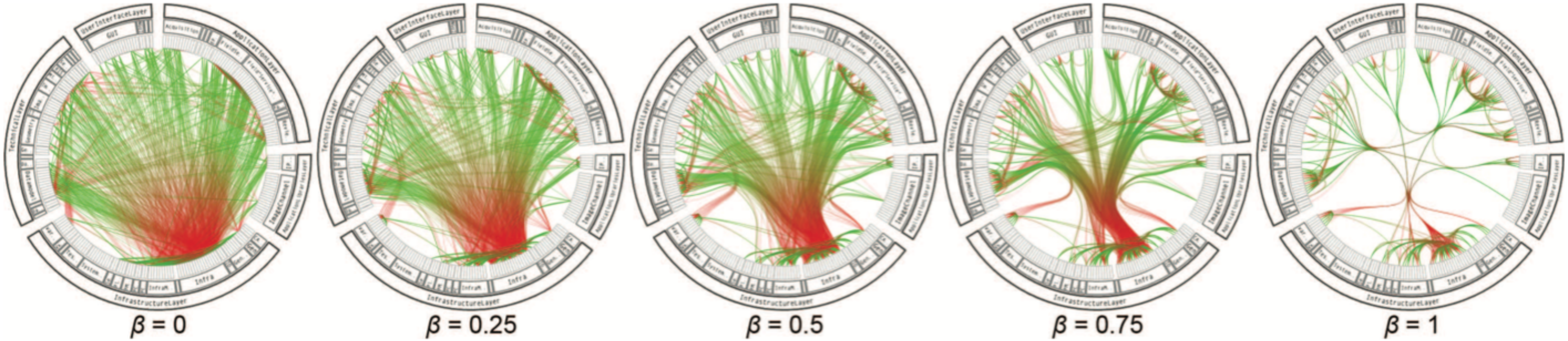


Fig. 14. Using the bundling strength β to provide a trade-off between low-level and high-level views of the adjacency relations. The value of β increases from left-to-right; low values mainly provide low-level, node-to-node connectivity information, whereas high values provide high-level information as well by implicit visualization of adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes.

Holten2006

Flow Map Layout

Flow map layout

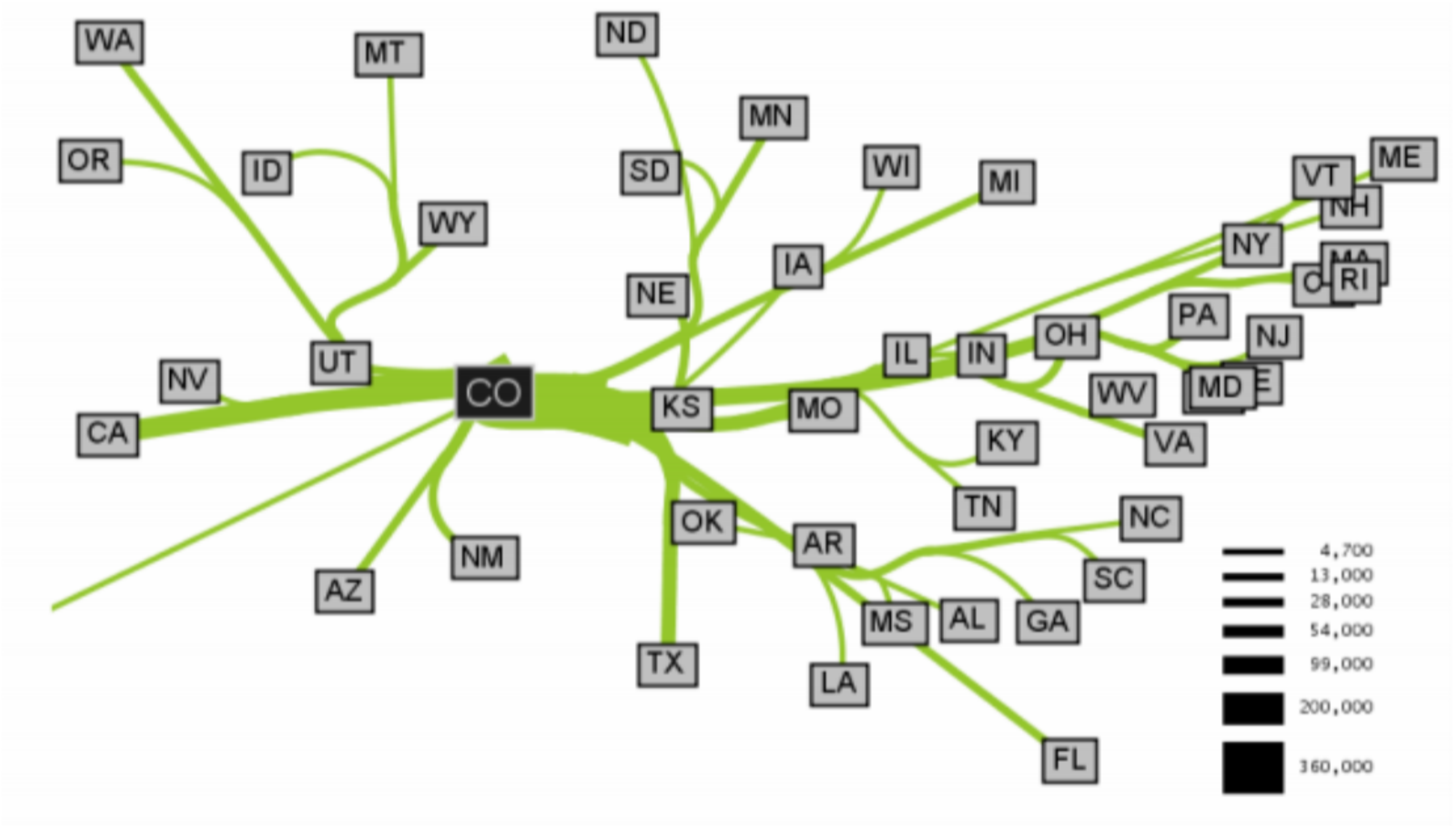


Fig. 1: Outgoing migration map from Colorado from 1995-2000, generated by Phan et al.'s algorithm

[Zhou2017]

Flow map layout

Fig. 2: Flow layout of embodied CO2 to the United Kingdom generated by Verbeek et al.'s techniques

[Zhou2017]

Coming up:
Force-directed, geometry-based, Image-based Edge bundling

More Edge Bundling Graph-theoretic Measures

NV

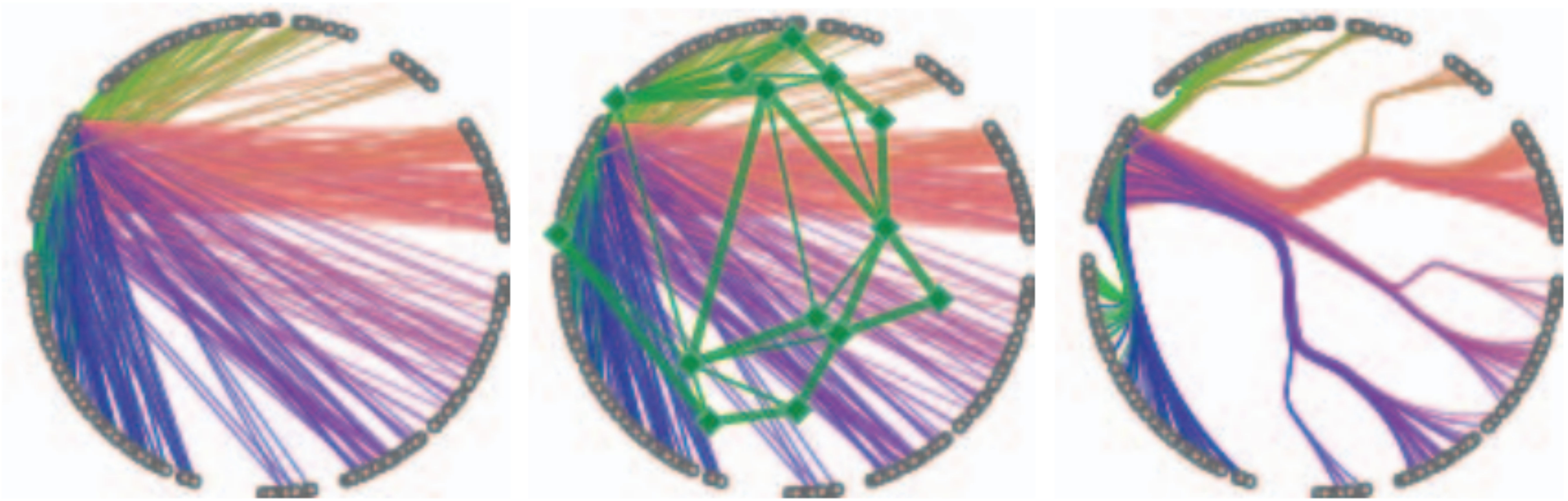
Edge Bundling

Survey: ZhouXuYuan2013

Survey: http://www.chaofz.me/asset/file/Edge%20Bundling%20Survey.pdf [Zhou2017]

Geometry-Based Edge Bundling

Geometry Based Edge Clustering



(a) (b) (c)

CuiZhouQu2008

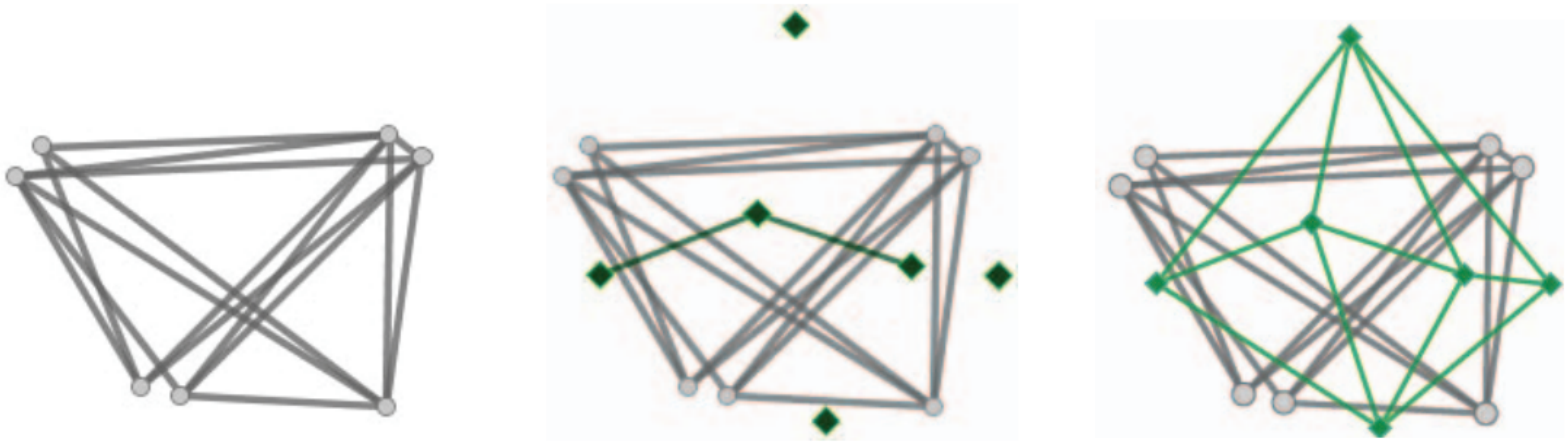
Geometry Based Edge Clustering

CuiZhouQu2008

- Graph analyzer: uses edge distribution patterns to figure out representative primary edge directions.
- Mesh generator: generates control-mesh edges perpendicular to each selected primary direction.
- Bundler: uses intersections between the original graph and the control mesh, it sets some control points on the control-mesh edges and curves the original graph edges to pass through these control points to form edge clusters.
- Edge smoother: curved edges with too many zigzags are further fine-tuned to become visually pleasing.

CuiZhouQu2008

Manual/automatic mesh generation



(a) (b) (c)

Fig. 3. Manual mesh generation: (a) a graph; (b) users click a set of vertices and edges; (c) a mesh is generated by Constrained Delaunay triangulation of the vertices and edges.

Animation

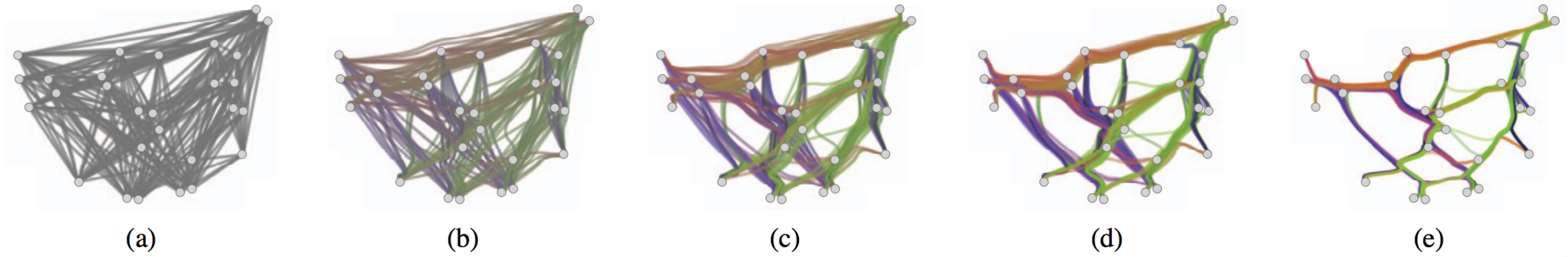
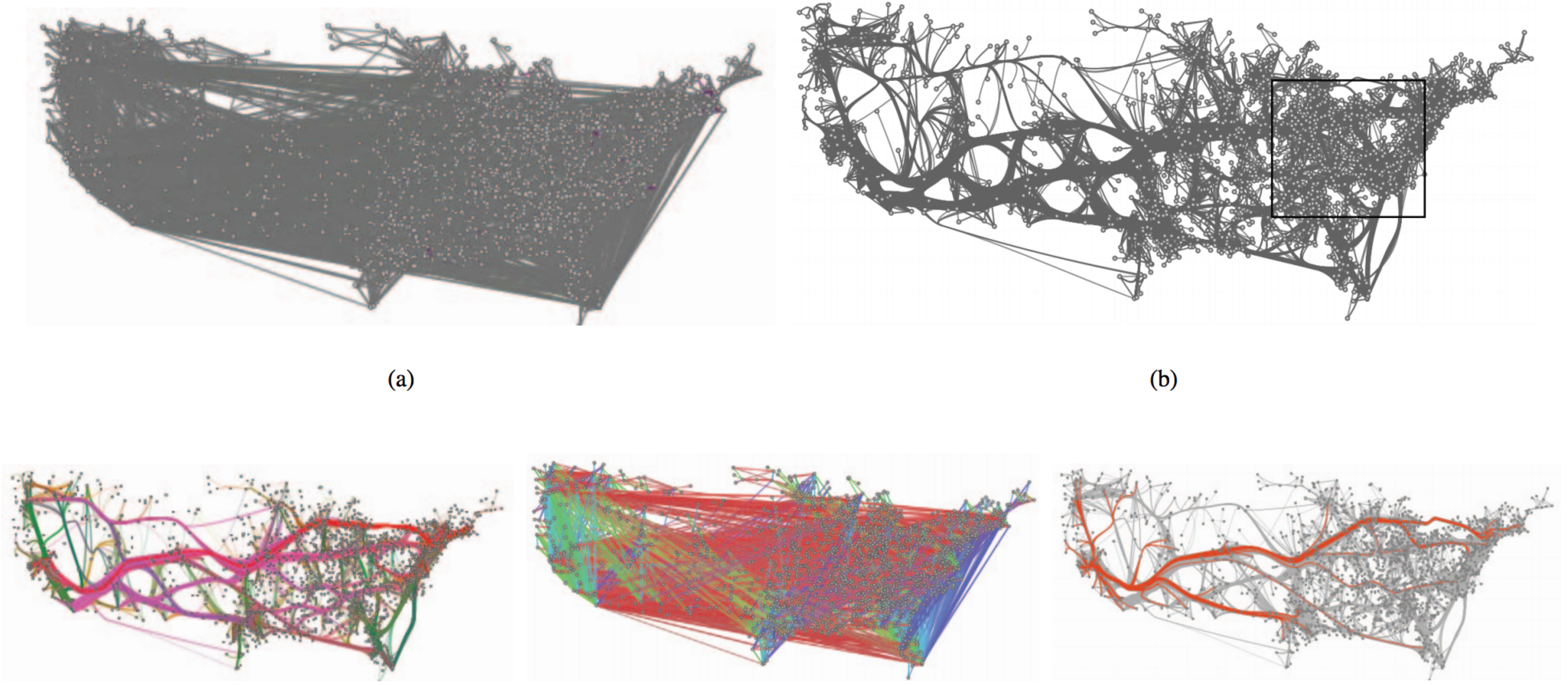


Fig. 10. An animation sequence for an edge-clustering process. The color is used to encode the edge directions.

