Grooming the Hairball - How to Tidy up Network Visualizations?

Hans-Jörg Schulz[1], Christophe Hurter[2]

VIS Tutorial 2013

1. University of Rostock, Rostock, Germany
2. French Civil Aviation University, Toulouse, France

PART II: EDGE SET SIMPLIFICATION

Speaker: Christophe Hurter

Objectives

At the end of the part II, you will be able to explain:

- *Algorithmic techniques* to reduce **edge** clutter
- *Interaction techniques* to reduce **edge** clutter
- More specifically ***Edge** Bundling* (EB) techniques

And you will be able to implement at least two EB techniques (Hierarchical **Edge** Bundling and Kernel Density Estimation **Edge** Bundling)

Outlines

Recall of the Dataflow model

Taxonomy of edge simplification

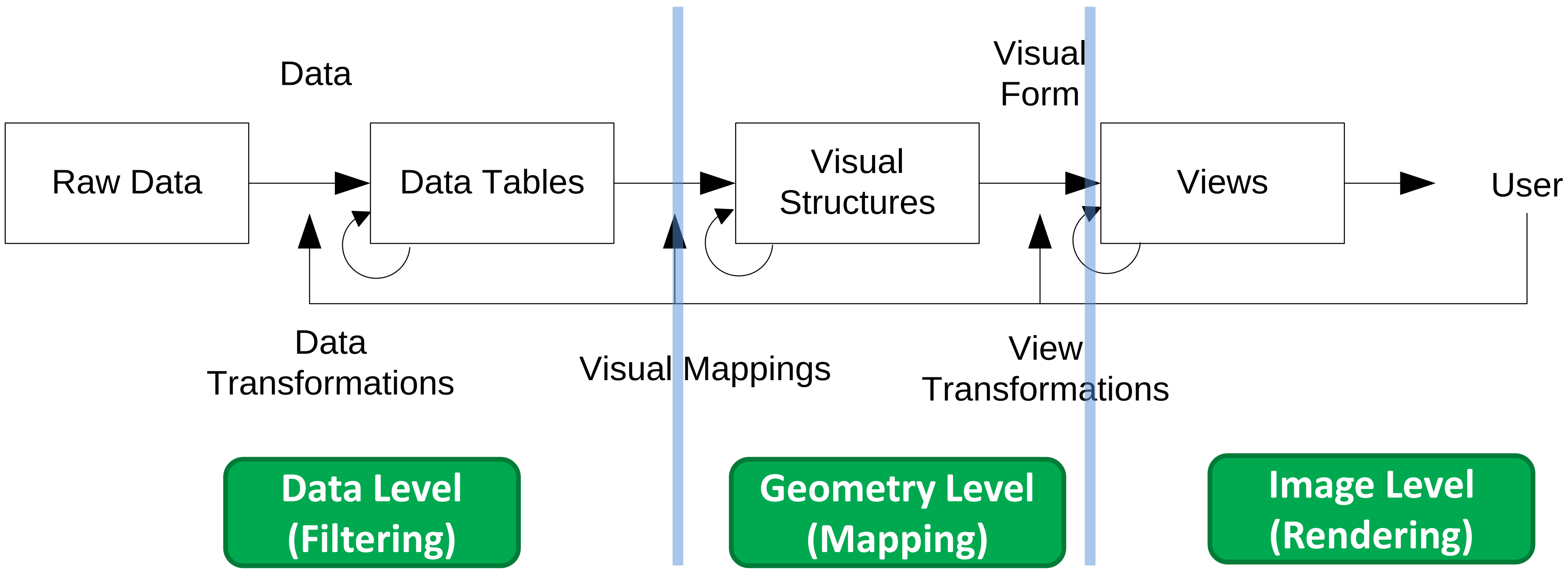
Edge simplification techniques

- Data Level
- Geometry Level
- Image Level

Taxonomy of edge simplification

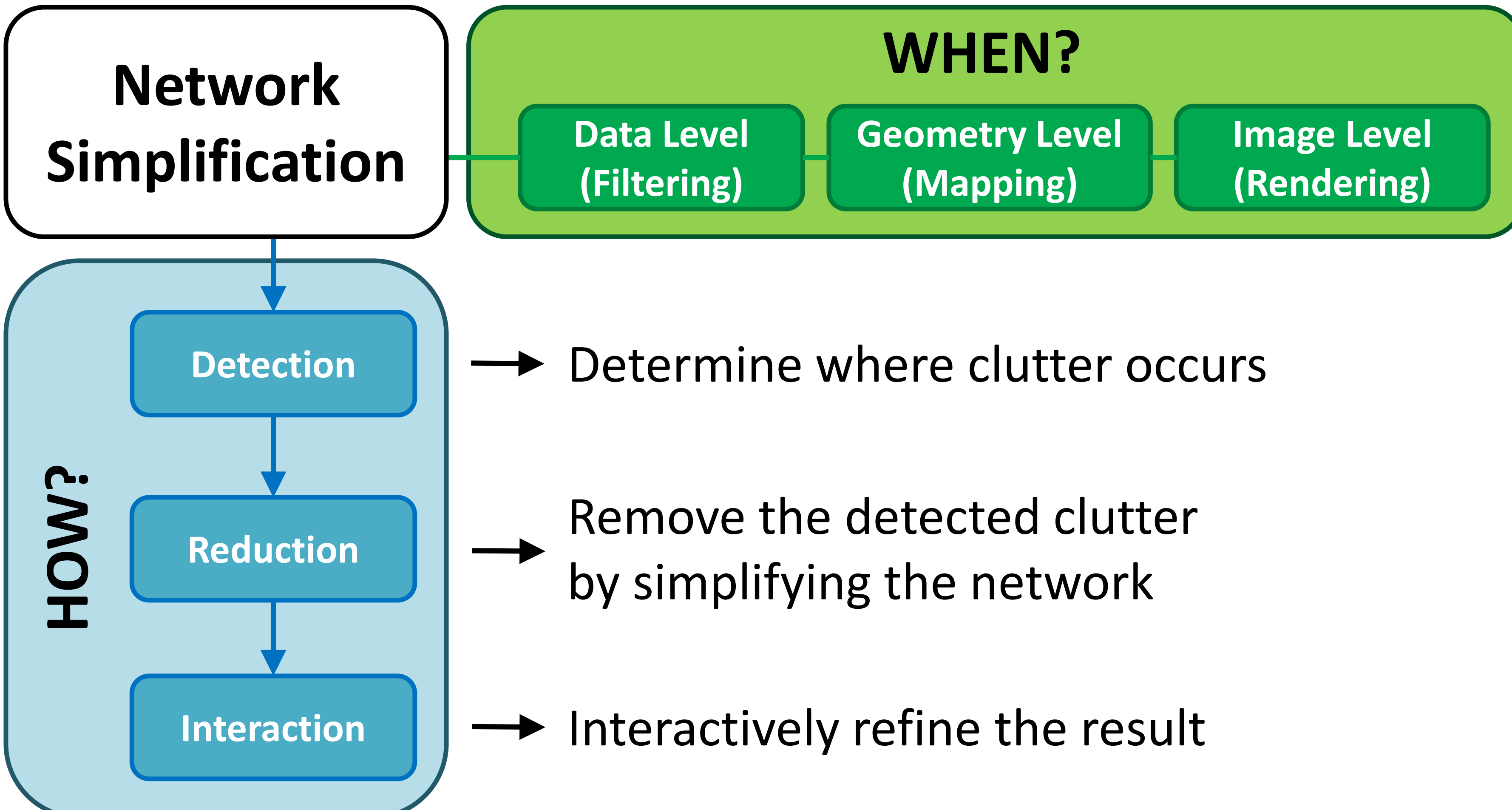
- Give a structure to better understand the design space
- Give criteria to better characterize existing techniques
- Give a tool to explore available technique
  (detection of new unexplored technique)

InfoVis pipeline



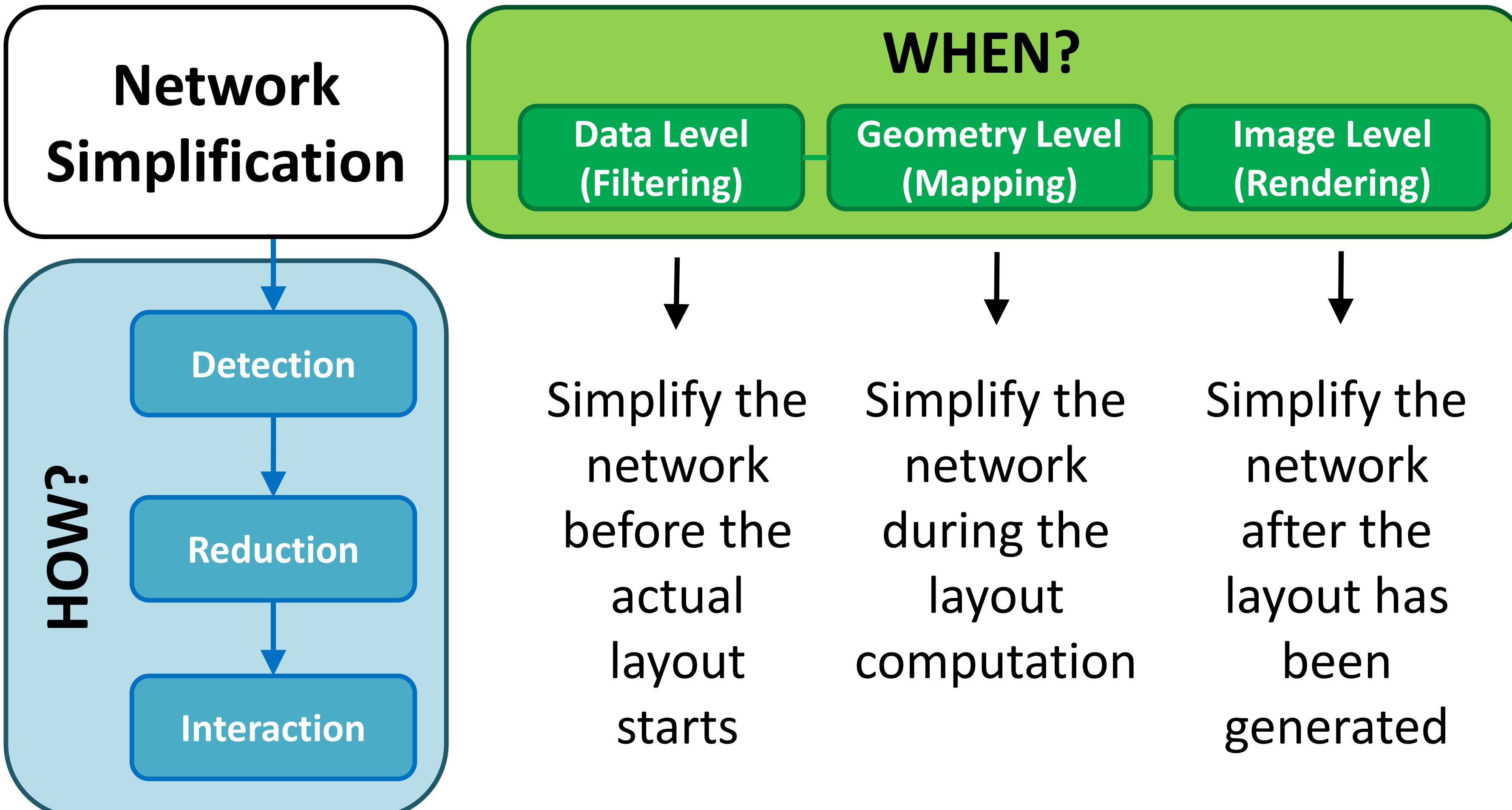
[S.K. Card, J. Mackinlay, B. Shneiderman, 1998]

Recall: taxonomy



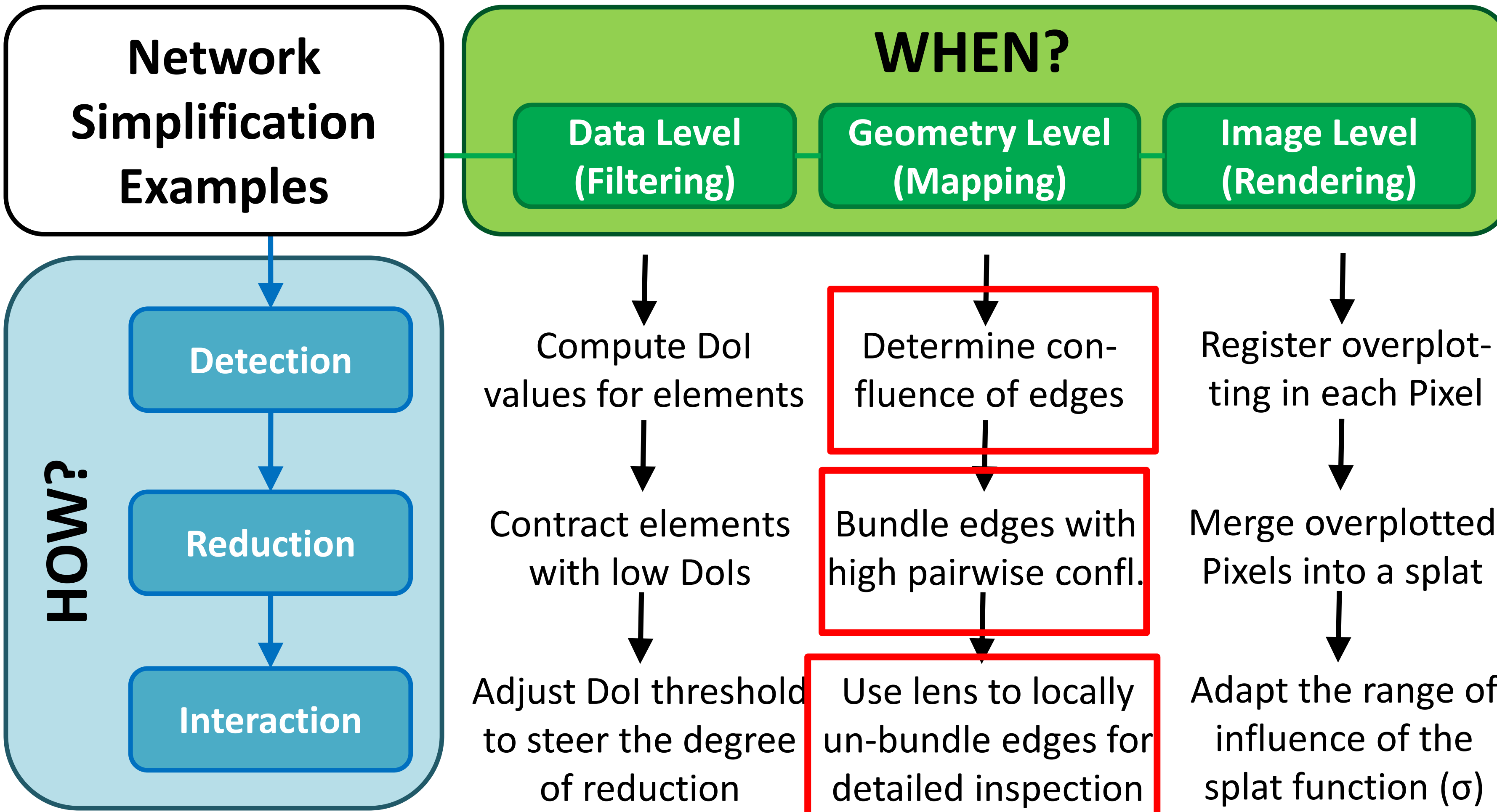
About the Tutorial Topic

A Conceptual Framework to Solve this Problem



About the Tutorial Topic

A Conceptual Framework to Solve this Problem



Instances

Data Level (Filtering)

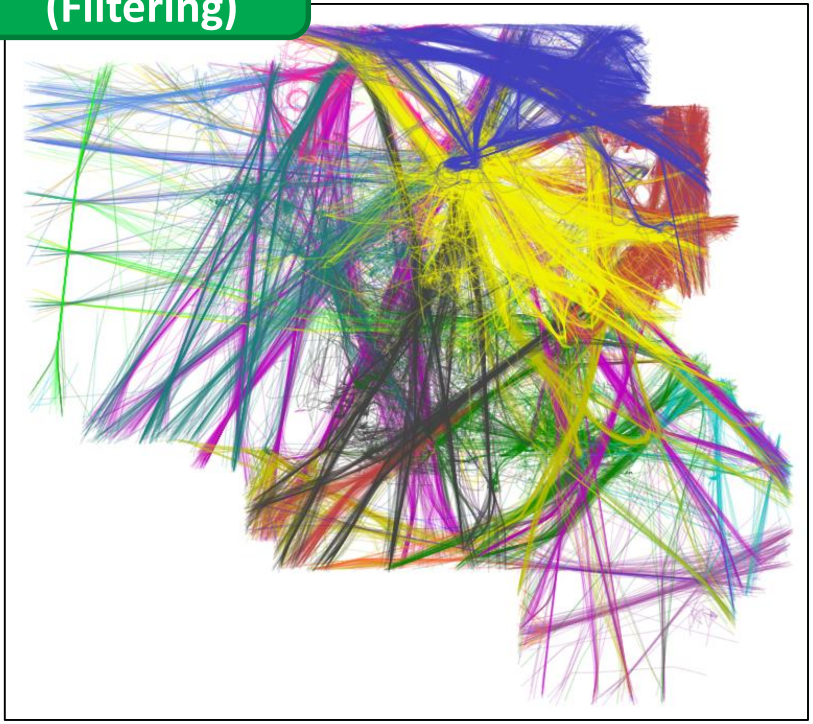
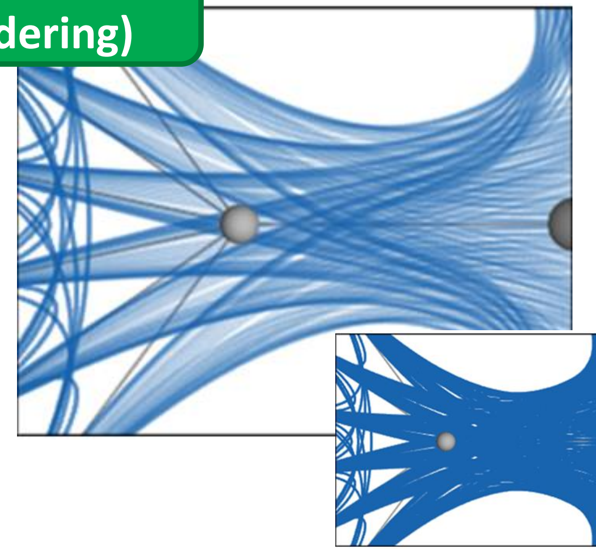
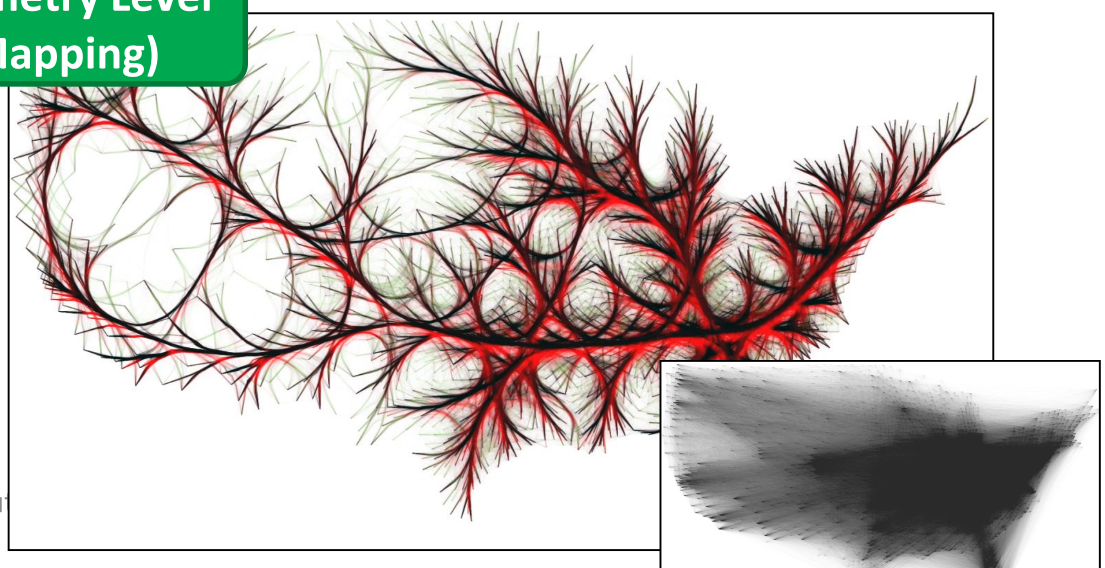


Image Level (Rendering)



Geometry Level (Mapping)



PART II:
EDGE SET SIMPLIFICATION

DATA LEVEL
(FILTERING)

WHEN?

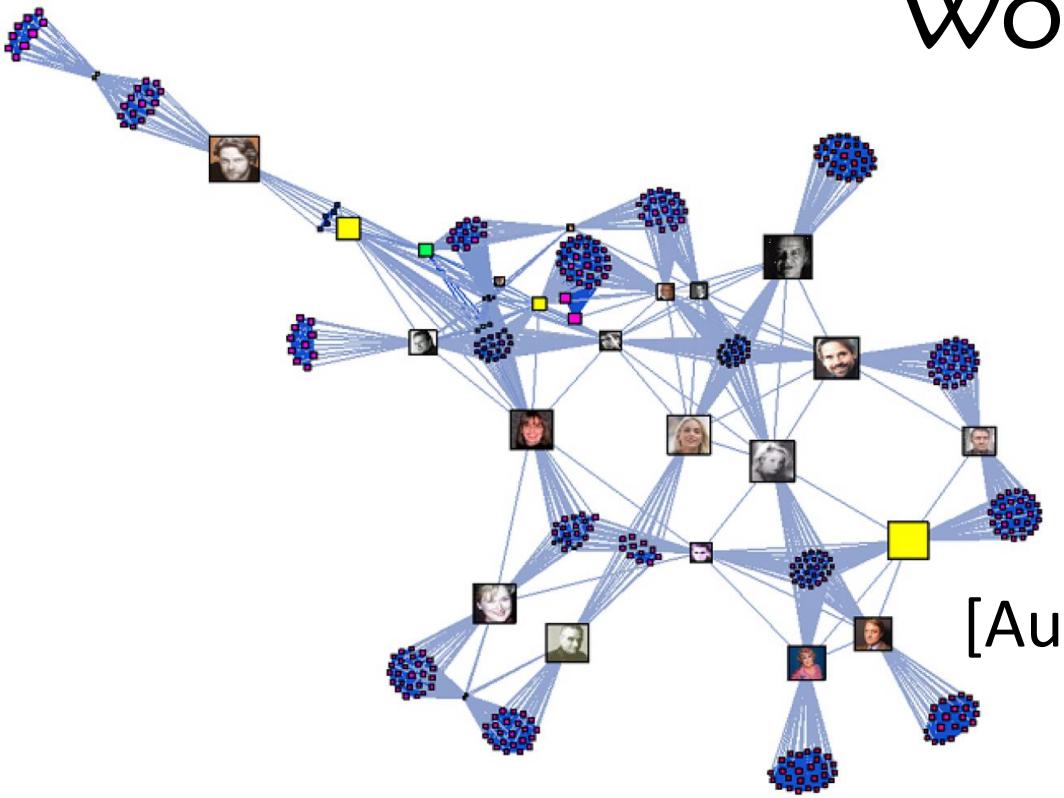
Data Level (Filtering)	Geometry Level (Mapping)	Image Level (Rendering)

Small world issue

The **small-world experiment** comprised several experiments conducted by Stanley Milgram and other researchers examining the average path length for social networks of people in the United States.

[Watts 1999]

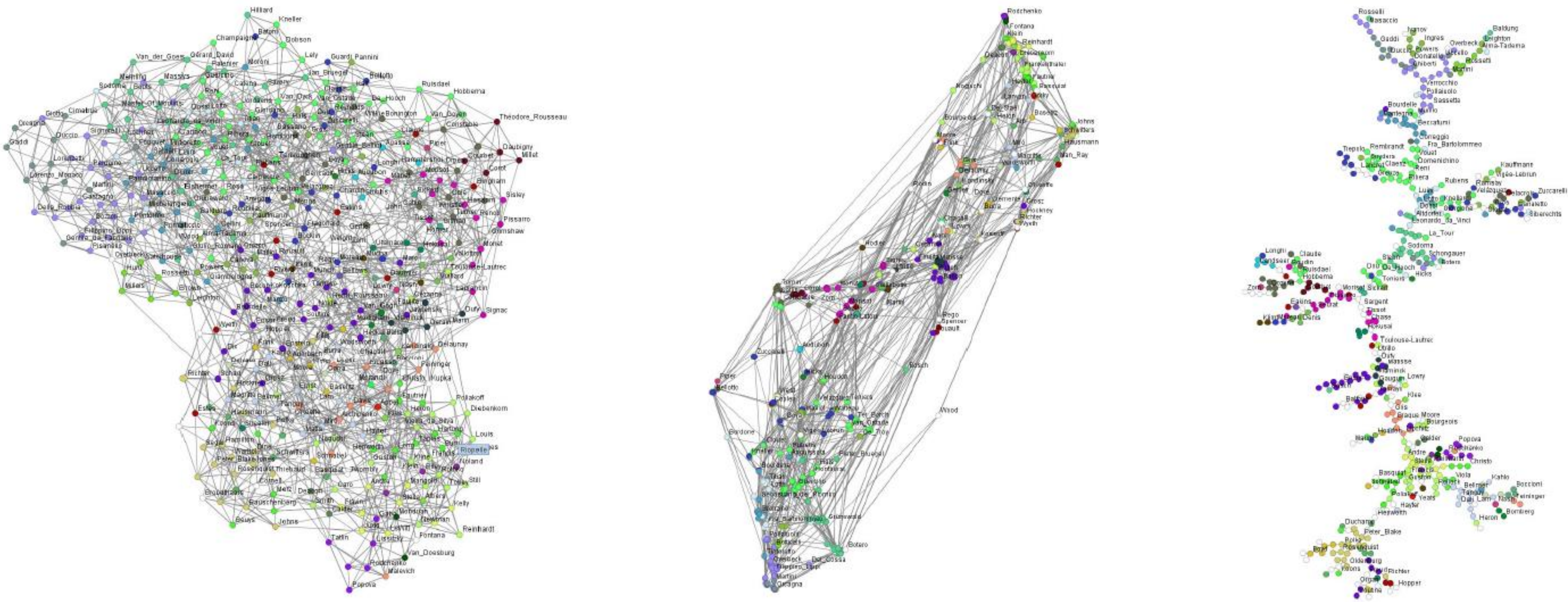
Multiscale Visualisation of Small World Networks



[Auber et al. Infovis 2003]

[Van Ham and Van Wijk, Infovis 2004] -> nodes simplification

Centrality Based Visualization of Small World Graphs

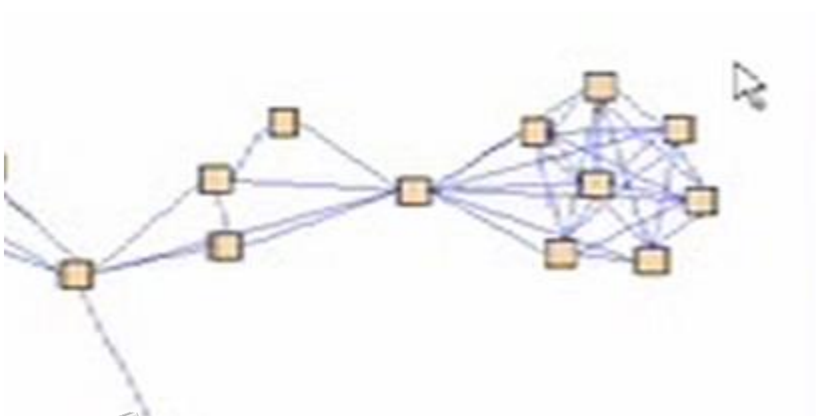


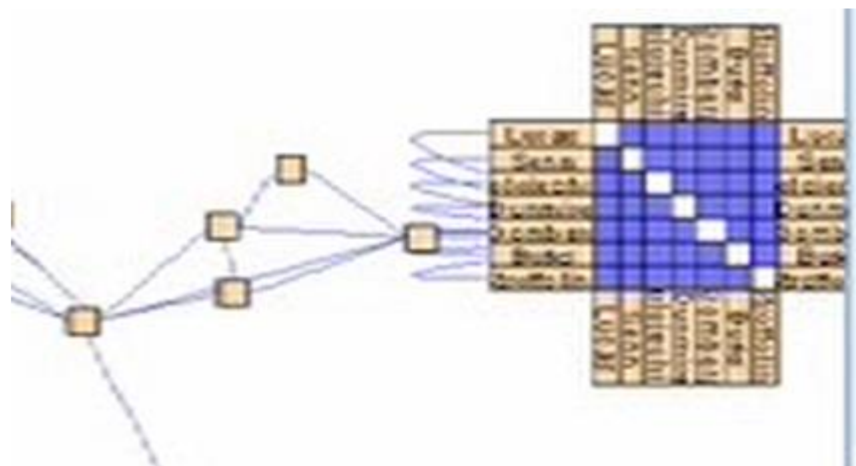
(a) (b) (c)

[van Ham and Wattenberg , Eurovis 08]

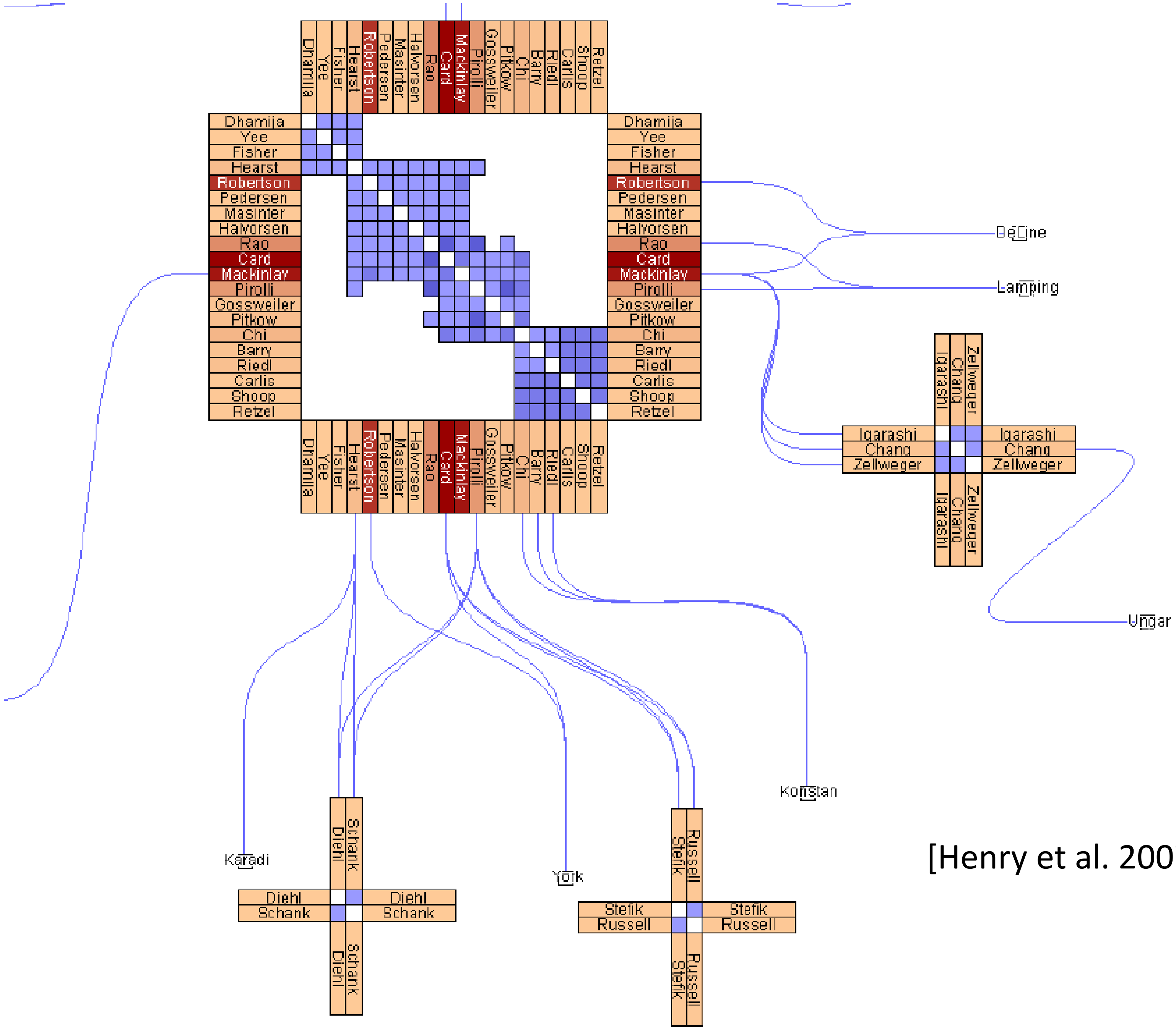
Data Level: interactive Methods

systems, such as Node Trix, compact dense subgraphs into matrix representations.