By viewing the animations, users will have
a better idea about the data and may detect some patterns that may
otherwise disappear in the final static layouts.

CuiZhouQu2008



(a) (b)

(c) (d) (e)

Fig. 14. U.S. immigration graph with 1790 nodes and 9798 edges: (a) original layout; (b) the edge-clustered result; (c) the result after applying edge clustering and transfer function; (d) the result after applying only transfer function; (e) a flow map layout highlighted in orange color.

Force-Directed Edge Bundling

Force-Directed Edge Bundling

- Highlight: A self-organizing approach: edges are modeled as flexible springs having attracting forces on other nodes while the node positions still keep fixed

HoltenWijk2009

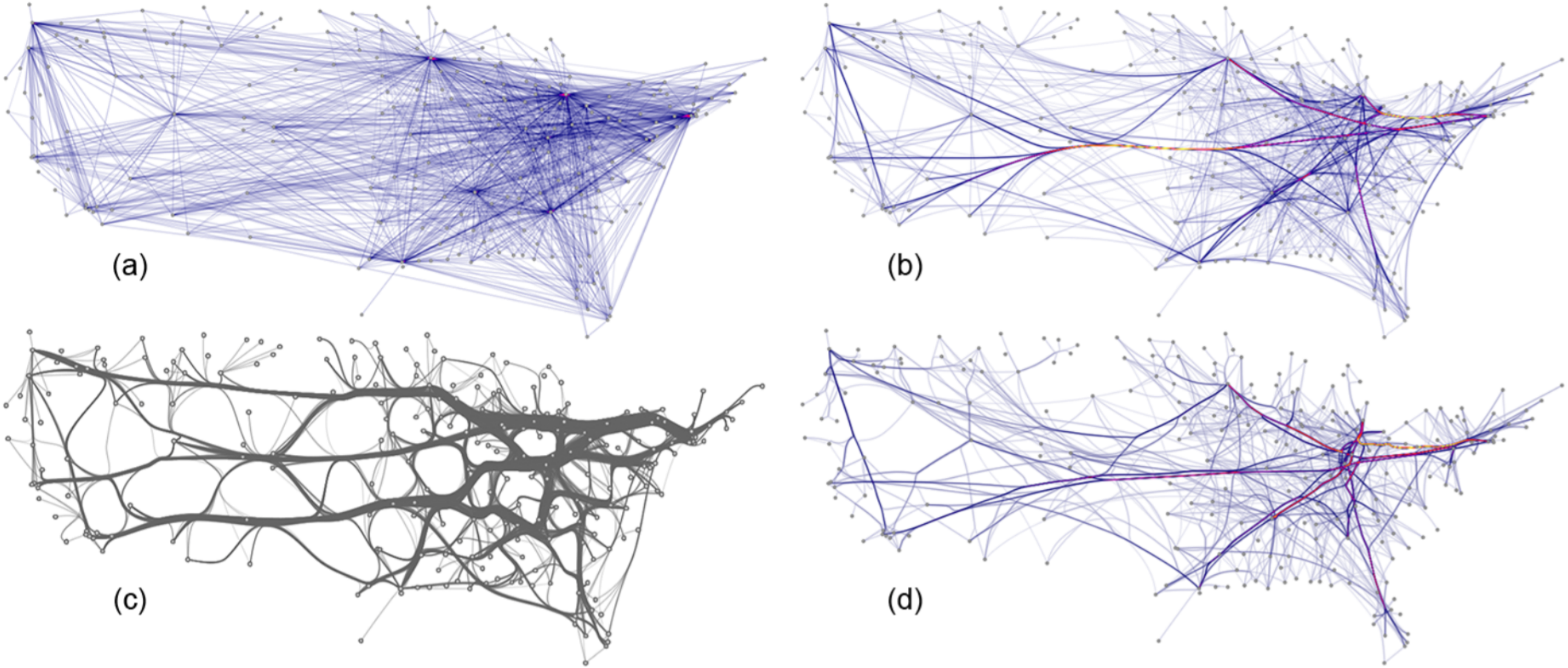


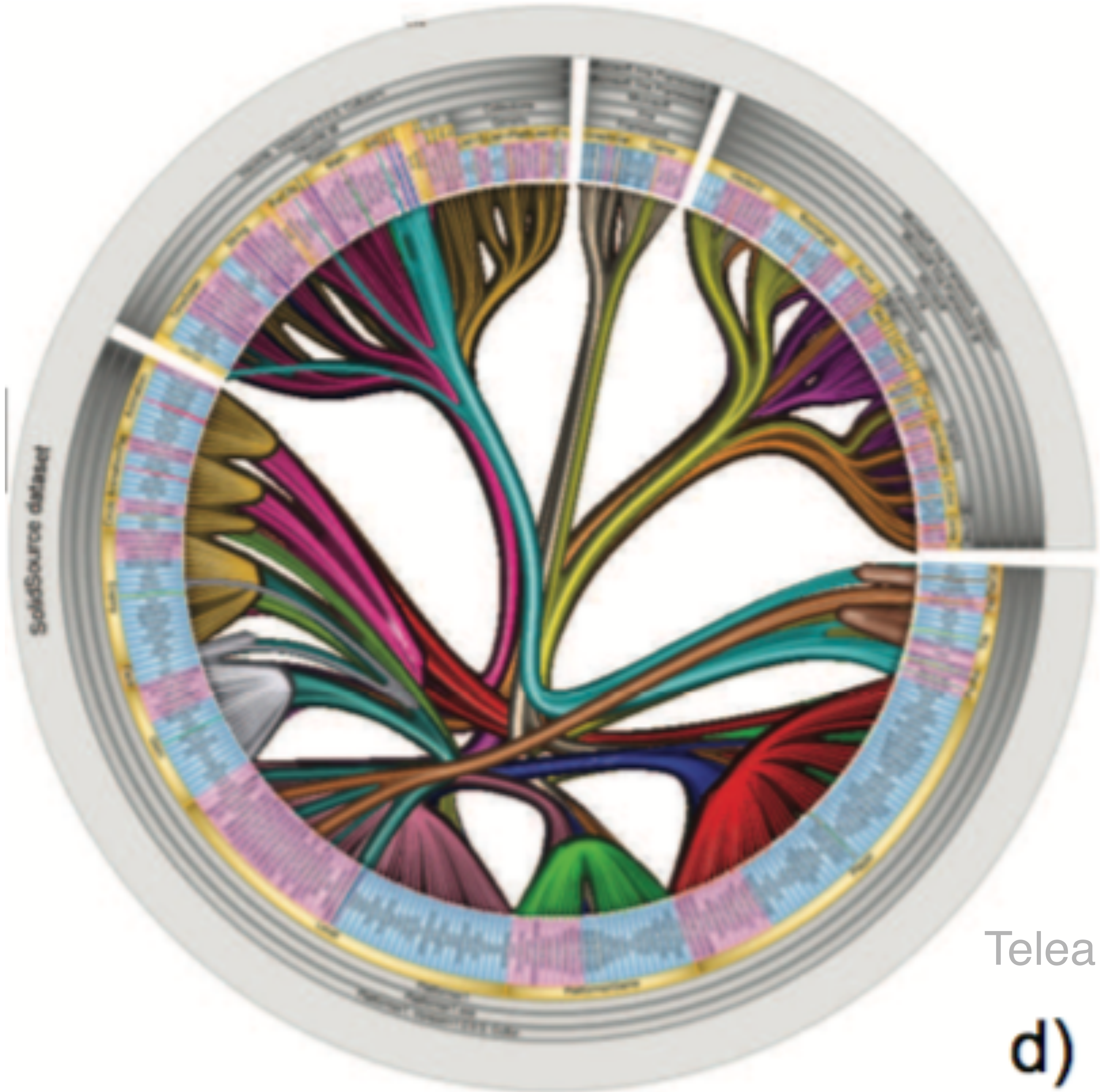
Figure 7: *US airlines graph (235 nodes, 2101 edges) (a) not bundled and bundled using (b) FDEB with inverse-linear model, (c) GBEB, and (d) FDEB with inverse-quadratic model.*

GBEB: Geometry based edge bundling

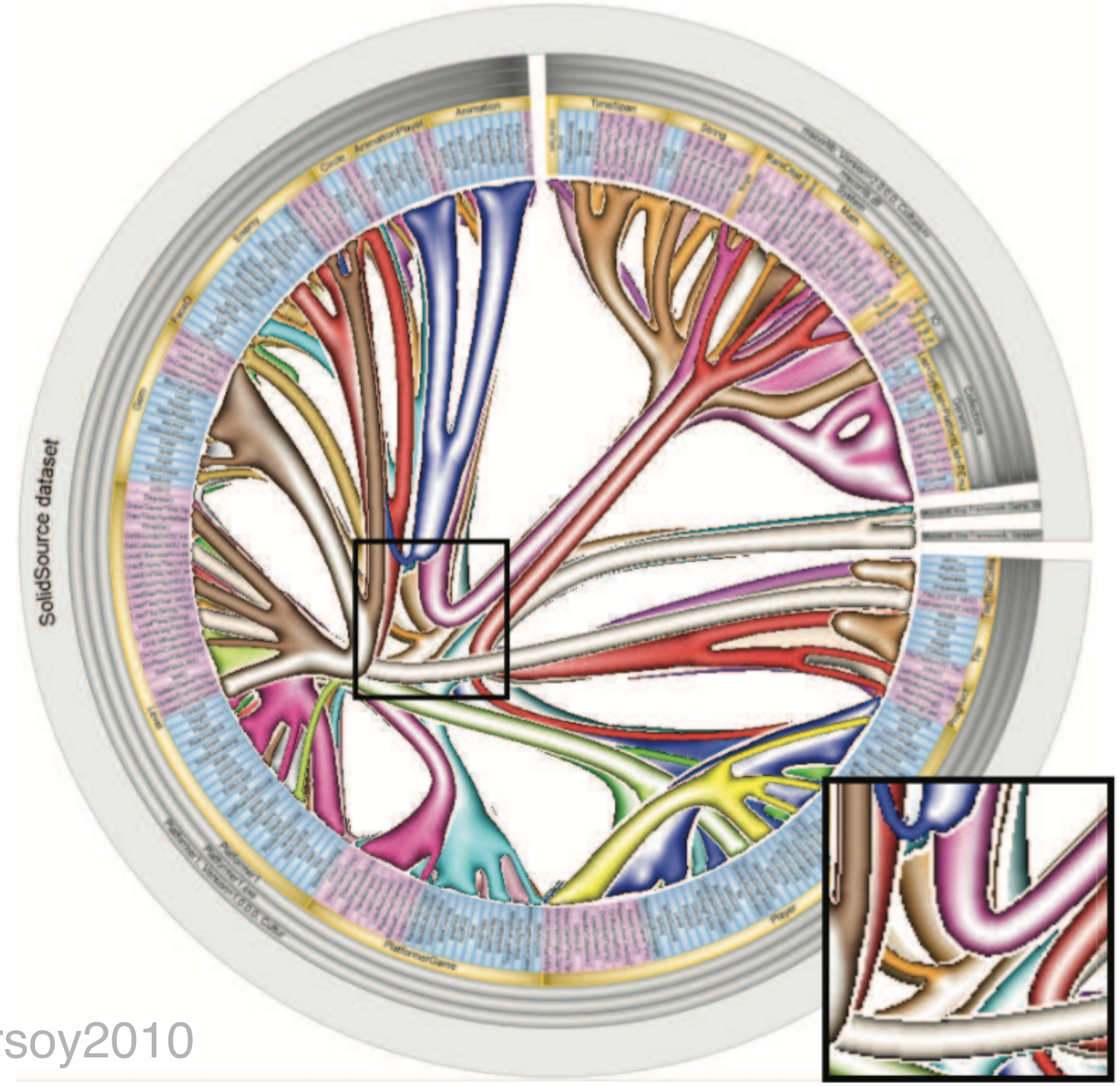
HoltenWijk2009

Image-Based Edge Bundling

Image Based Edge Clustering



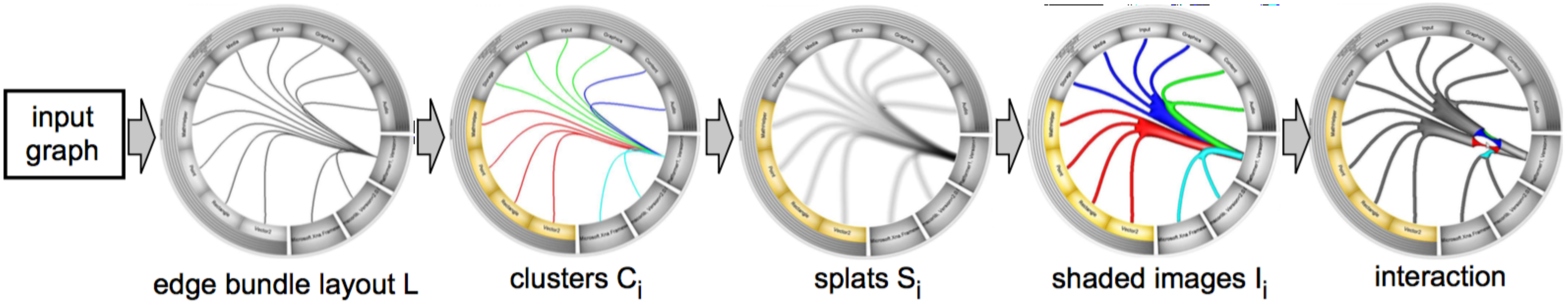
d)