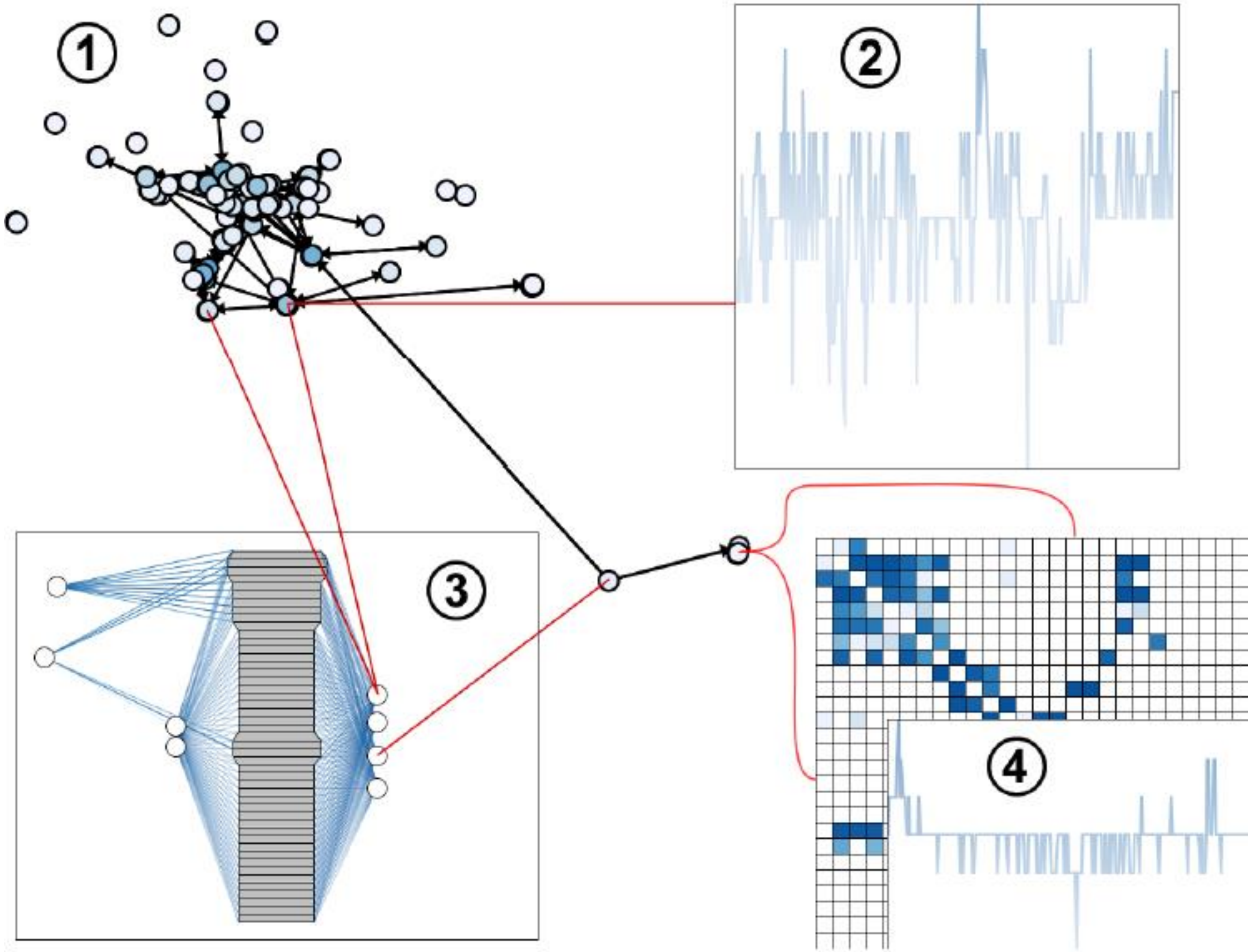


[Henry et al. , InfoVis 2007]



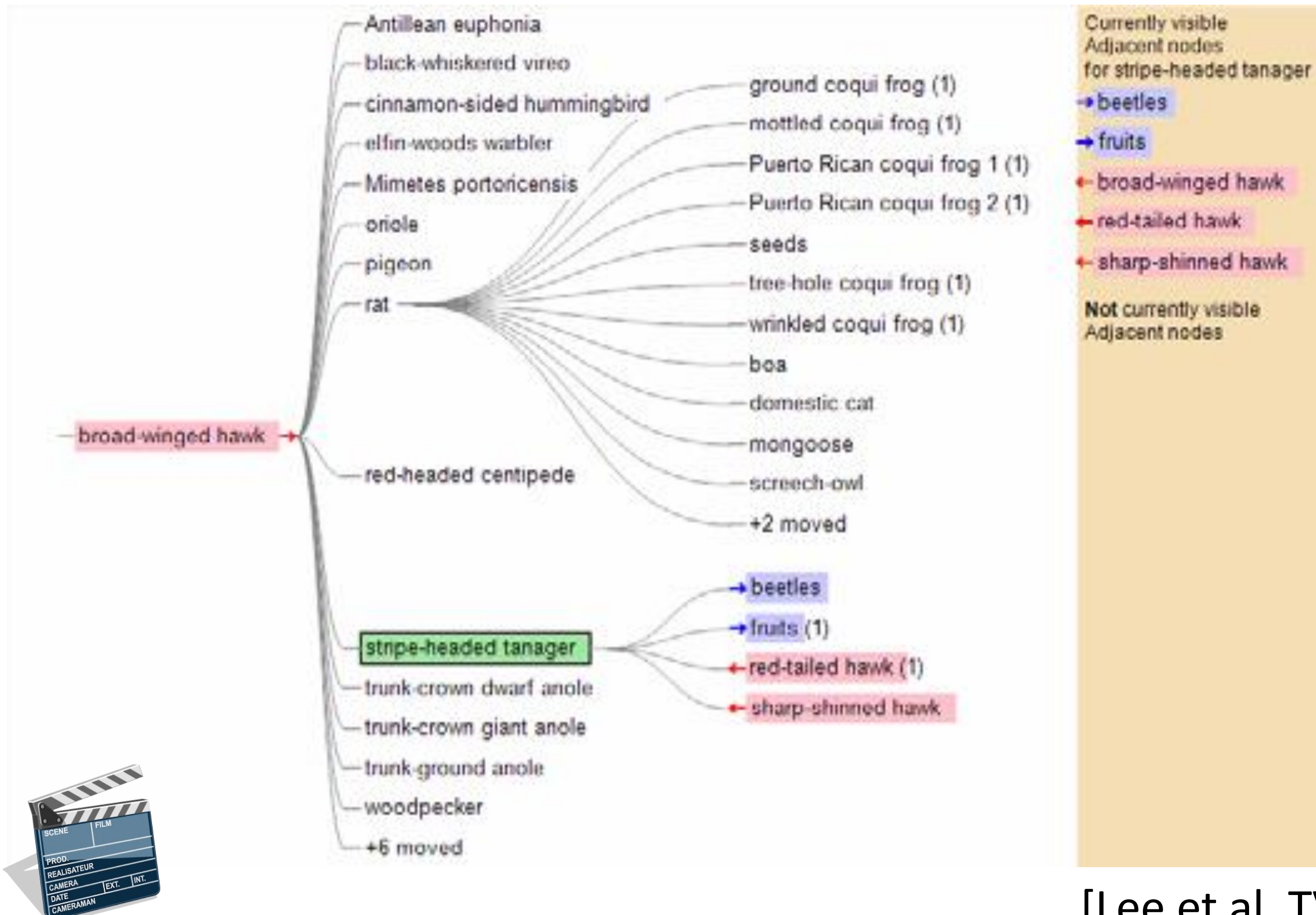
[Henry et al. 2007]

In Situ Exploration of Large Dynamic Networks



[Hadlak et al. TVCG 2011]

TreePlus: Interactive Exploration of Networks with Enhanced Tree Layouts



[Lee et al. TVCG 2006]

Data Level: clustering Methods

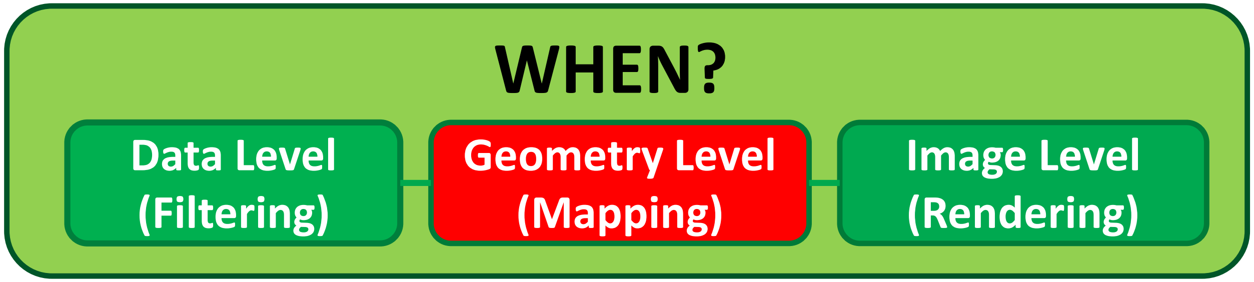
Clutter Reduction Techniques

G. Ellis and A. Dix TVCG 2007

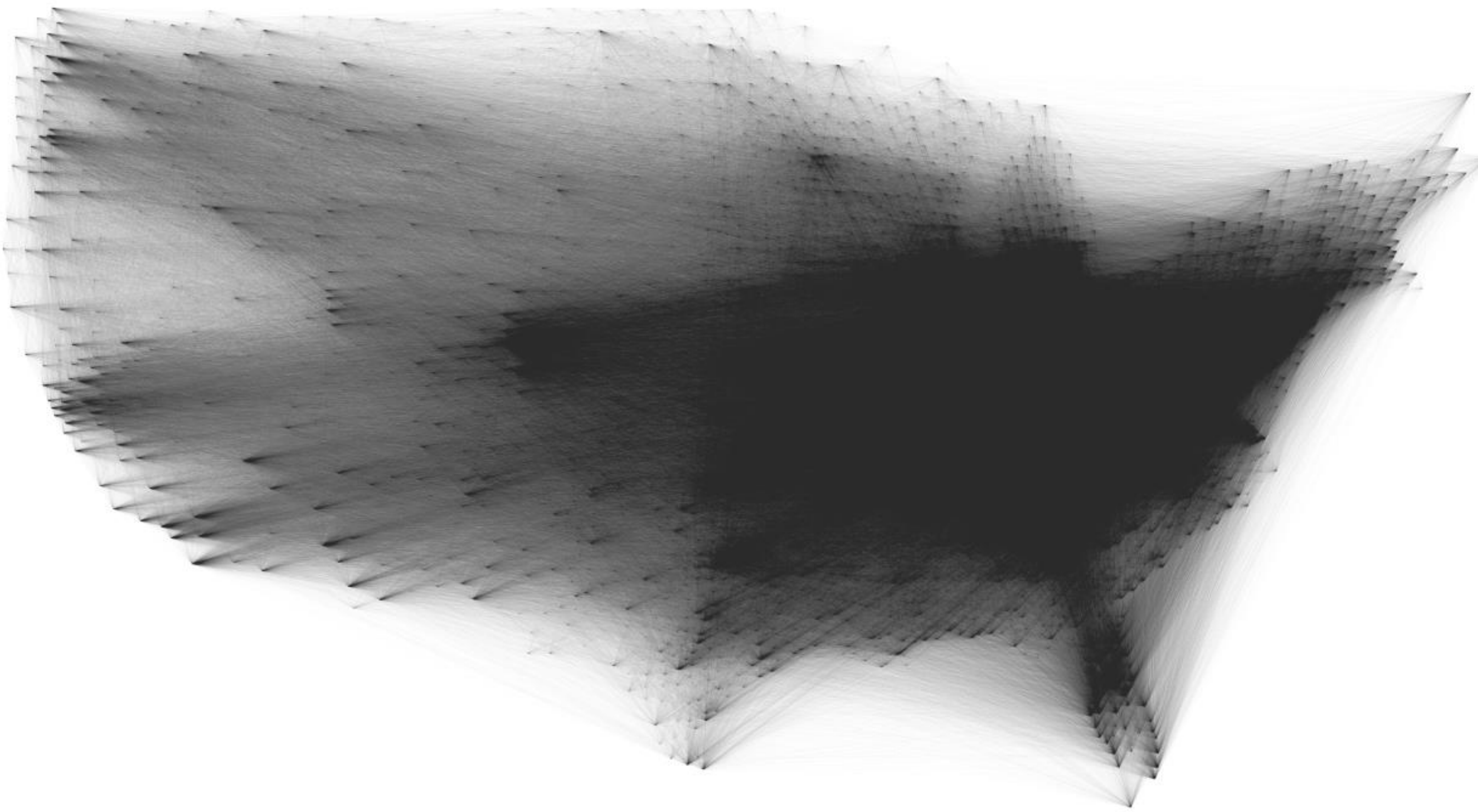
	clutter reduction technique
appearance	sampling
	filtering
	change point size
	change opacity
	clustering
spatial distortion	point/line displacement
	topological distortion
	space-filling
	pixel-plotting
	dimensional reordering
temporal	animation

PART II:
EDGE SET SIMPLIFICATION

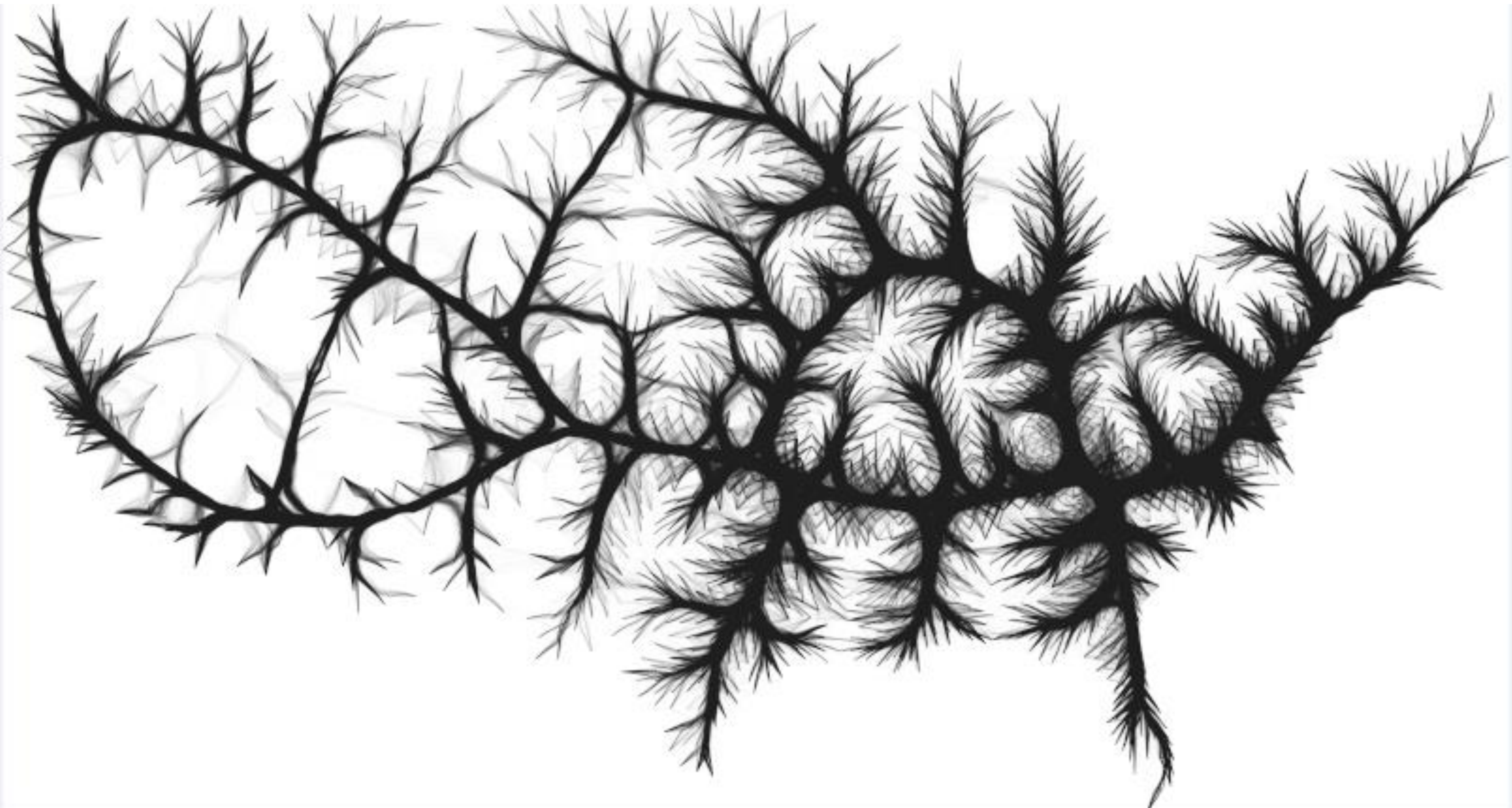
GEOMETRY LEVEL



Us migration
Original
545 881 edges

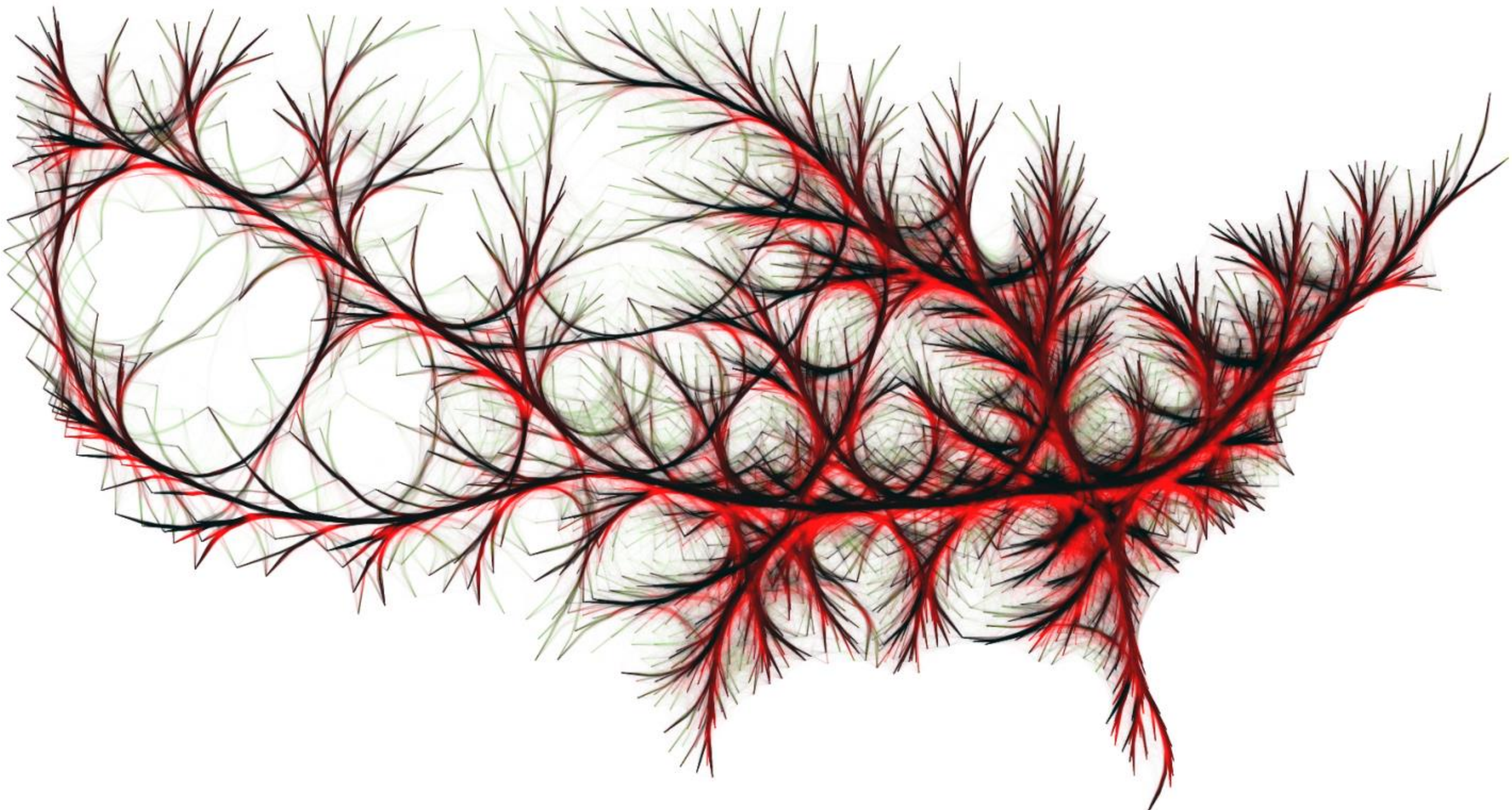


Bundled



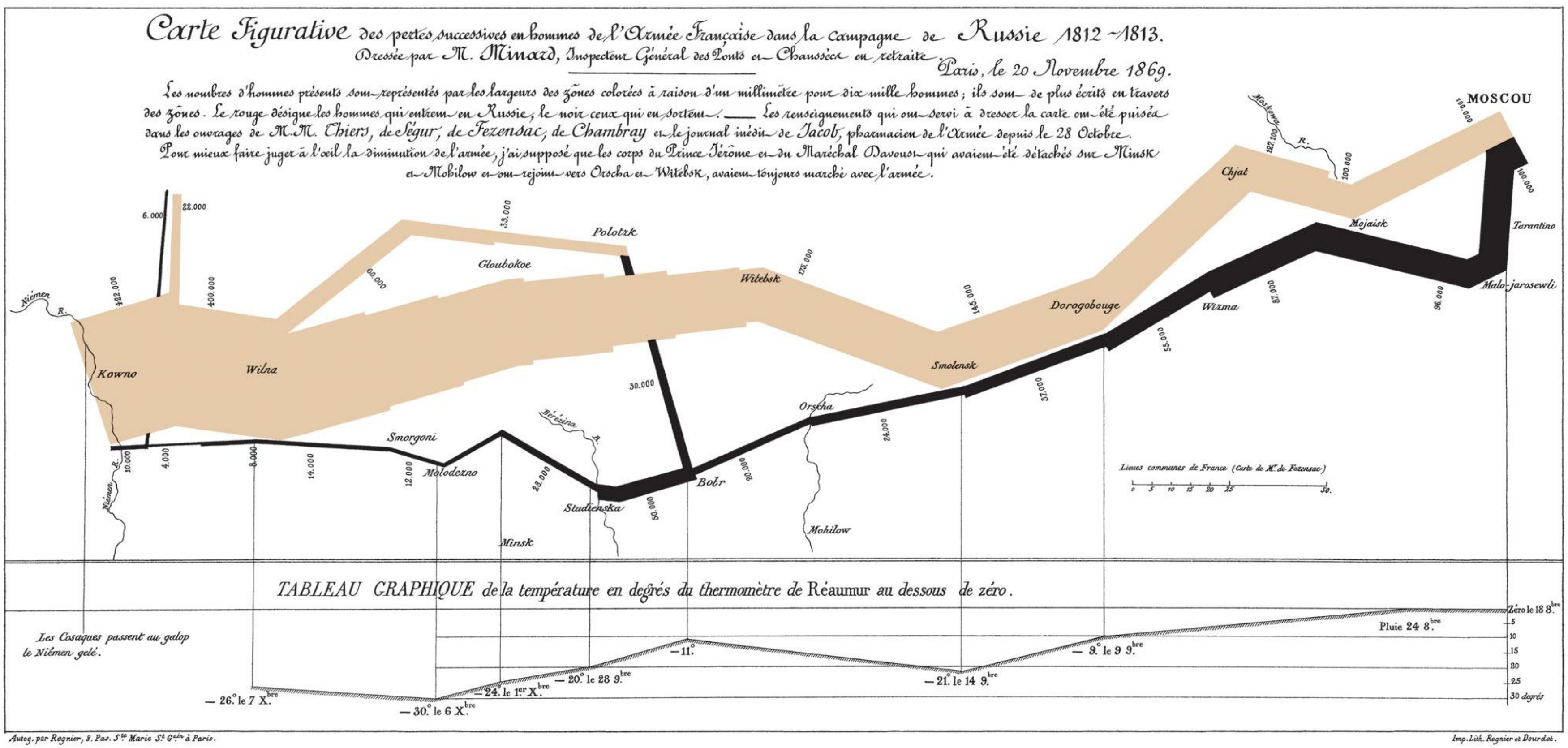
Us migration
Kernel Density Based Edge Bundling
22 million vertexes

Shading

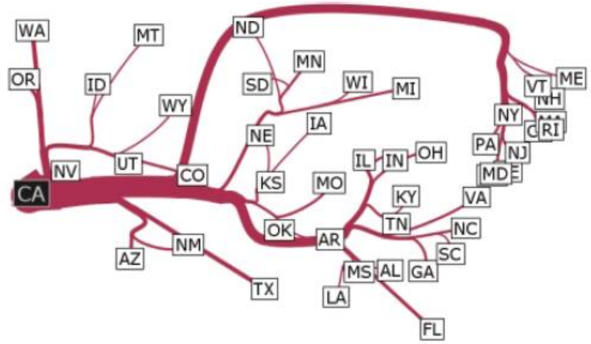


Flow map/ Sankee graphs

Sankey diagrams are a specific type of flow diagram, in which the width of the arrows is shown proportionally to the flow quantity.

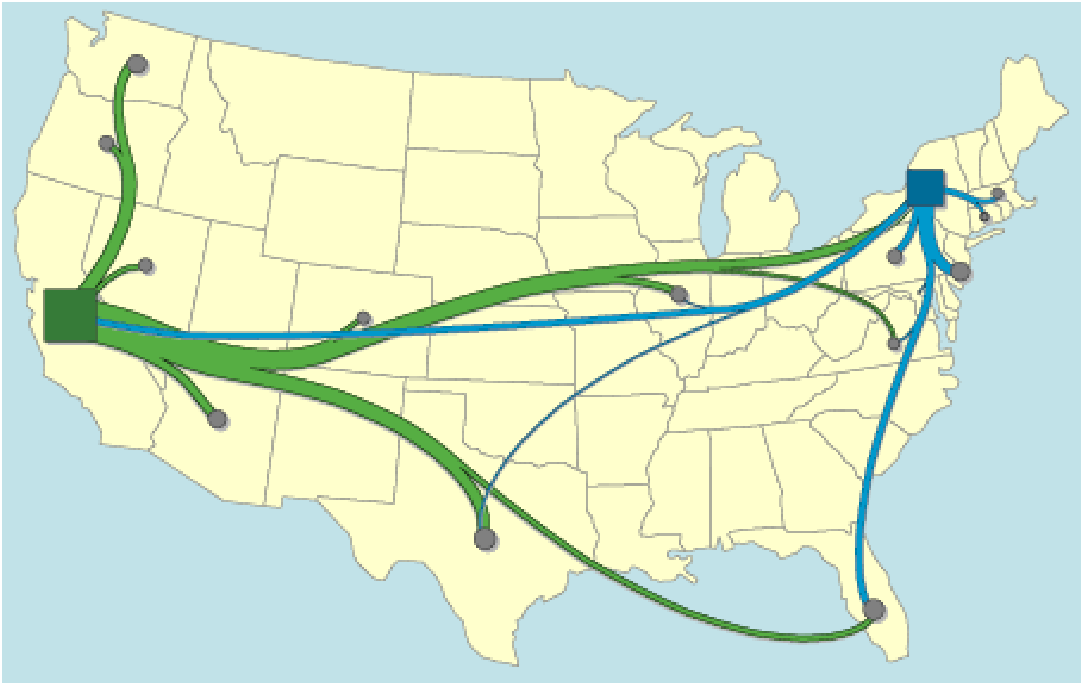


[Tobler 1987]



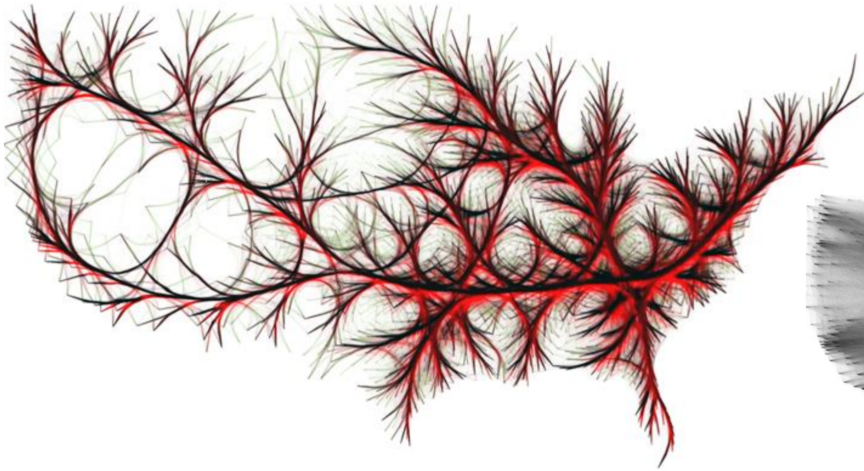
[Phan et al. InfoVis 05]

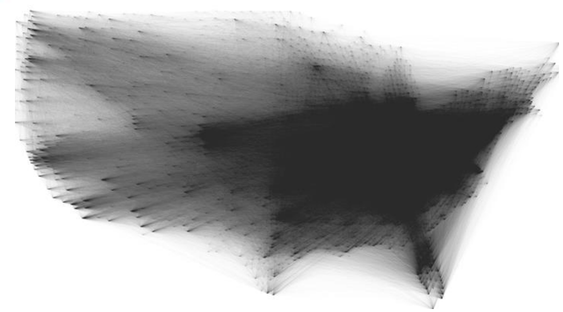
[Verbeek et al. infovis 11]



[Nocaj and Brandes,
Stub Bundling and Confluent Spirals for Geographic Networks
Graph Drawing 13]

Edge Bundling
definitions





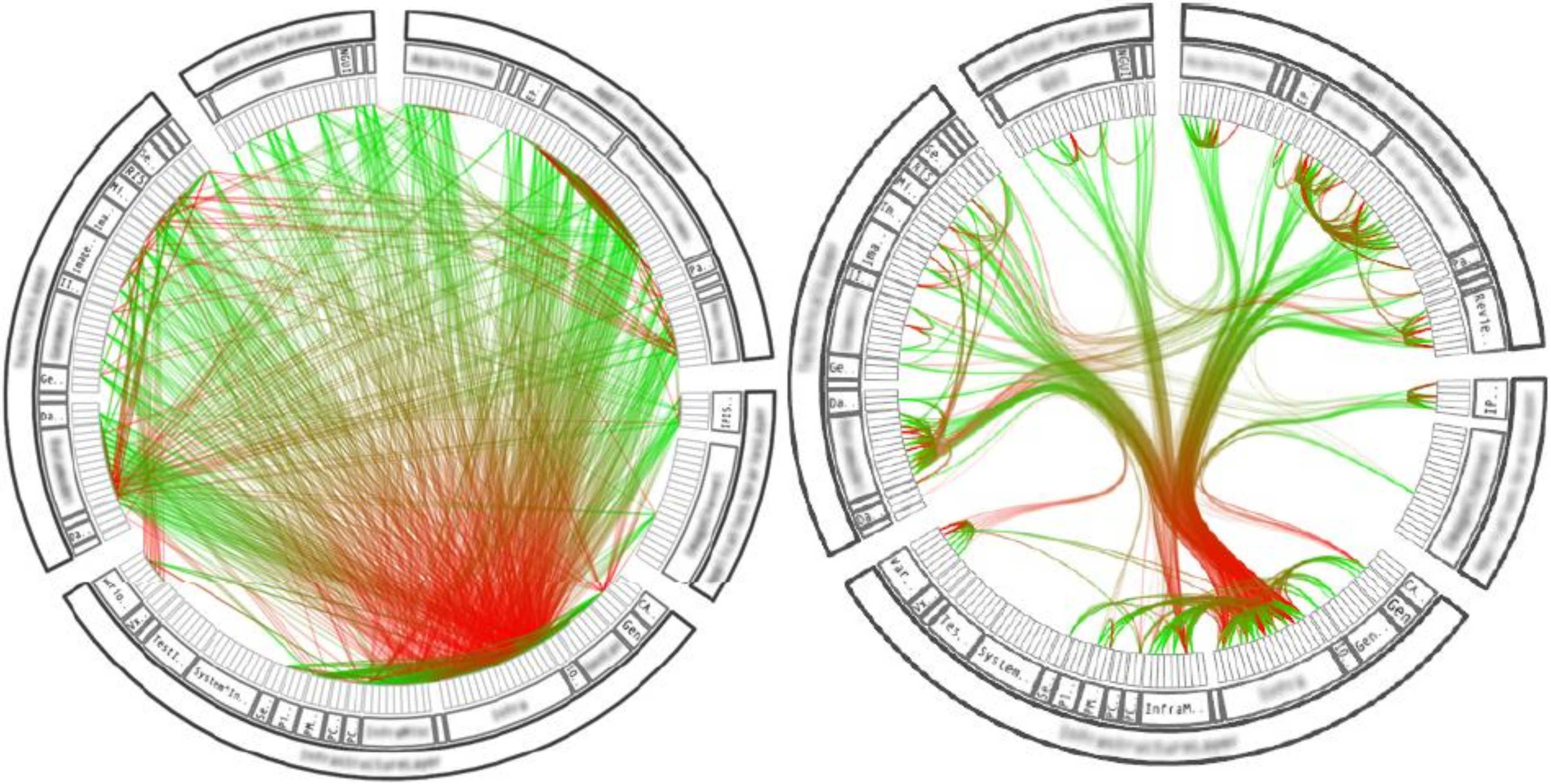
Edge bundling techniques trade clutter for overdraw by routing related edges along similar paths.

Bundling can be seen as sharpening the edge spatial density, by making it high along bundles and low elsewhere.

Bundling improves readability for finding node-groups related to each other by edge-groups (bundles) which are separated by white space.

Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data

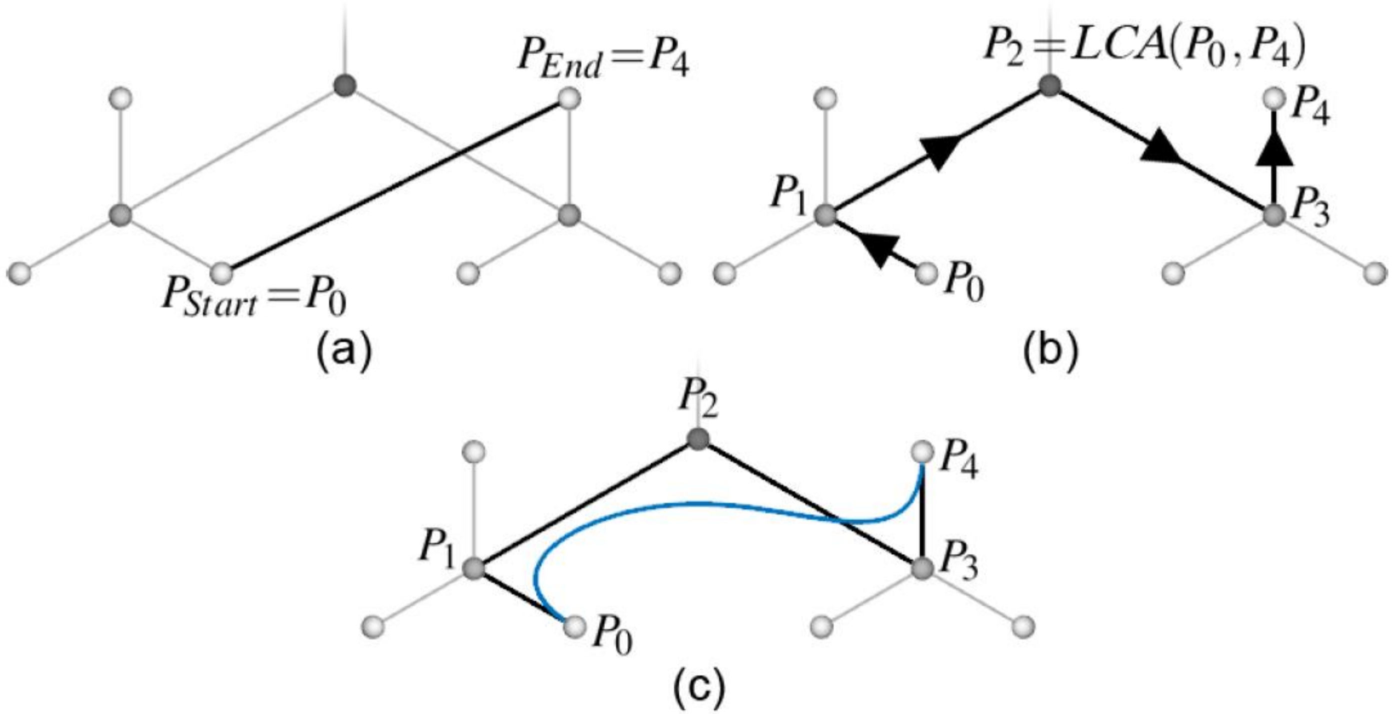
Danny Holten. 2006. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (September 2006), 741-748.



A software system and its associated call graph (caller = green, callee = red). (a) and (b) show the system without bundling using a radial and a squarified treemap layout (node labels disabled), respectively.

[Holten Infovis 2006]

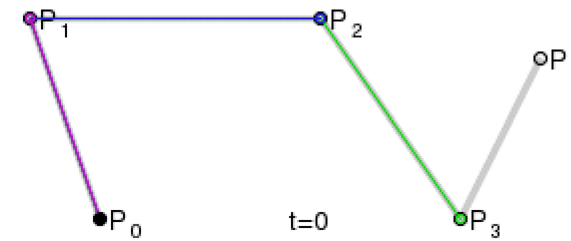
Hierarchical Edge Bundles principle



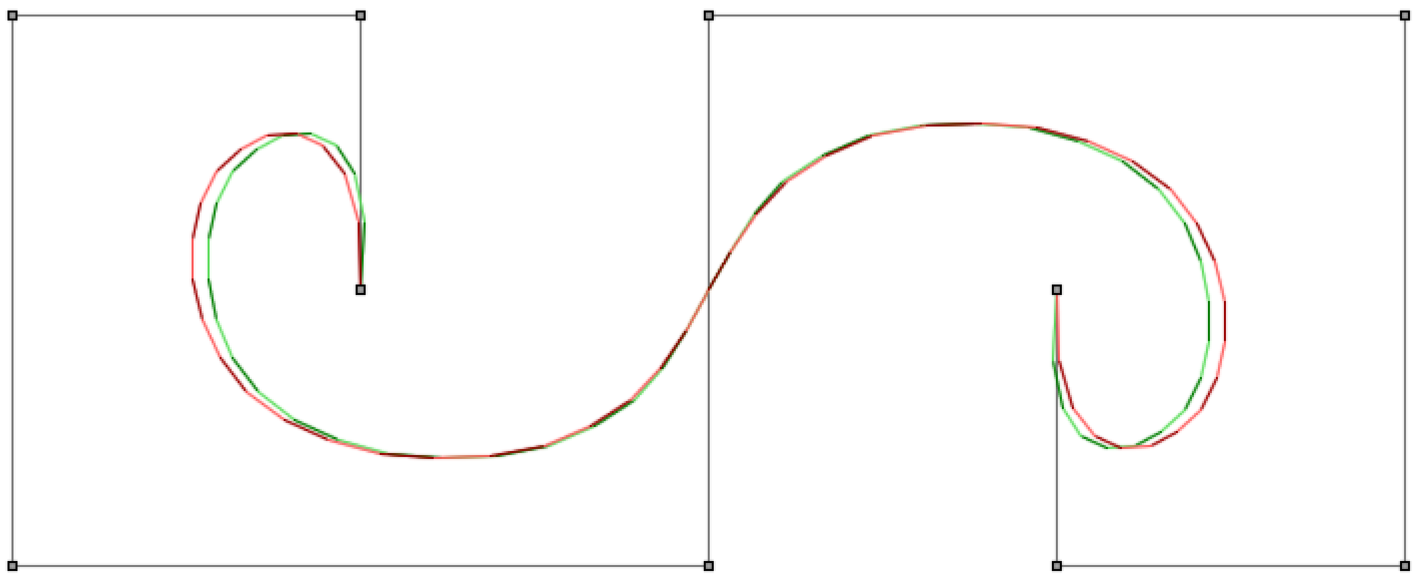
The approach is to use the path along the hierarchy between two nodes having an adjacency relation as the control polygon of a spline curve; the resulting curve is subsequently used to visualize the relation.

[Holten Infovis 2006]

Spline computation
Demo

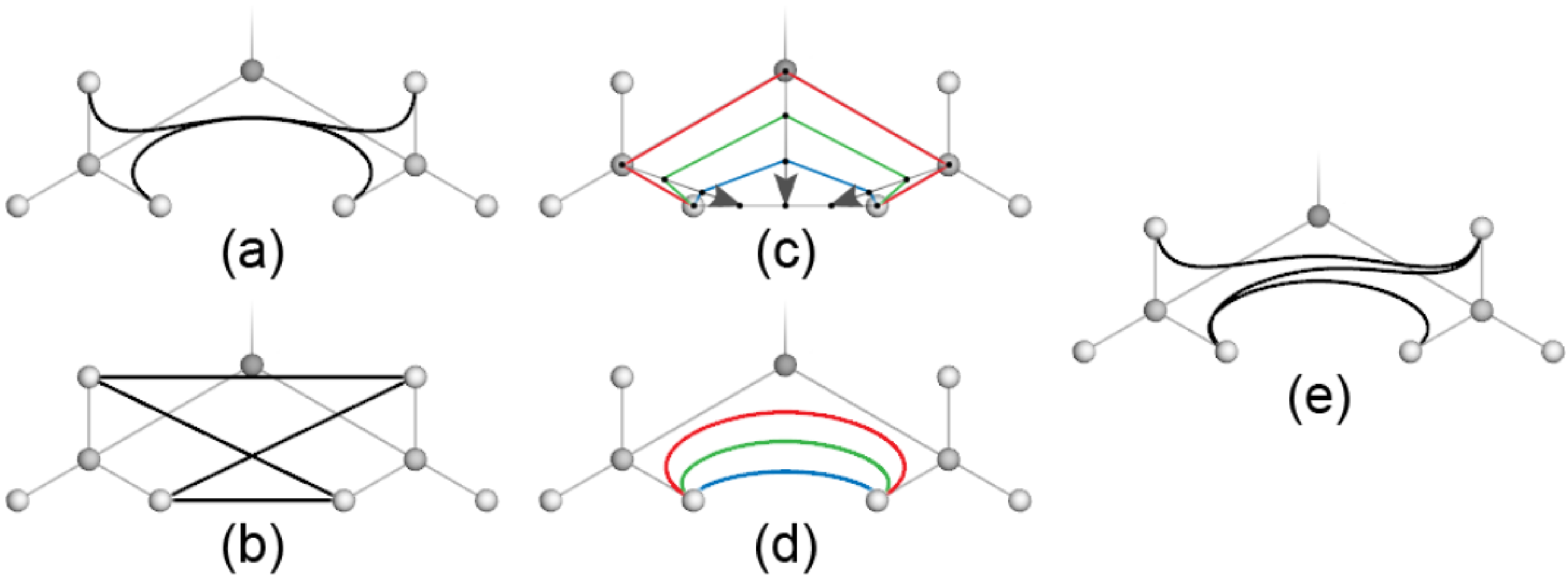


B-spline



[B. A. Barsky. *The Beta-Spline: A Local Representation based on Shape Parameters and Fundamental Geometric Measures*. PhD thesis, University of Utah, 1981]

Resolving ambiguity

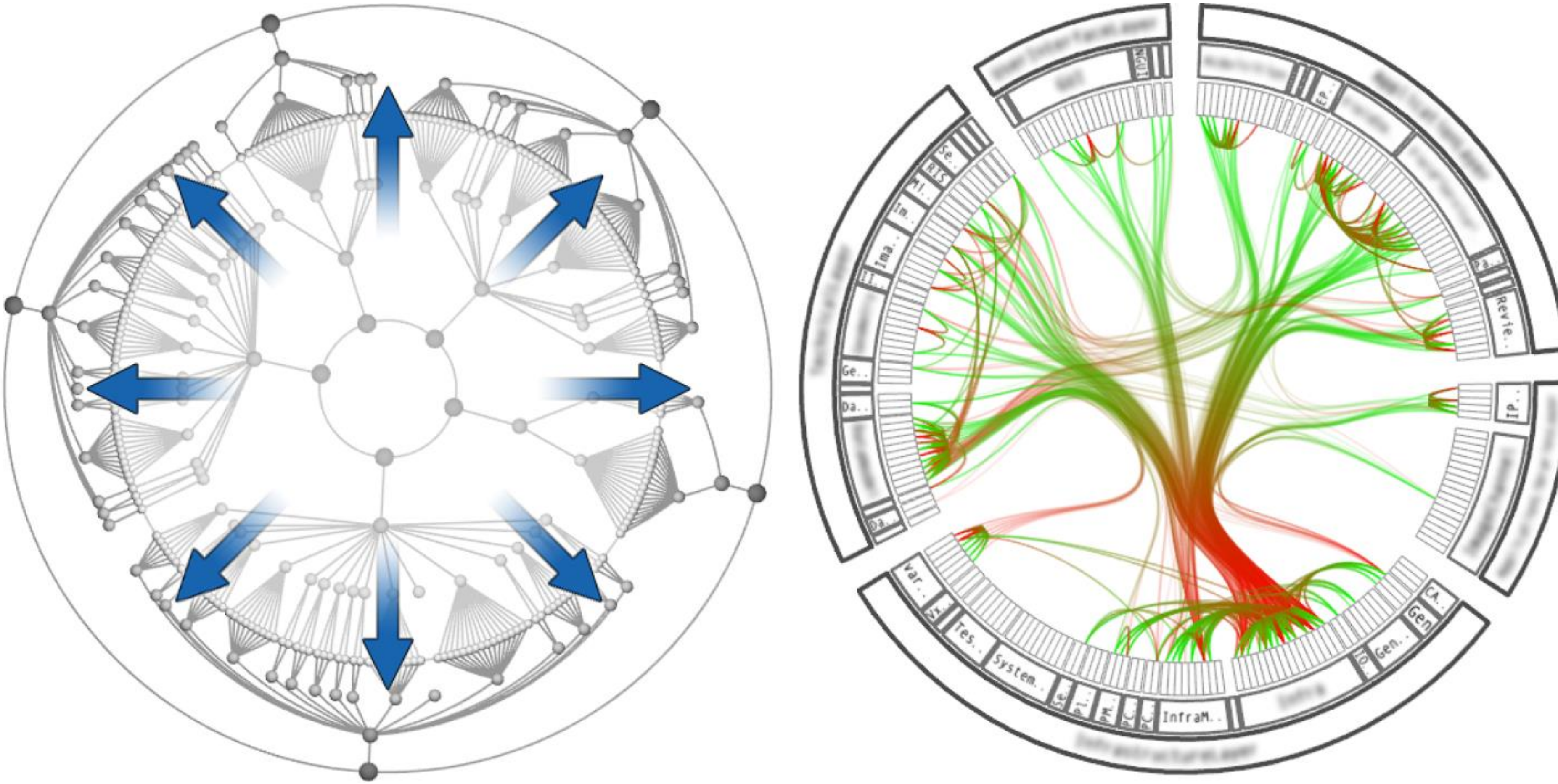


The bundle in (a) might contain each edge depicted in (b).

(c) and (d) show how different values of spline curves can be used.

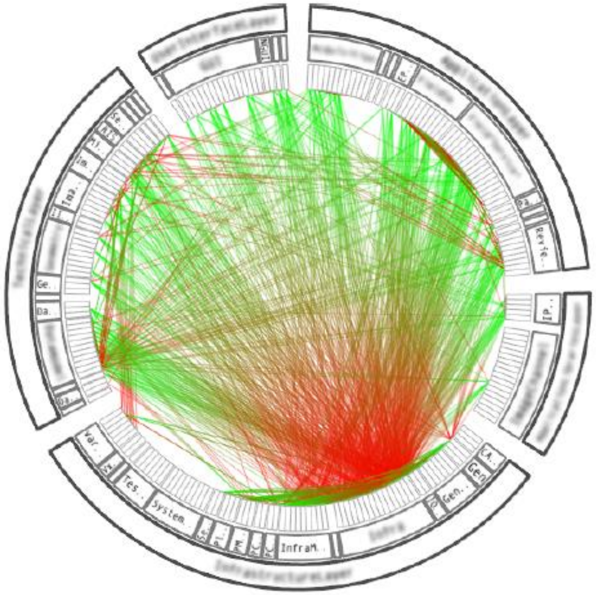
As shown in (e), a fairly high bundling strength (b = 0.8) can be chosen to retain visual bundles while still resolving ambiguity.

[Holten Infovis 2006]

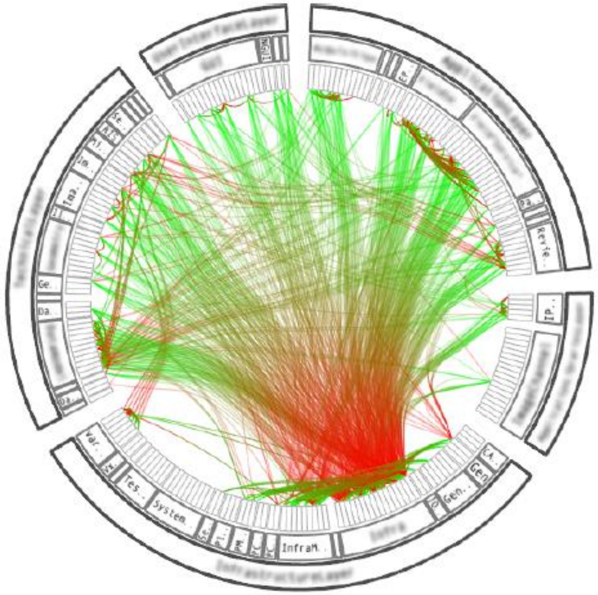


Radial layout construction. A radial tree layout is used for the inner circle and subsequently mirrored to the outside;
the inner layout is hidden and its structure is used to guide the adjacency edges.

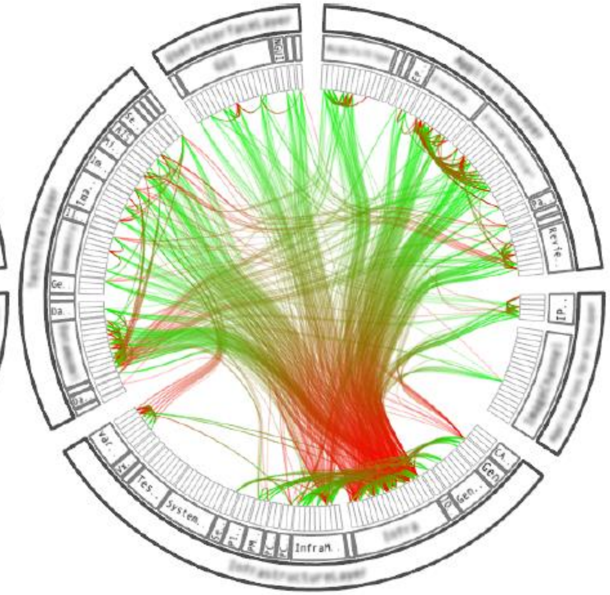
[Holten Infovis 2006]



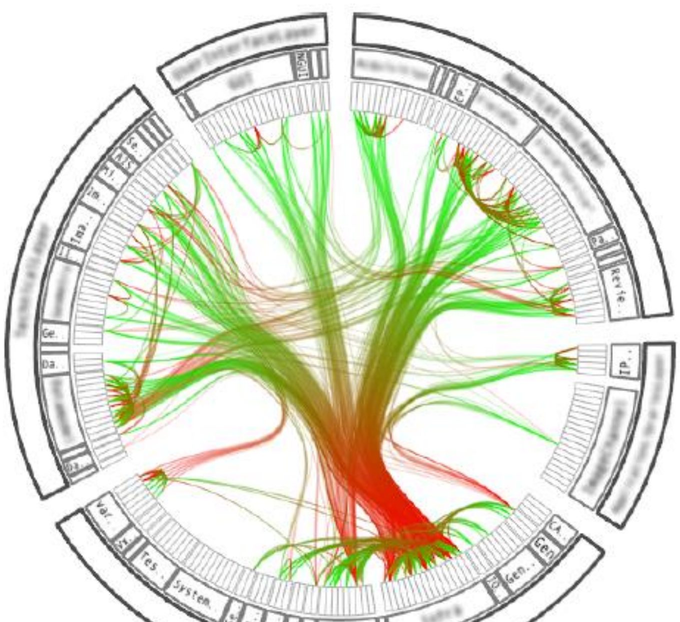
$\beta = 0$



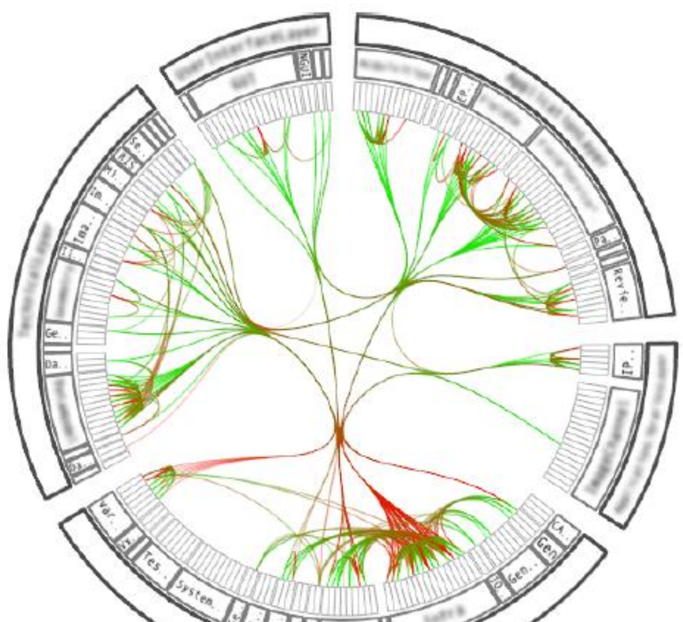
$\beta = 0.25$



$\beta = 0.5$



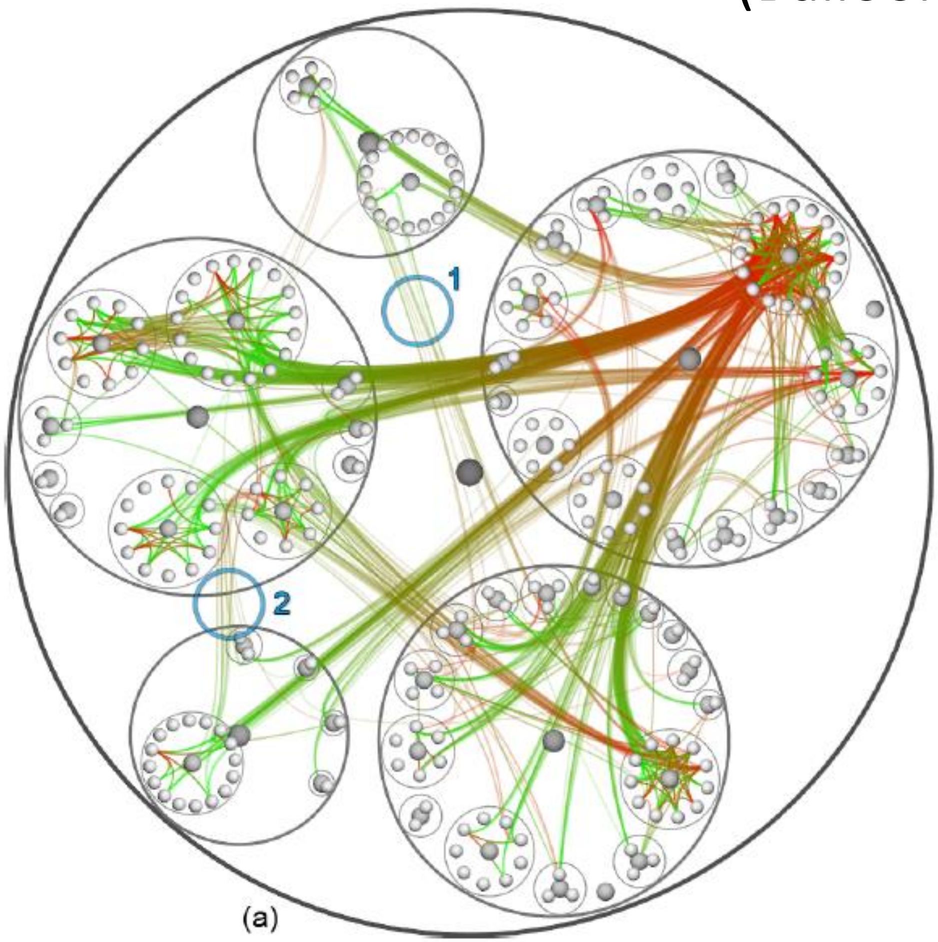
$\beta = 0.75$

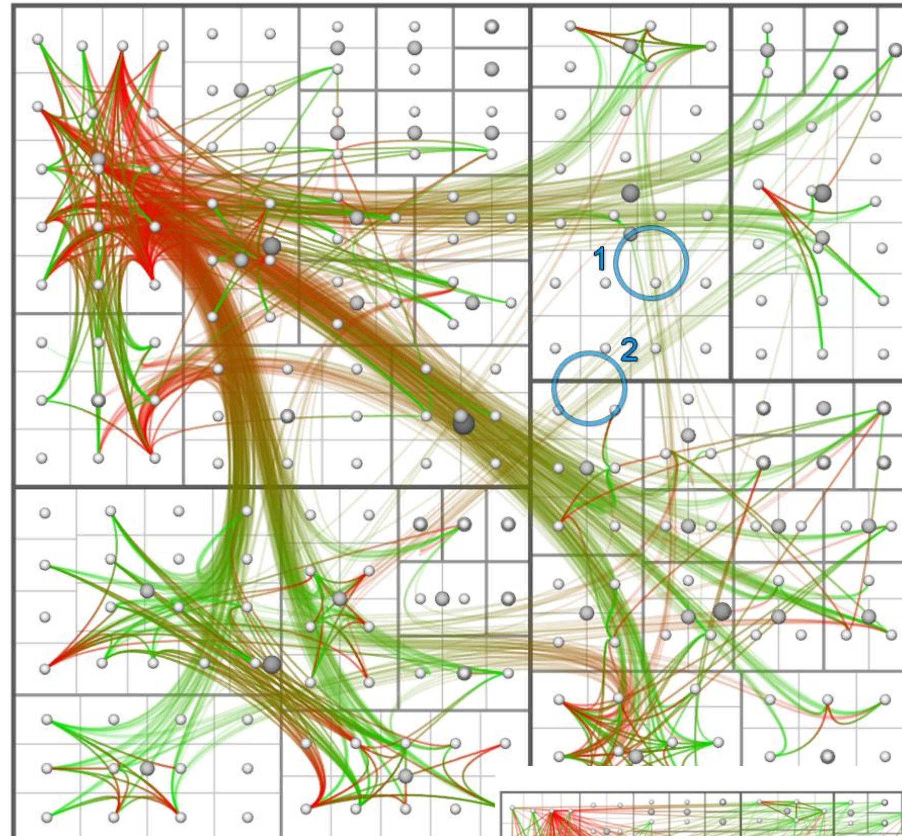


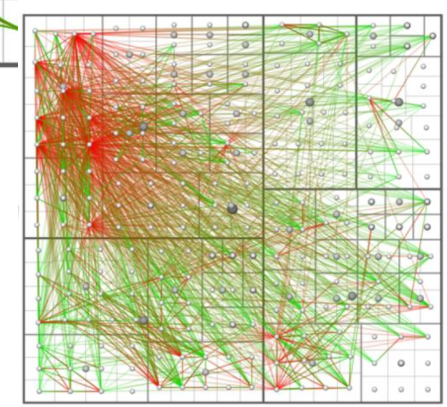
$\beta = 1$

[Holten Infovis 2006]

Hierarchical Edge Bundling with different graph layout (Balloon, Treemap)





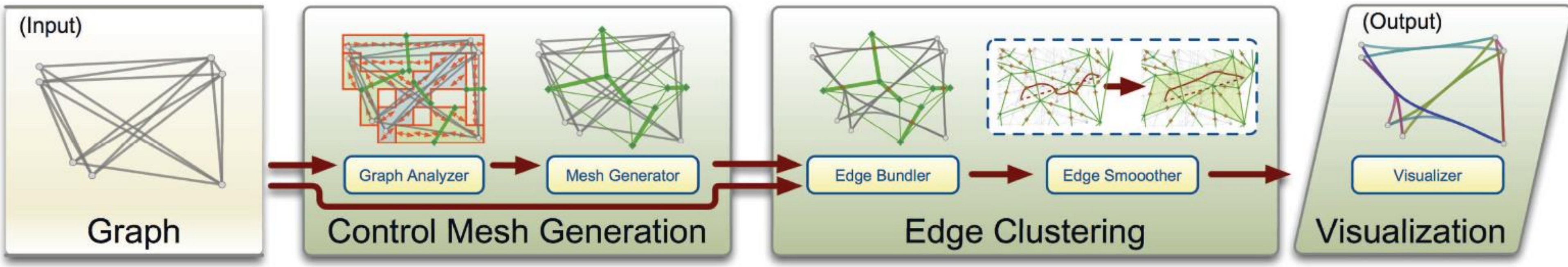


[Holten Infovis 2006]

Geometry-Based Edge Clustering for Graph Visualization

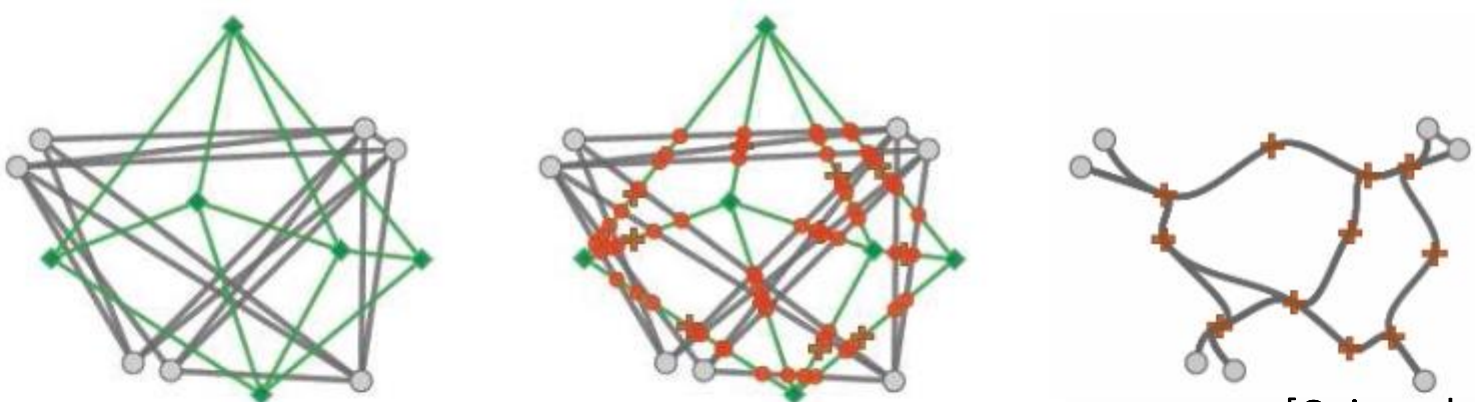
Weiwei Cui, Hong Zhou, Huamin Qu, Pak Chung Wong, and Xiaoming Li

InfoVis 08

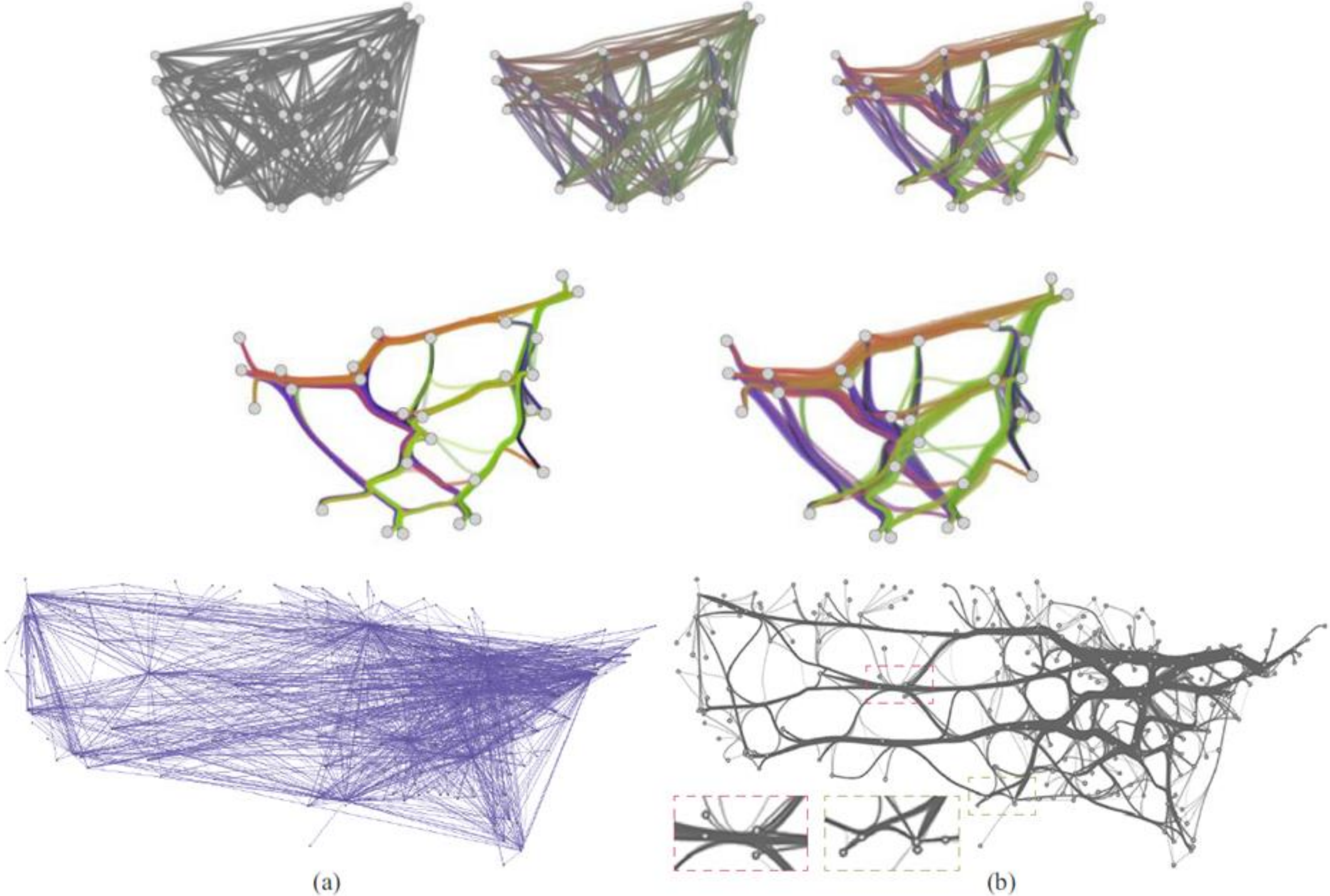


Control meshes are used to route curved edges

Control points: center of the intersection of graph edges and mesh



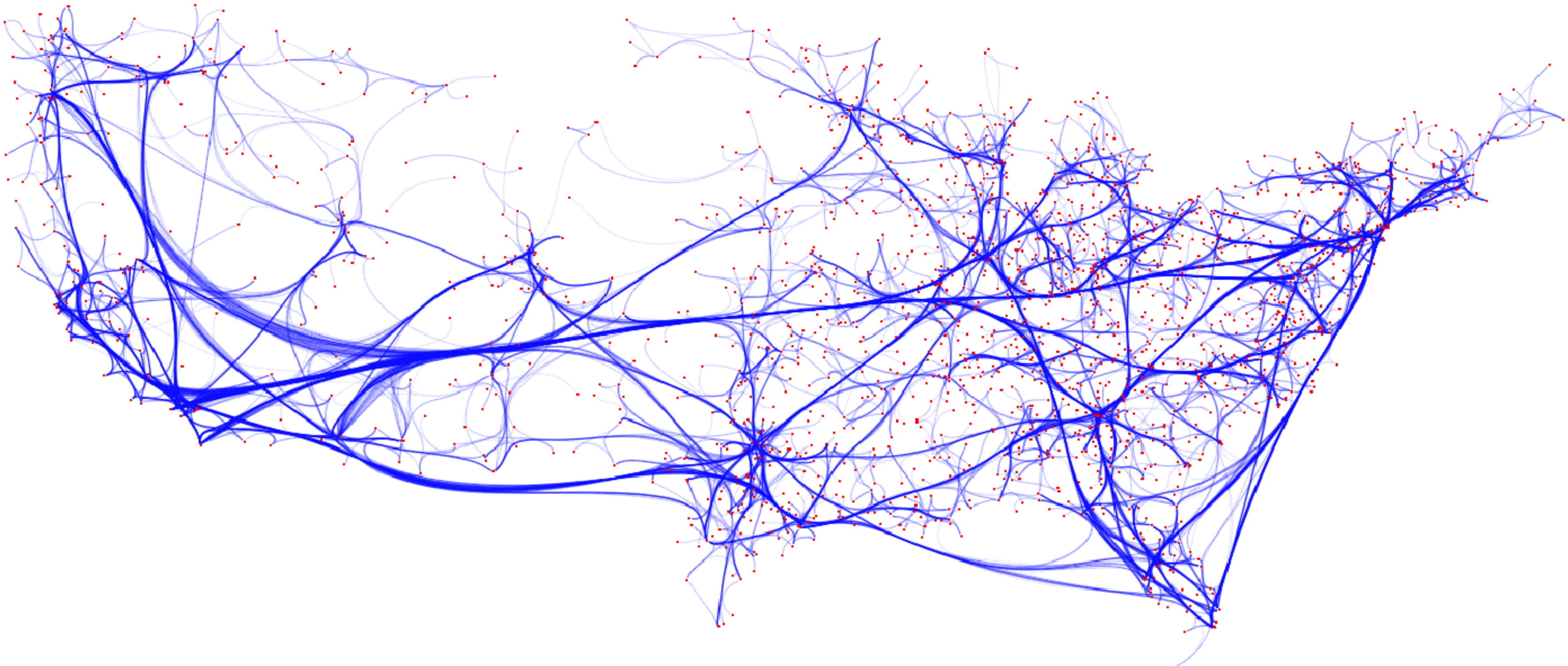
[Cui et al. Infovis 08]



[Cui et al. Infovis 08]

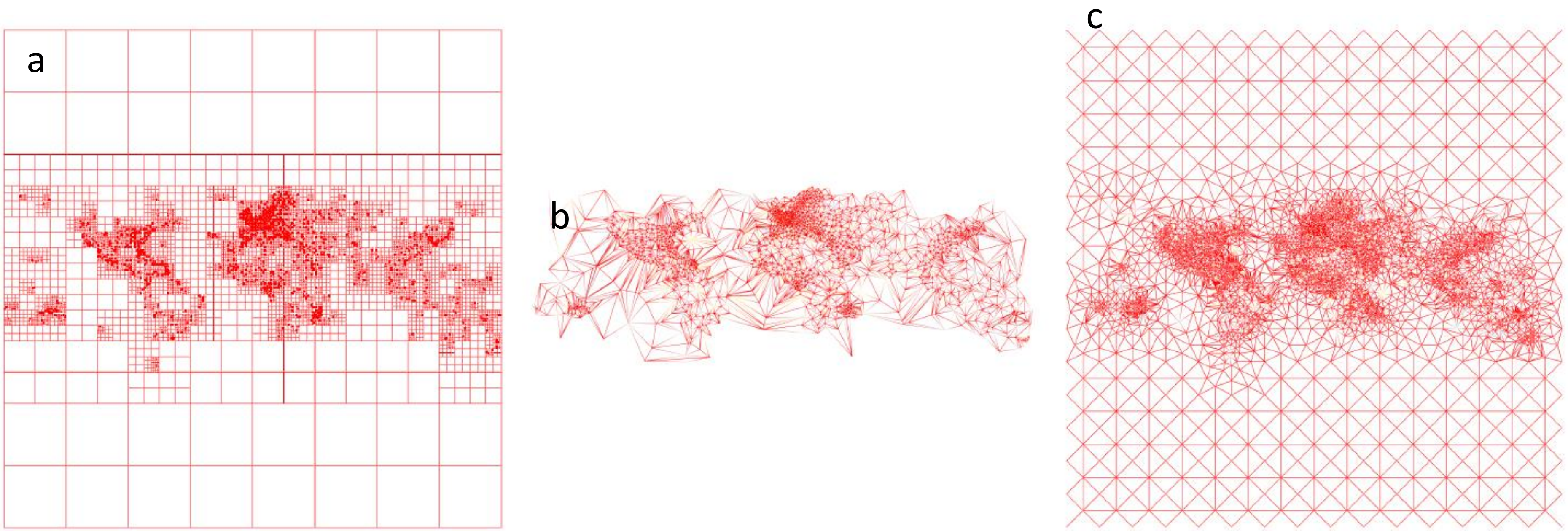
Winding Roads
Routing edges into bundles

A. Lambert and R. Bourqui and D. Auber
Eurographics/ IEEE-VGTC Symposium on Visualization 2010



[A. Lambert et al. EuroVis 2010]

[A. Lambert et al. EuroVis 2010]

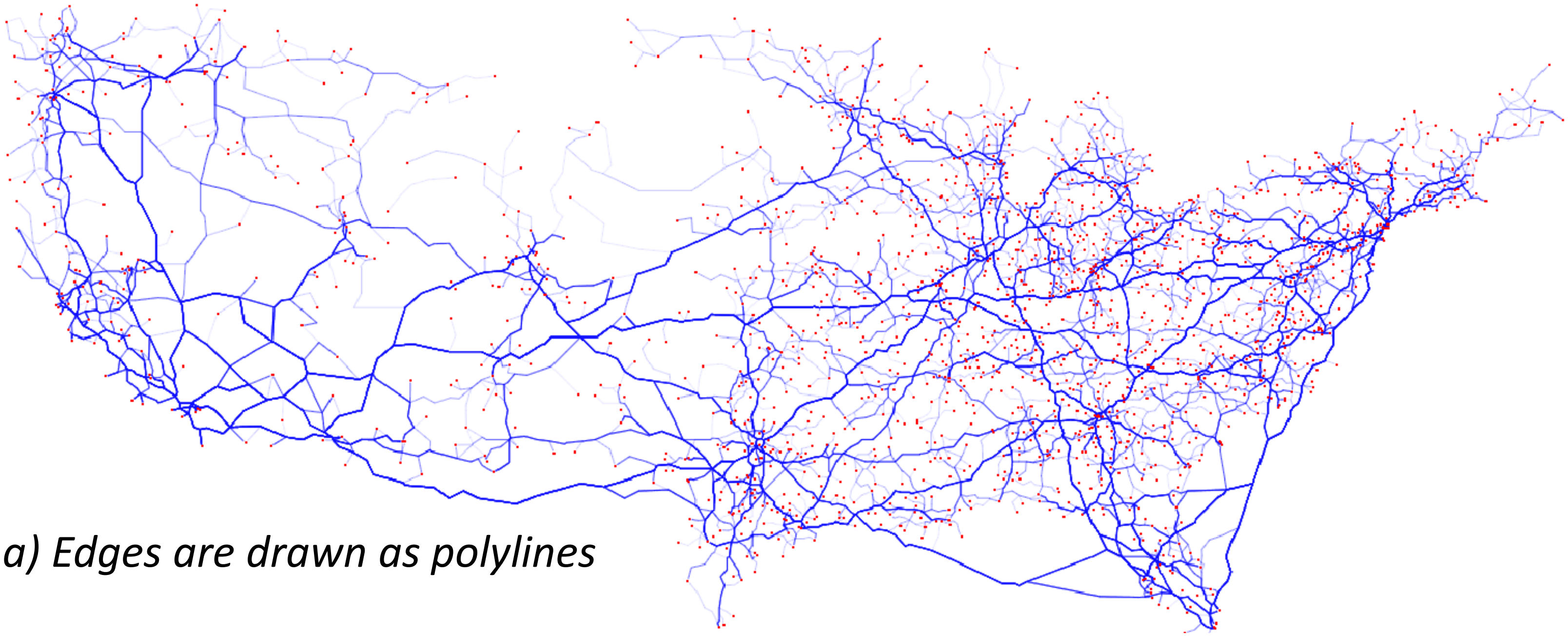


Grid graphs generated on the 2000 Air Traffic (AT) network with

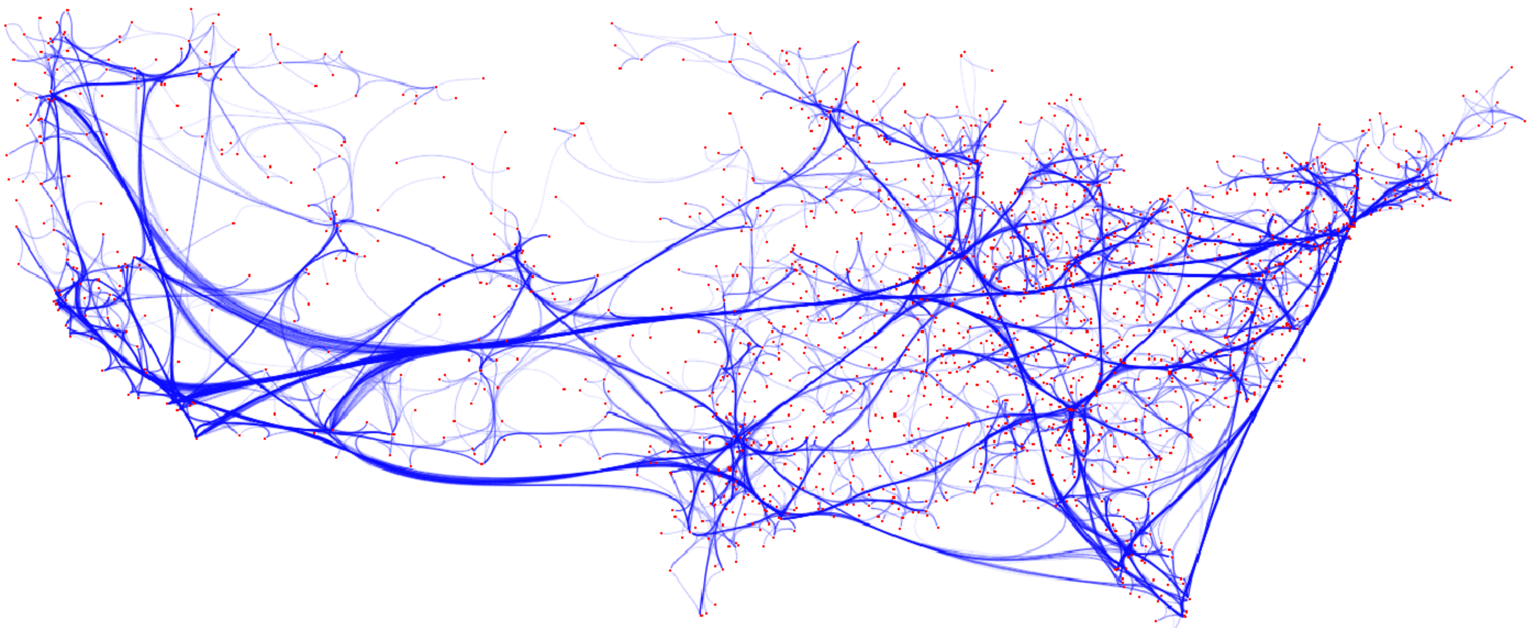
*(a) a quad tree (*37395 *nodes/*69102 *edges),*

*(b) A Voronoi diagram (*4531 *nodes/*13558 *edges)*

*(c) the hybrid quad tree/Voronoï approach (*10146 *nodes/*30315 *edges).*



a) Edges are drawn as polylines



(b) Edges are drawn as Bézier curves

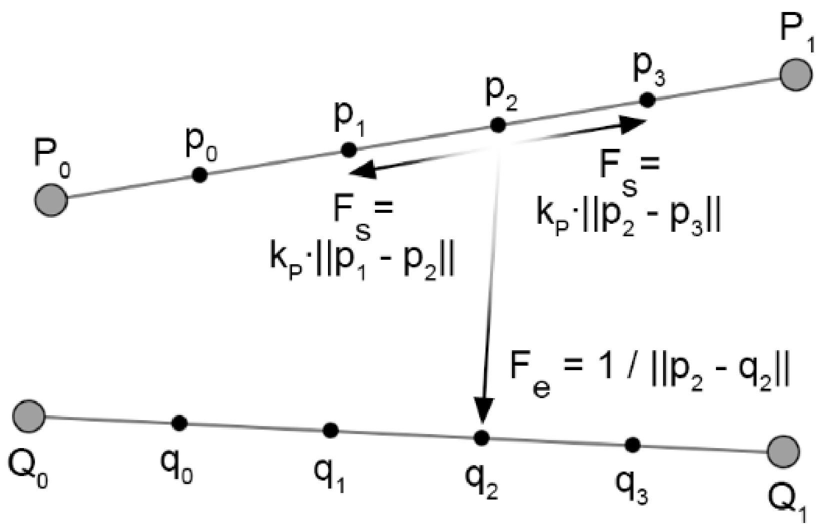
[A. Lambert et al. EuroVis 2010]

Force-Directed Edge Bundling for Graph Visualization

Danny Holten and Jarke J. van Wijk

Eurovis 2009

Force-Directed Edge Bundling: string and electrostatic force



Two interacting edges P and Q.