TeleaErsoy2010

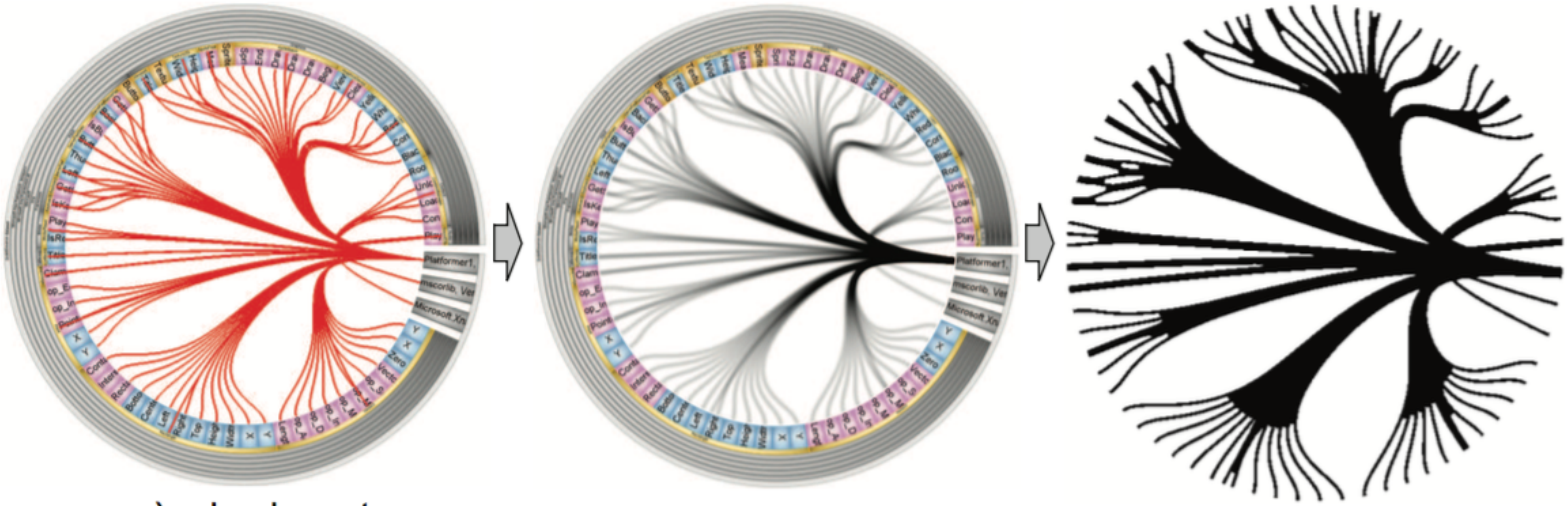
Figure 6: *Bundle visual separation using halos*