The spring forces **Fs** and the electrostatic force **Fe** that are exerted on subdivision point p2 by p1, p3, and q2 are shown.

$$\mathbf{F}_{\mathbf{p}_i} = k_P \cdot (\|p_{i-1} - p_i\| + \|p_i - p_{i+1}\|) + \sum_{Q \in E} \frac{1}{\|p_i - q_i\|},$$

with

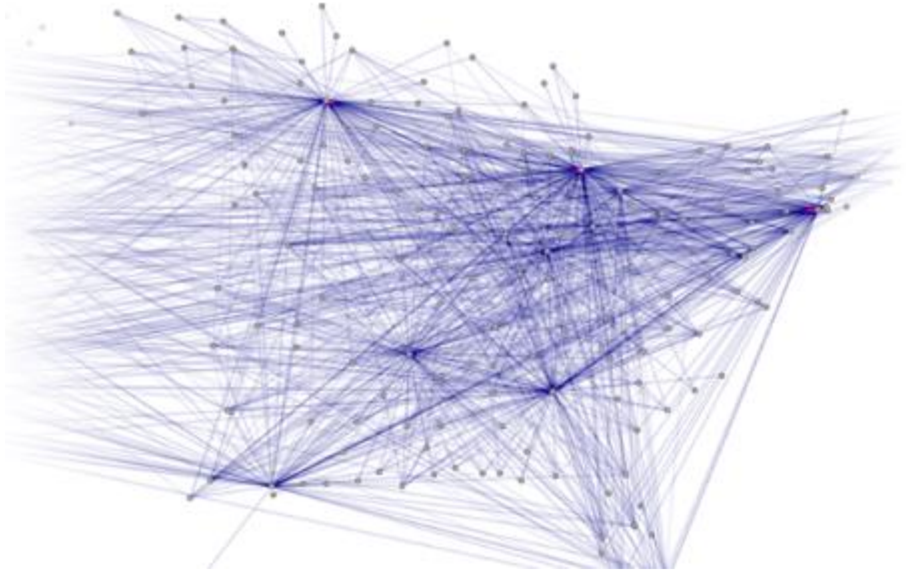
k_p : spring constant for each segment of edge P,

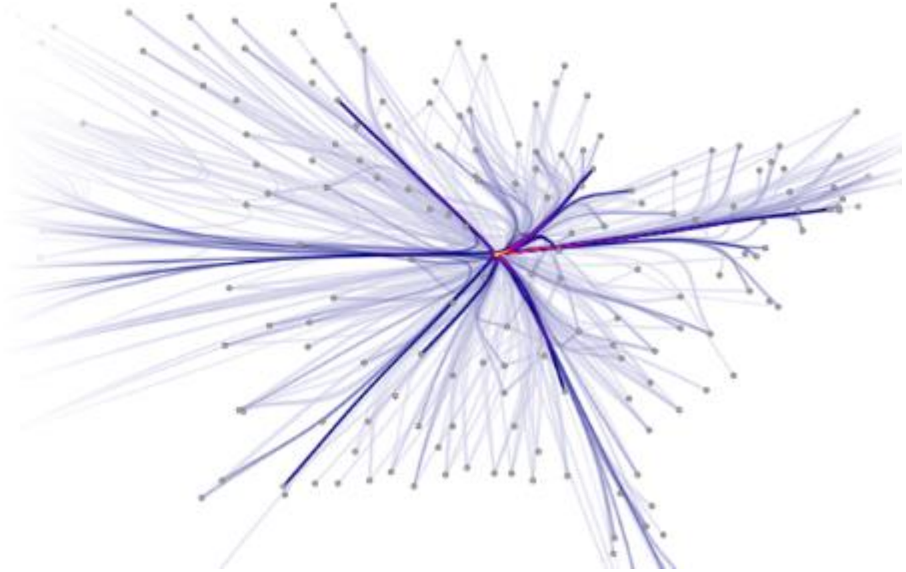
E : set of all interacting edges except edge P.

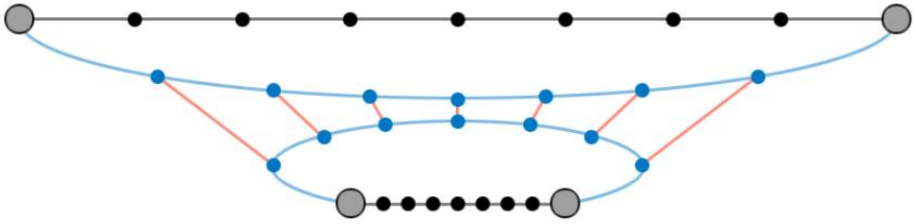
[Holten and Wijk EuroVis 2009]

Main issues

Initial algorithm produces edge-bundled graph, but the amount of bundling is often too high:



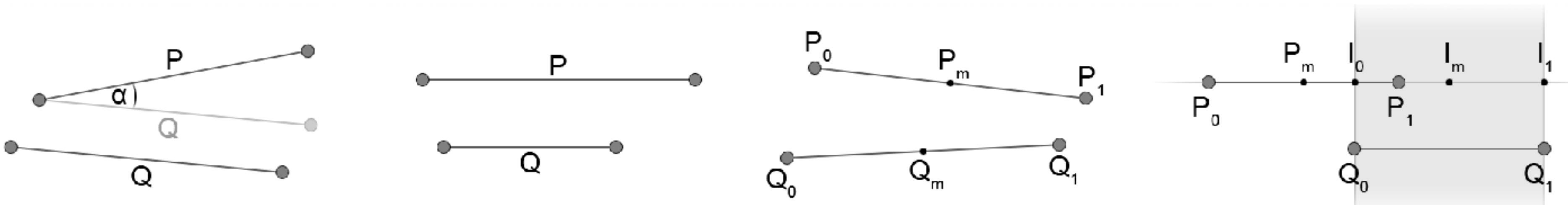




Bundling edges that differ considerably in length can result in noticeable stretching and curving of short edges. Original edges, curved edges and attracting forces are shown in black, blue, and red, respectively.

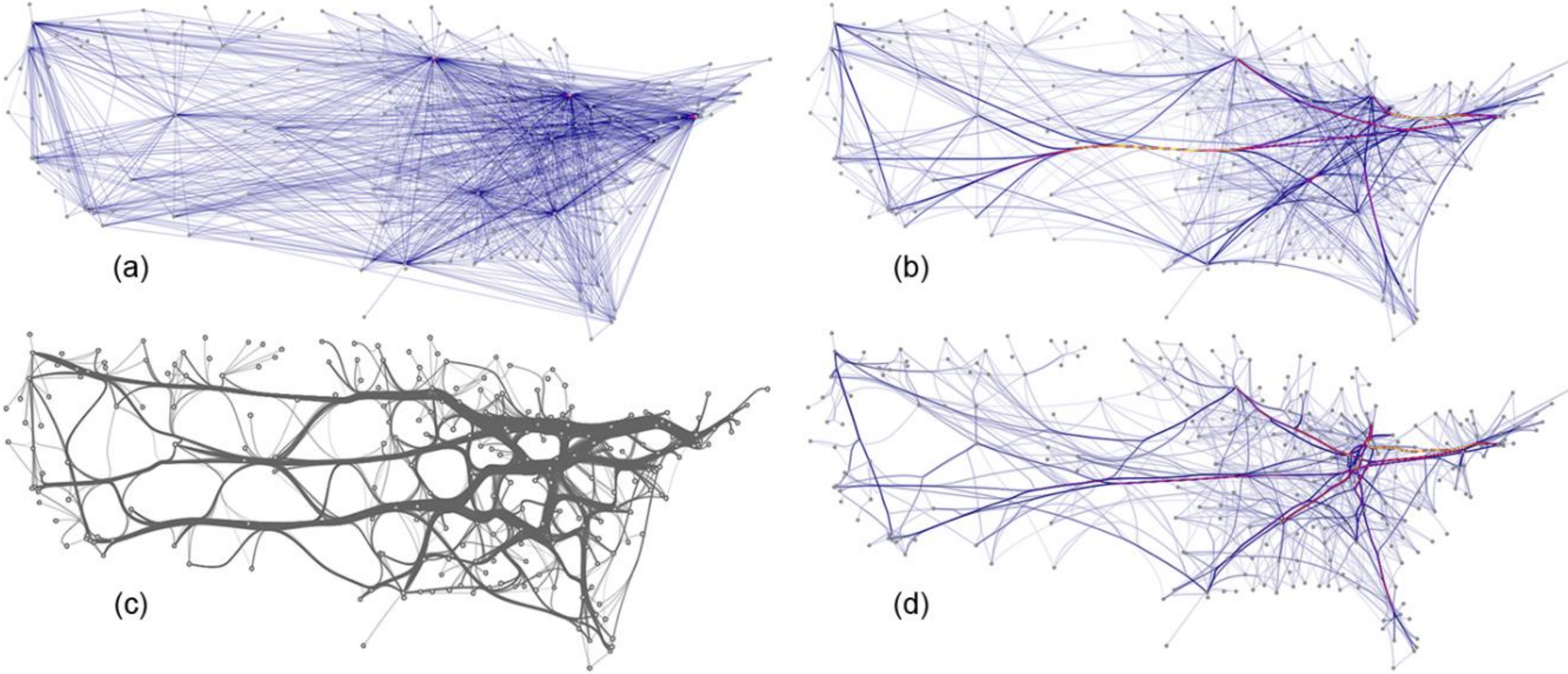
[Holten and Wijk EuroVis 2009]

Solution: edge compatibility



- Angle compatibility
- Scale compatibility
- Position compatibility
- Visibility compatibility

[Holten and Wijk EuroVis 2009]



	Air Line graph
FDEB	18.8 s
GBEB	2.5 s

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.

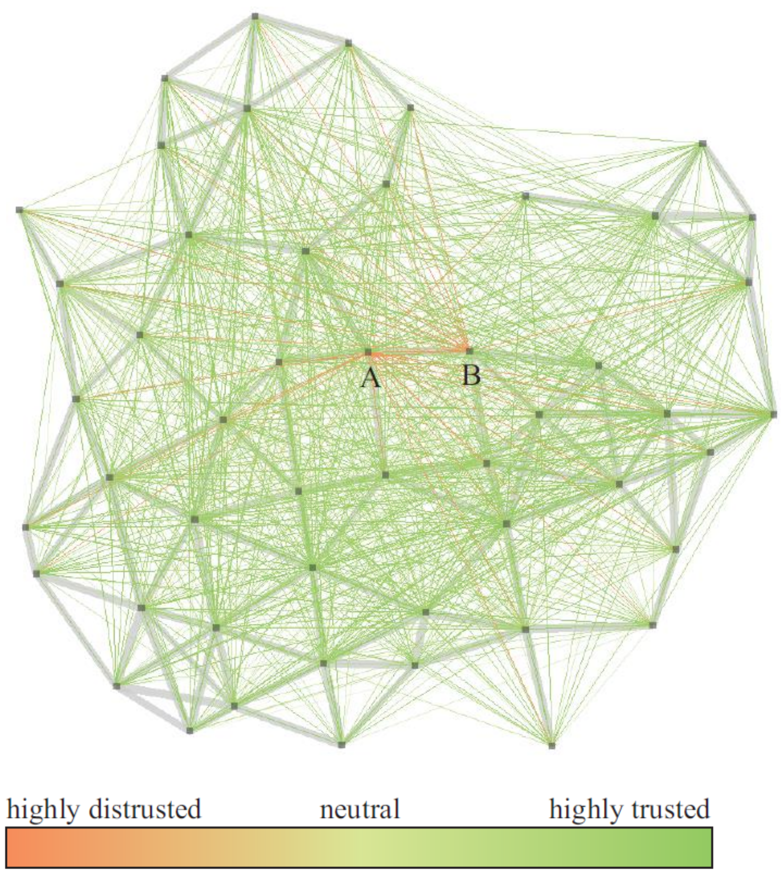
[Holten and Wijk EuroVis 2009]

SideKnot: Revealing Relation Patterns for Graph Visualization

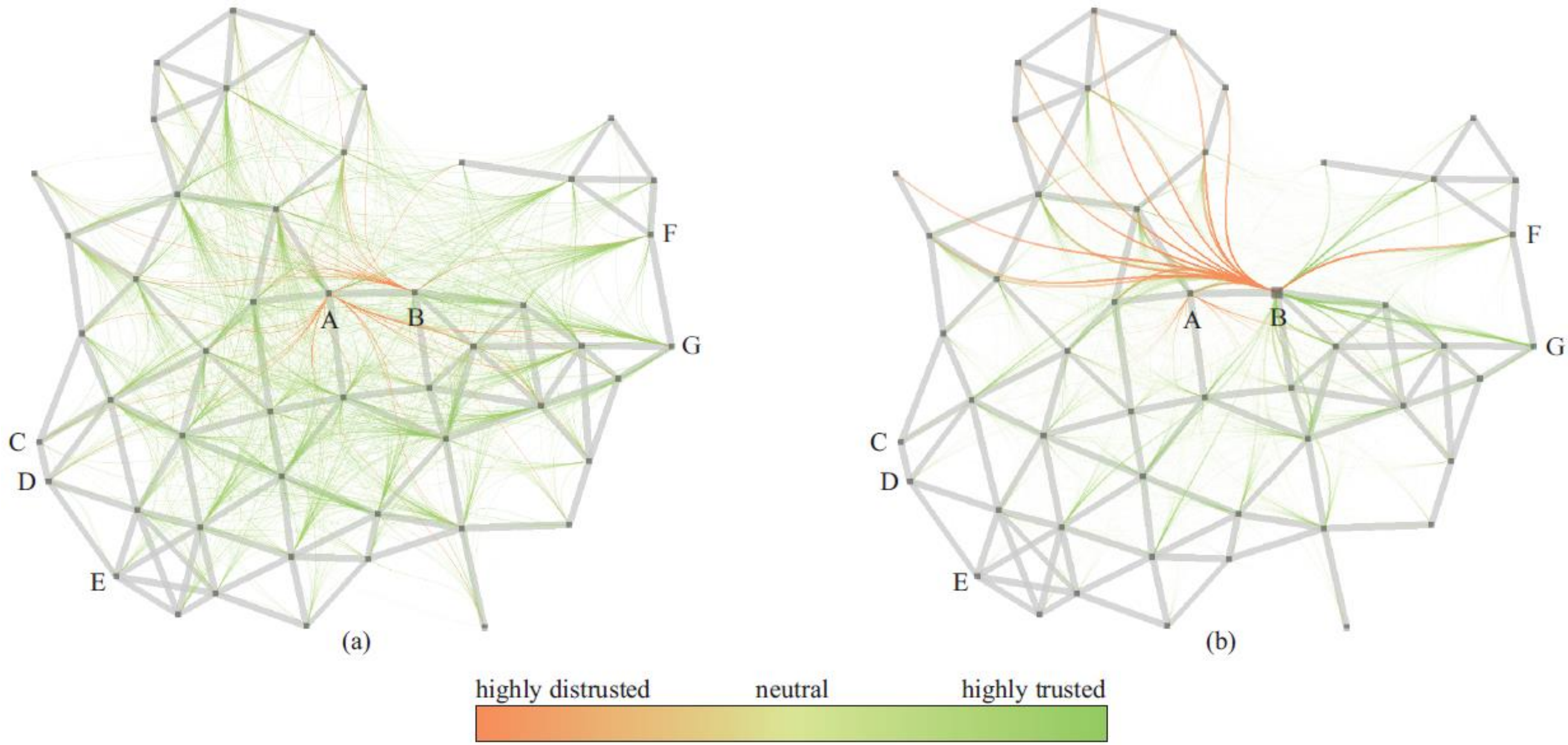
Dichao Peng Neng Lu Wei Chen Qunsheng Peng PacificVis 2012

SideKnot: Revealing Relation Patterns for Graph Visualization

Dichao Peng Neng Lu Wei Chen Qunsheng Peng



Original unbundled graph of 2263 trust relations between 50 nodes in a wireless network.
The grey grids in the background shows wireless connectivity between the nodes, while the colored lines are their trust relations established upon packet transmission along the wireless links.
The lower color gradient bar is used to encode the low-to-high trust level.



Bundled result for trust graph

Node B is selected by user and its trust relations get highlighted

The SideKnot Algorithm

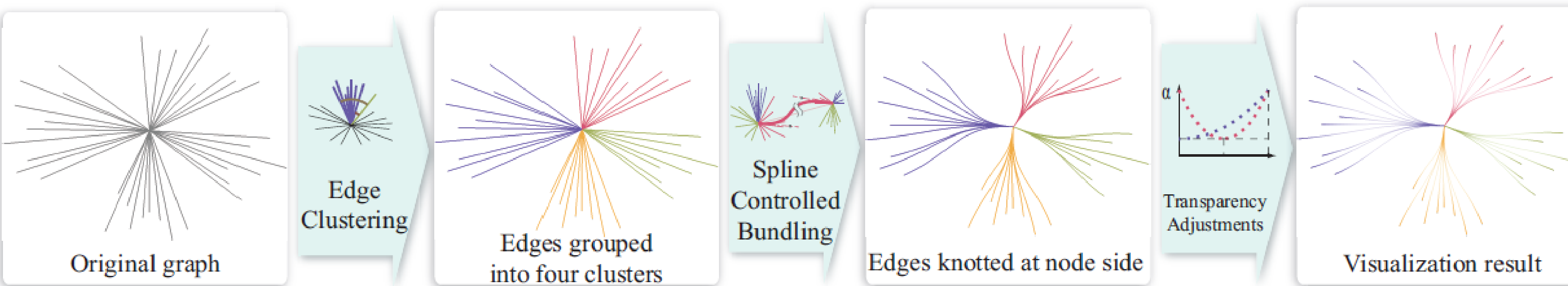
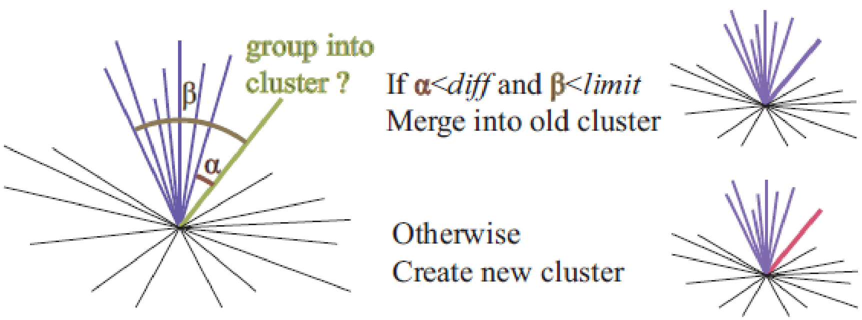
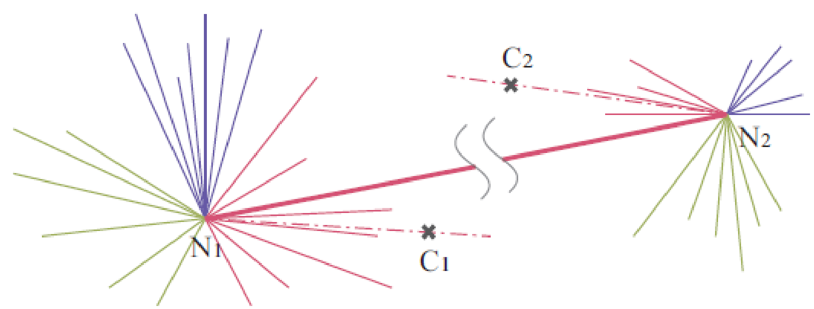
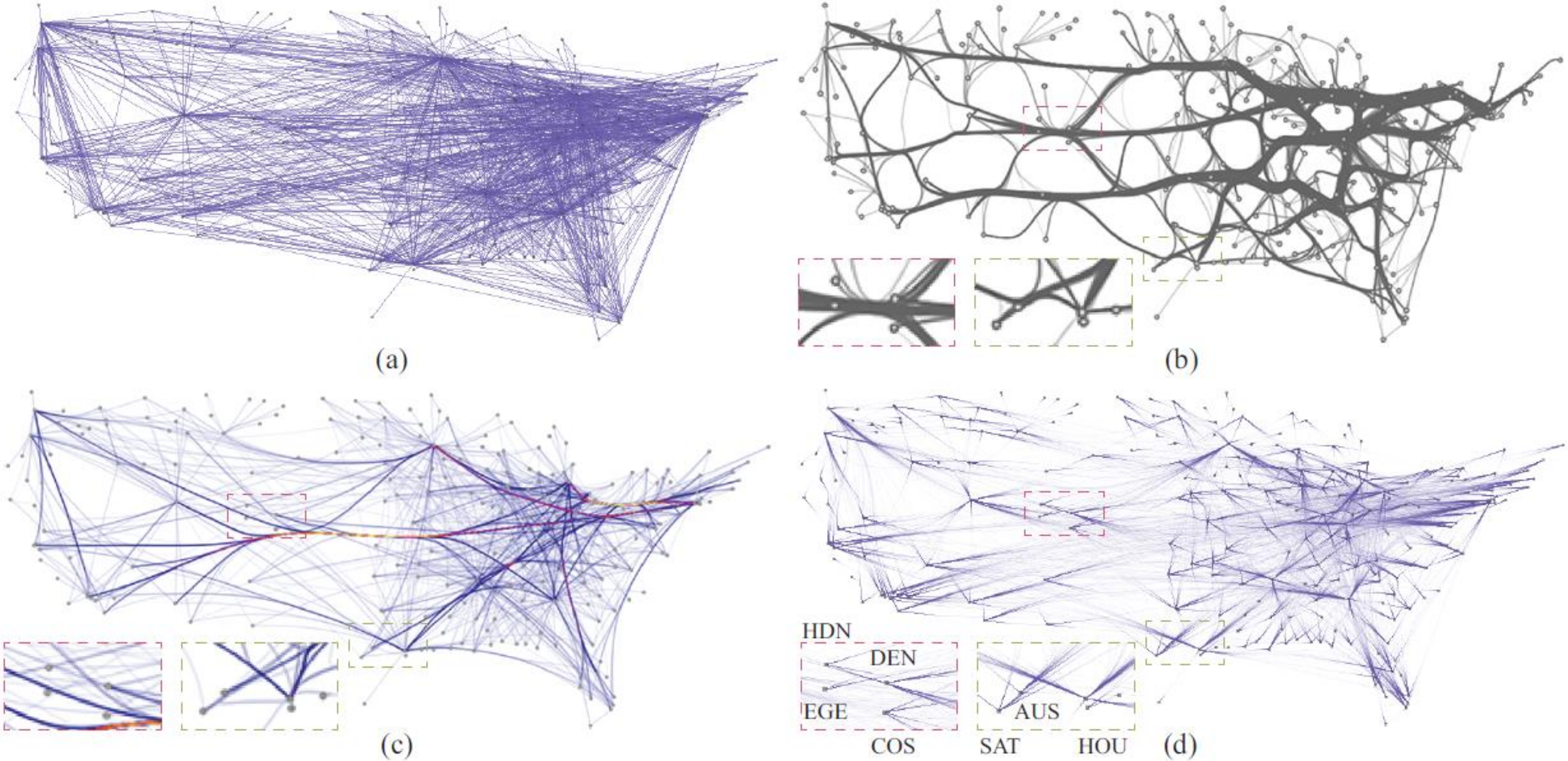


Figure 2: Processing pipeline of SideKnot. Edges are colored for illustration purpose only.







Edge bundling for U.S. airline route graph.

(a) Original graph with 235 nodes and 2101 edges.
(b) Bundled with GBEB
(c) Bundled with FDEB.
(d) Bundled with our method. Our visualization shows a number of middle-western cities (e.g. HDN, EGE, DEN, COS and more) have more routes to the east than to the west, suggesting their stronger connections with the eastern part of U.S

Multilevel Agglomerative Edge Bundling for Visualizing Large Graphs

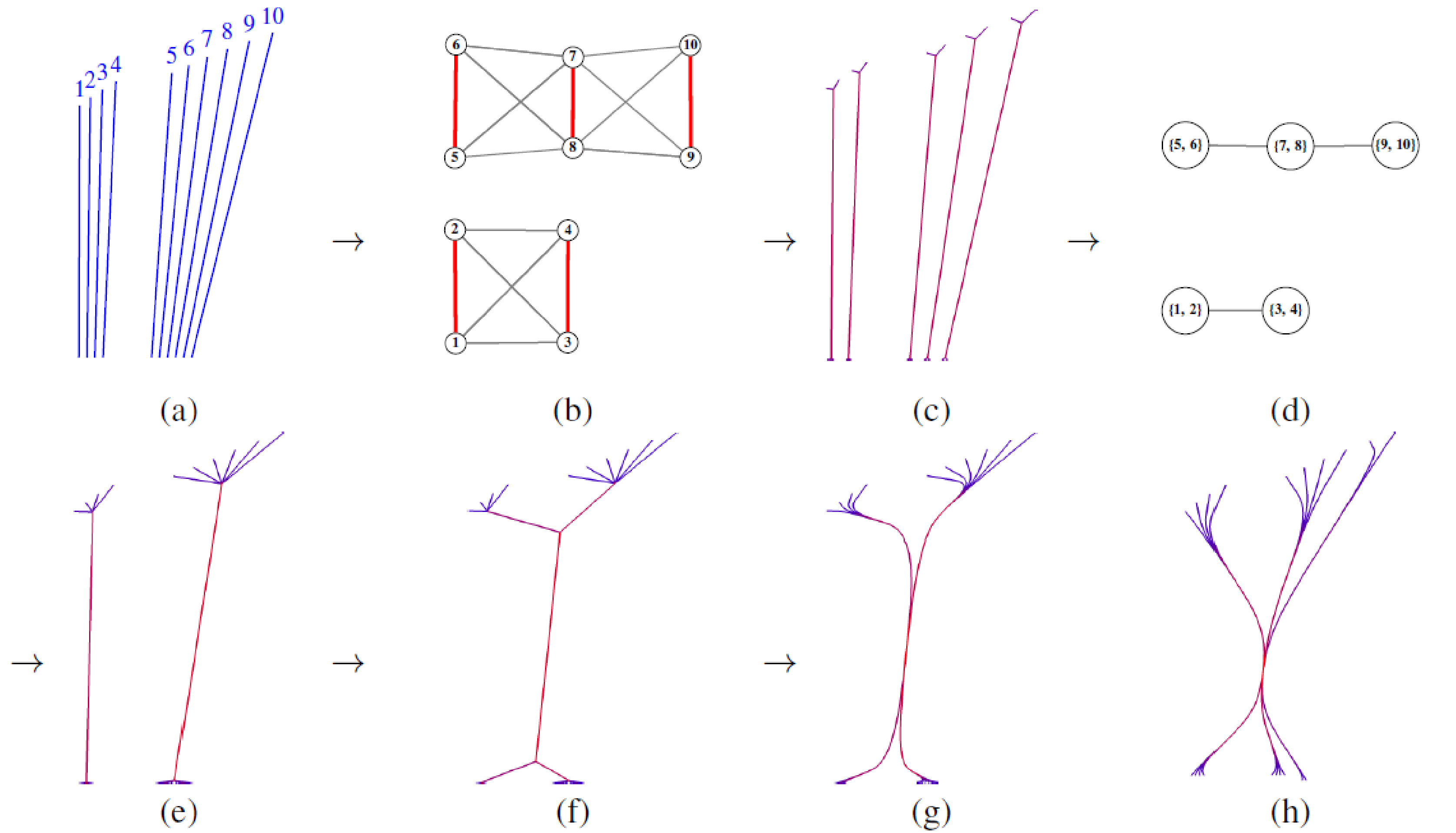
Emden R. Gansner Yifan Hu Stephen North Carlos Scheidegger

PacificVis 2011

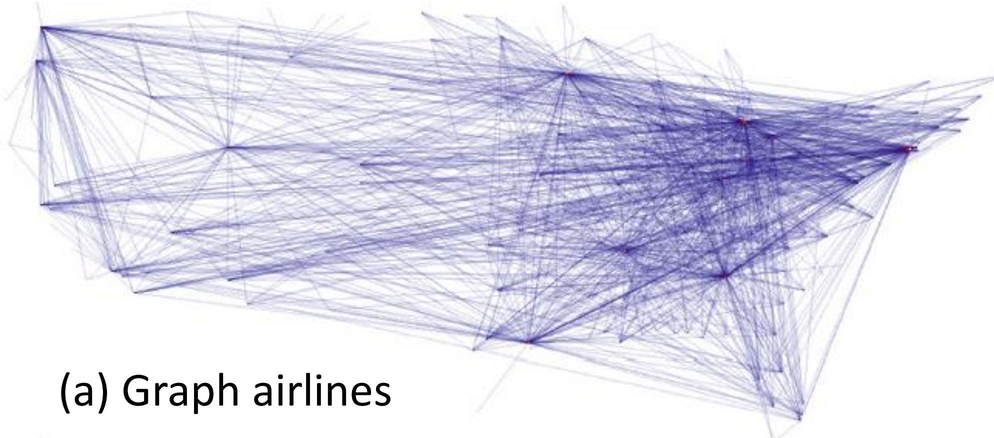
Ink ratio

“Data graphics (Data Visualization) visually display measured quantities by means of the combined use of points, lines, a coordinate system, numbers, symbols, words, shading, and color.”

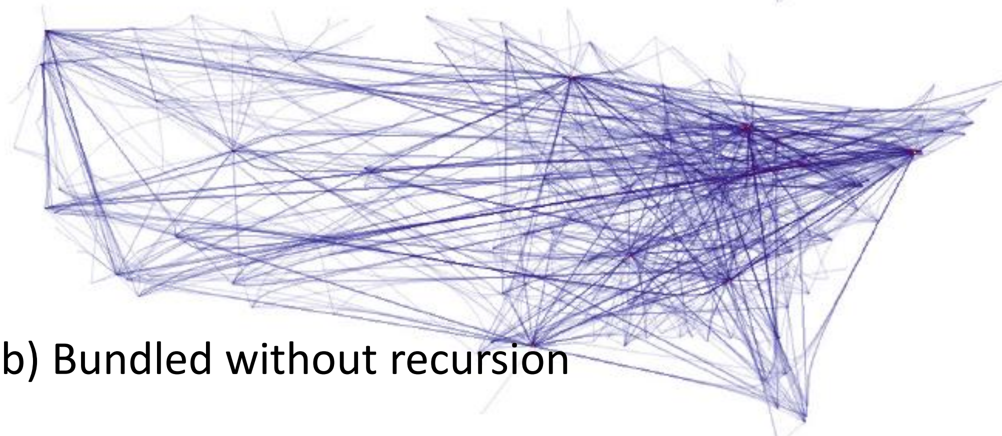
Edward R. Tufte



[Emden et at. 2011]



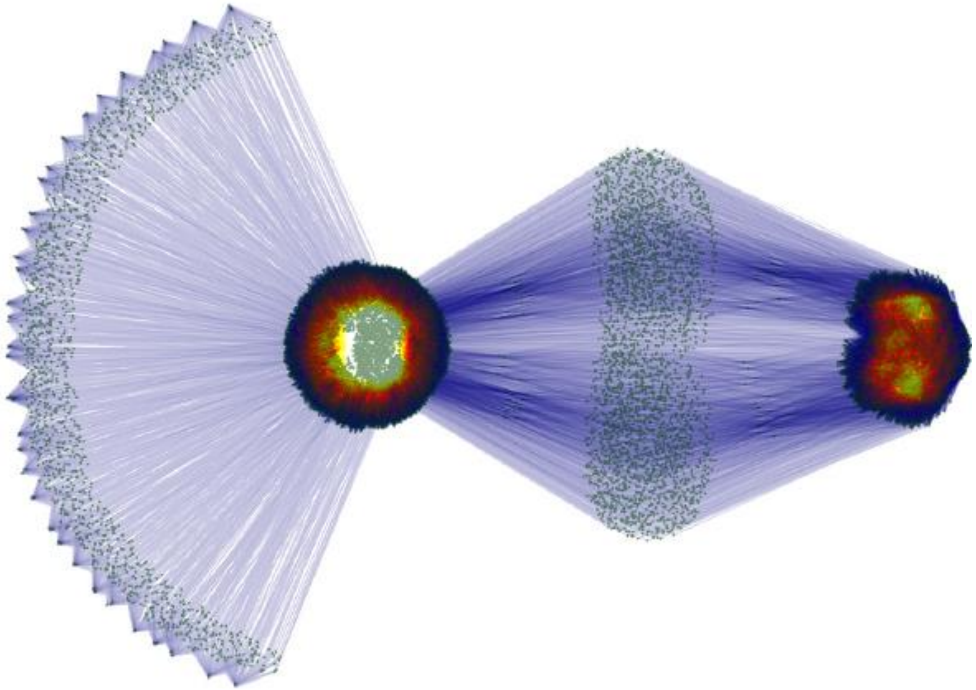
(a) Graph airlines



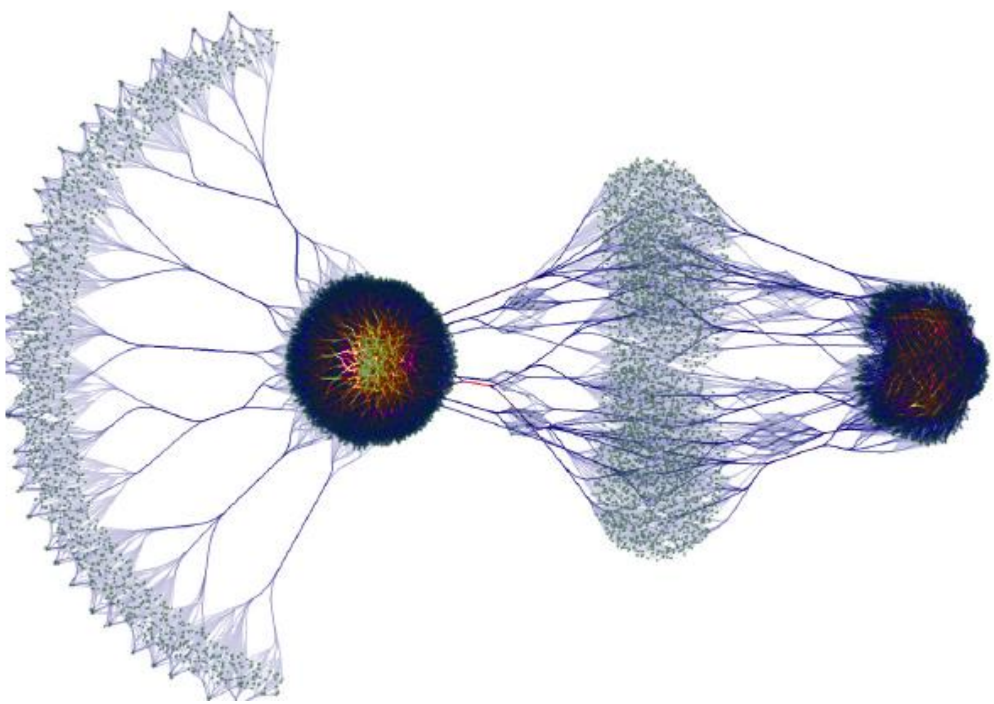
(b) Bundled without recursion

(c) With recursion.