Image Based Edge Clustering



TeleaErsoy2010

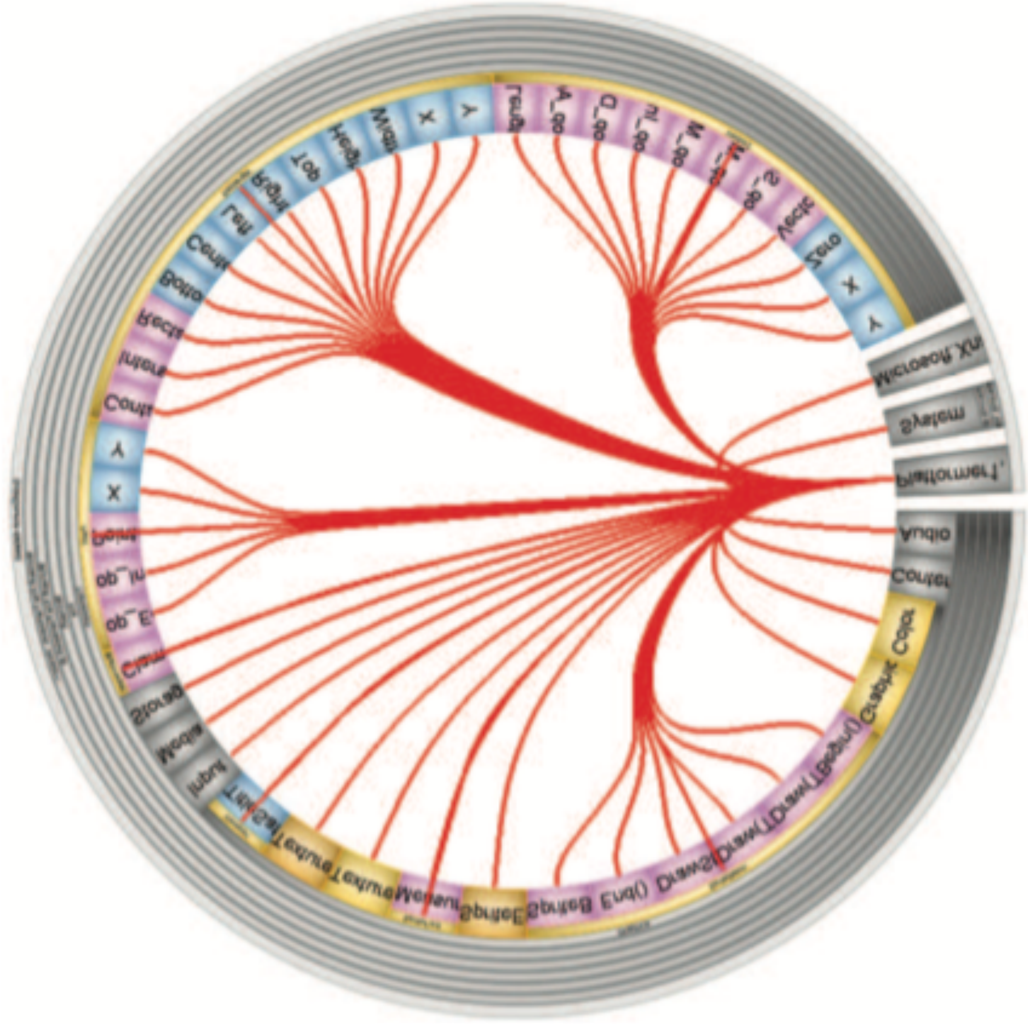
Shape Construction



a) edge layout b) splatted image c) binary shape

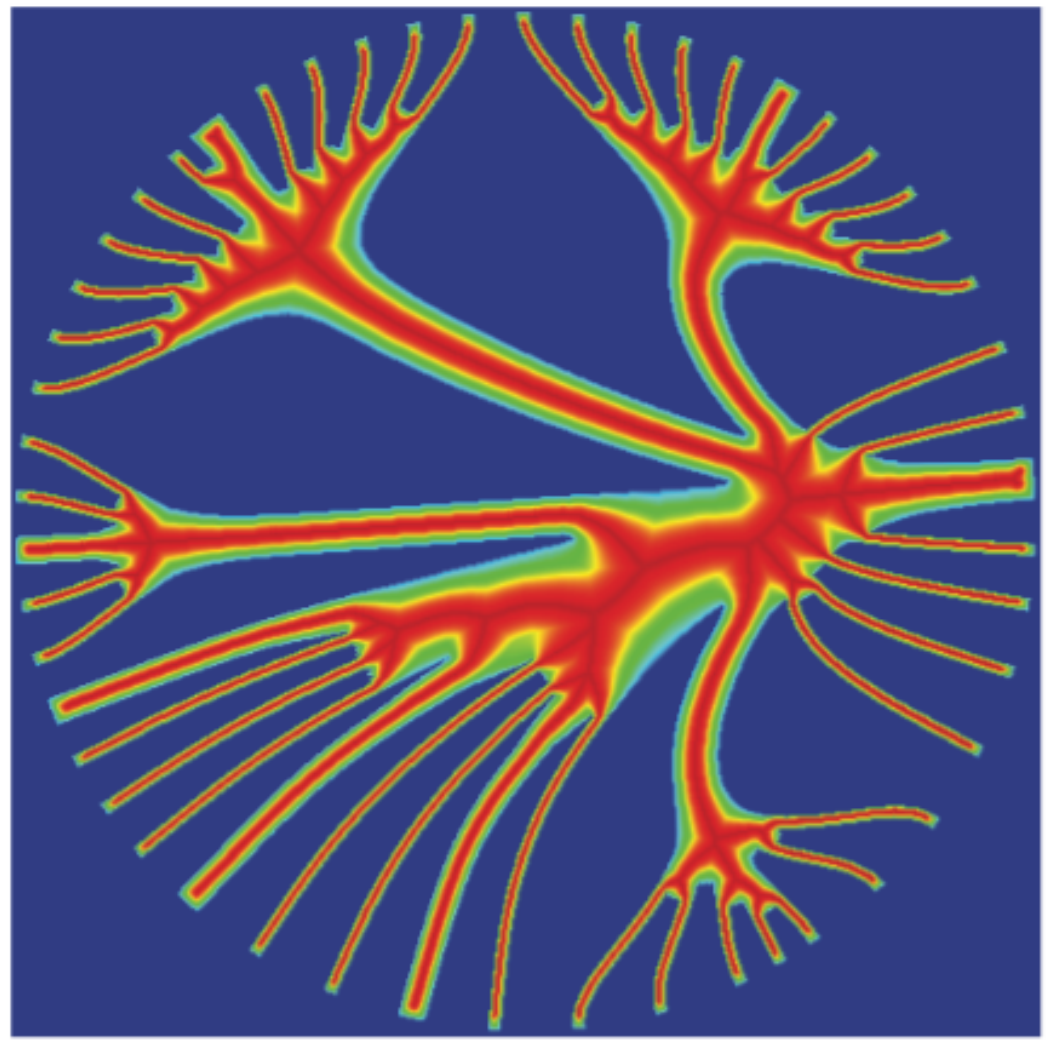
TeleaErsoy2010

Shading

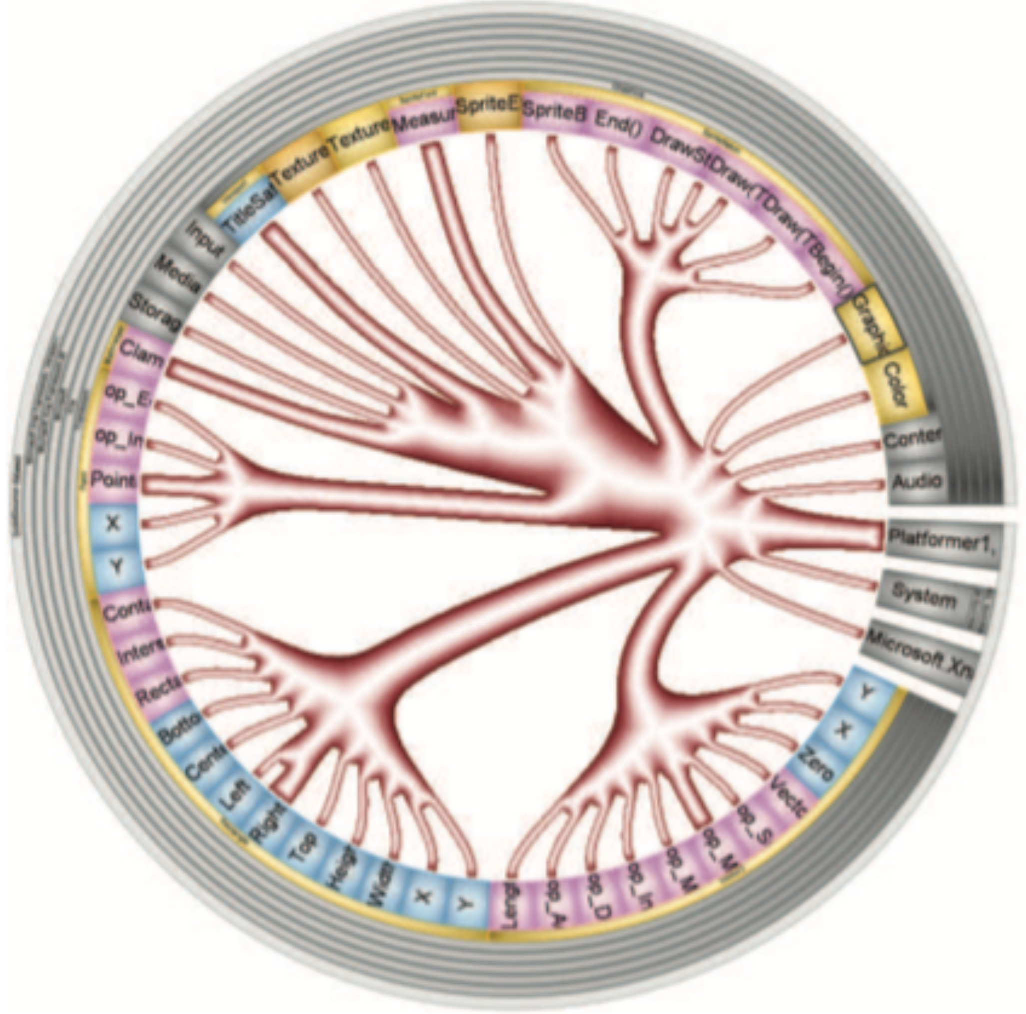


a) edge layout

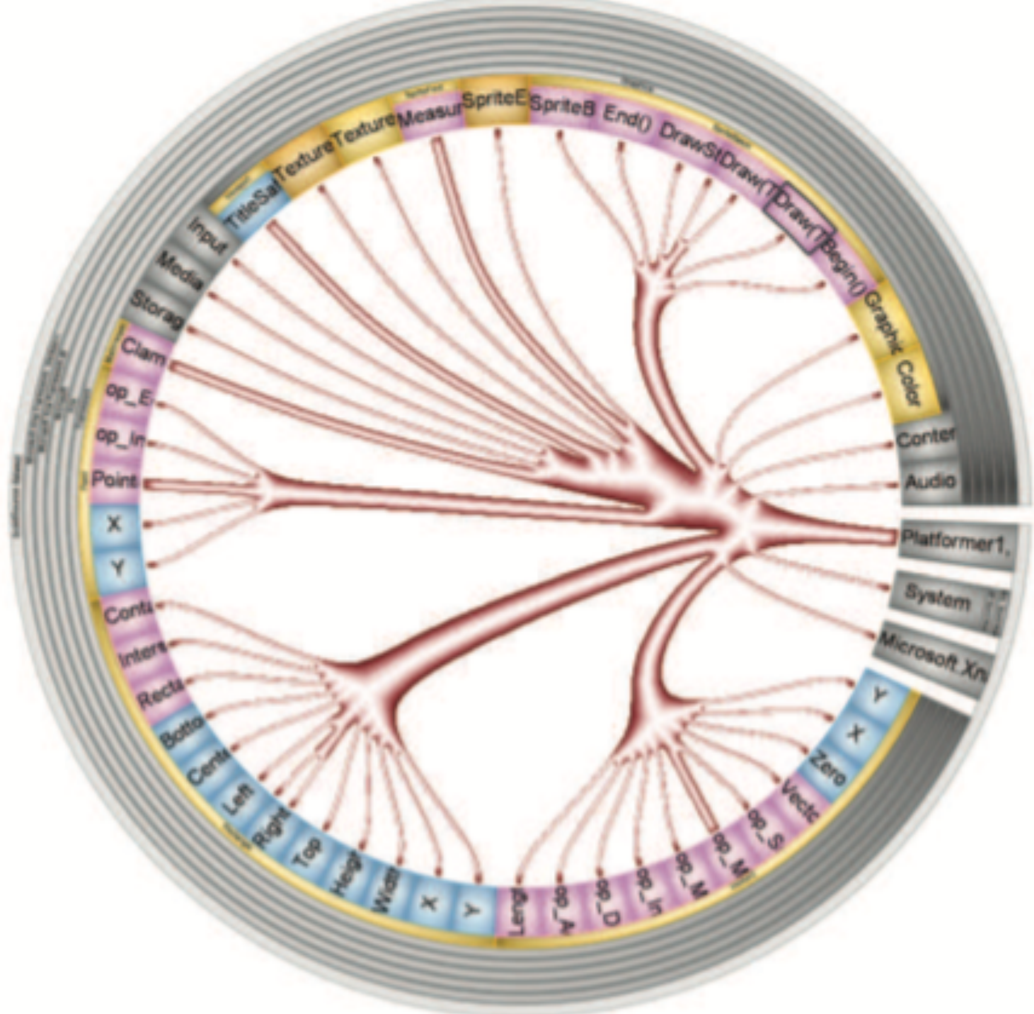
b) shape I and skeleton Sk



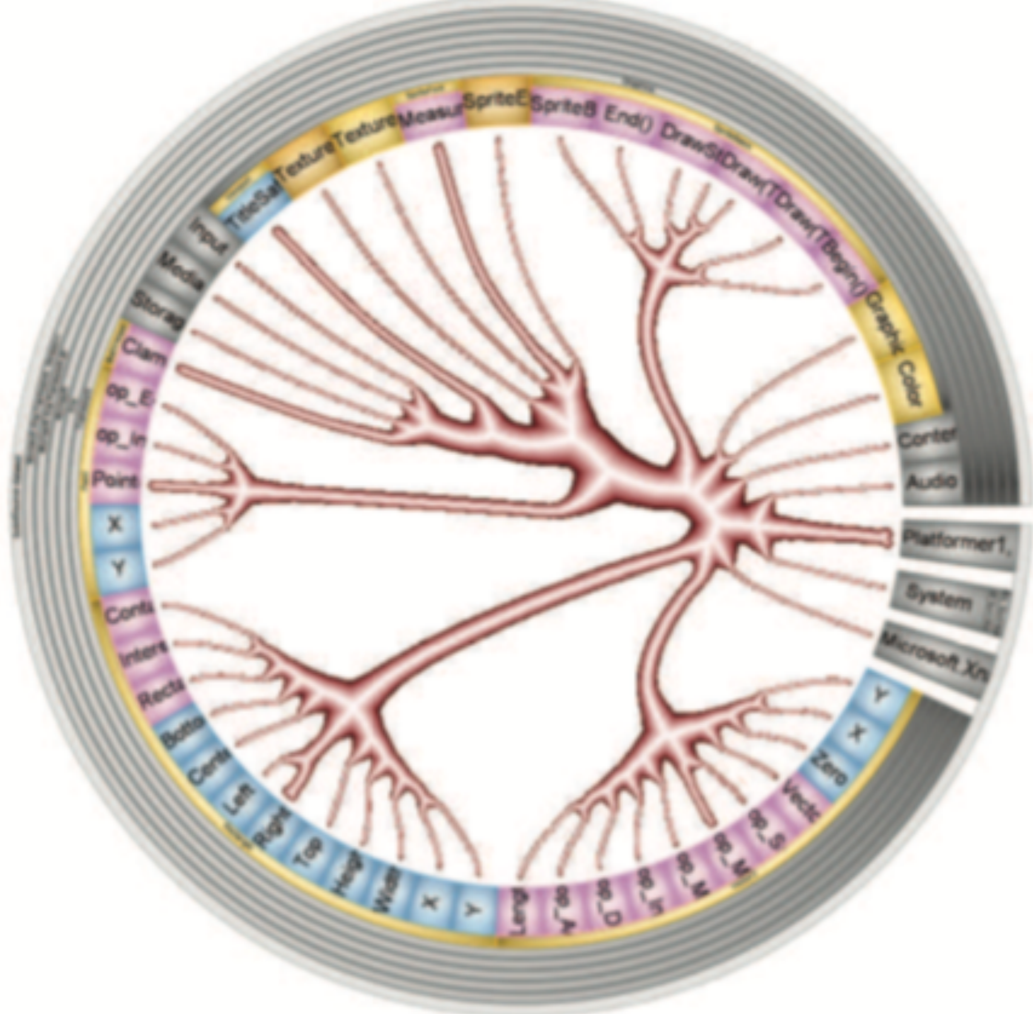
c) height profile H



d) convex shading
(large splat size)



e) convex shading
(small splat size)



f) convex shading
(thin shapes)

TeleaErsoy2010

Different rendering styles

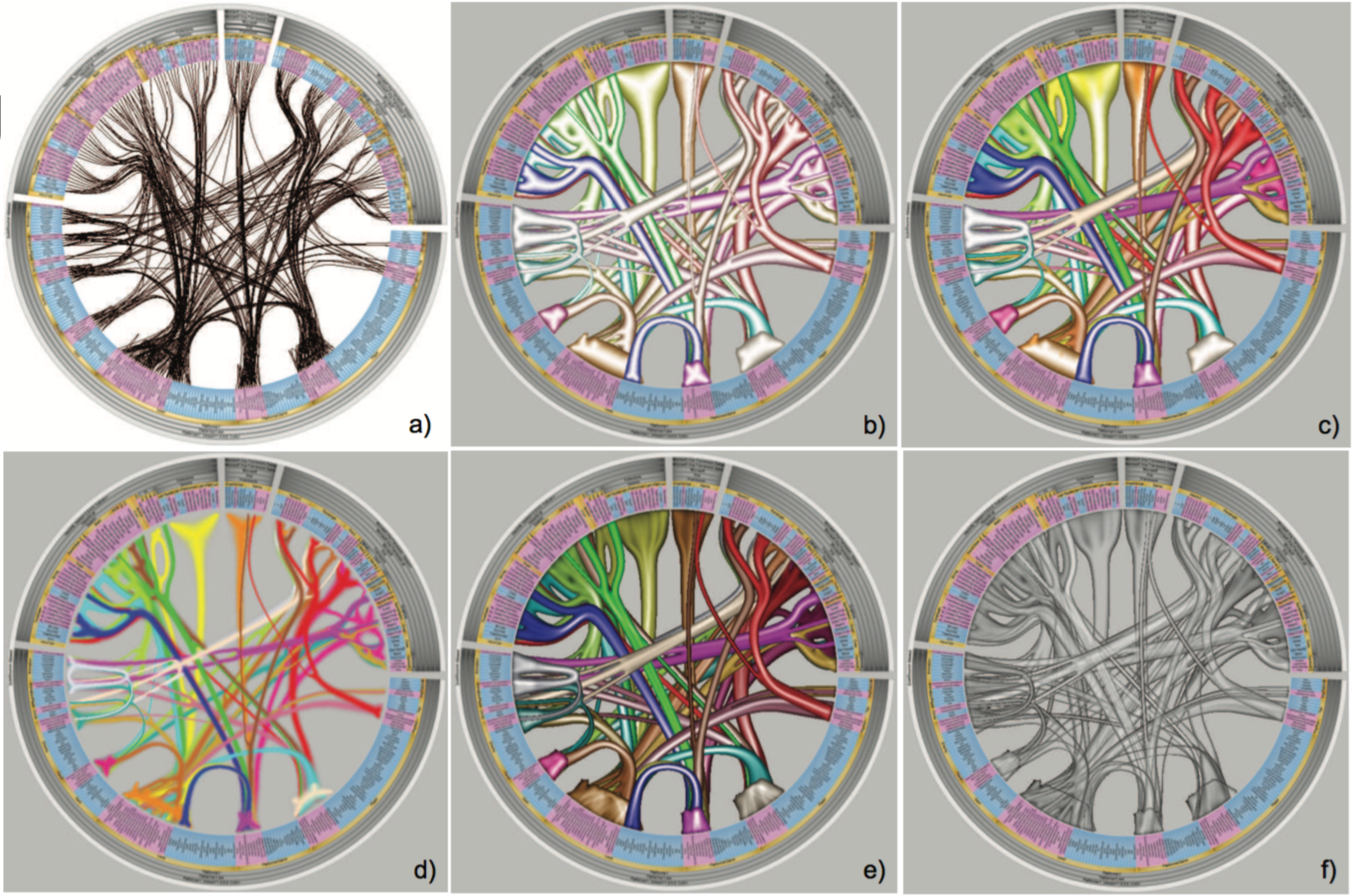


Figure 5: *Rendering styles: convex shapes (b), density-luminance (c), density-saturation (d), bi-level (e), and outlines (f).*

TeleaErsoy2010

Interaction: Digging Lens

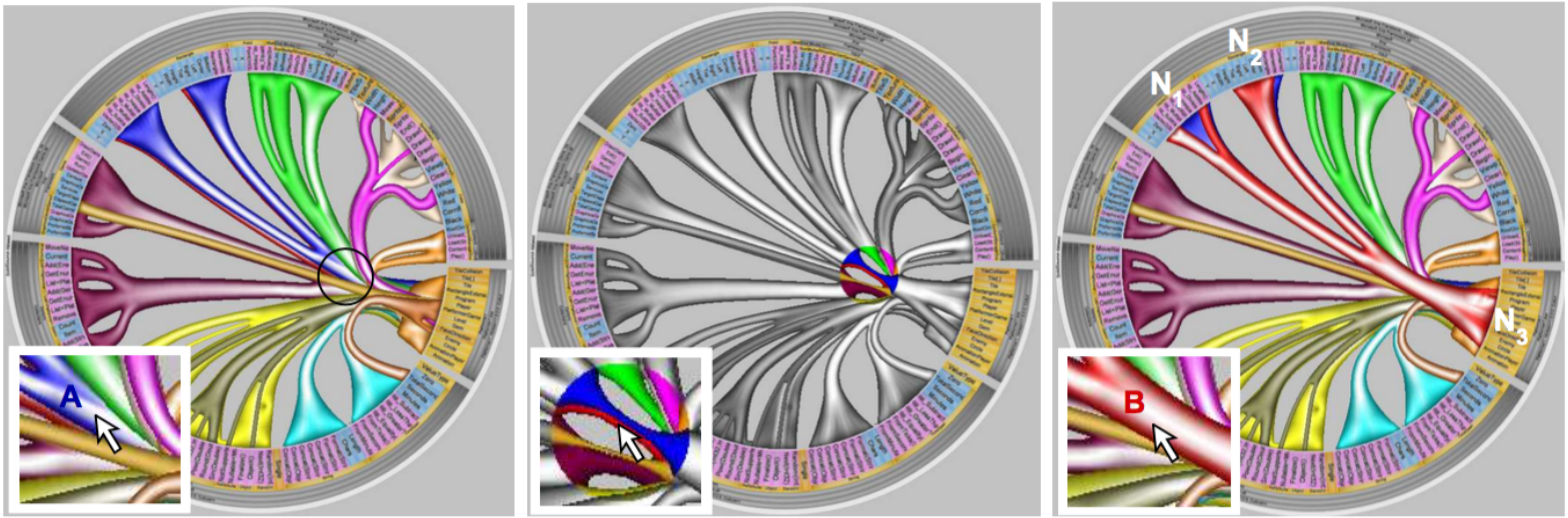
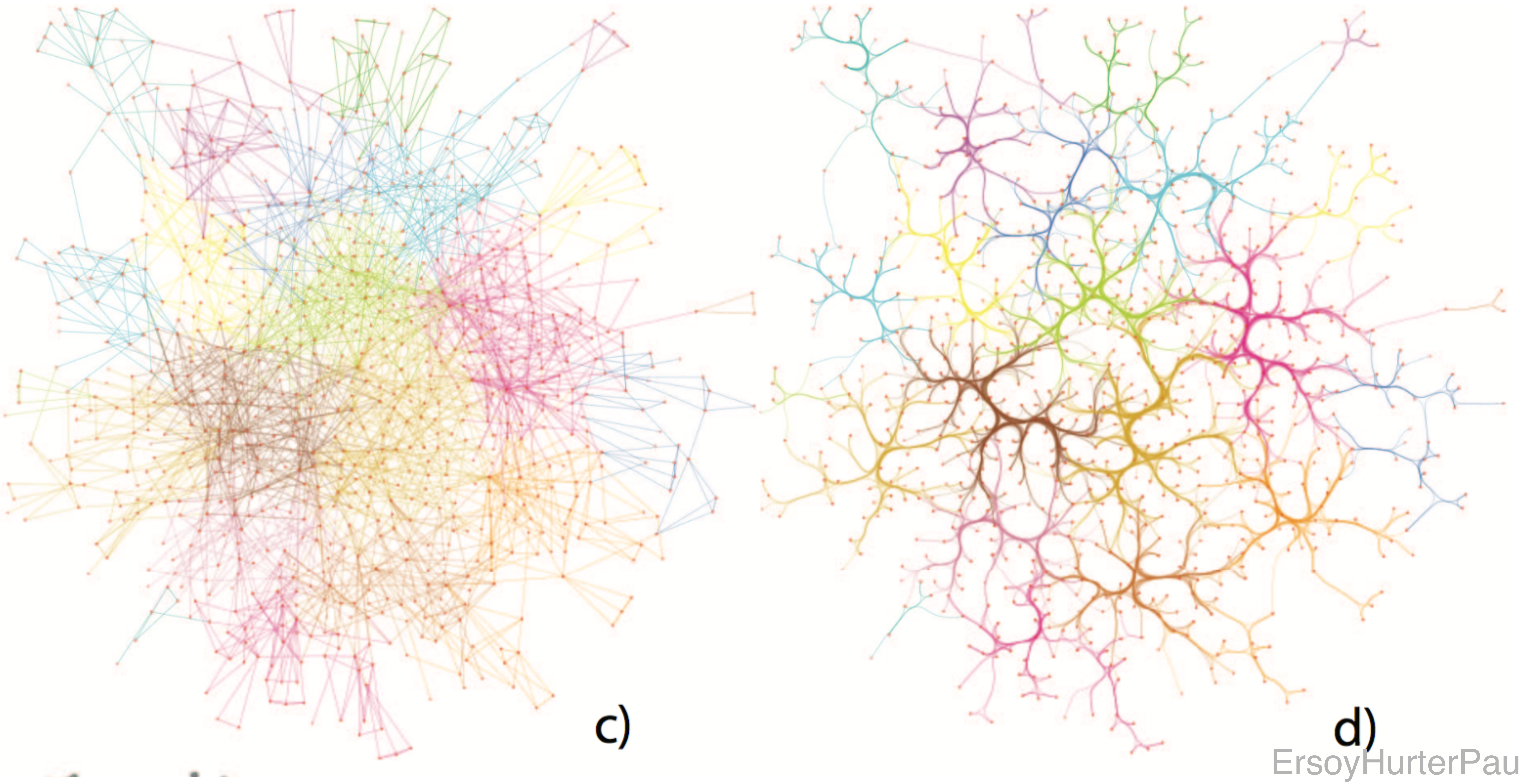


Figure 7: *The digging lens is used to interactively explore areas where shapes overlap. Insets show zoomed-in details.*

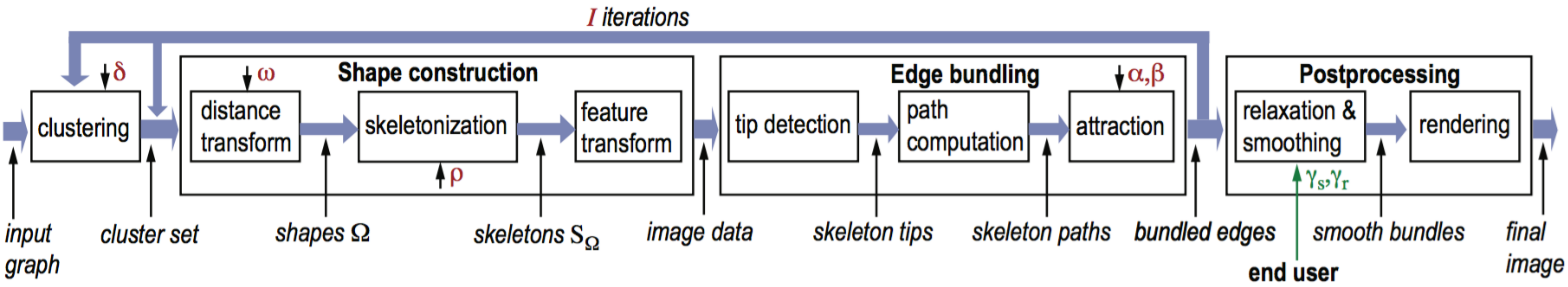
TeleaErsoy2010

Skeleton-Based Edge Bundling

Skeleton-Based Edge Bundling



ErsoyHurterPaulovich2011



1. we *cluster* edges into groups, or clusters, C_i which have strong geometrical and optionally attribute-based similarity;

2. for each cluster C, we compute a thin shape Ω surrounding its edges using a distance-based method;

3. for each shape Ω, we compute its skeleton S_Ω and feature transform of the skeleton FT_S;

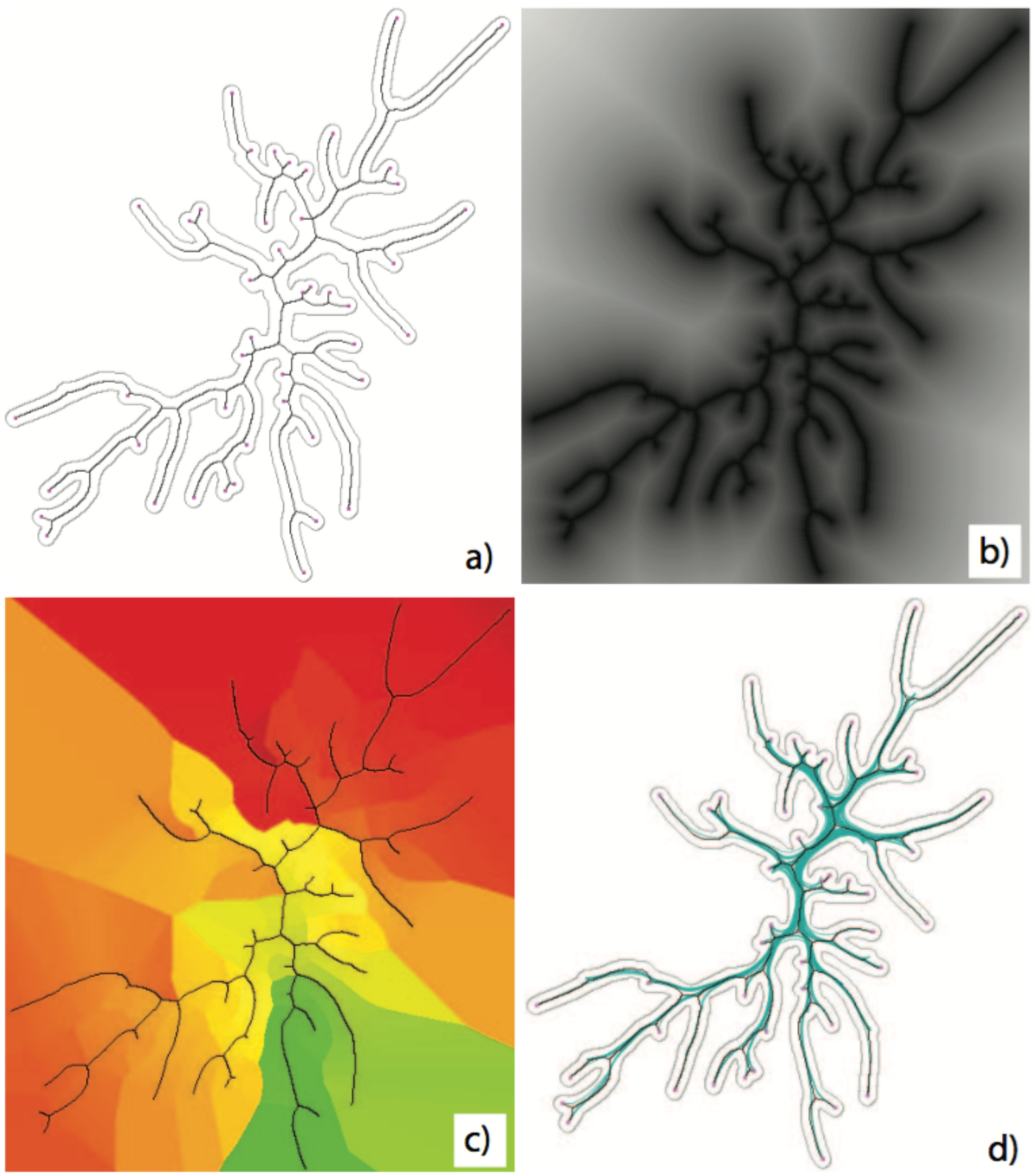
4. for each cluster C, we attract its edges towards S_Ω using FT_S;

5. we repeat the process from step 1 or step 2 until the desired bundling level is reached;

6. we perform a final smoothing and next render the graph using a cushion-like technique to help understanding bundle overlaps.

ErsoyHurterPaulovich2011

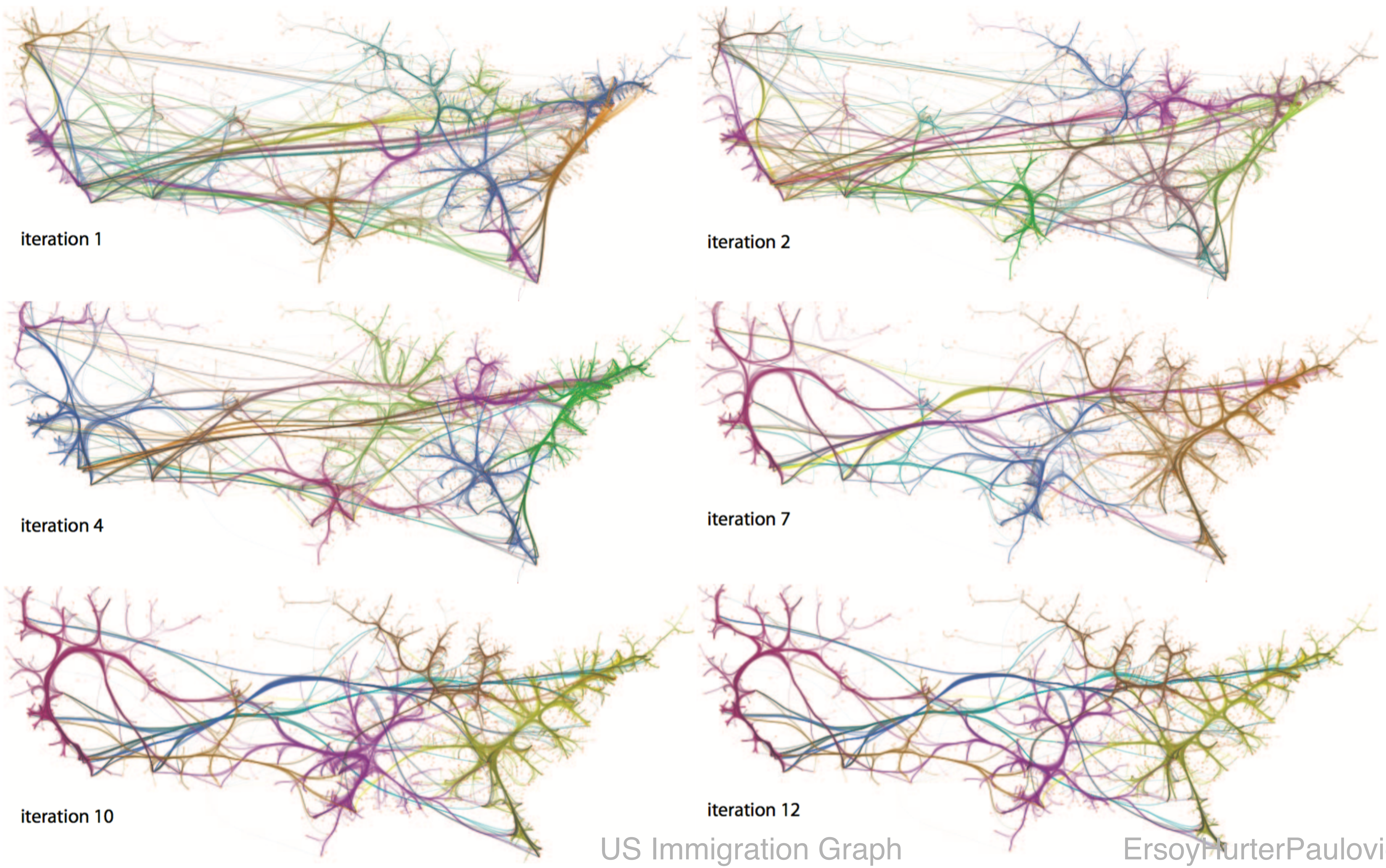
Shape Construction



ErsoyHurterPaulovich2011

Relaxation and Smoothing

ErsoyHurterPaulovich2011



US Immigration Graph ErsoyHurterPaulovich2011

Comparison

Type	Technique	Computation	Advantages	Disadvantages
Flow Map Layout	[PXY*05]	low time complexity	1. intuitive 2. fast computation	1. not clear how to extend their method to general graphs [CZQ*08] 2. edge splits are binary [Hol06]
	[VBS11]	good computational costs	1. crossing-free [VBS11] 2. automated	limited usages to apply to general graphs
Hierarchical Edge Bundles	[Hol06]	good computational costs	1. significantly reduces visual clutter 2. suitable for software analysis	can only work on hierarchical structures
Geometry-Based Edge Bundling	[CZQ*08]	good computational costs	provides a clear visual pattern of densely bundled edges [LLCM12]	1. highly relies on the quality of the control meshes [LLCM12] 2. edges might create high curving variations [HvW09]
Force-Directed Edge Bundling	[HvW09]	high computational costs	able to be used on general graphs	1. difficult to add interactions because of high time complexity 2. does not effectively show the semantic properties of nodes and edges [KS10]
	[SHH11]	high computational costs	shows the direction, graph connectivity and weights in the bundled edges [SHH11]	difficult to add interactions because of high time complexity
Image-Based Edge Bundling	[TE10]	high computational costs	expresses coarse-scale structures [TE10]	limited by the resolution of the intermediate image [BSD13]
	[EHP*11]	massively accelerates bundling with GPU	1. Increased bundling speed 2. emphasis on structure of the bundled layout [EHP*11]	needs some works on bundle crossing minimization and node-edge overlap reduction [EHP*11]

[Zhou2017]

Graph-Theoretic Measures

Brain Connectivity Toolbox

https://sites.google.com/site/bctnet/measures/list

Incomplete list of measures

1 Degree and Similarity
2 Density and Rentian Scaling
3 Clustering and Community Structure
4 Assortativity and Core Structure
5 Paths and Distances
6 Efficiency and Diffusion
7 Centrality
8 Motifs

https://sites.google.com/site/bctnet/measures/list Brain Connectivity Toolbox

Modularity

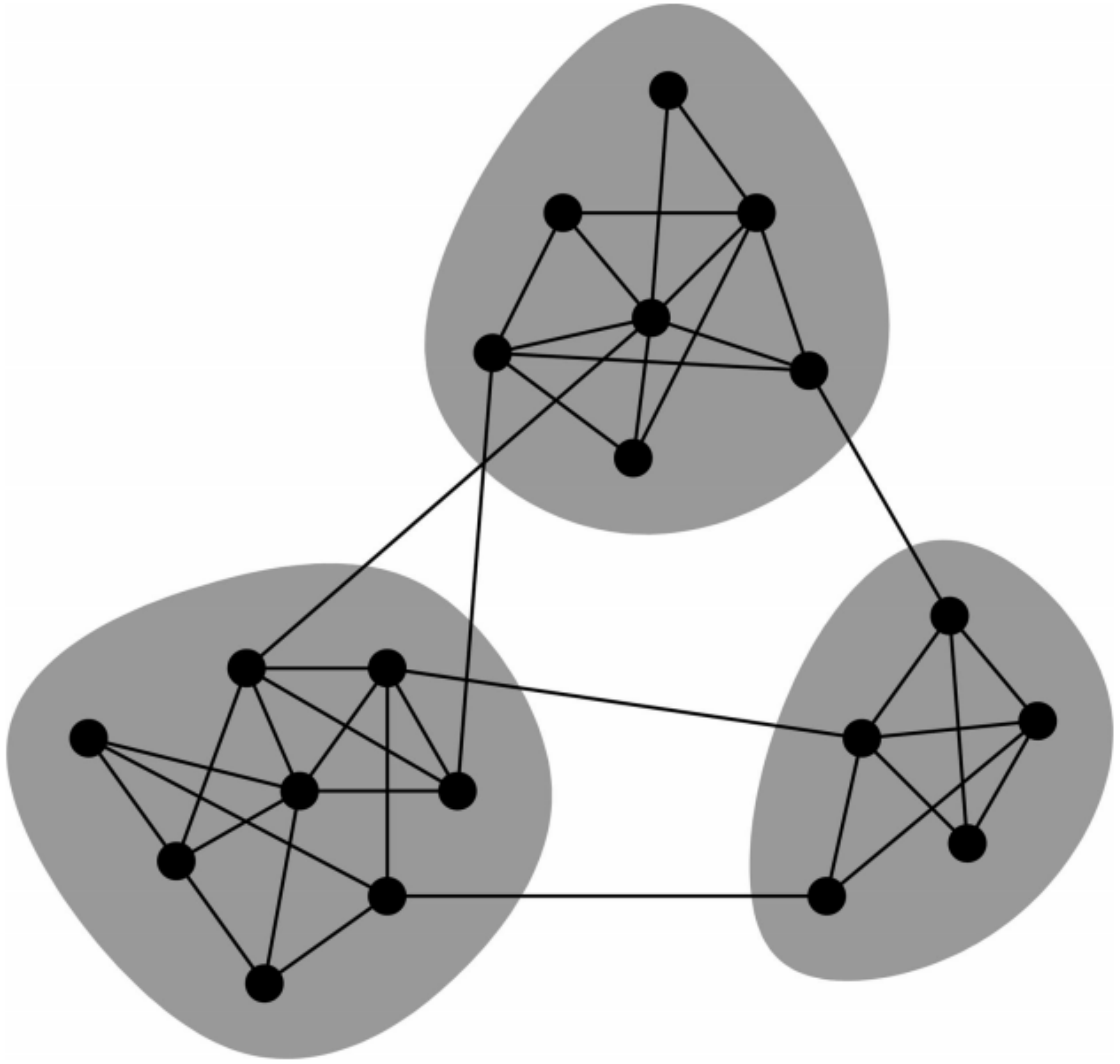


Fig. 1. The vertices in many networks fall naturally into groups or communities, sets of vertices (shaded) within which there are many edges, with only a smaller number of edges between vertices of different groups.