[Emden et at. 2011]



Net50, 429 760 edges (87 sec)

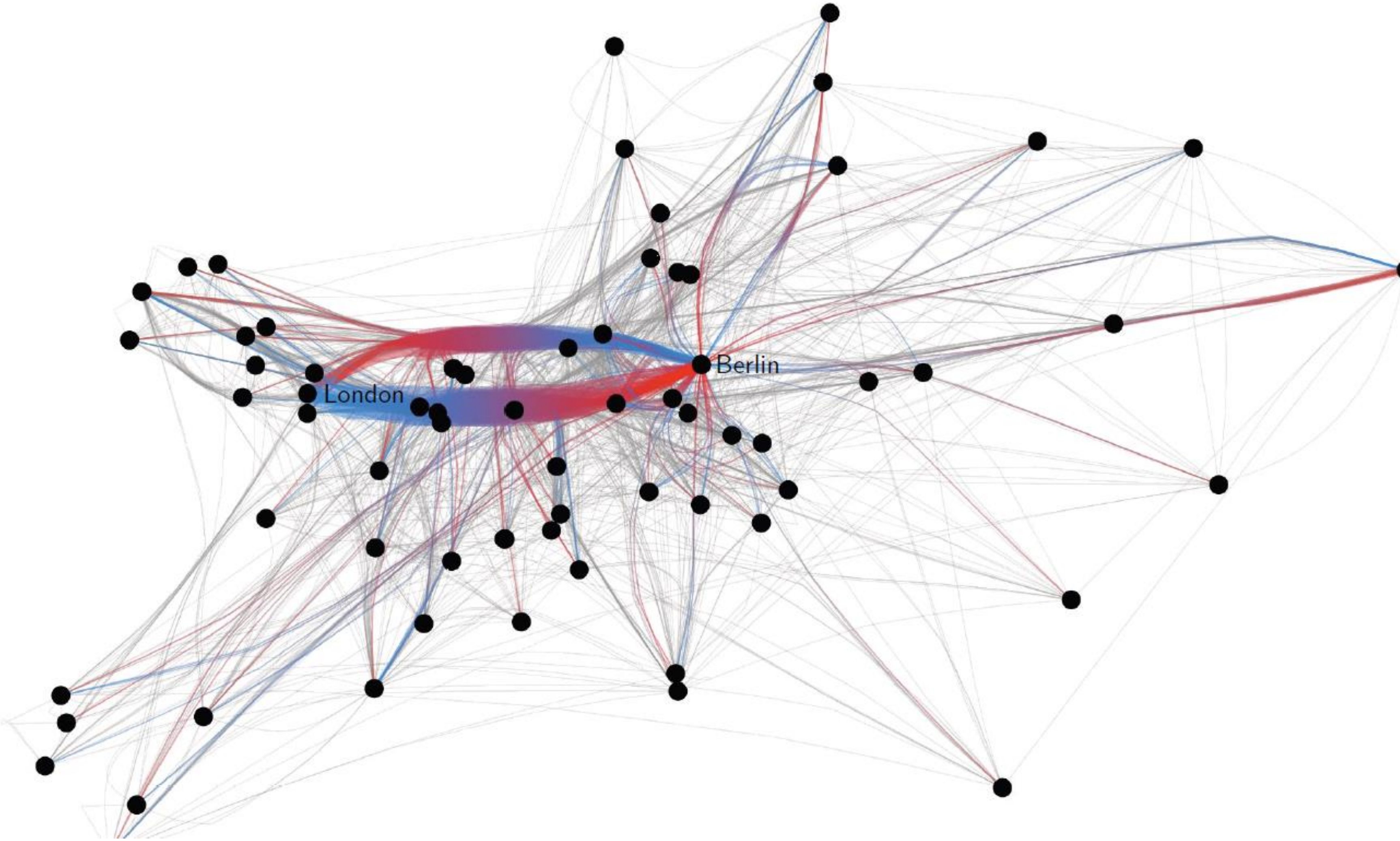


[Emden et at. 2011]

Divided Edge Bundling for Directional Network Data

David Selassie, Brandon Heller and Jeffrey Heer

Infovis 2011



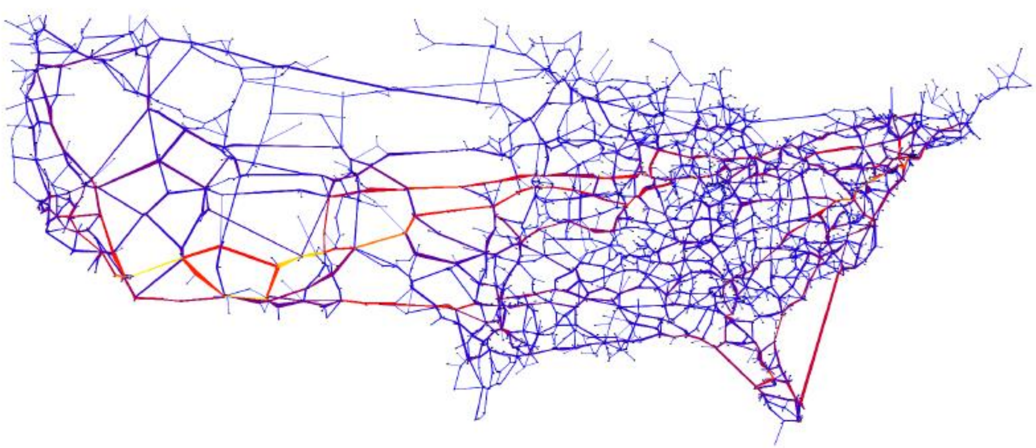
[Selassie et at. 2011]

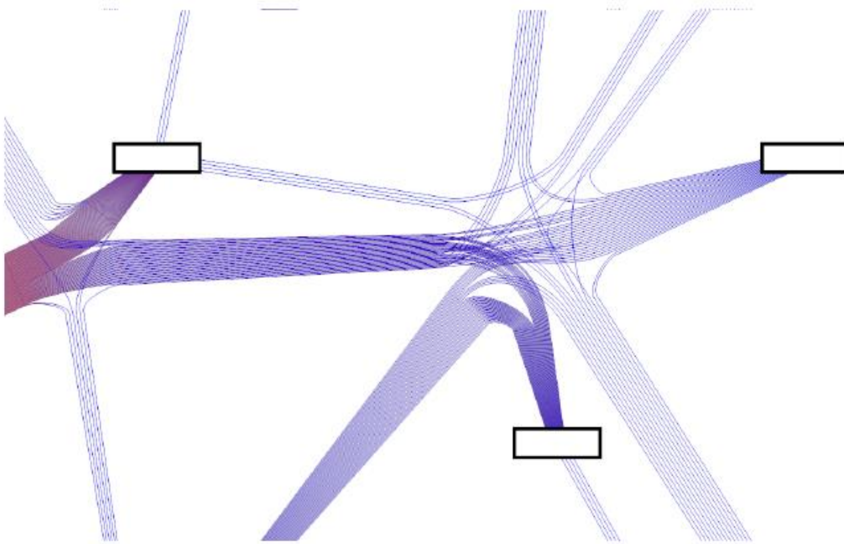
(a) Unbundled (c) Force-directed edge bundling (d) Divided edge bundling

Edge Routing with Ordered Bundles

Sergey Pupyrev, Lev Nachmanson,
Sergey Bereg, and Alexander E. Holroyd

Graph Drawing 2012





Sum it up!

Holten bundled edges in compound graphs by routing edges along the hierarchy layout using B-splines [HEB].

Control meshes are used to route curved edges, a Delaunay-based extension called geometric-based edge bundling (GBEB).

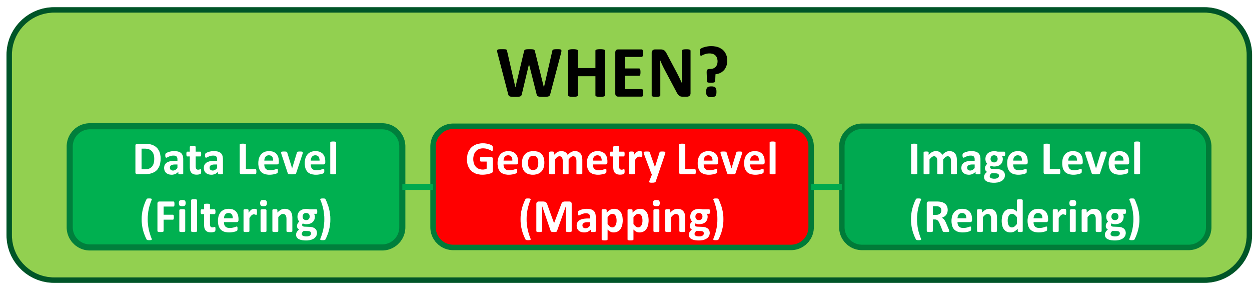
Winding roads' (WR) which use Voronoi diagrams for 2D and 3D layouts.

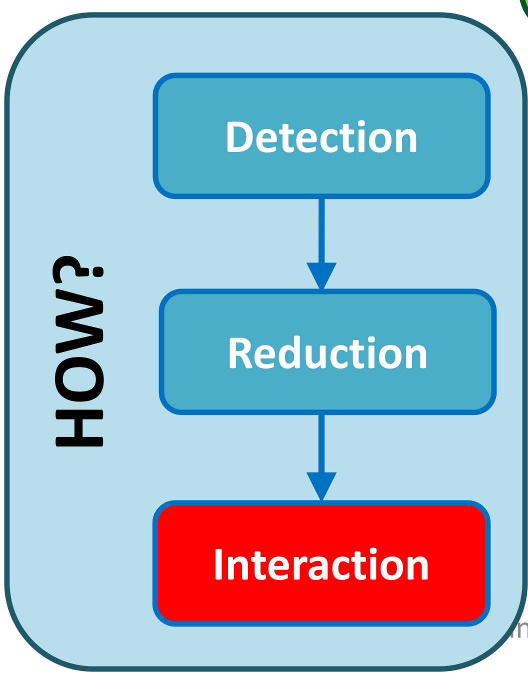
Force-directed edge bundling (FDEB) creates bundles by attracting edge control points, and was adapted to separate opposite-direction bundles [Divided edge bundling for directional network data].

The MINGLE method uses multilevel clustering to accelerate the bundling process.

Edge Routing uses optimization to produce bundles

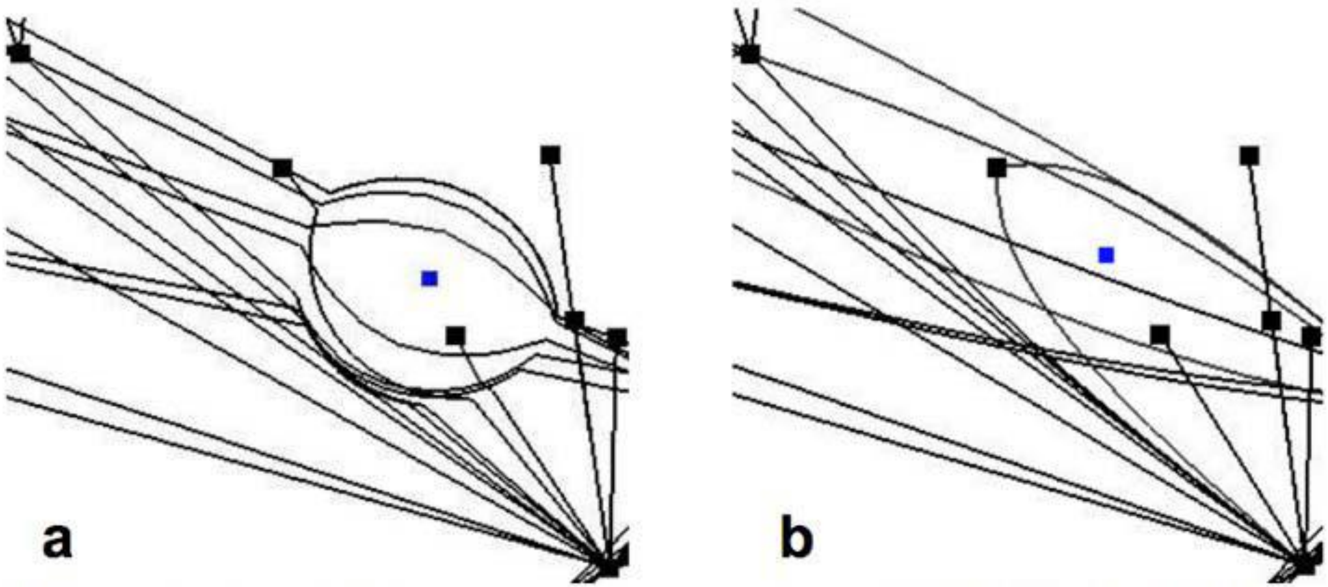
GEOMETRY LEVEL INTERACTION





EdgeLens: an interactive method for managing edge congestion in graphs

Wong, Nelson and Carpendale, Sheelagh
and Greenberg, Saul
Infovis2003

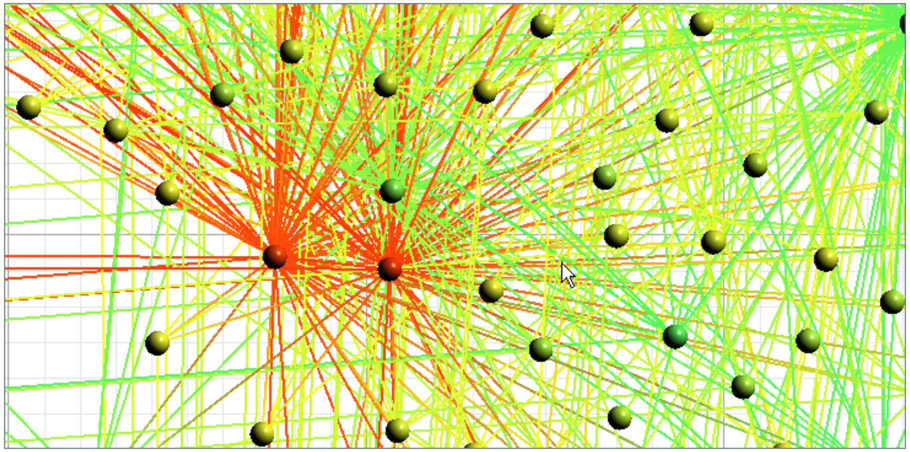


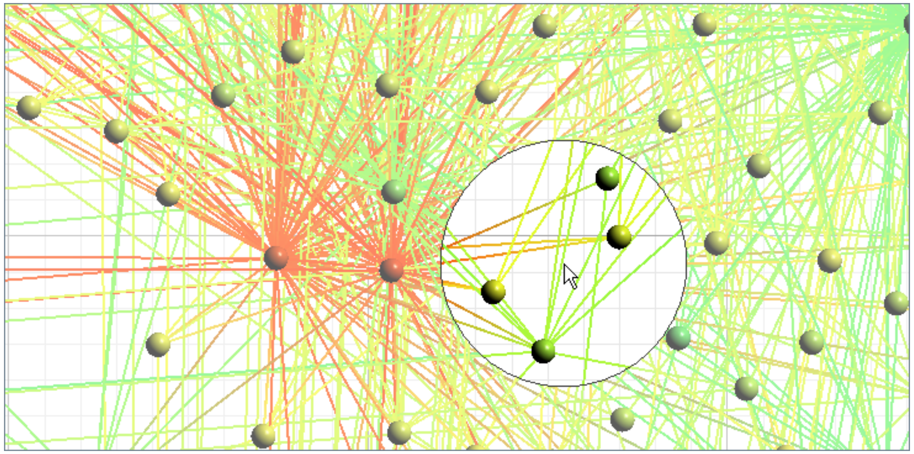
Interaction-based techniques such as EdgeLens and Edge Plucking can be used to curve graph edges away from a user's focus of attention without changing node positions.

Fisheye Tree Views and Lenses for Graph Visualization

Christian Tominski, James Abello, Frank van Ham, Heidrun Schumann

IV 2006

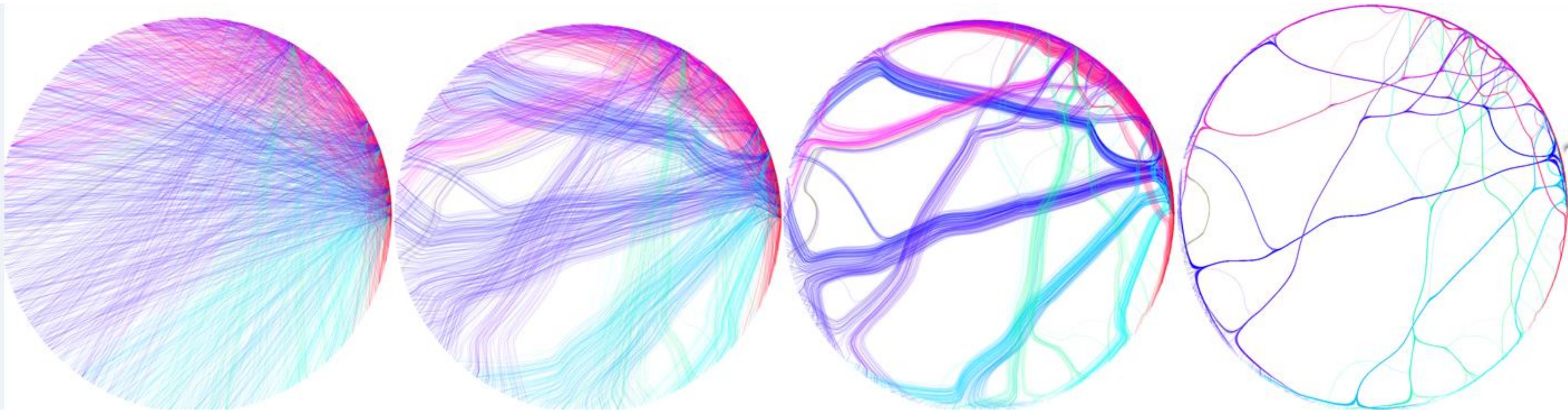




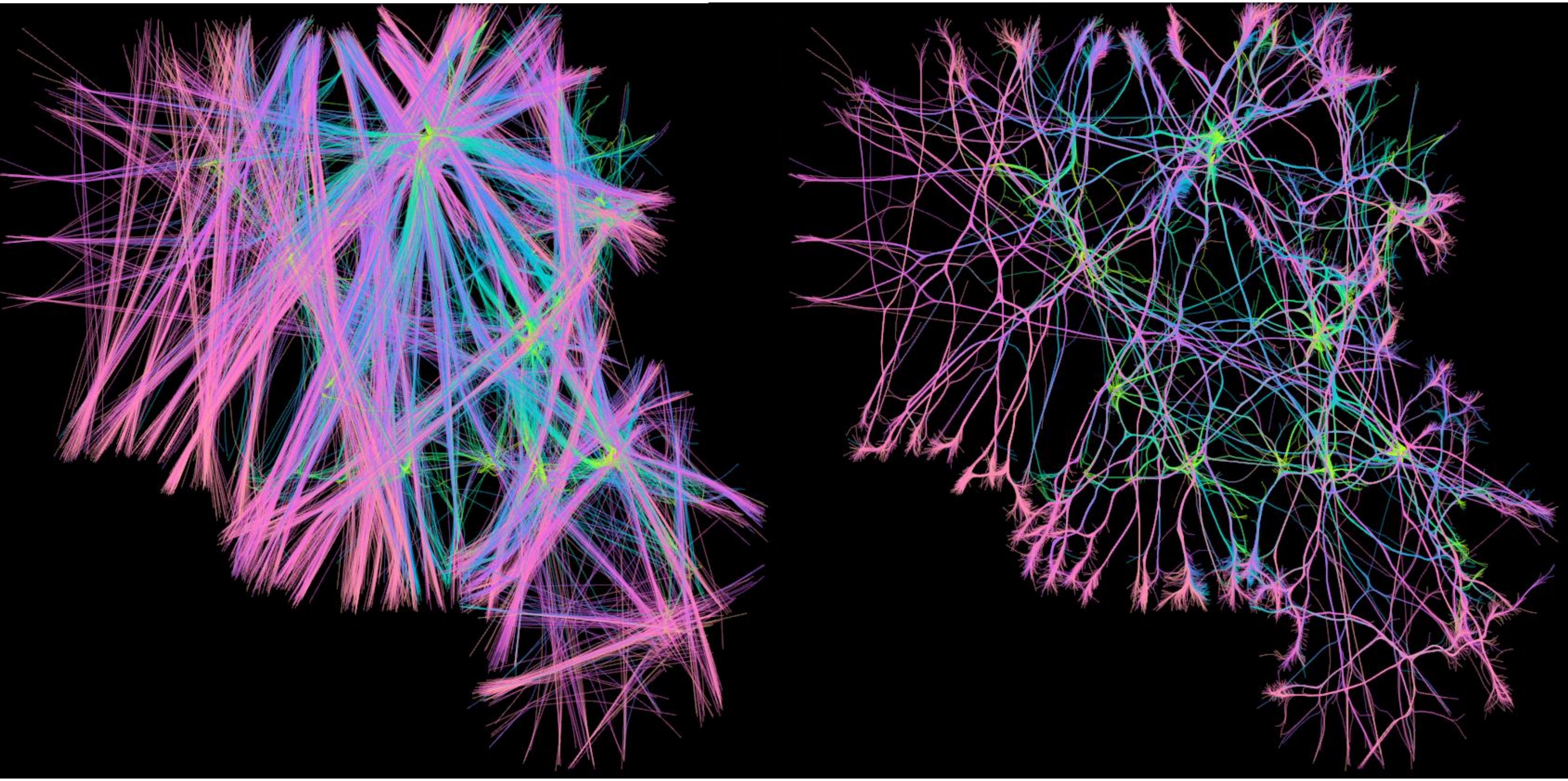
MoleView: An Attribute and Structure-based Semantic Lens for Large

C. Hurter, O. Ersoy, A. Telea

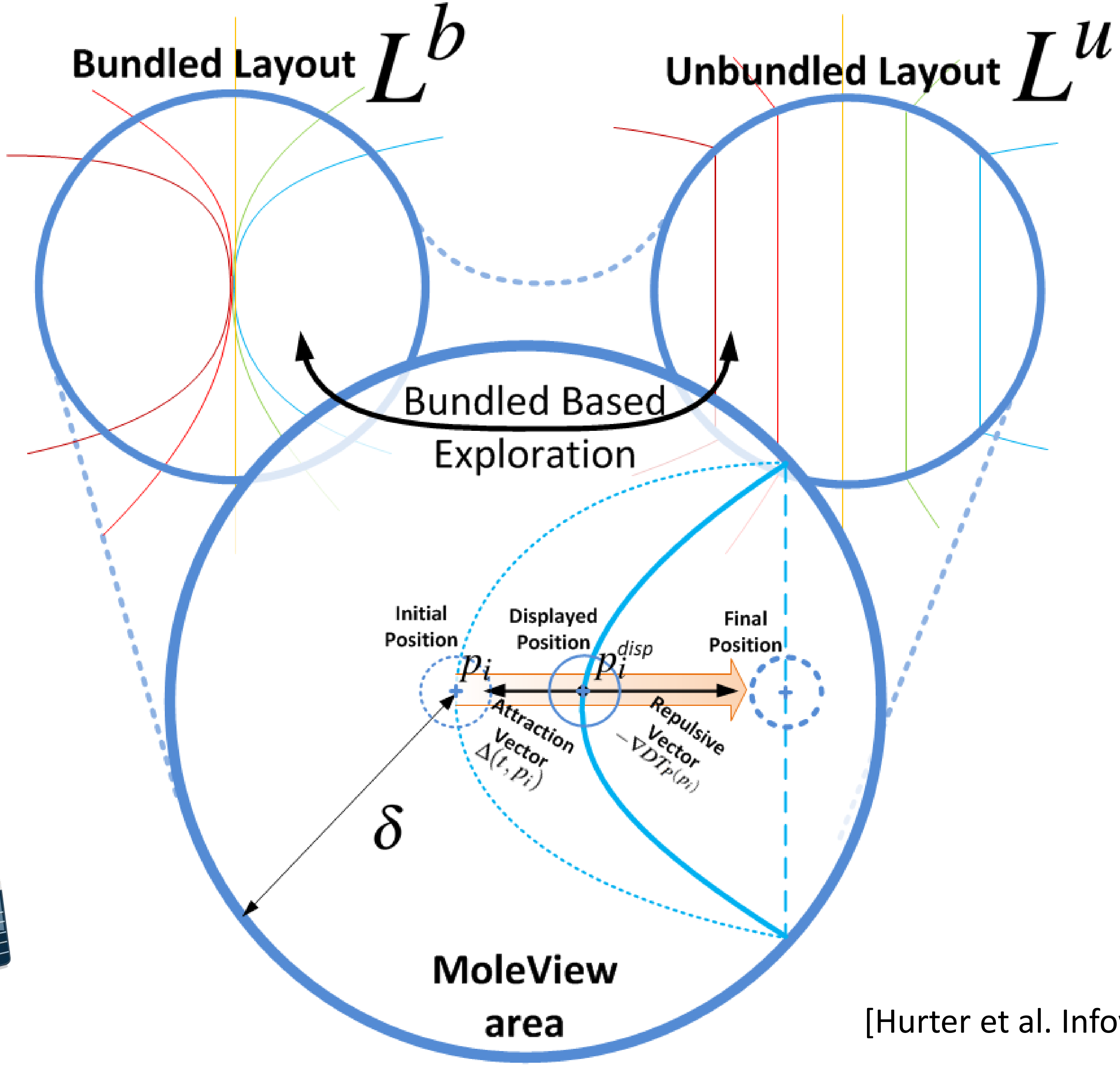
Infovis 2011



Demo: Bundle-based exploration



[Hurter et al. Infovis 2011]



[Hurter et al. Infovis 2011]

Exploring the Design Space of Interactive Link Curvature in Network Diagrams

Nathalie Henry Riche, Tim Dwyer,
Bongshin Lee ,Sheelagh Carpendale

AVI 2012

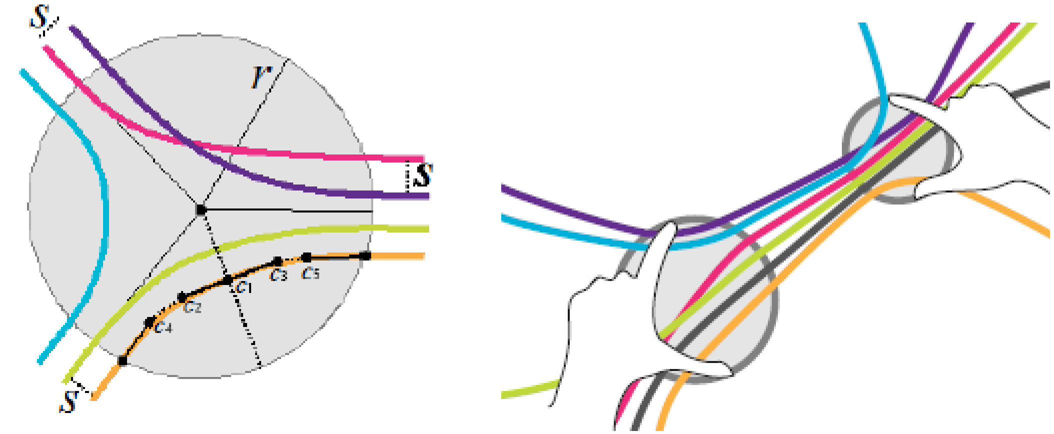
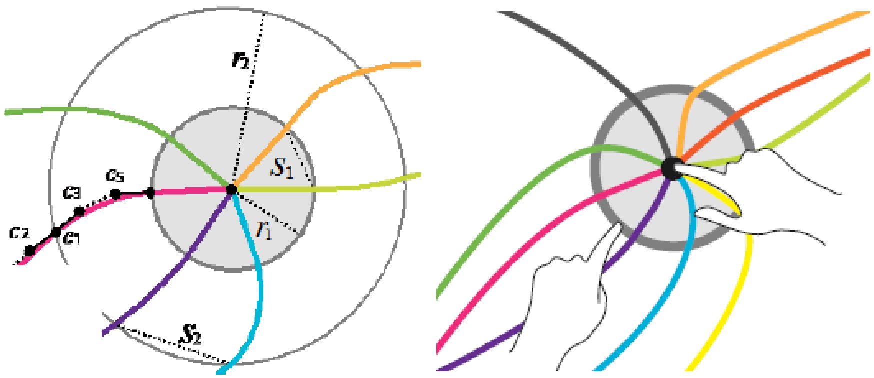


Figure 4. (a) Constructution of a bundling control point, (b) example multi-touch gesture to control link bundles.



5. (a) Construction of link fanning, (b) example of ›uch gesture to control link fanning.

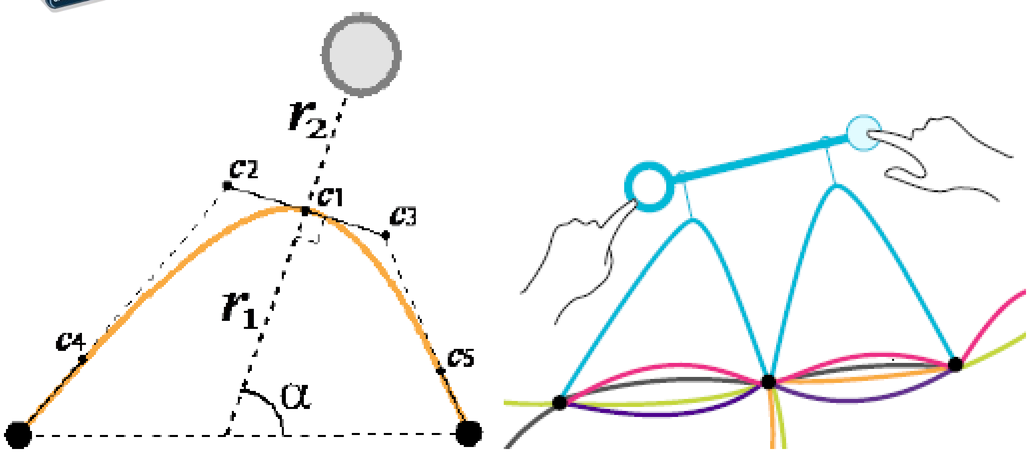


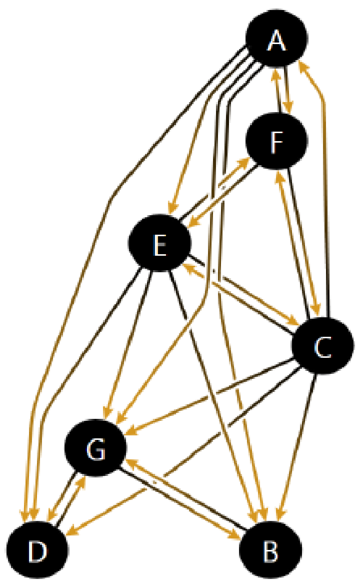
Figure 6. (a) Construction of a curve attracted by a magnet, (b) example of multi-touch gesture to control a line-magnet.

[Henry Riche et al. AVI 2012]

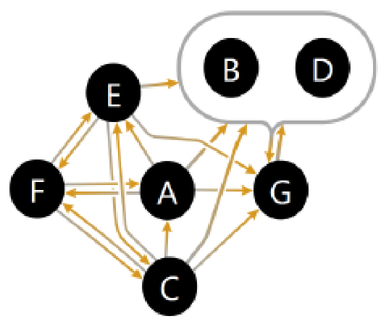
Edge Compression Techniques for Visualization of Dense Directed Graphs

Tim Dwyer, Nathalie Henry Riche,
Kim Marriott, and Christopher Mears

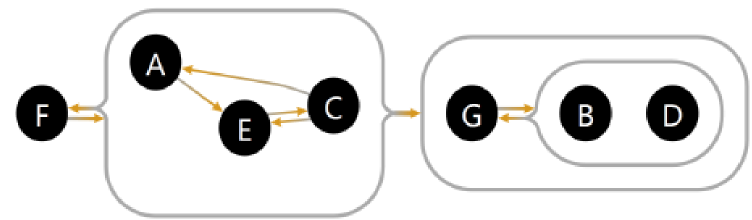
Infovis 2013



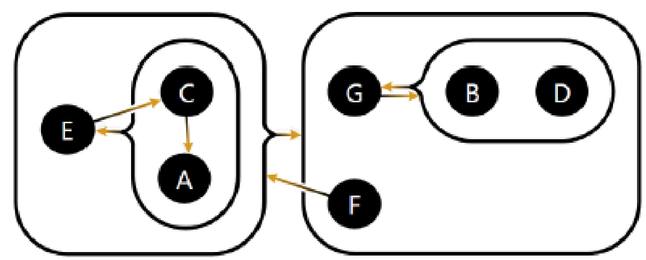
(a) A standard "flat" node-link diagram of a graph with 23 directed edges.



(b) Since B and D have exactly the same sets of neighbors they can be grouped leaving 18 edges using a simple *Neighbor Matching*.



(c) A *Modular Decomposition* allows internal structure within modules and nesting. Nine edges remain.



(d) A *Power-Graph Decomposition* further relaxes the definition of a module to allow edges to cross module boundaries, allowing the same graph to be drawn with only 7 edges.

Ambiguity-Free Edge-Bundling for Interactive Graph Visualization

Sheng-Jie Luo, Chun-Liang Liu, Bing-Yu Chen, and Kwan-Liu Ma

TVCG 2012

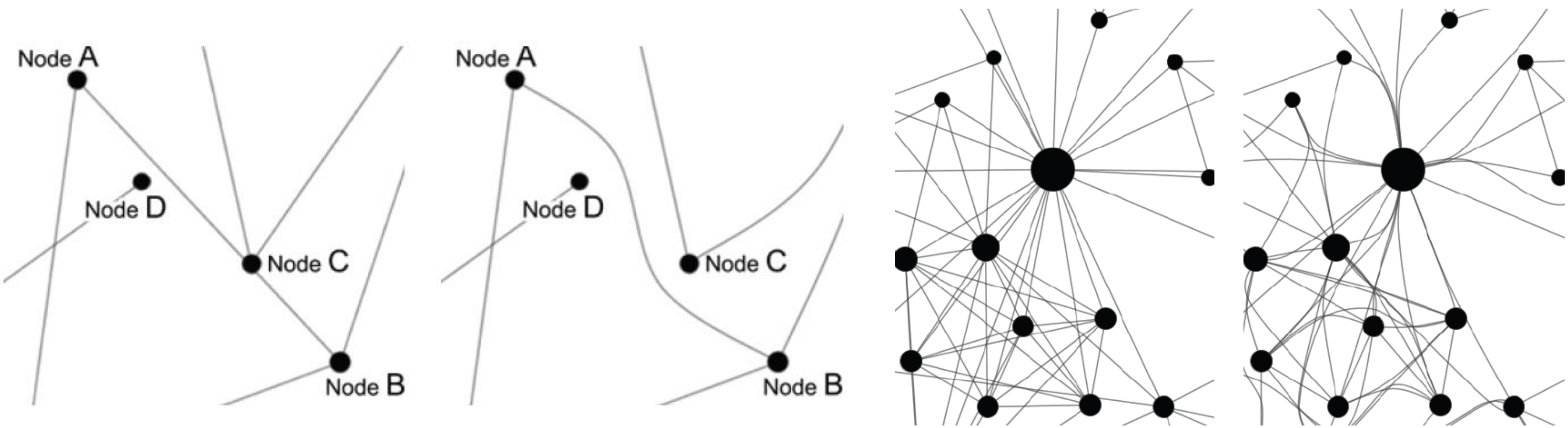


IMAGE LEVEL

WHEN?