https://en.wikipedia.org/wiki/Modularity_(networks)

Newman2006

Modularity

$$Q = \frac{1}{4m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)(s_i s_j + 1) = \frac{1}{4m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)s_i s_j, \quad [1]$$

$$Q = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s}, \quad [2]$$

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}, \quad [3]$$

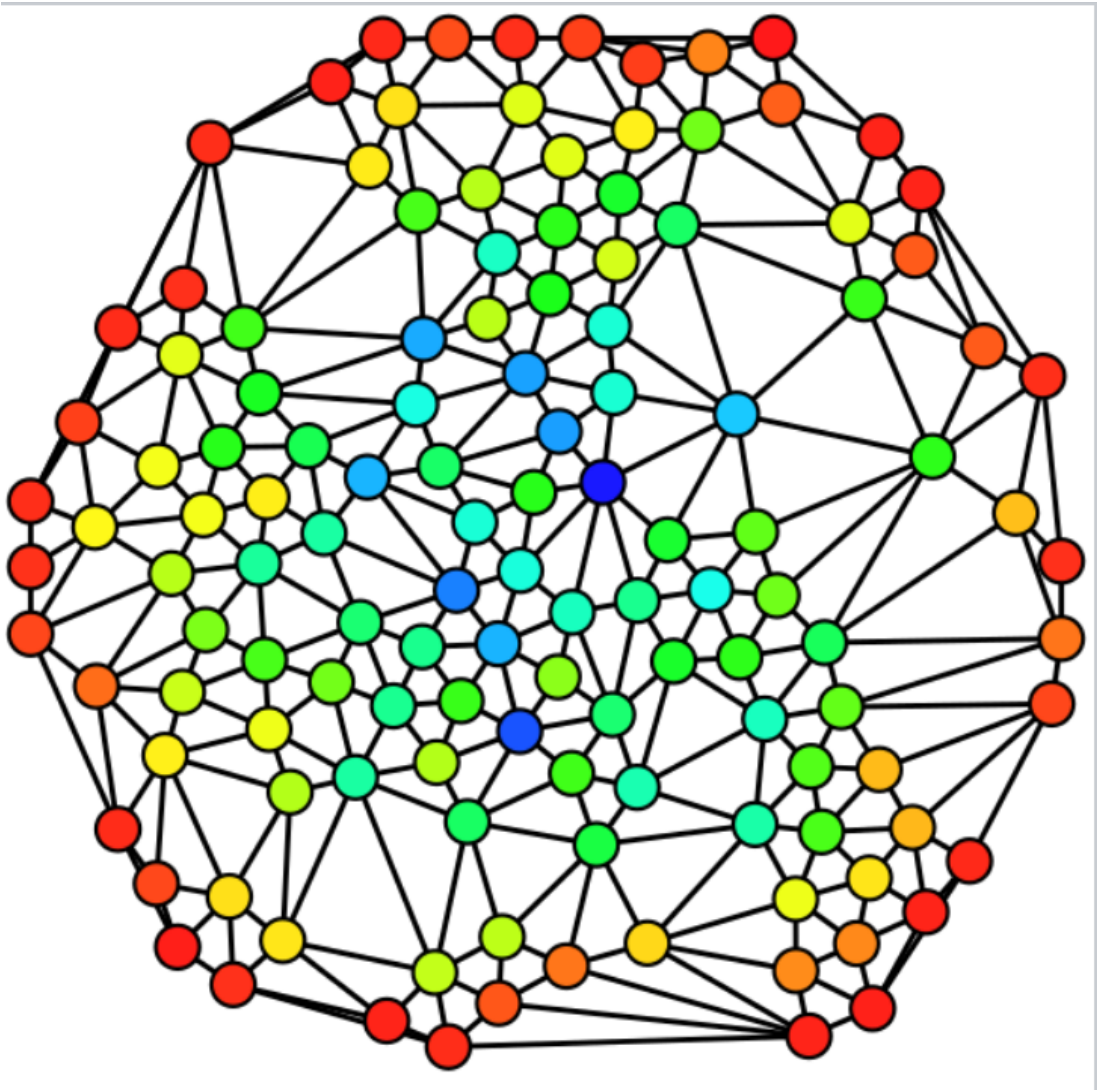
Newman2006

Modularity

- Network contains n vertices.
- For a division of the network into two groups:
  - si = 1 if vertex i belongs to group 1
  - si =-1 if it belongs to group 2
- Aij: adjacency matrix, number of edges between vertices i and j (0 or 1).
- ki, kj: degree of vertices i and j.
- ki*kj/2m: The expected number of edges between vertices i and j if
- edges are placed at random.
- m: total number of edges.
- Q: sum of Aij - ki*kj/2m over all pairs of vertices i,j that fall in the same group.

Newman2006

Betweenness Centrality



https://en.wikipedia.org/wiki/Betweenness_centrality

Betweenness Centrality

The betweenness centrality of a node v is given by the expression:

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths from node s to node t and $\sigma_{st}(v)$ is the number of those paths that pass through v.

https://en.wikipedia.org/wiki/Betweenness_centrality

Global Clustering Coefficient

$$C = \frac{3 \times \text{number of triangles}}{\text{number of connected triplets of vertices}}$$

https://en.wikipedia.org/wiki/Clustering_coefficient

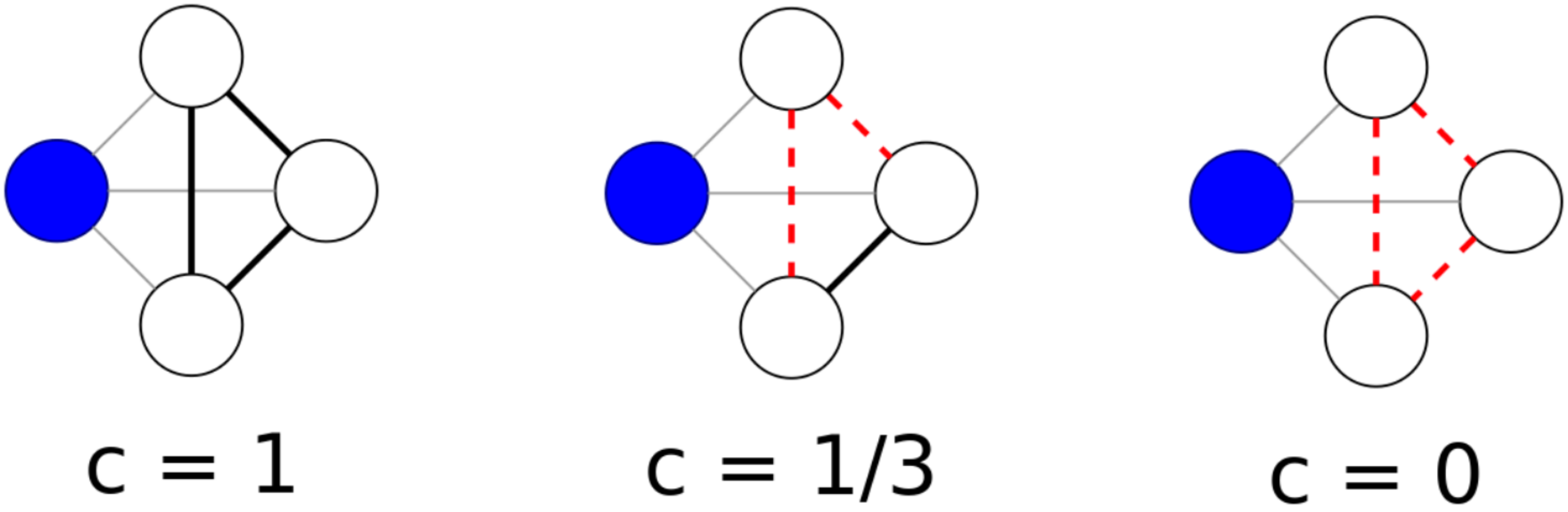
Local Clustering Coefficient

- The local clustering coefficient for a vertex is given by the proportion of links between the vertices within its neighborhood divided by the number of links that could possibly exist between them.

$$C_i = \frac{2|\{e_{jk} : v_j, v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i - 1)}.$$

https://en.wikipedia.org/wiki/Clustering_coefficient

Local Clustering Coefficient



https://en.wikipedia.org/wiki/Clustering_coefficient

State-of-the-Art Network Visualization VIS 2019

[I] A Deep Generative Model for Graph Layout (J)
Authors: Oh-Hyun Kwon, Kwan-Liu Ma
Video Preview | VIS 2019 Talk | DOI

[I] DeepDrawing: A Deep Learning Approach to Graph Drawing (J)
Authors: Yong Wang, Zhihua Jin, Qianwen Wang, Weiwei Cui, Tengfei Ma, Huamin Qu
Video Preview | VIS 2019 Talk | DOI

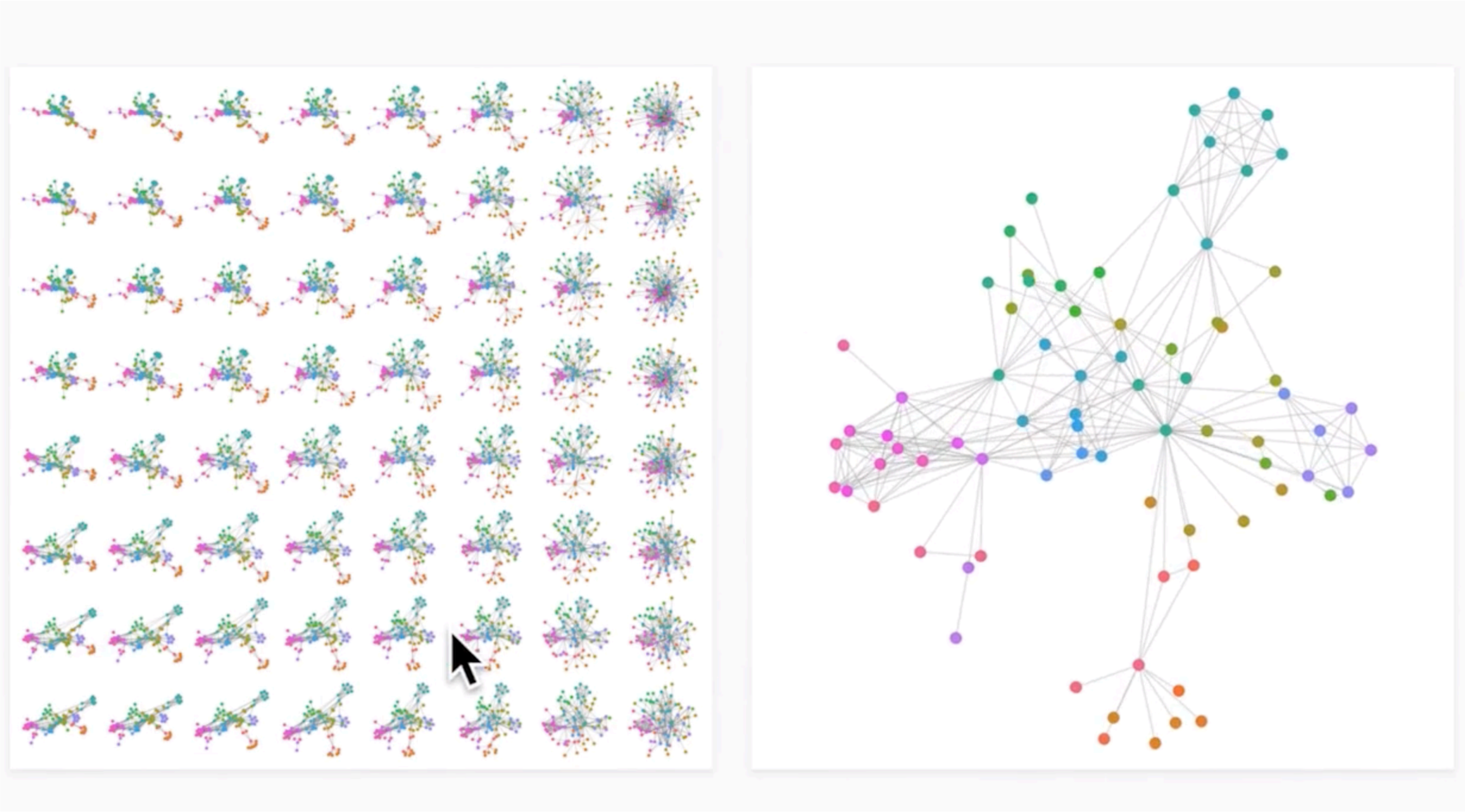
[I] Interactive Structure-aware Blending of Diverse Edge Bundling Visualizations (J)
Authors: Yunhai Wang, Mingliang Xue, Yanyan Wang, Xinyuan Yan, Baoquan Chen, Chi-Wing Fu, Christophe Hurter
Video Preview | VIS 2019 Talk | DOI

[I] Persistent Homology Guided Force-Directed Graph Layouts (J)
Authors: Ashley Suh, Mustafa Hajij, Bei Wang, Carlos Scheidegger, Paul Rosen
Video Preview | VIS 2019 Talk | DOI