Data Level (Filtering) – Geometry Level (Mapping) – Image Level (Rendering)

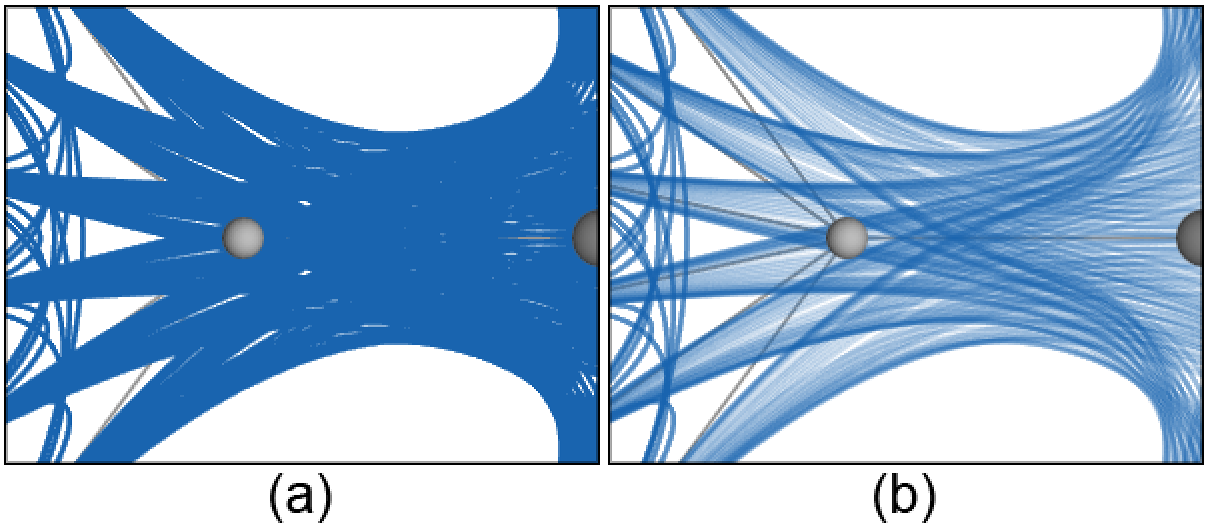
Thanks to the recent improvements regarding graphic hardware and its flexible usage, image level methods are nowadays very popular. Graphic cards can be used to improve rendering aesthetics and to address scalability issues.

Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data

Danny Holten. 2006. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (September 2006), 741-748.

Pixel based technique

to distinguish individual curves or subbundles within a bundle

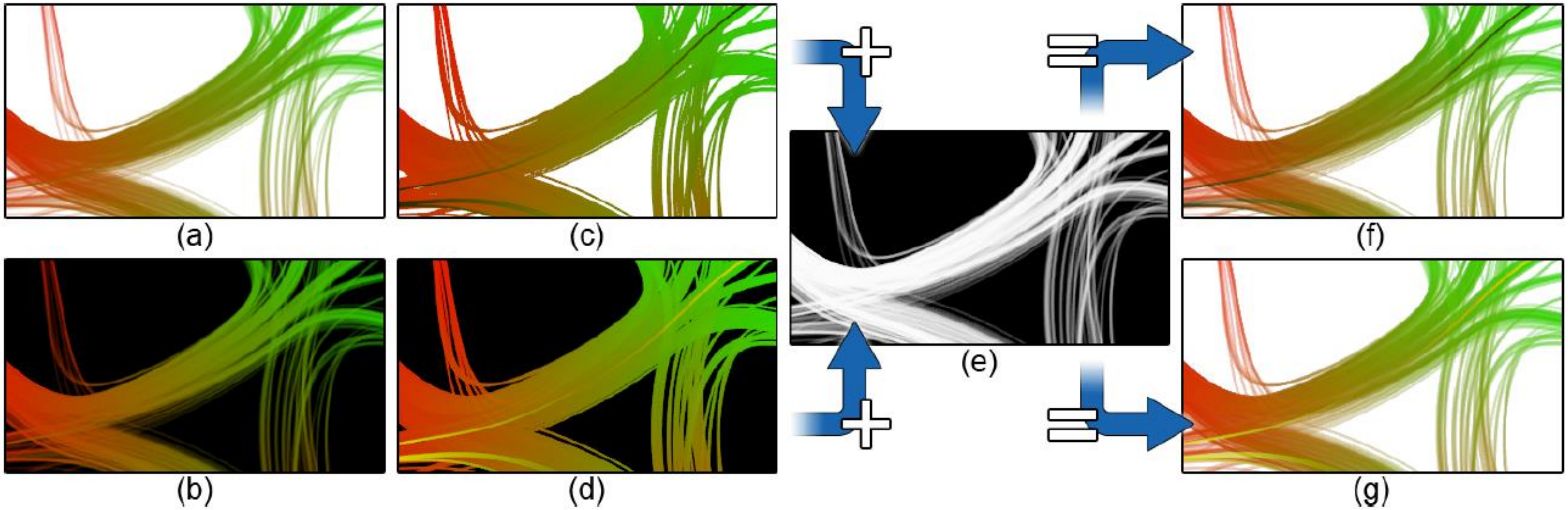


(a) Alpha blending disabled

(b) Alpha blending enabled

Holten Infovis 2006

Pixel based technique

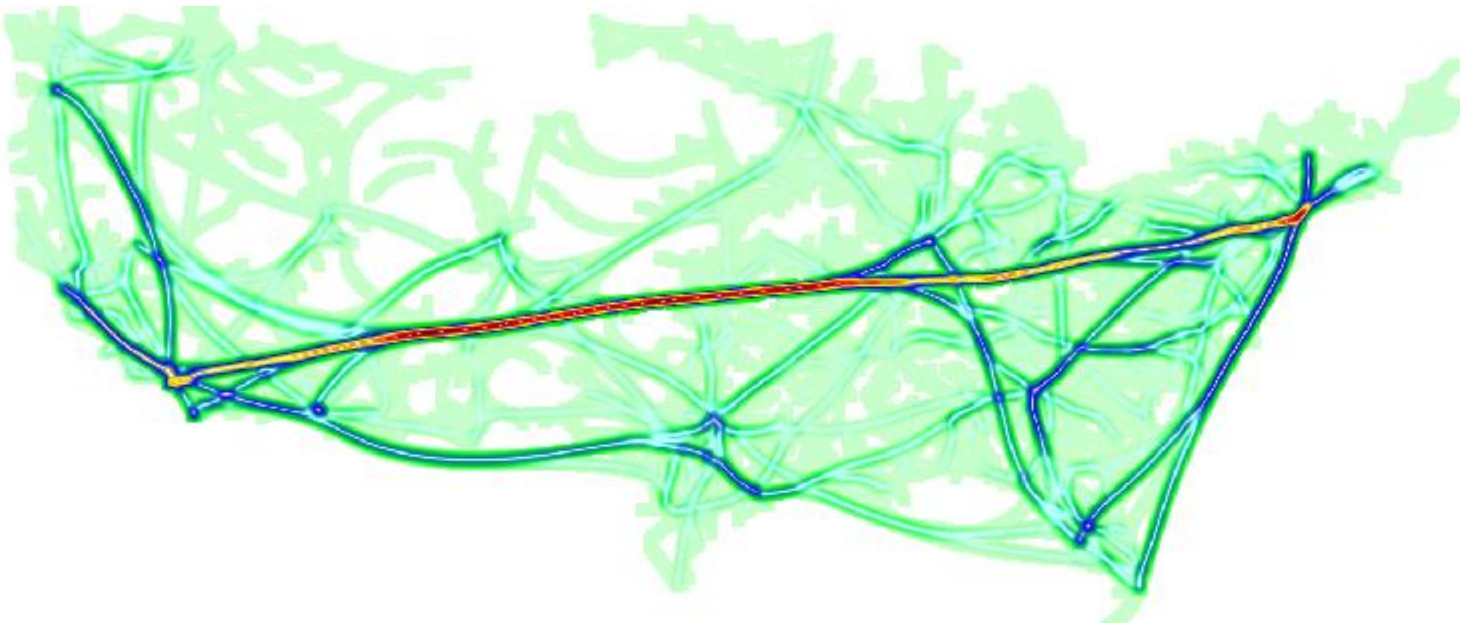


Using OpenGL's EXT blend minmax extension:

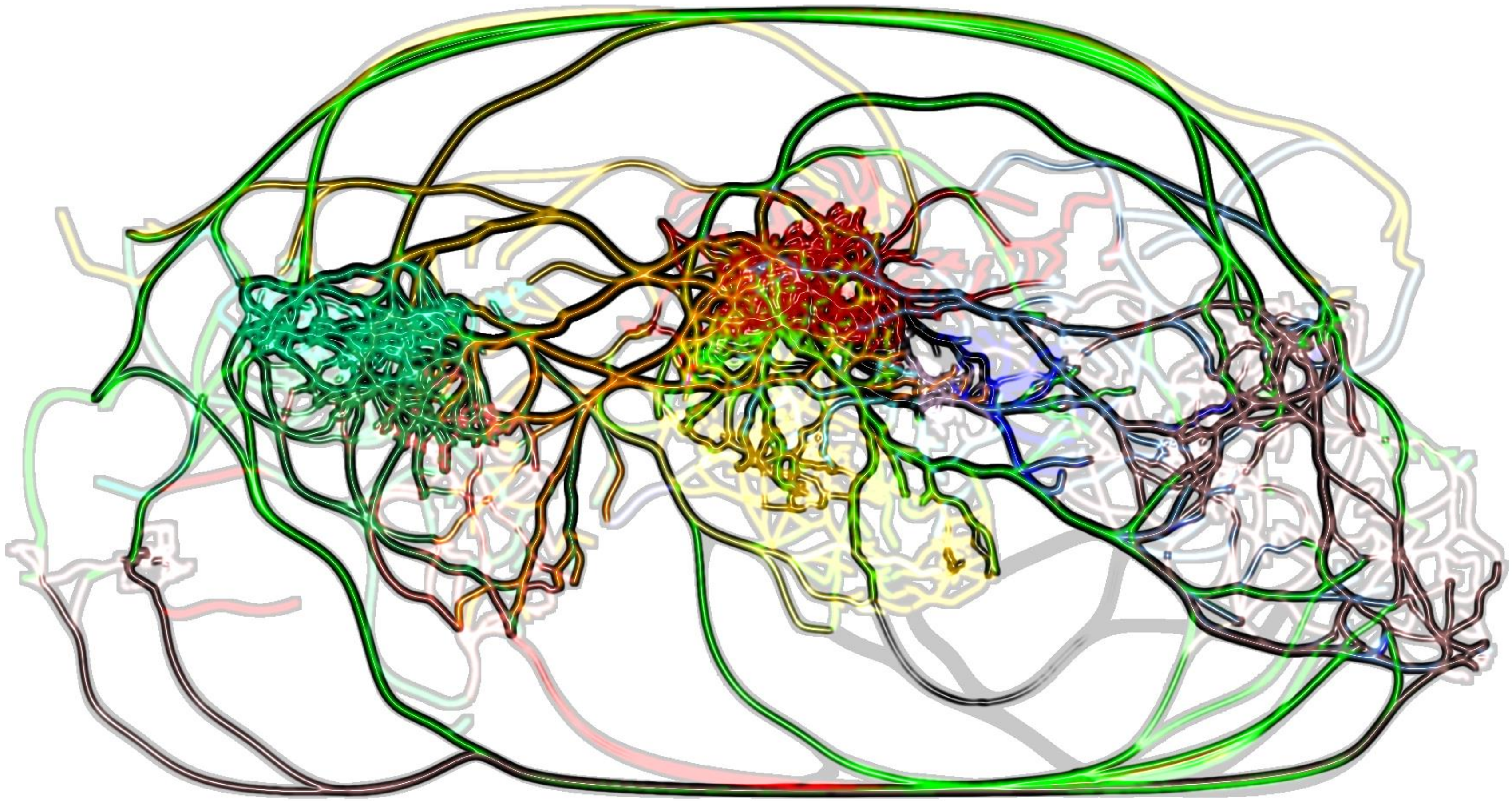
- (a) and (b) show normal alpha blending
- (c) and (d) show the usage of standard MIN EXT (minimum) and MAX EXT (maximum) blending
- (e) shows a transparency mask generated using normal alpha blending
- f) and (g) show how this provides these results with additional levels of opacity

Winding Roads
Routing edges into bundles

A. Lambert and R. Bourqui and D. Auber
Eurographics/ IEEE-VGTC Symposium on Visualization 2010



[A. Lambert et al. EuroVis 2010]



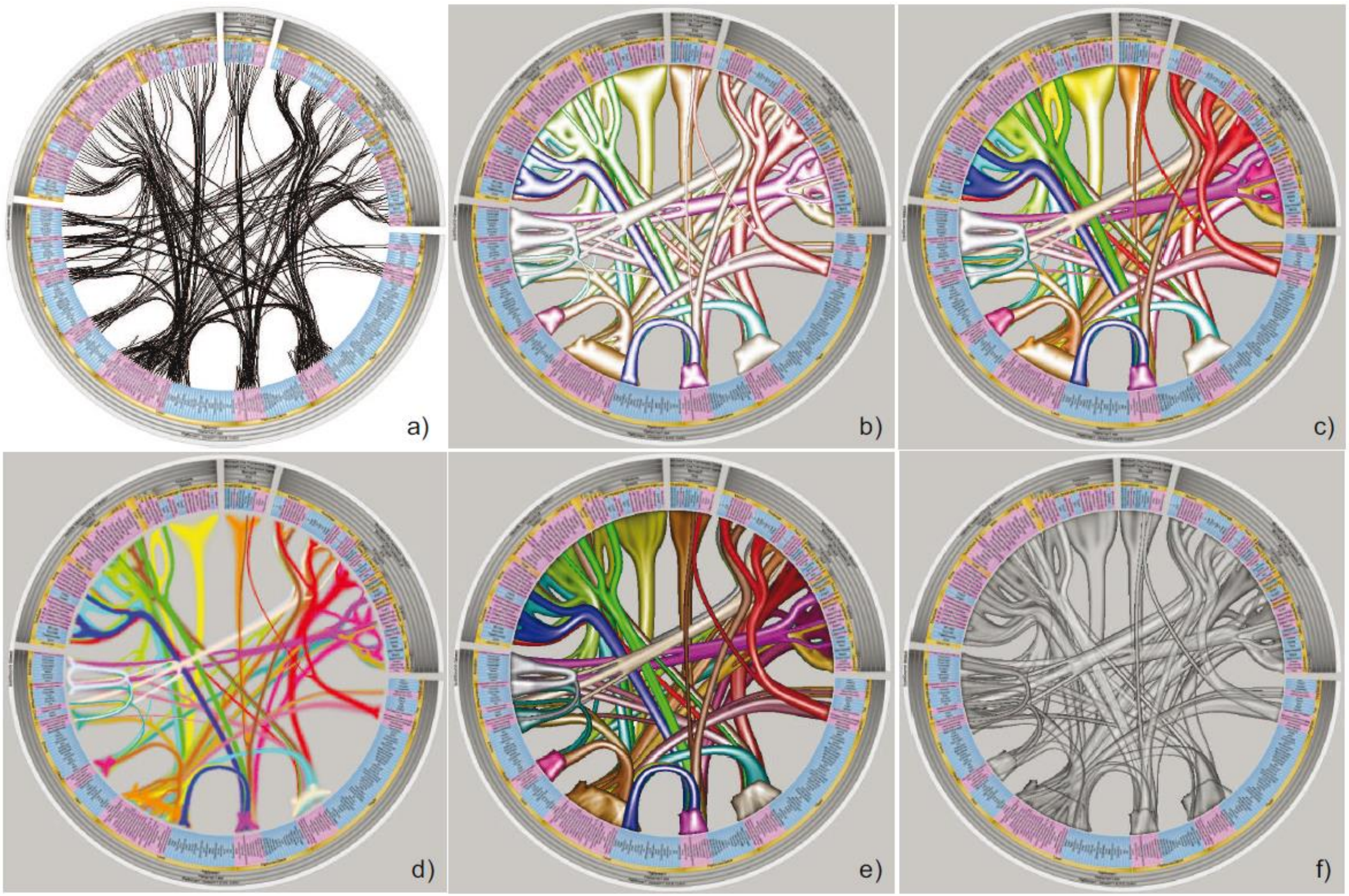
[A. Lambert et al. EuroVis 2010]

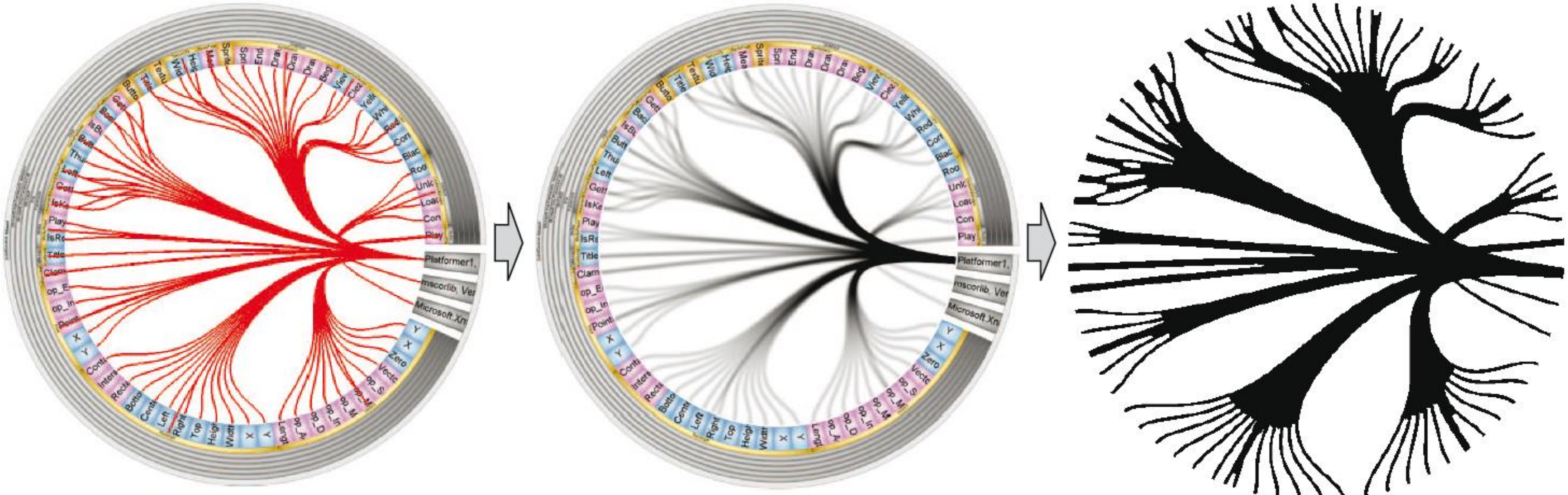
Image-Based Edge Bundles: Simplified Visualization of Large Graphs

A. Telea and O. Ersoy

Eurovis 2010

Image-Based Edge Bundles: Simplified Visualization of Large Graphs

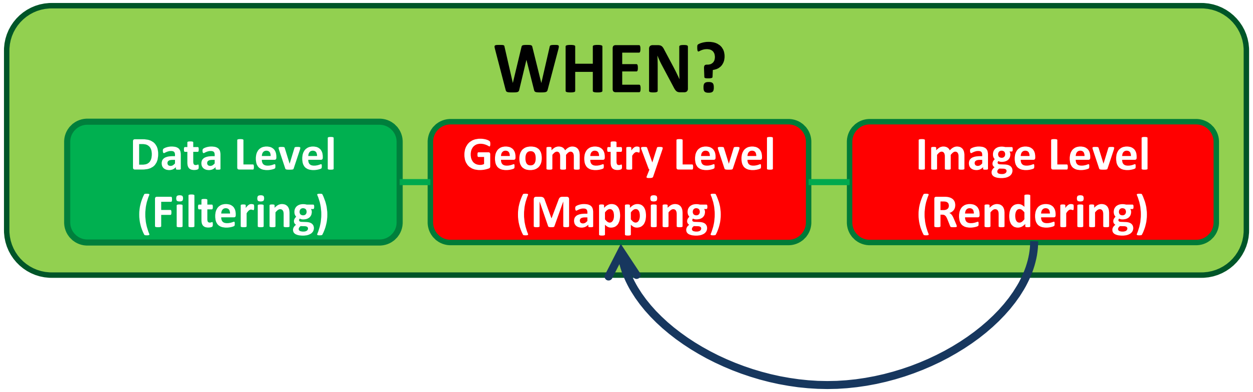




a) edge layout b) splatted image c) binary shape

a) edge layout b) shape I and skeleton Sk c) height profile H

GEOMETRY LEVEL
USING
IMAGE LEVEL

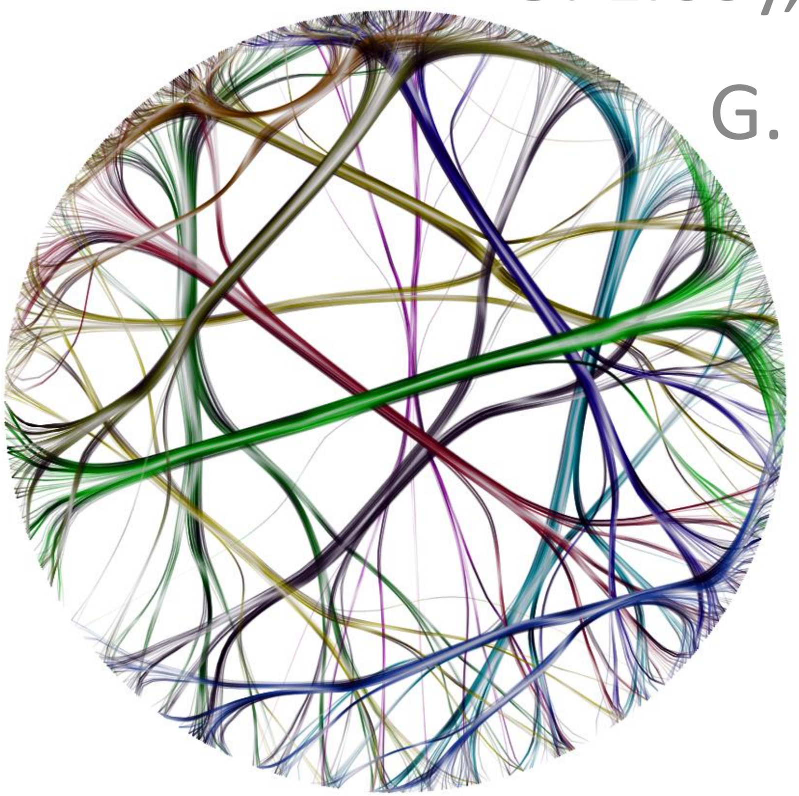


SBEB
Skeleton-based Edge Bundling

O. Ersoy, C. Hurter, F. V. Paulovich,

G. Cantareira, A. Telea

Infovis 2011



Edge Bundling

Node-link diagrams are widely used

Large number of nodes + edges = visual clutter!

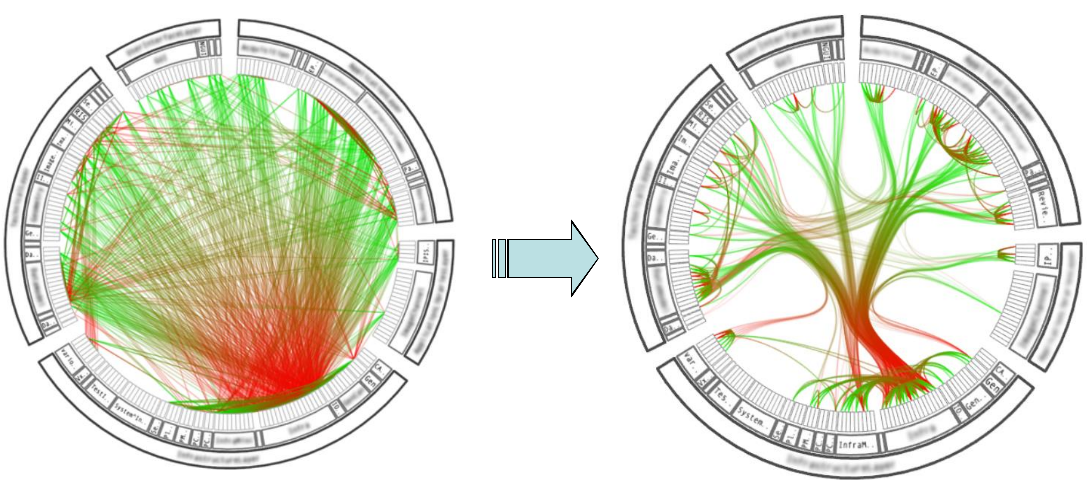
Edge bundling,

draws related edges close to each other increases white space between bundles.

trades clutter for overdraw

This technique is applicable to general graphs

(no hierarchy required)



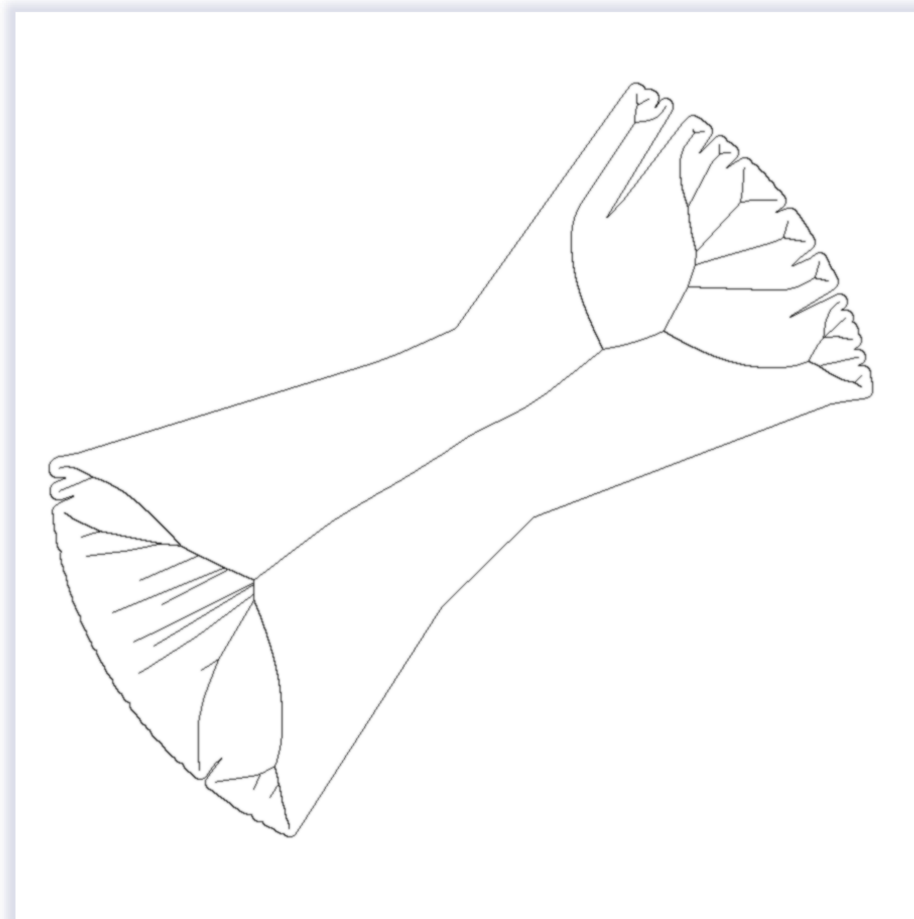
Our Idea: Use Skeletons

The skeleton,

is nicely centered in the shape

obeys the shape's topology
(branches, loops, etc)

"Use skeletons
to direct edge bundling"



Skeleton-based Edge Bundling

At each iteration:

cluster the edges,

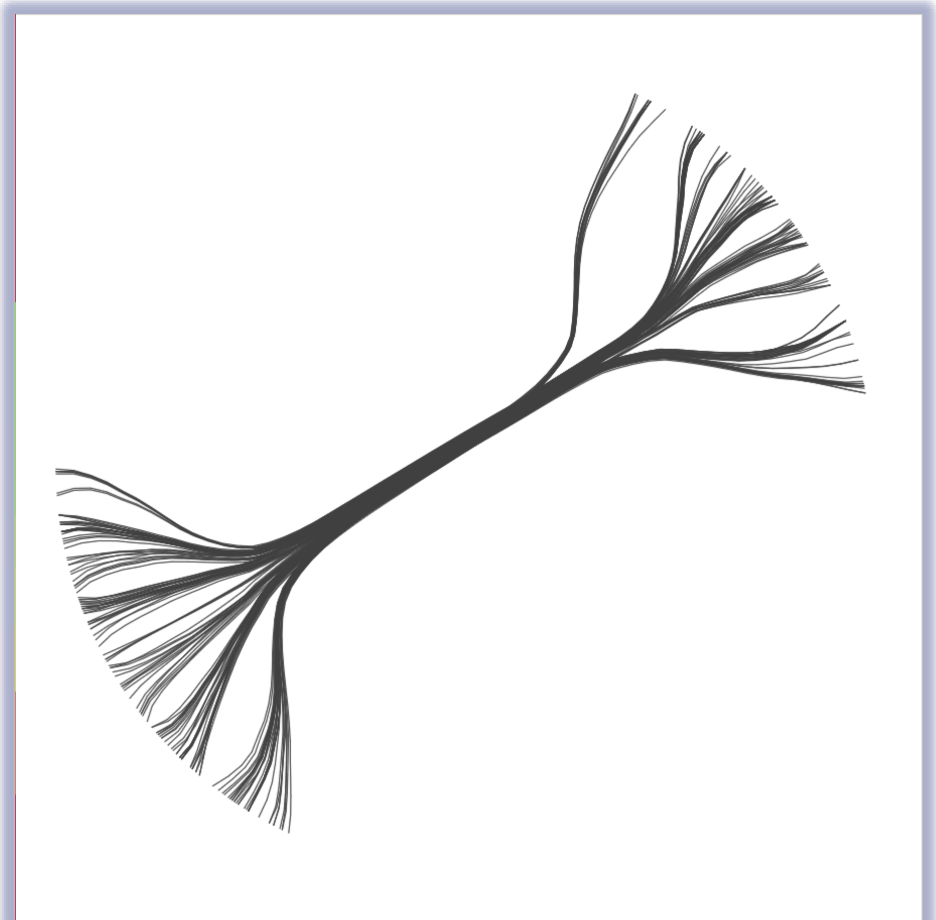
Then for each cluster:

inflate + threshold edges to get a thin shape Ω using DT,

compute the skeleton S_Ω,

compute the feature transform FT_S,

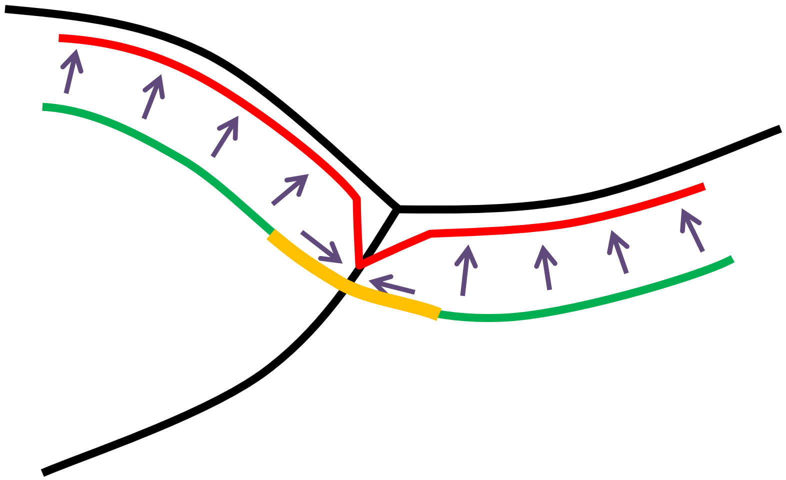
attract the edges to S_Ω using FT_S



Edge Attraction: Singularities

When we attract all edge points to their closest skeleton point, singularities appear

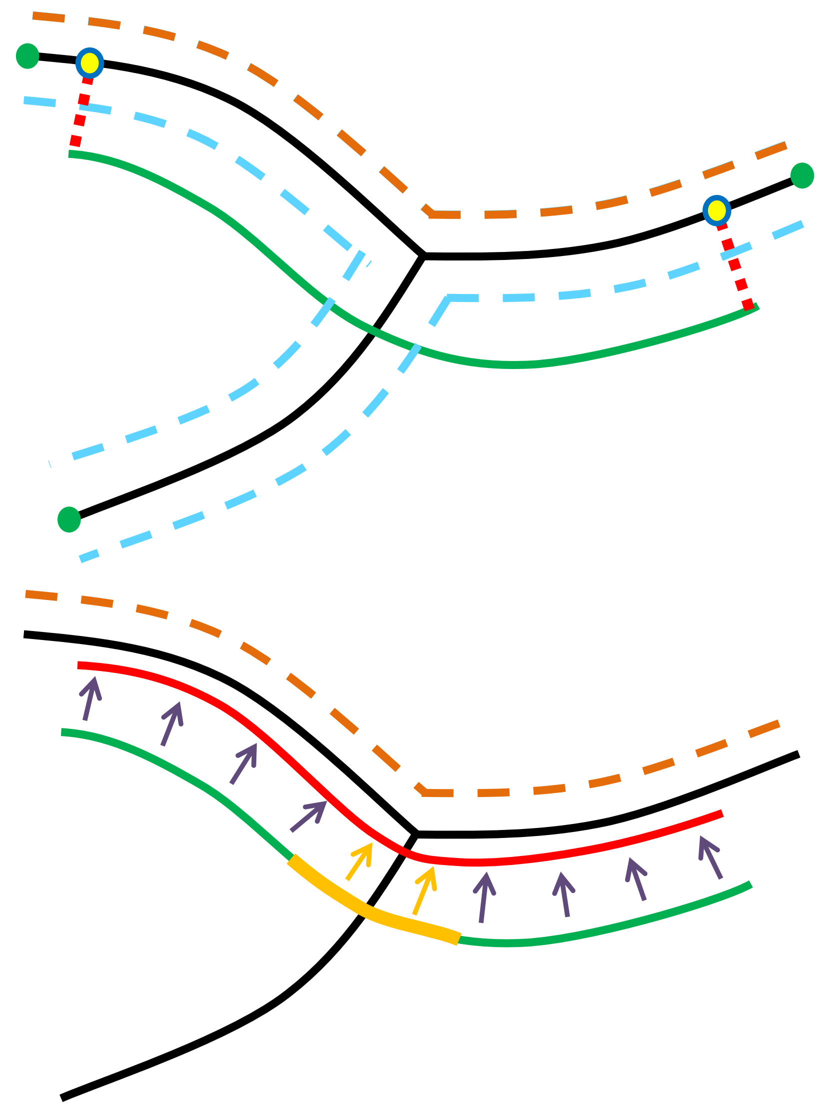
The singularities cause undesired kinks



Edge Attraction: Singularities

For edge attraction:

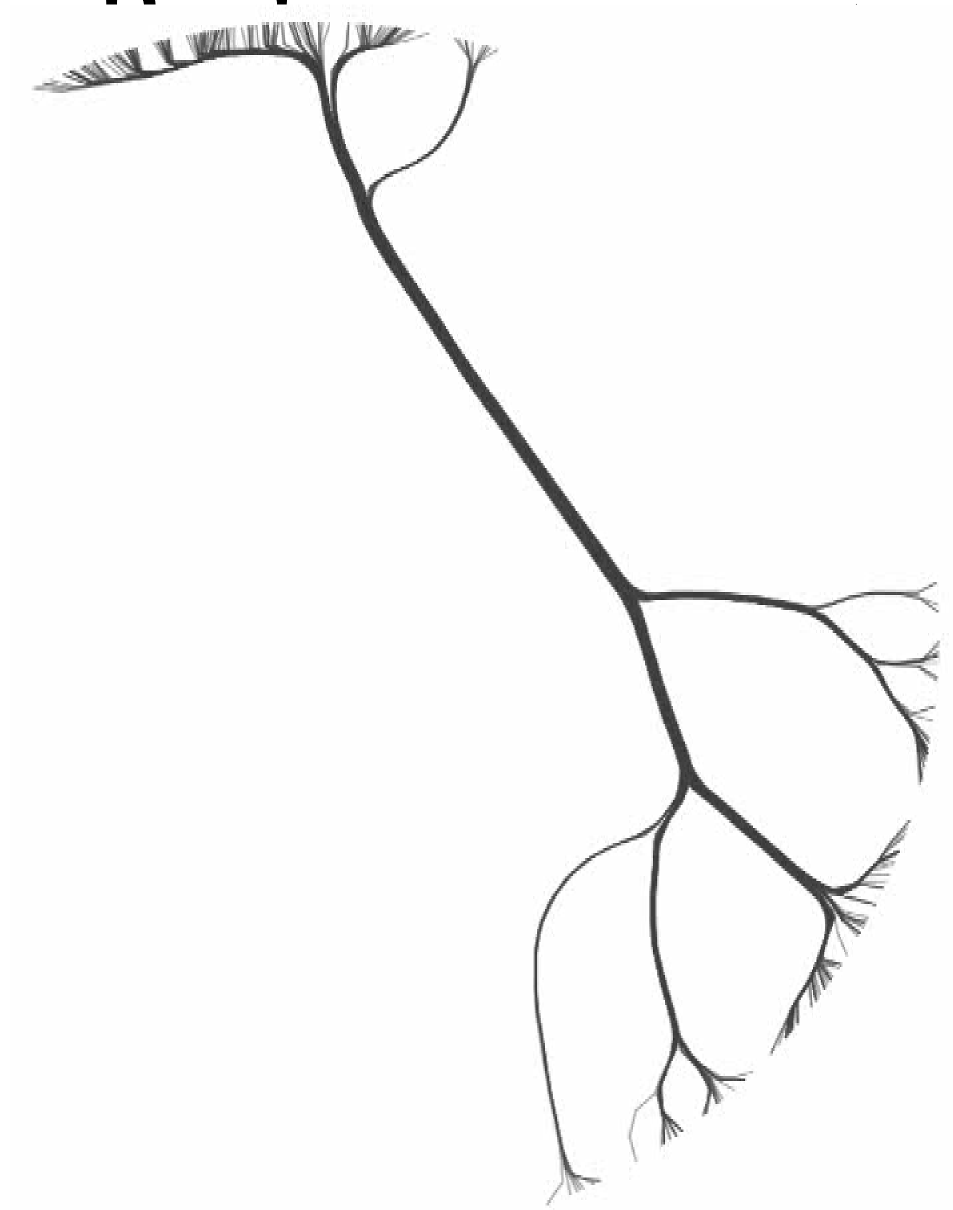
- compute all branch termination points, i.e. tips
- compute all skeleton paths
- for each edge, select a skeleton path passing through the FT_S of both edge end points
- move all edge points towards the selected path