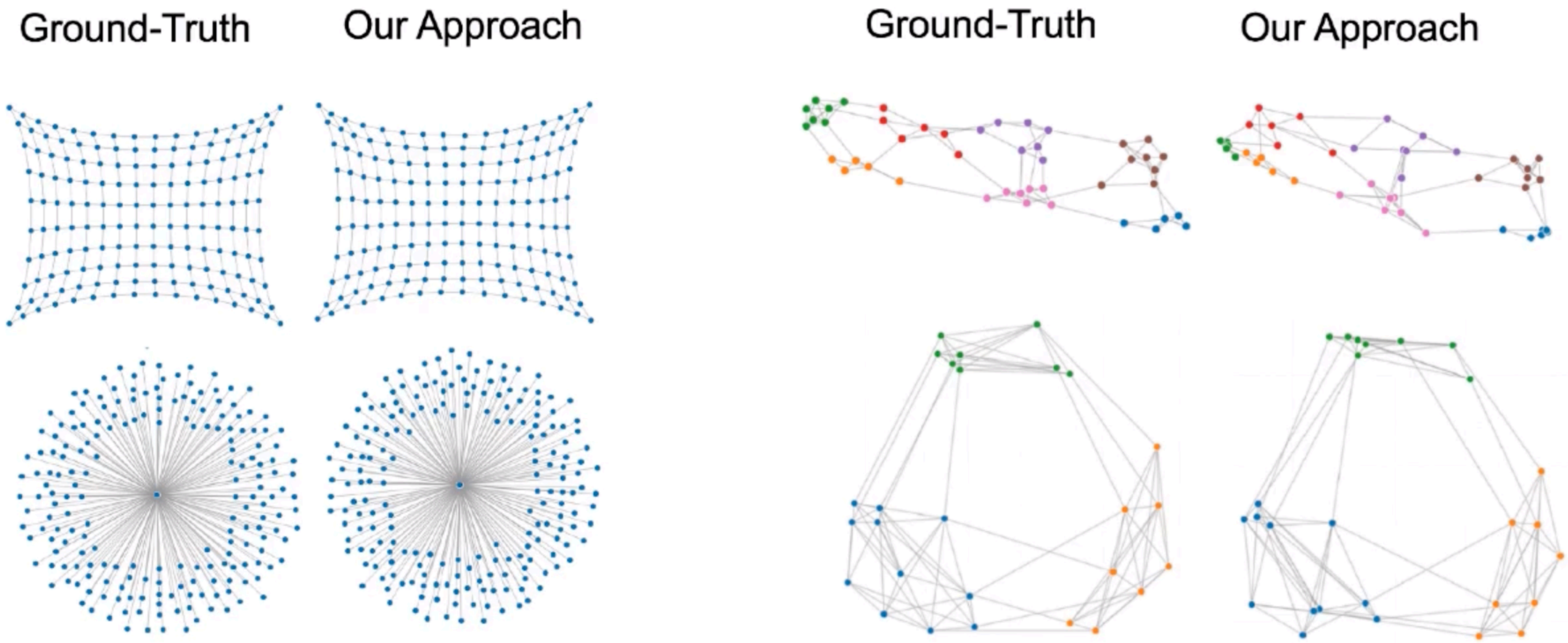
[I] Graph Drawing by Stochastic Gradient Descent (T)
Authors: J X Zheng, D F M Goodman, S Pawar
Video Preview | DOI

[V] The Effect of Edge Bundling and Seriation on Sensemaking of Biclusters in Bipartite Graphs (T)
Authors: Maoyuan Sun, Jian Zhao, Hao Wu, Kurt Luther, Chris North, Naren Ramakrishnan
Video Preview | DOI