Edge smoothing

Each point is moved towards the average of its neighboring points

Removes sharp bundle turns



Edge relaxation

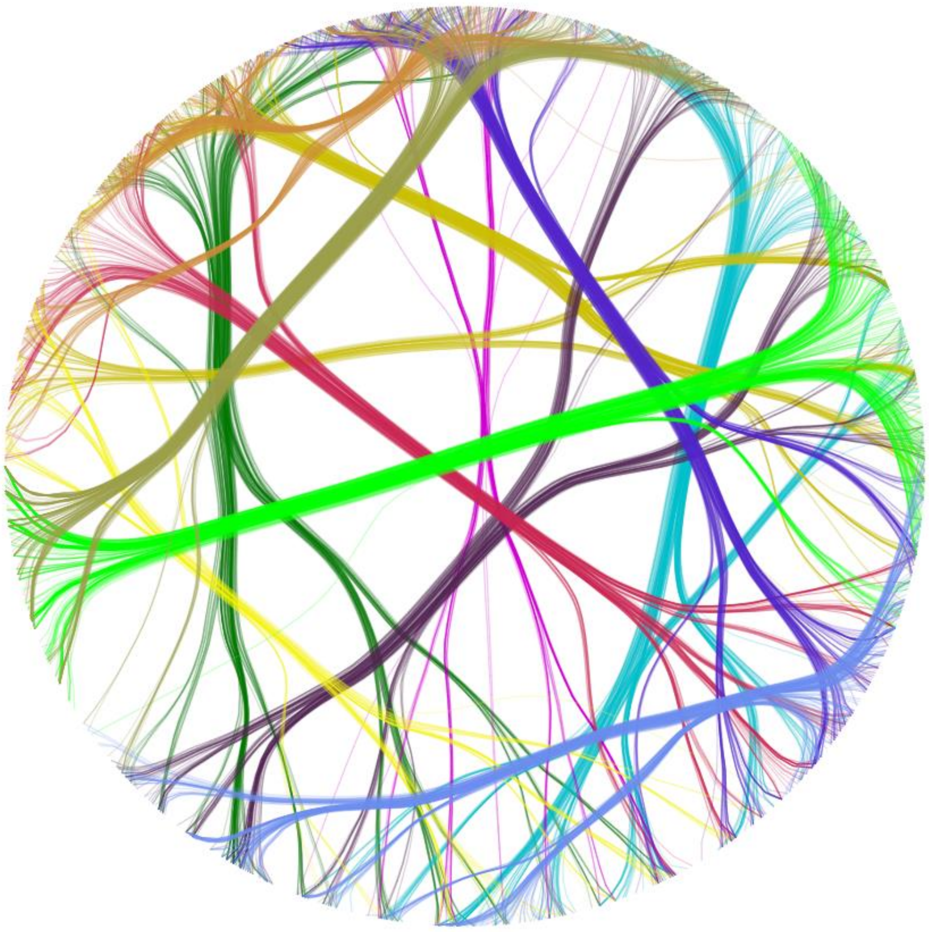
Edges are interpolated between their fina
positions and initial positions

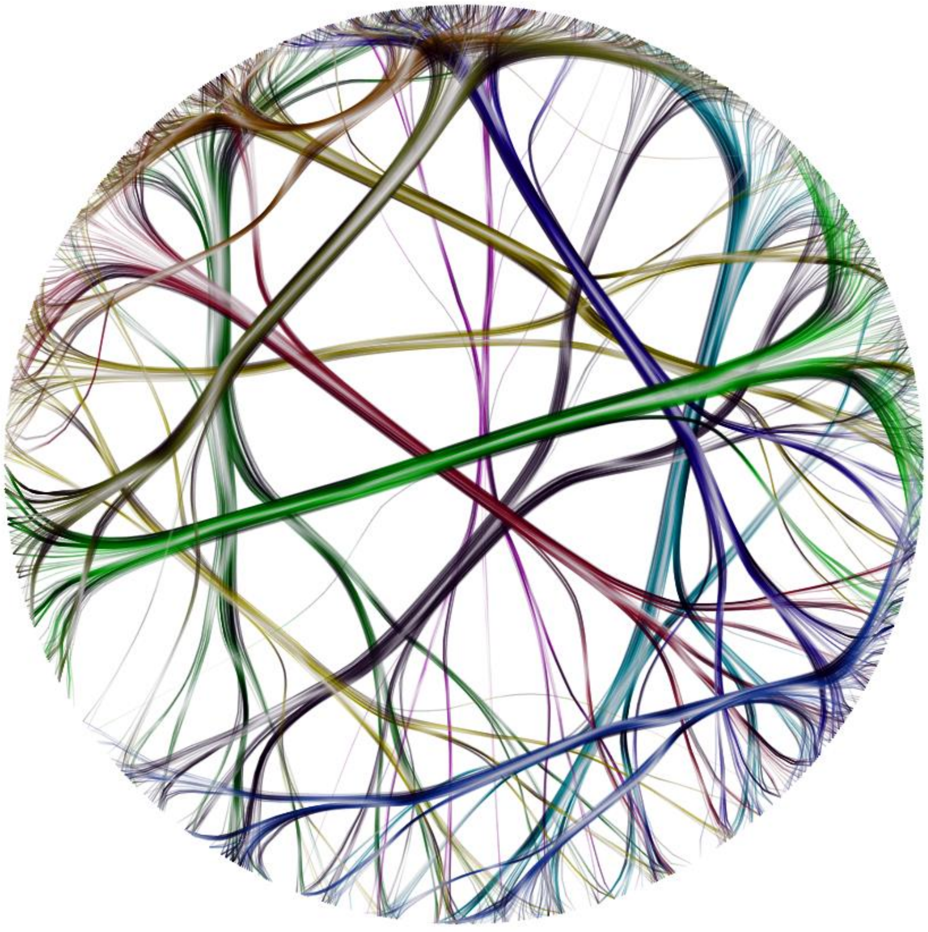
Shows individual edges within a bundle
and where they come from

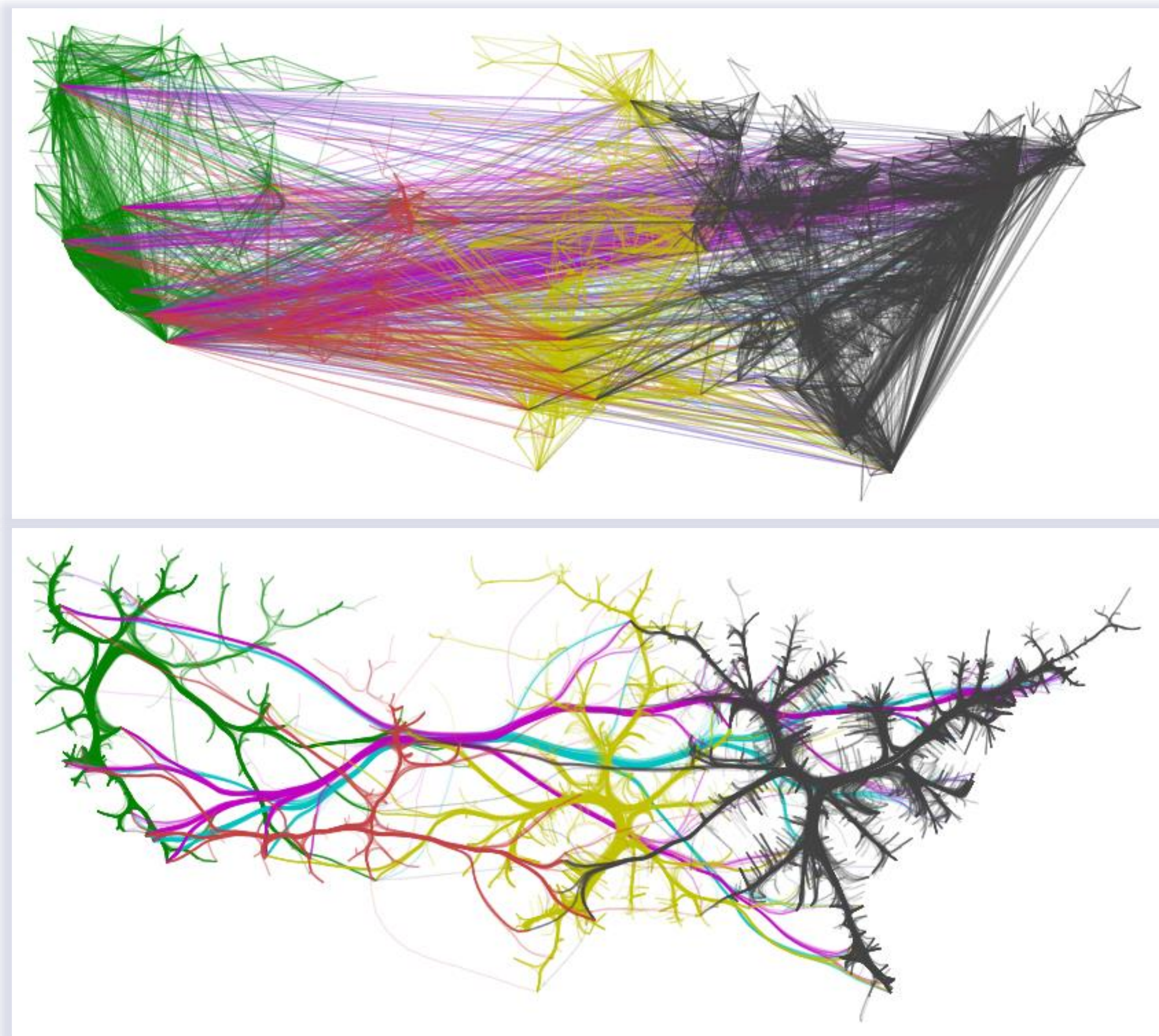
Rendering

We use a 3D cushion profile: bright at the bundle's center, dark at the boundary

Set the saturation and brightness of edge points based on distances to the skeleton

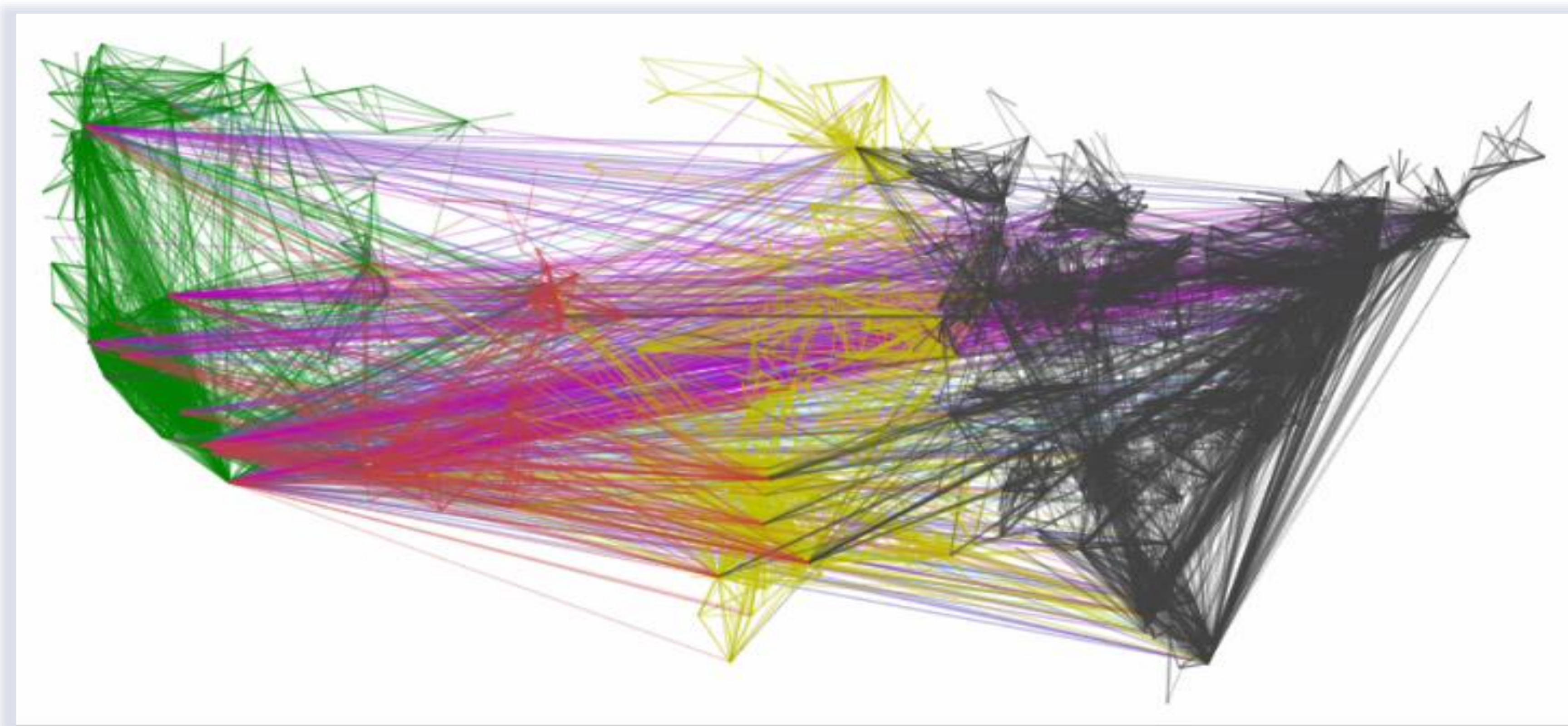






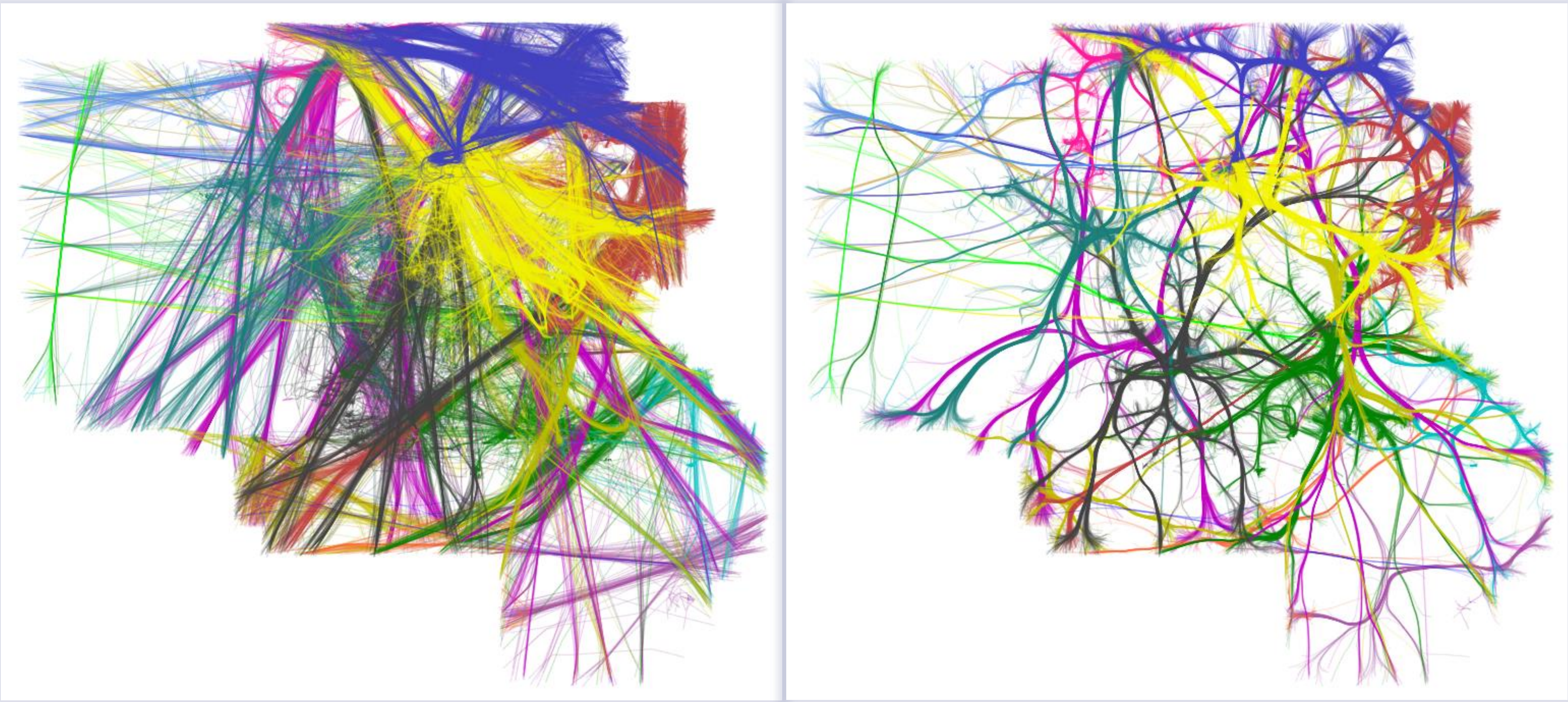
US Migrations, 1715 nodes, 9780 edges, 6 clusters, 4.1 sec.

Results

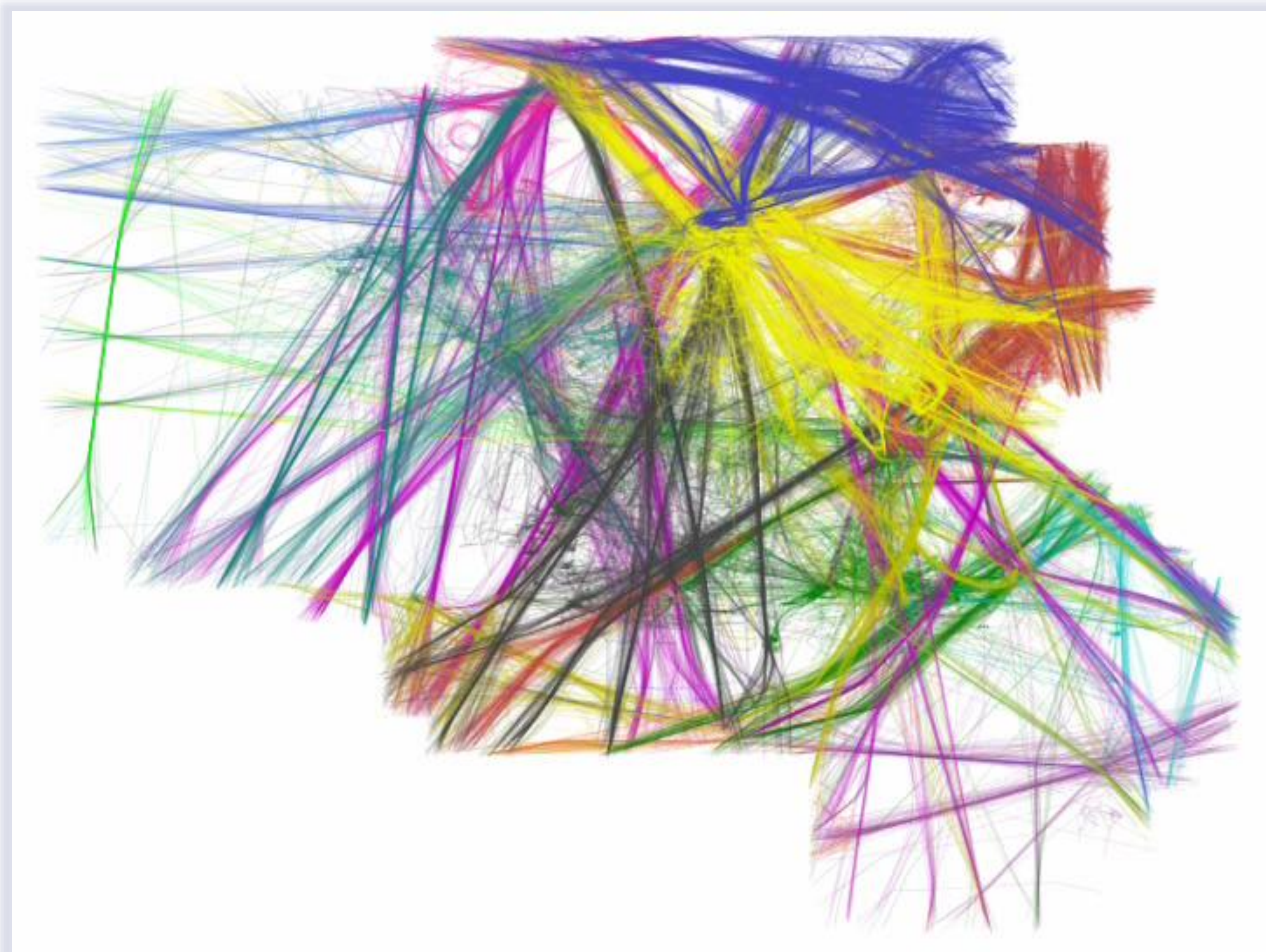


US Migrations, 1715 nodes, 9780 edges, 6 clusters, 4.1 sec.

Results

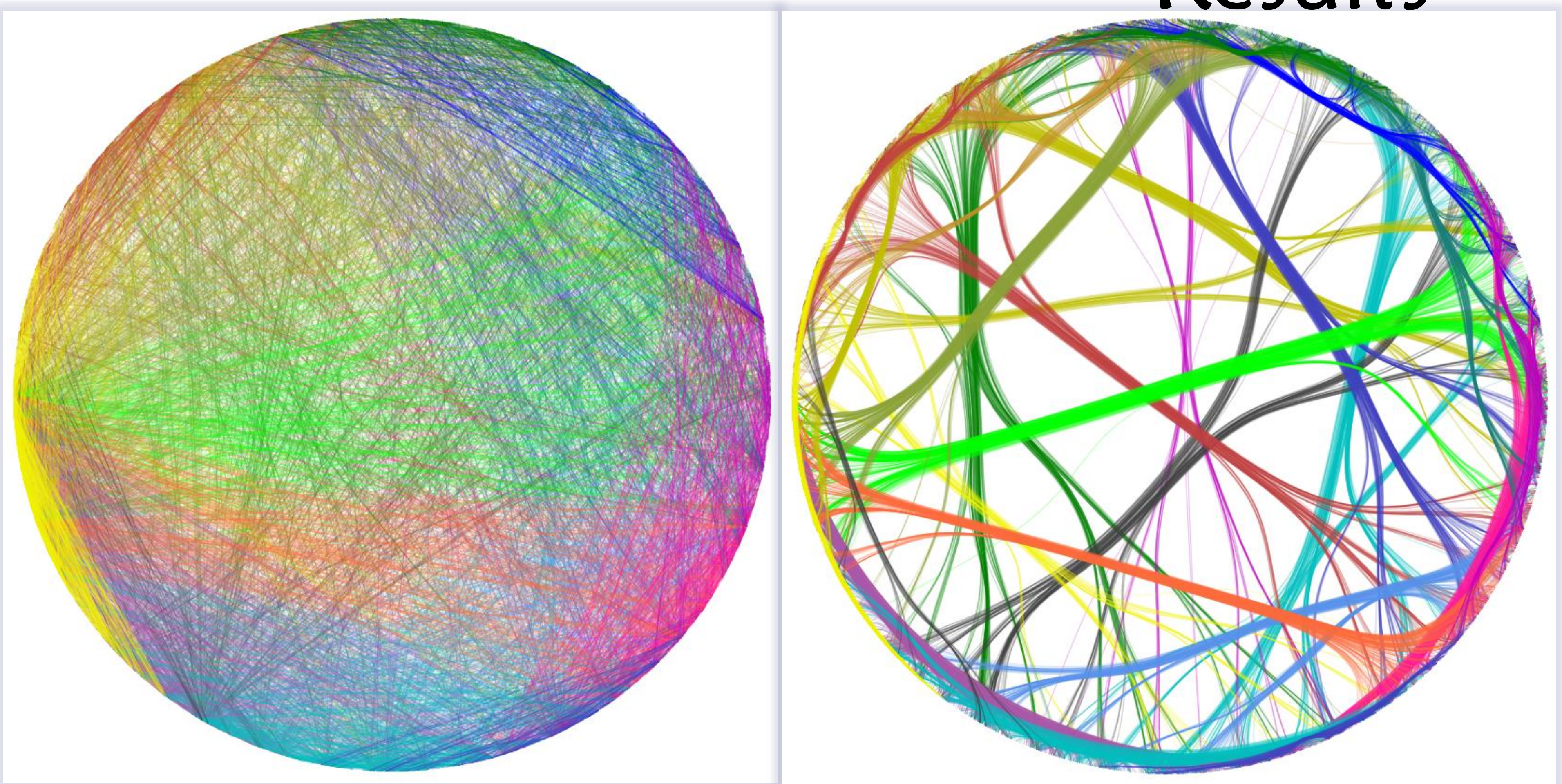


France Air, 34544 nodes, 17272 edges, 207 clusters, 29.2 sec.

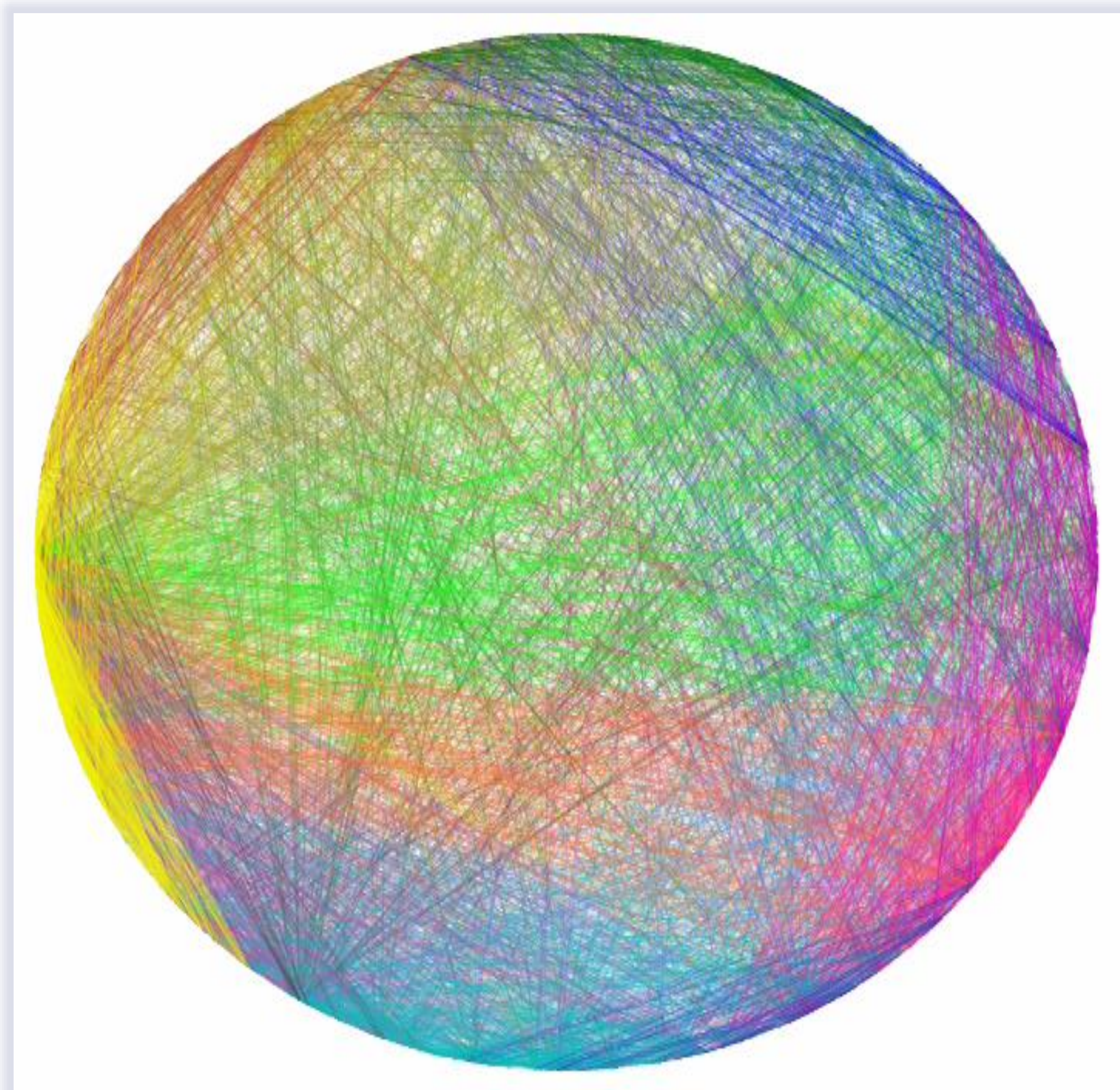


France Air, 34550 nodes, 17272 edges, 207 clusters, 29.2 sec.

Results



Radial, 1024 nodes, 4021 edges, 94 clusters, 7.4 sec.



Radial, 1024 nodes, 4021 edges, 94 clusters, 7.4 sec.

Skeleton-based Edge Bundling

Pros

- Generality: No hierarchy required, applicable to general graphs, directed or undirected
- Structured look control: Generates smooth or organic-looking bundles
- Speed and simplicity: CUDA implementation, entirely image-based

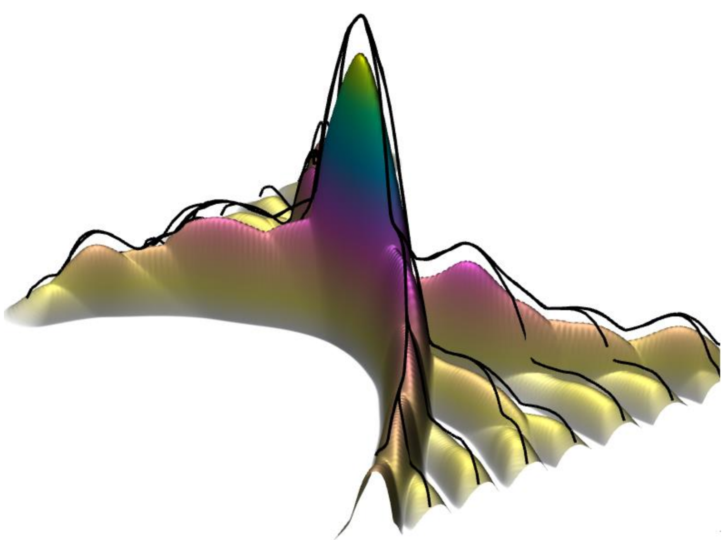
Cons

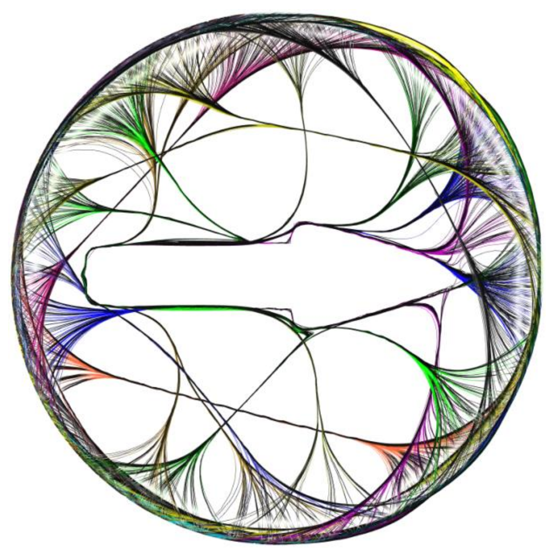
- Susceptible to clustering results

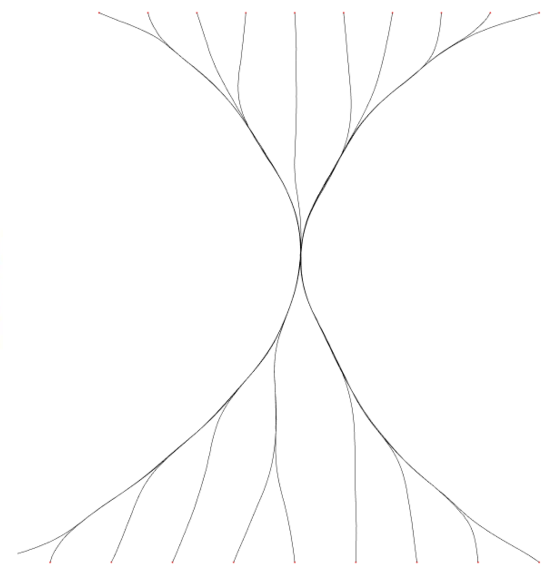
Graph Bundling by Kernel Density Estimation