http://ieeevis.org/year/2019/info/papers-sessions



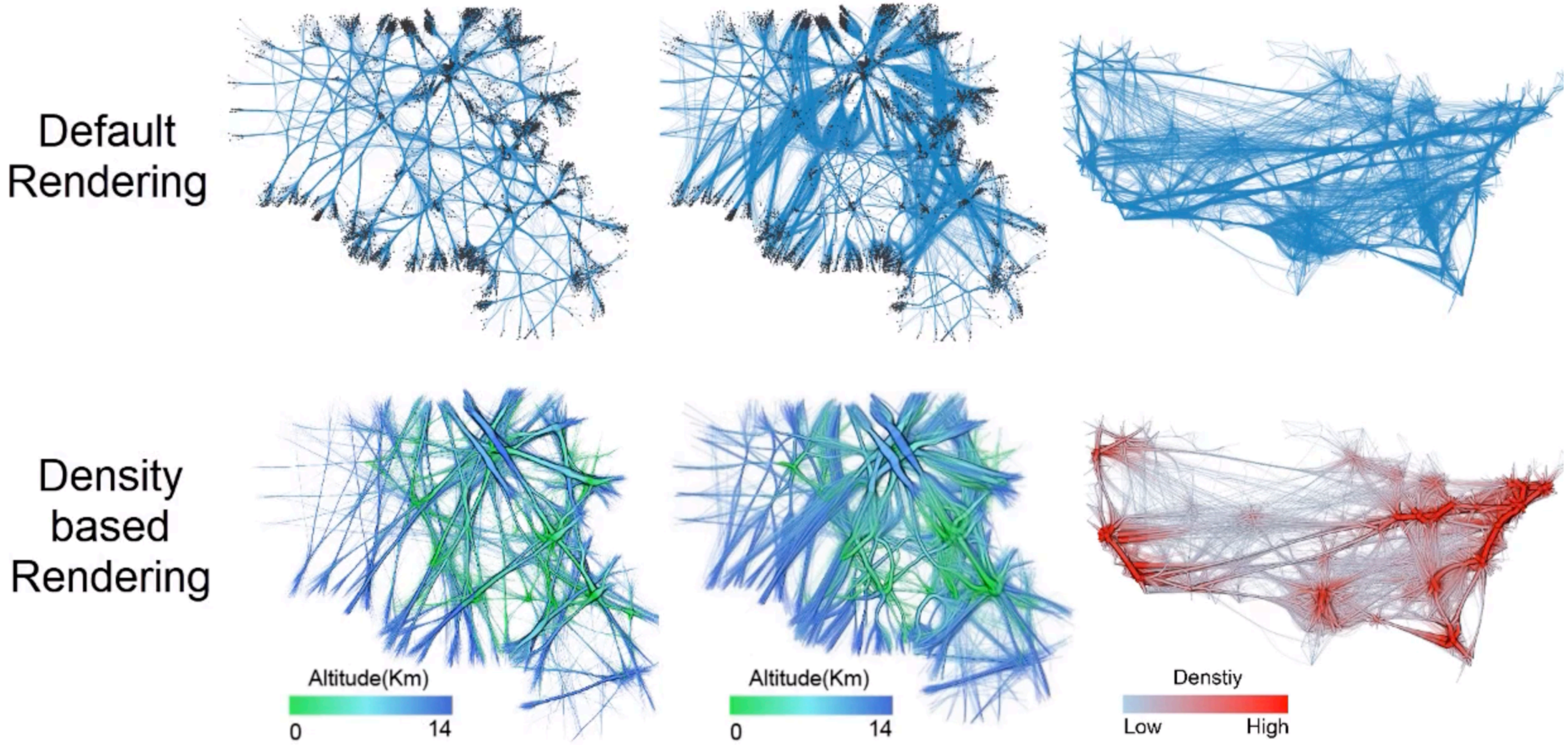
https://vimeo.com/360049688



Example Galleries

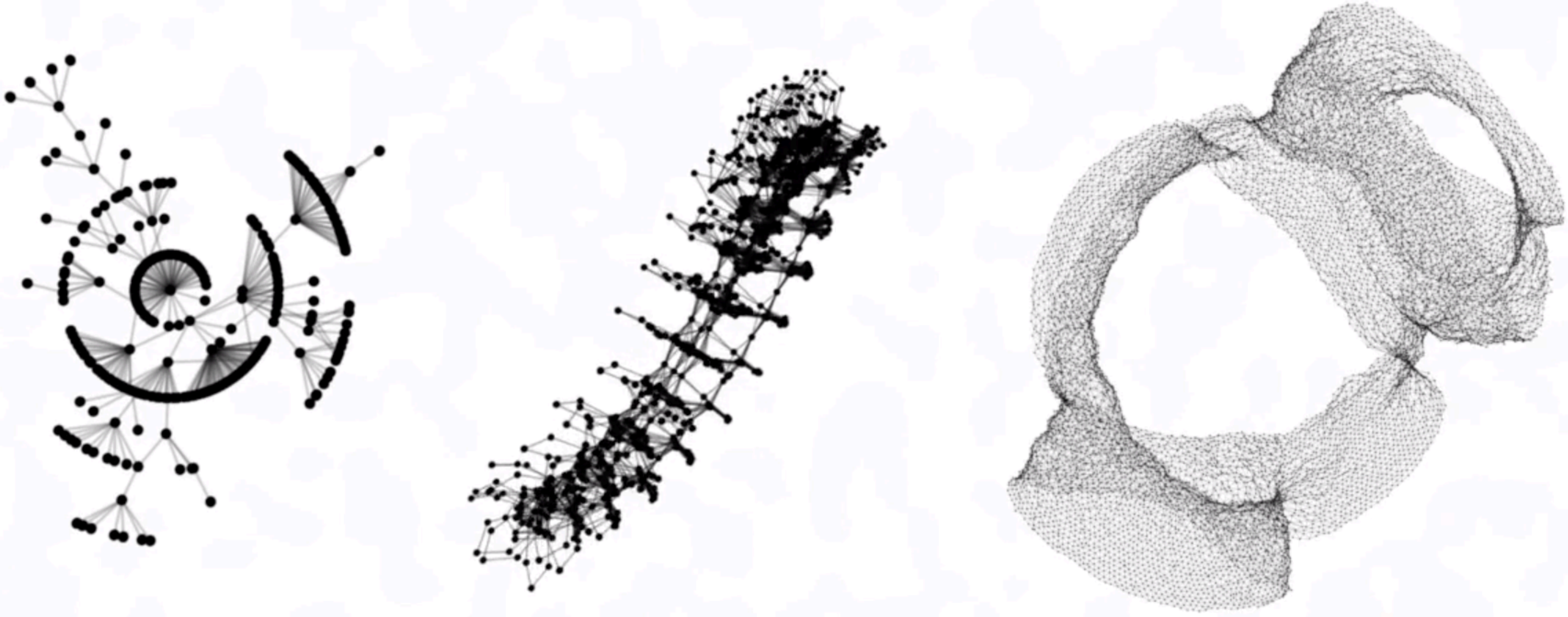
https://vimeo.com/360049919

Rendering styles



https://vimeo.com/360050223

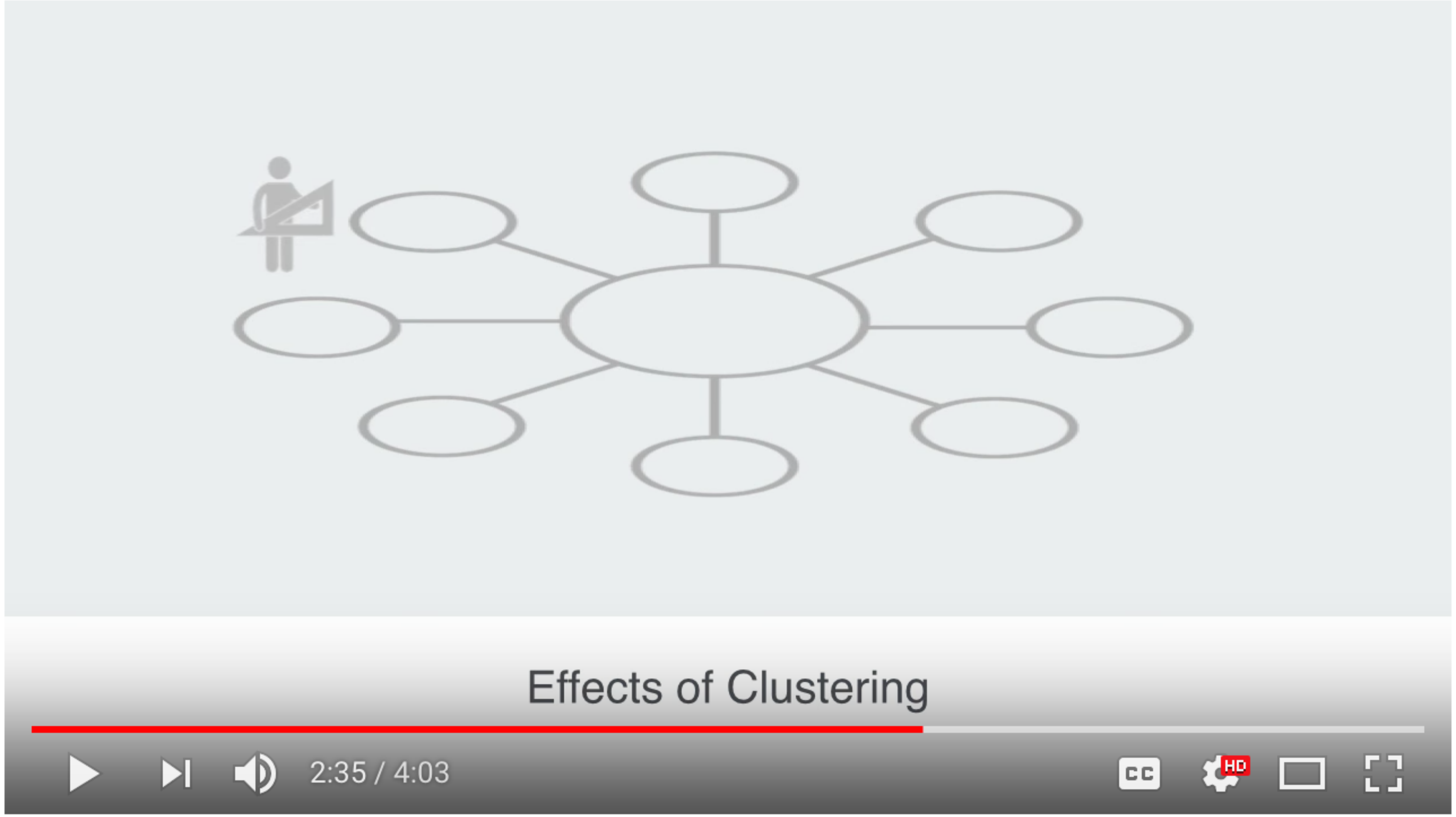
Imperial College London



https://vimeo.com/364569741

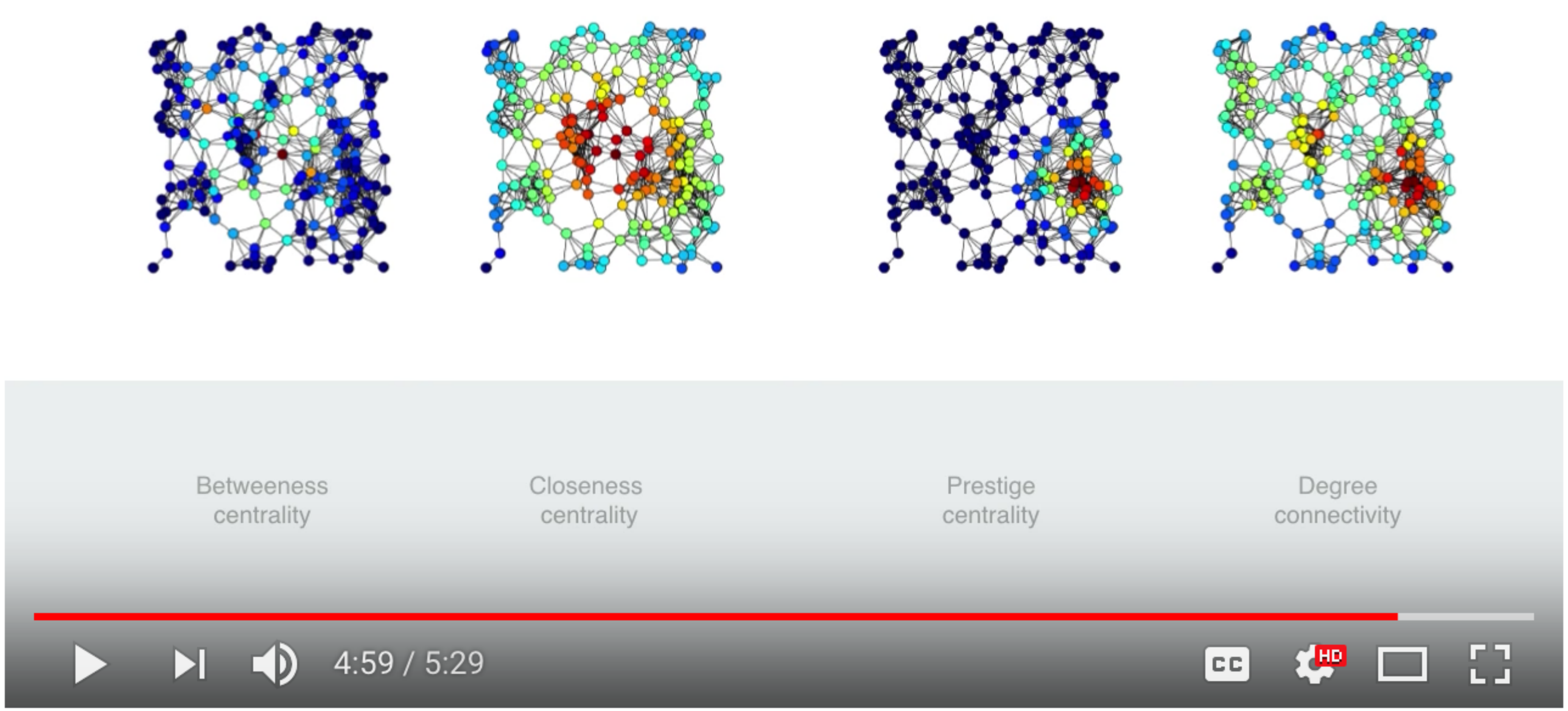
Additional Materials

Clustering Coefficient



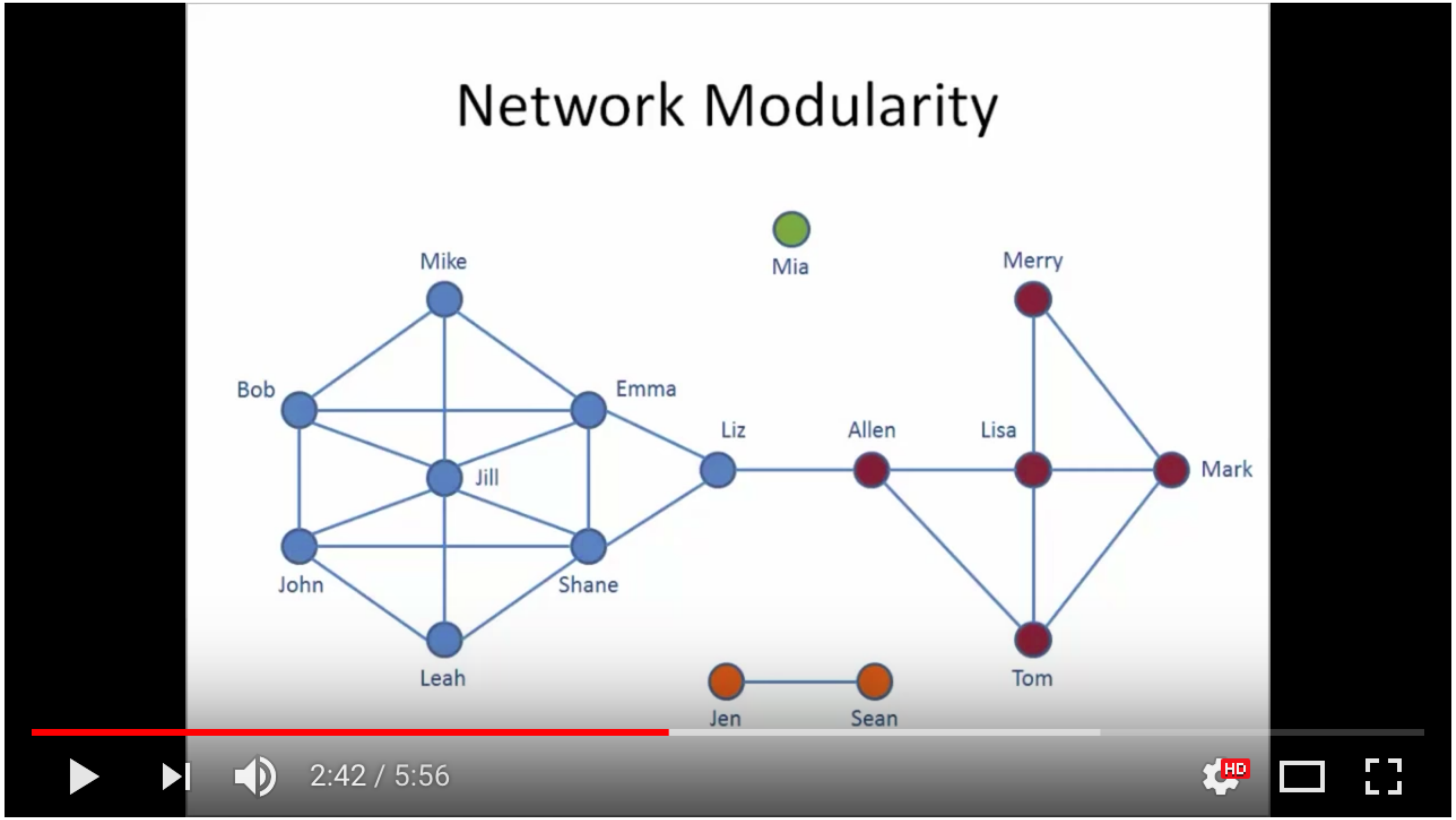
Network Clustering & Connectedness https://www.youtube.com/watch?v=2Oa7mef77nM

Centrality

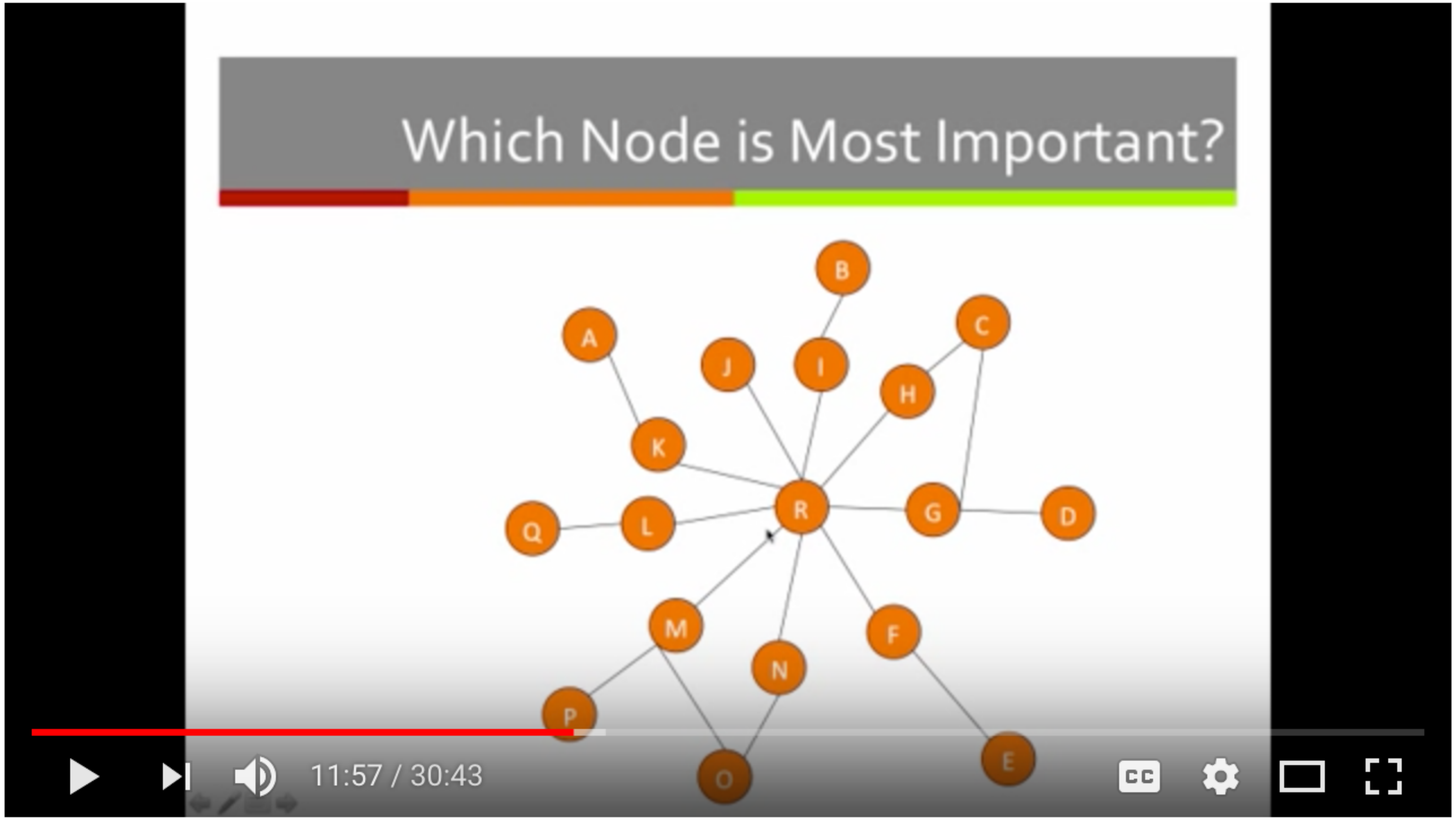


Network Centrality https://www.youtube.com/watch?v=NgUj8DEH5Tc

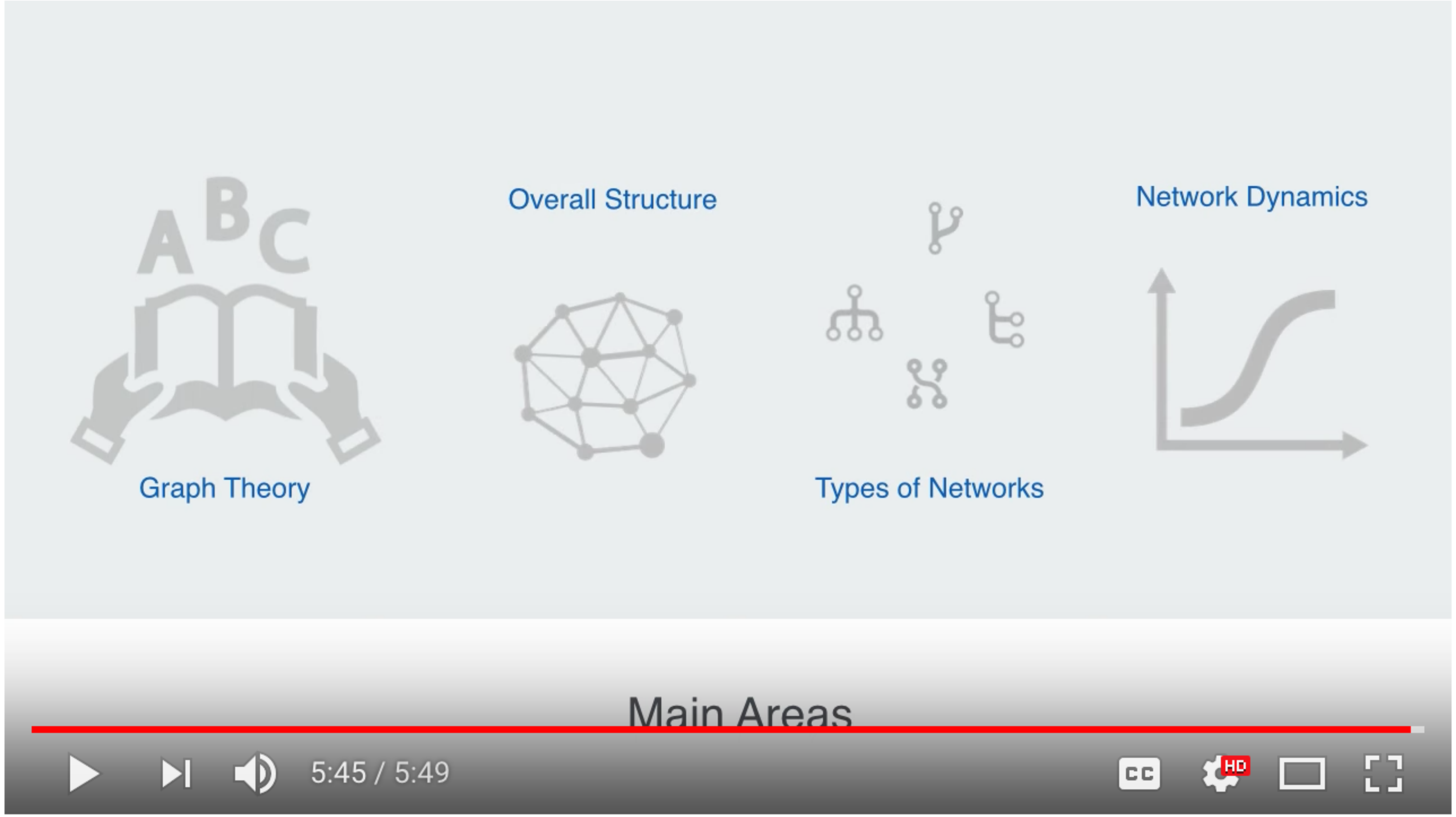
Network Modularity



https://www.youtube.com/watch?v=2_Q7uPAl34M



https://www.youtube.com/watch?v=89mxOdwPfxA



Network Theory Overview

https://www.youtube.com/watch?v=qFcuovfgPTc

https://www.youtube.com/watch?v=7HkXkAZye1Y&list=PLsJWgOB5mIMAuH3cHa-MXukX6-RPpDXgI

Thanks!

Any questions?

You can find me at: beiwang@sci.utah.edu

CREDITS

Special thanks to all people who made and share these awesome resources for free:

- Presentation template designed by Slidesmash
- Photographs by unsplash.com and pexels.com
- Vector Icons by Matthew Skiles

Presentation Design

This presentation uses the following typographies and colors:

Free Fonts used:

http://www.1001fonts.com/oswald-font.html

https://www.fontsquirrel.com/fonts/open-sans

Colors used