C. Hurter, O.Ersoy, A.Telea

Eurovis 2012



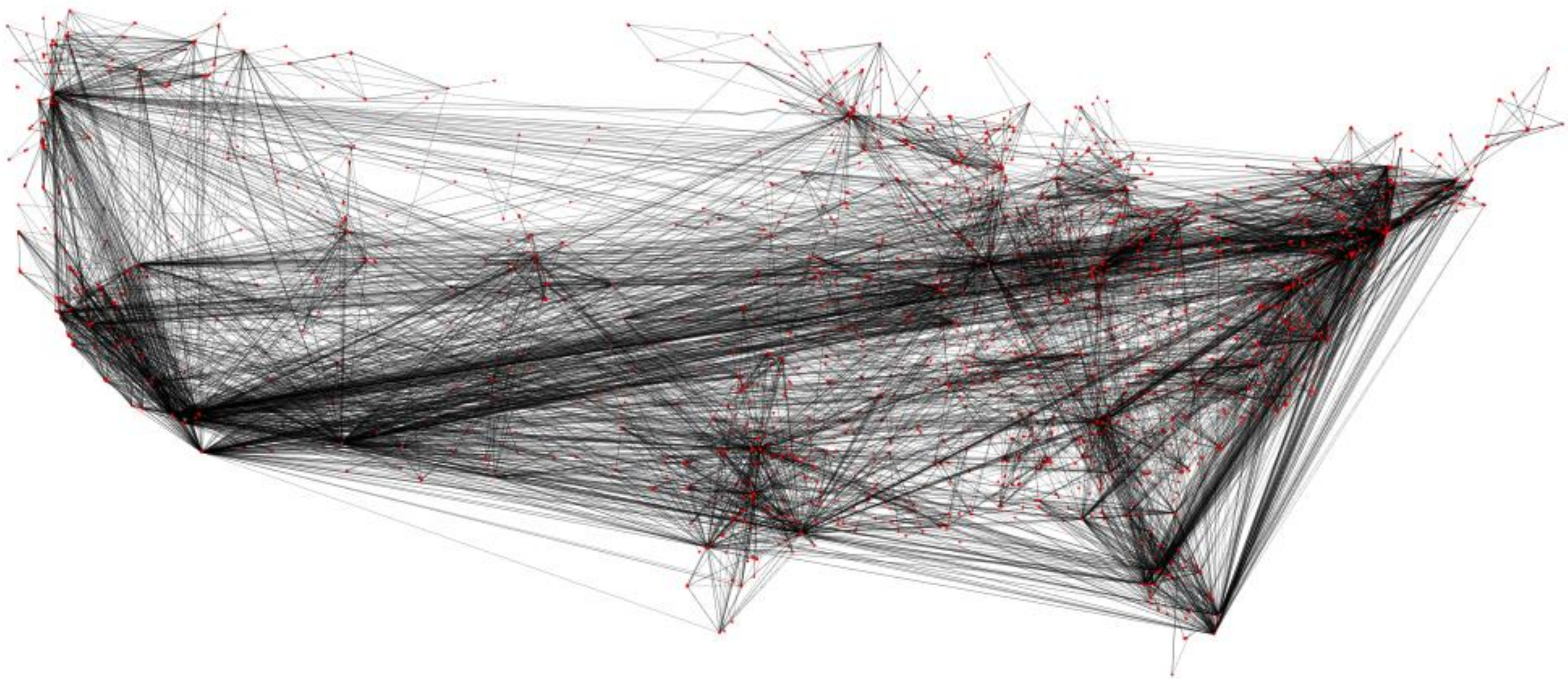




An unbundled graph

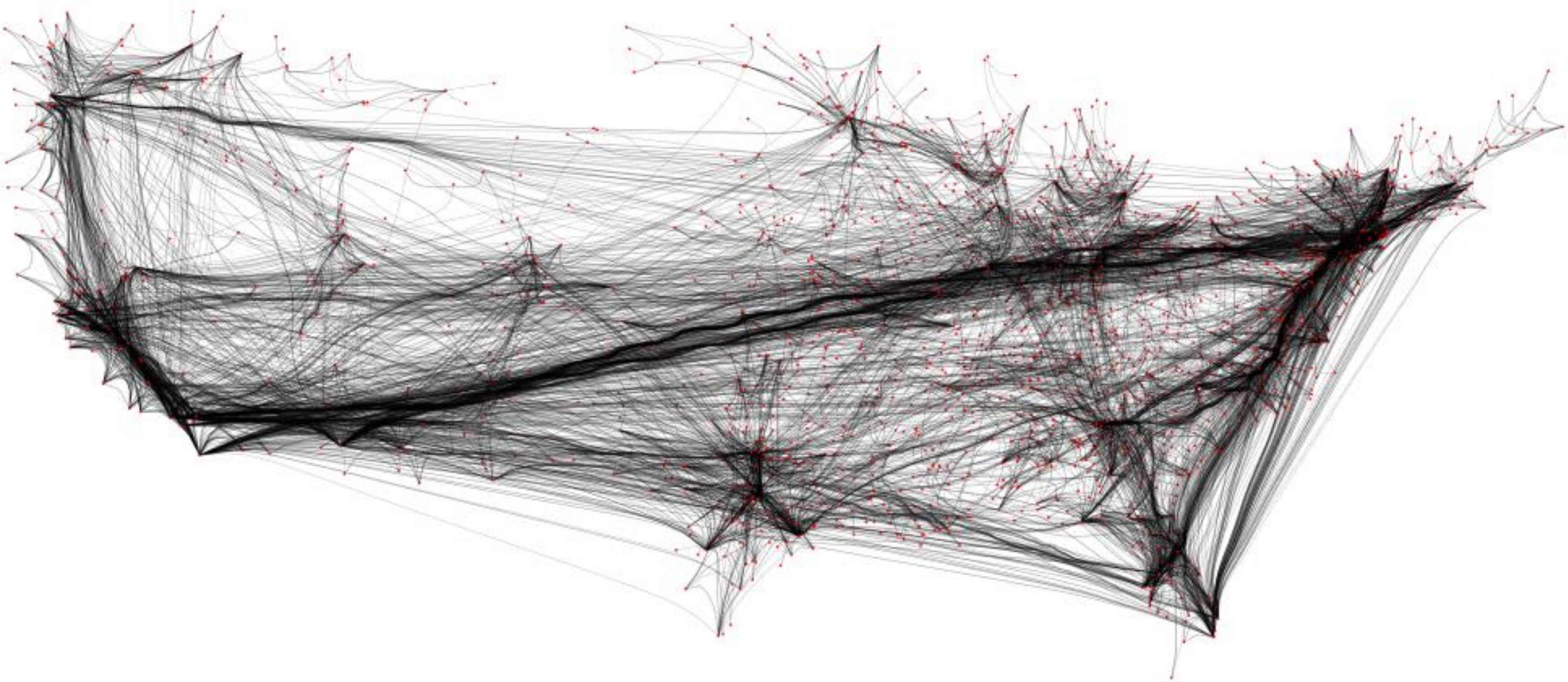
Consider the edge density

it is relatively **uniform**



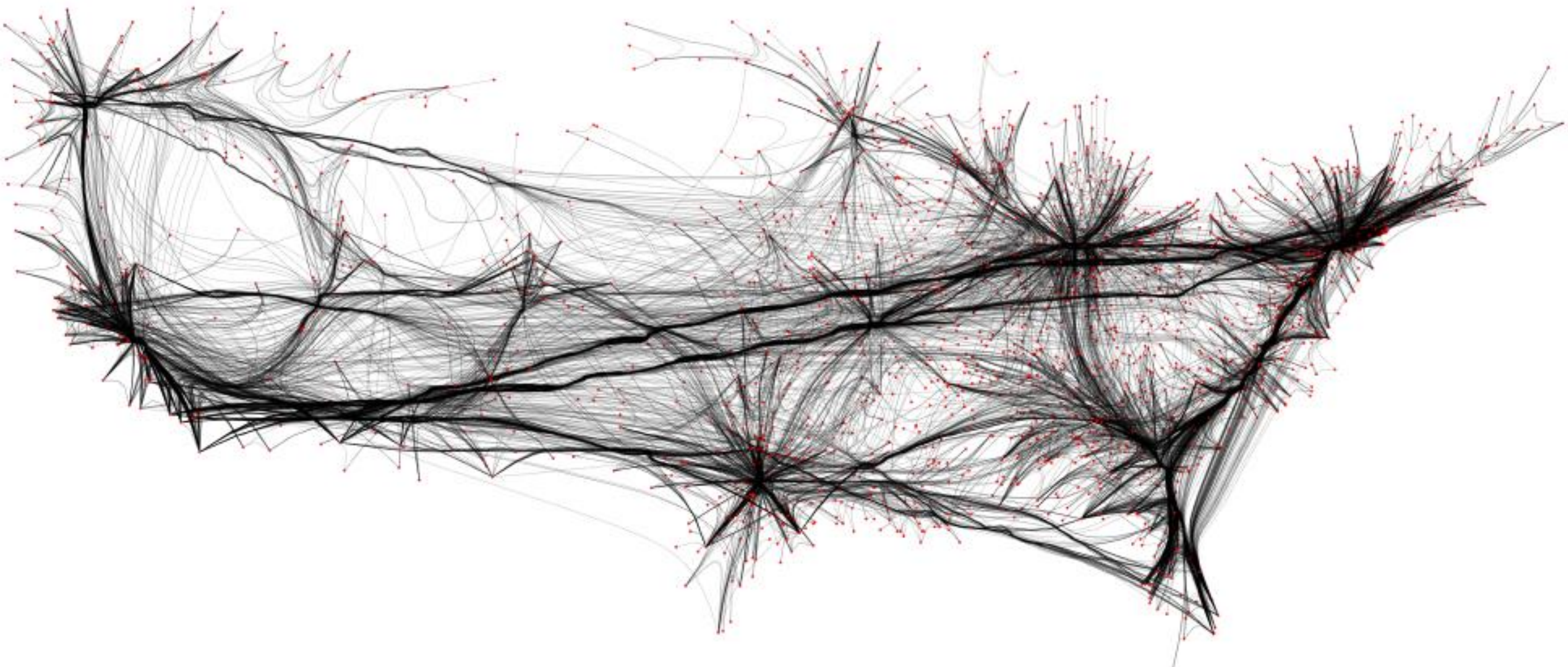
One bundling step

Edges get closer **-> density gets sharpened**



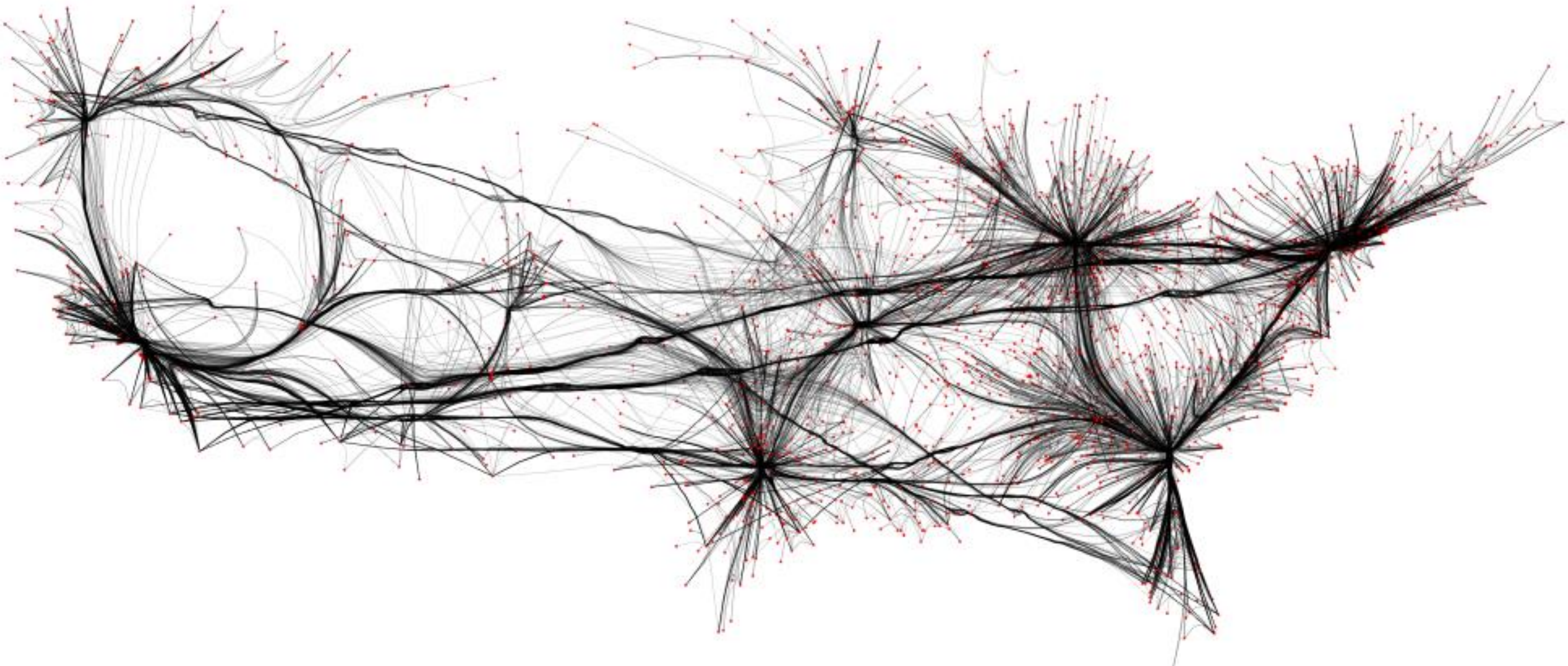
More bundling…

Edges get closer **-> density gets sharper**



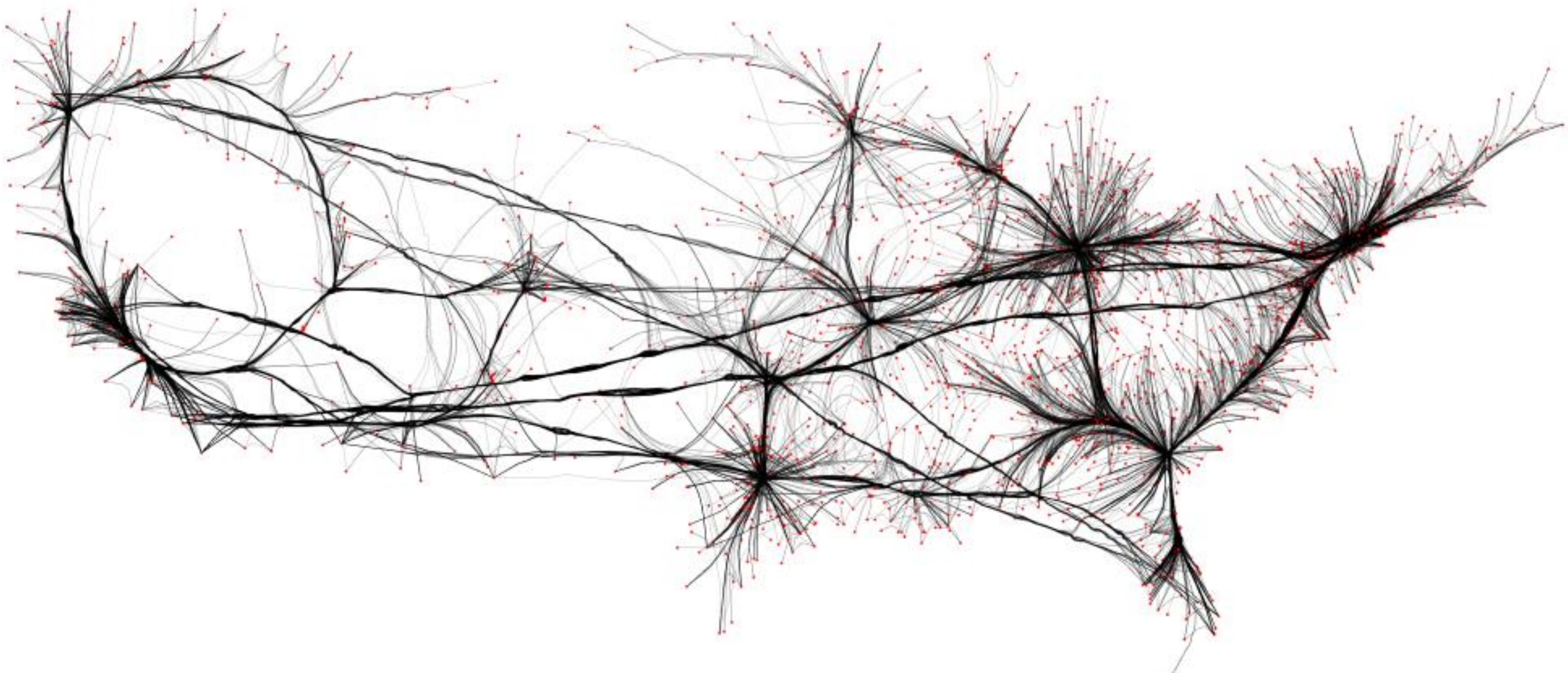
Even more bundling…

Edges get closer **-> density gets even sharper**



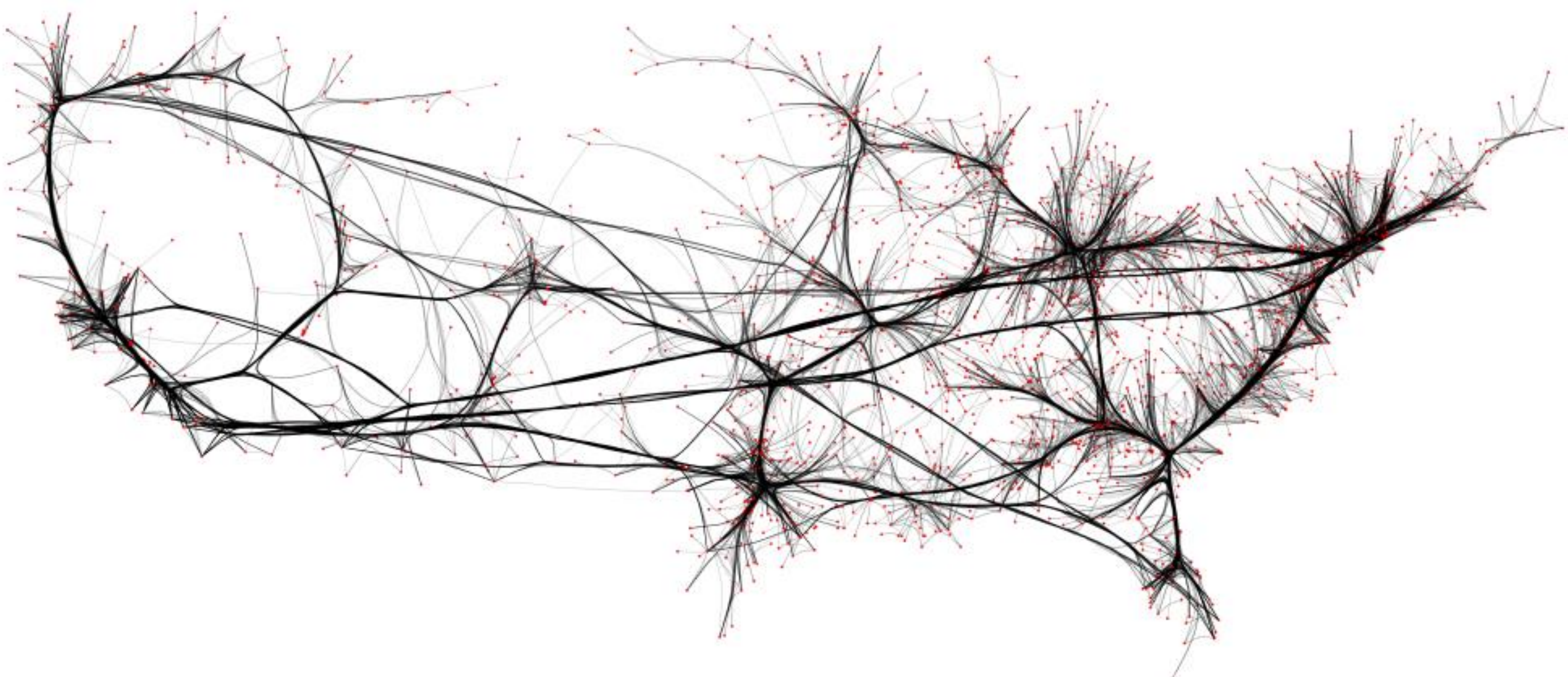
Even more bundling…

Edges get closer **-> density gets even sharper**

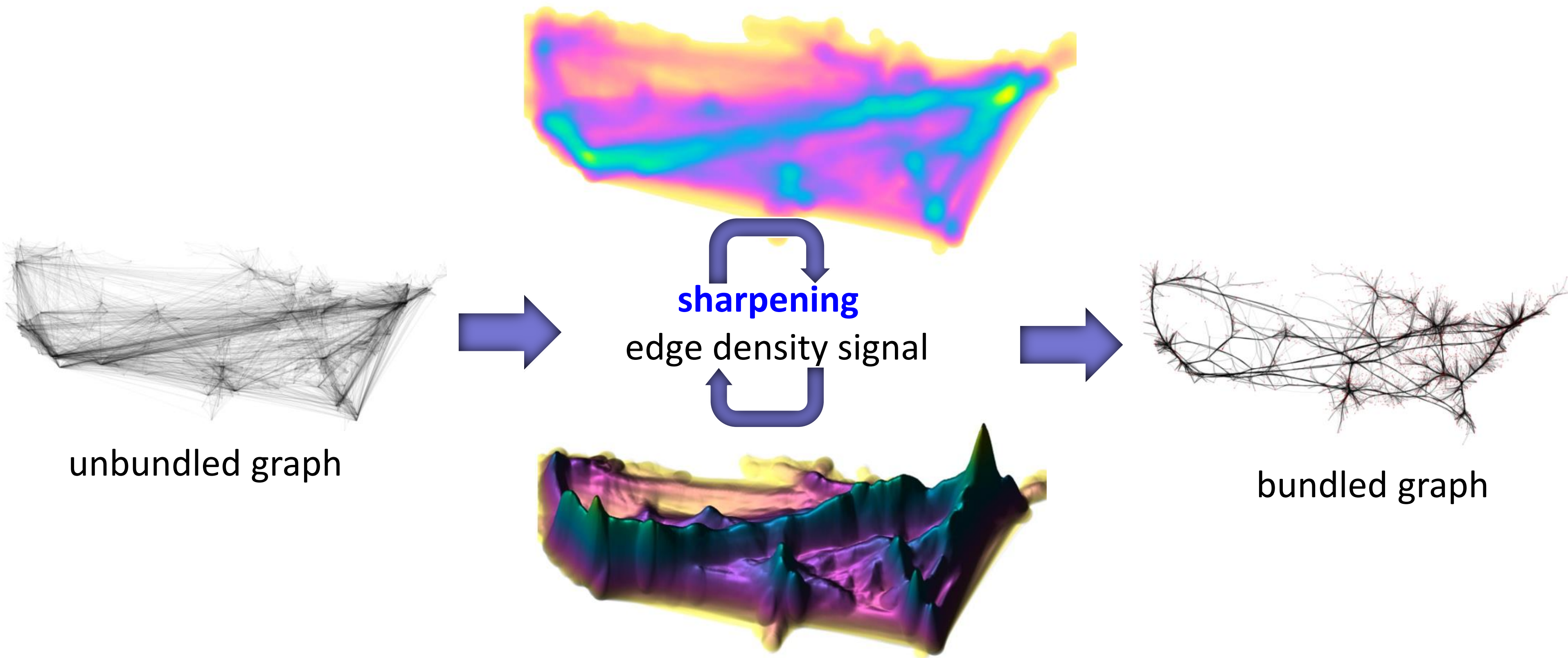


Even more bundling…

Edges get closer **-> density gets even sharper**



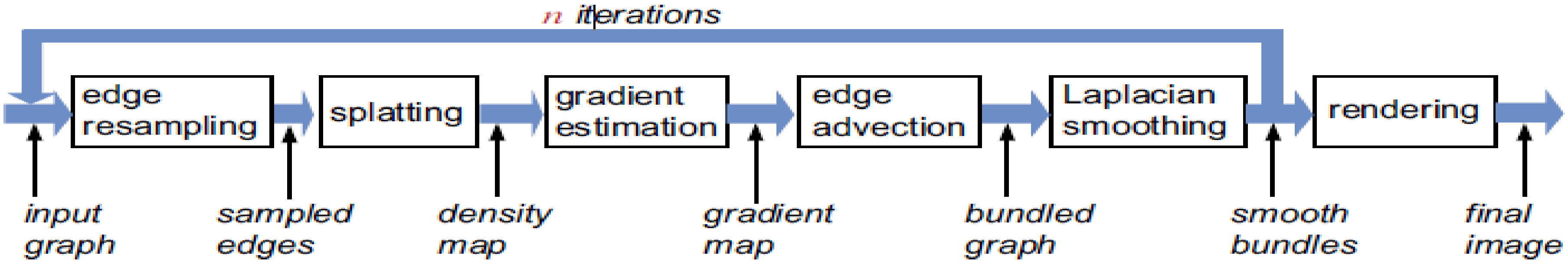
KDEEB : Mean Shift and Bundling [1]

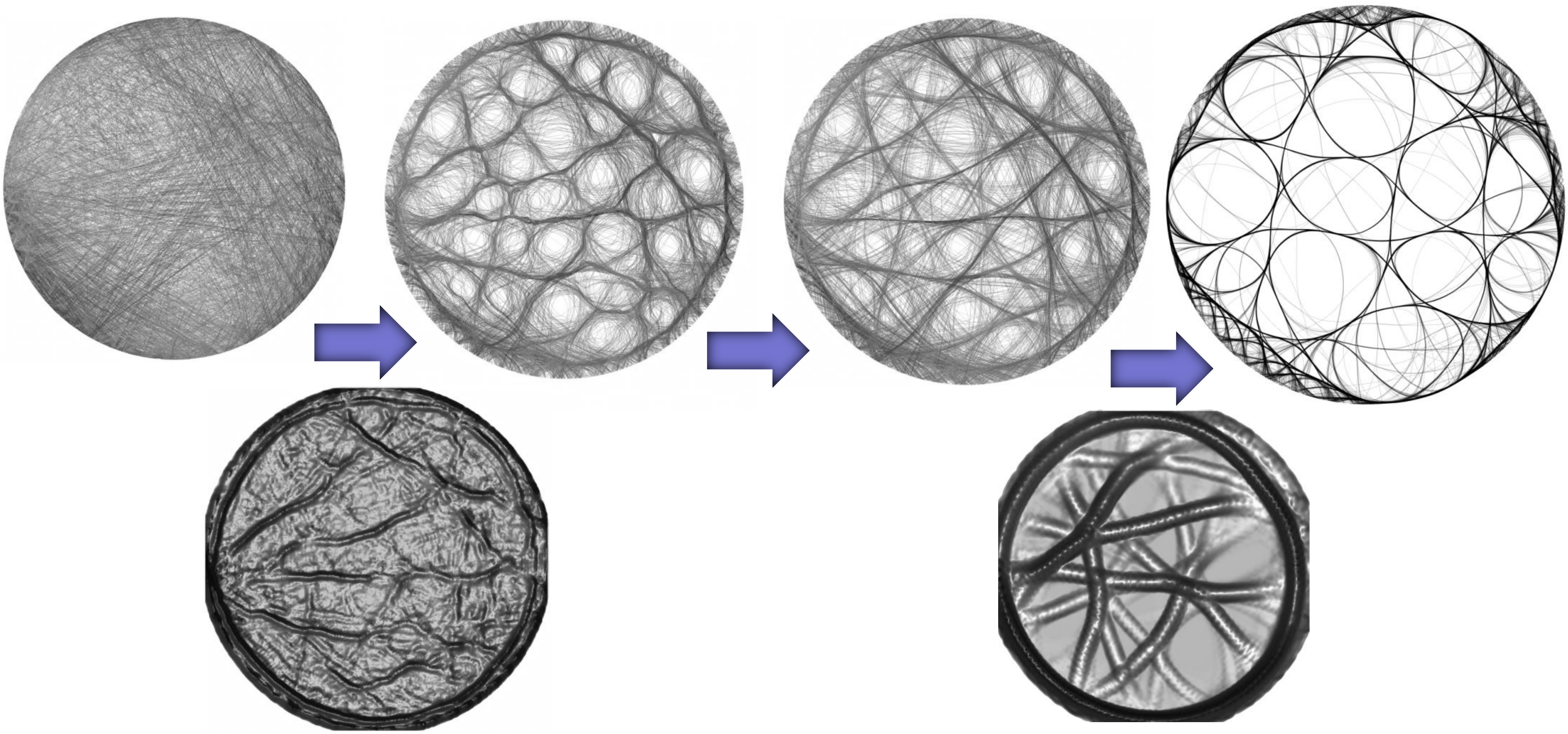


If bundling **sharpens** the **edge density**, then **sharpening** the edge density should **bundle**

[1]Christophe Hurter, Ozan Ersoy, Alexandru Telea. **Graph Bundling by Kernel Density Estimation.** *(EuroVis 2012). Computer Graphics Forum 31, 3pt1 (june 2012)* p865-874.

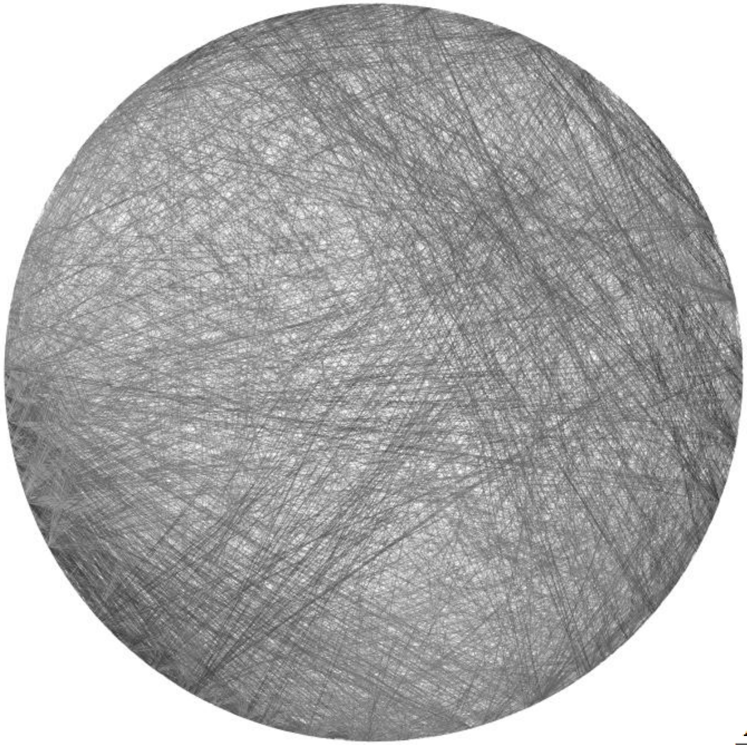
KDEEB pipeline





Step 1: Computing the density map

Initial graph



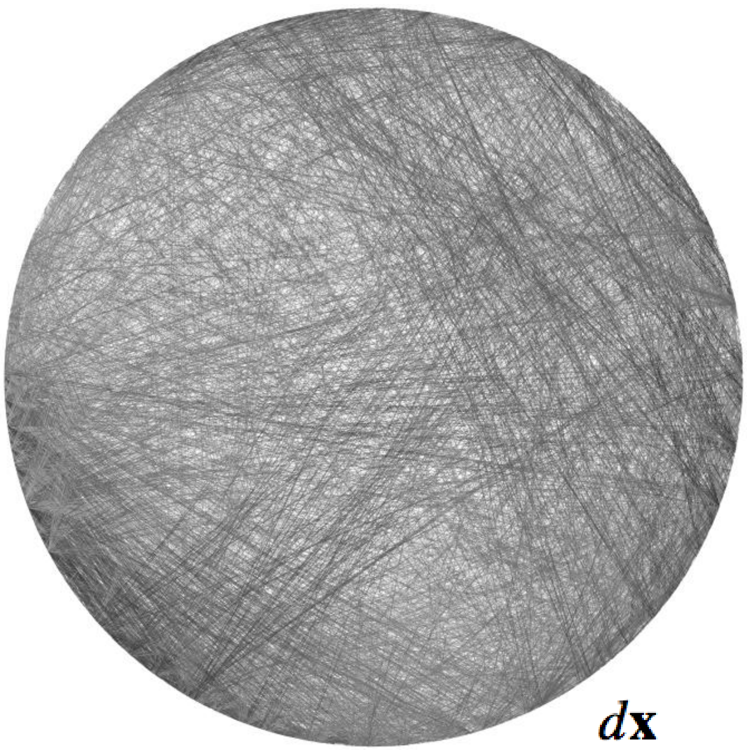
Density map

$$\rho(\mathbf{x}) = \sum_{i=1}^{N} \int_{\mathbf{y}\in e_i} K\left(\frac{\mathbf{x}-\mathbf{y}}{h}\right)$$

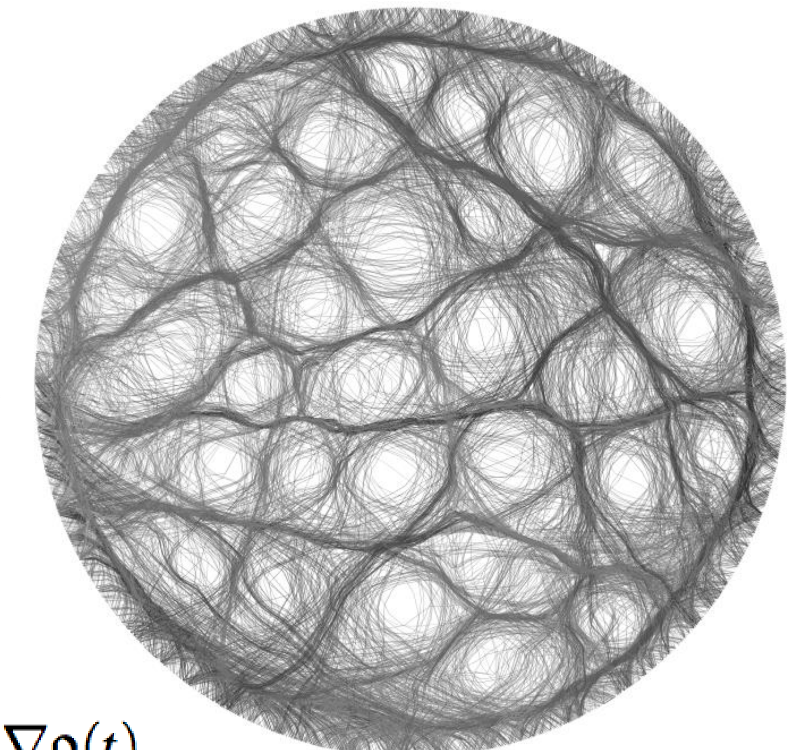
Efficient computation: graph edge splatting

Step 2: Edge advection

Initial graph



Advected edges (iteration 1)



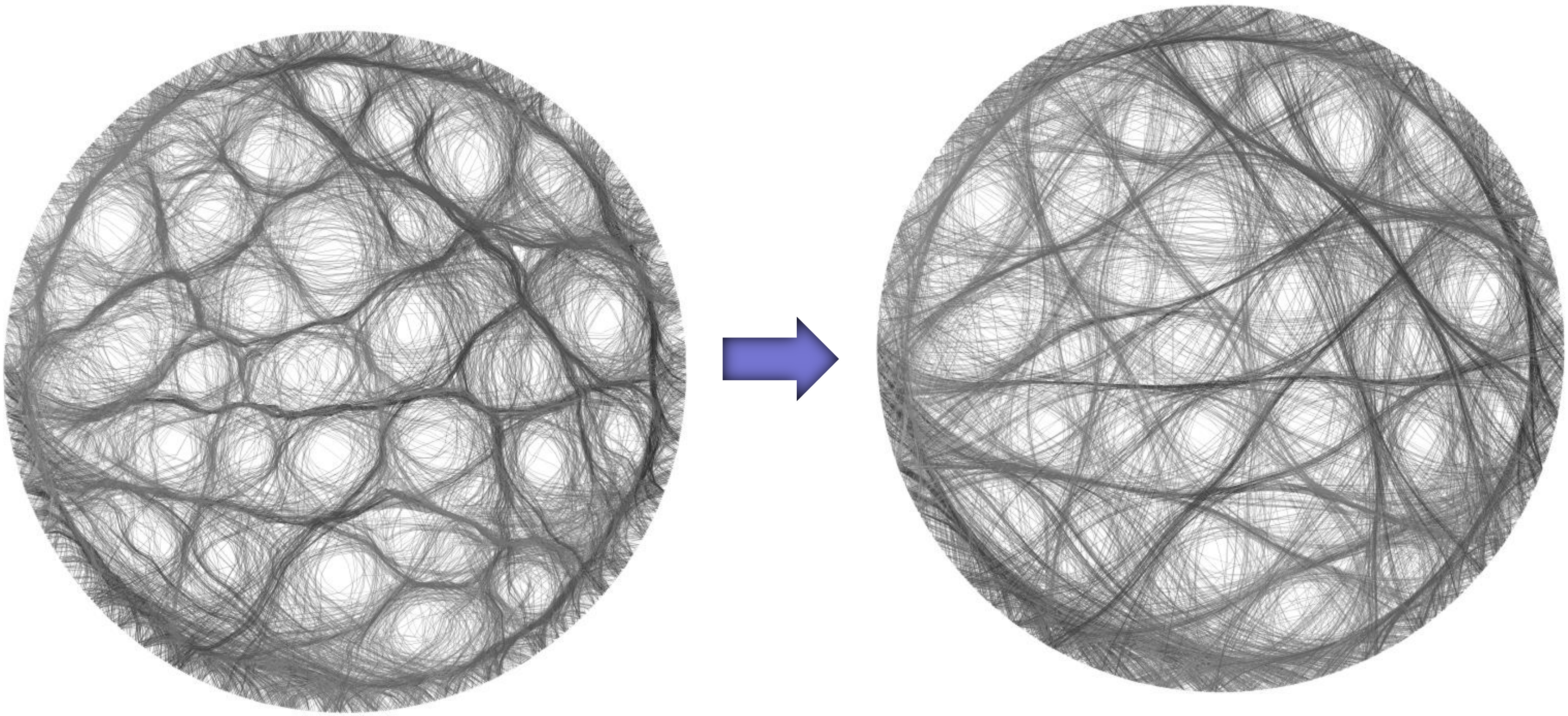
$$\frac{d\mathbf{x}}{dt} = \frac{h(t)\nabla\rho(t)}{\max(\|\nabla\rho(t)\|,\varepsilon)}$$

Move edges in the **normalized** gradient of the density map

Step 3: Edge smoothing

Advected edges

Smoothed edges

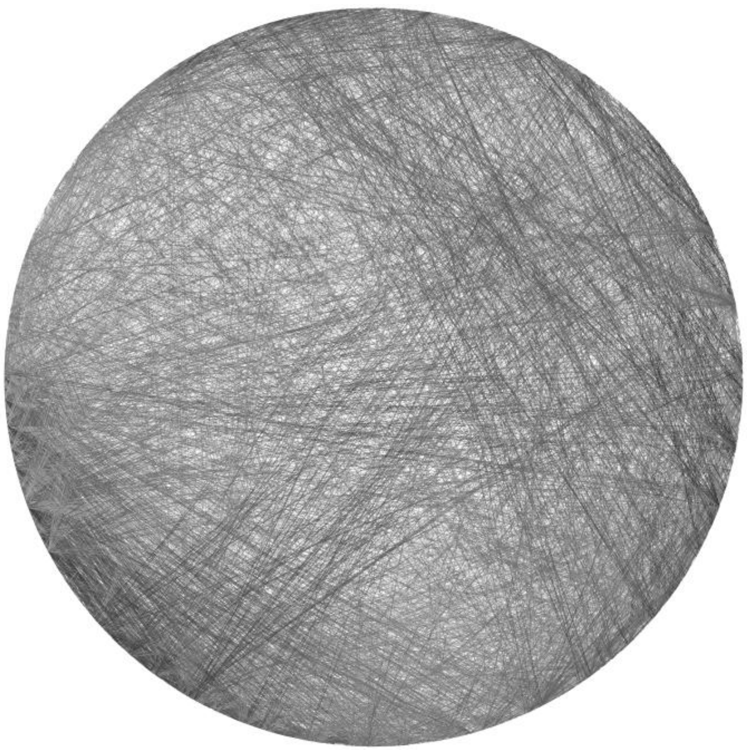


Eliminates advection **artifacts** caused by

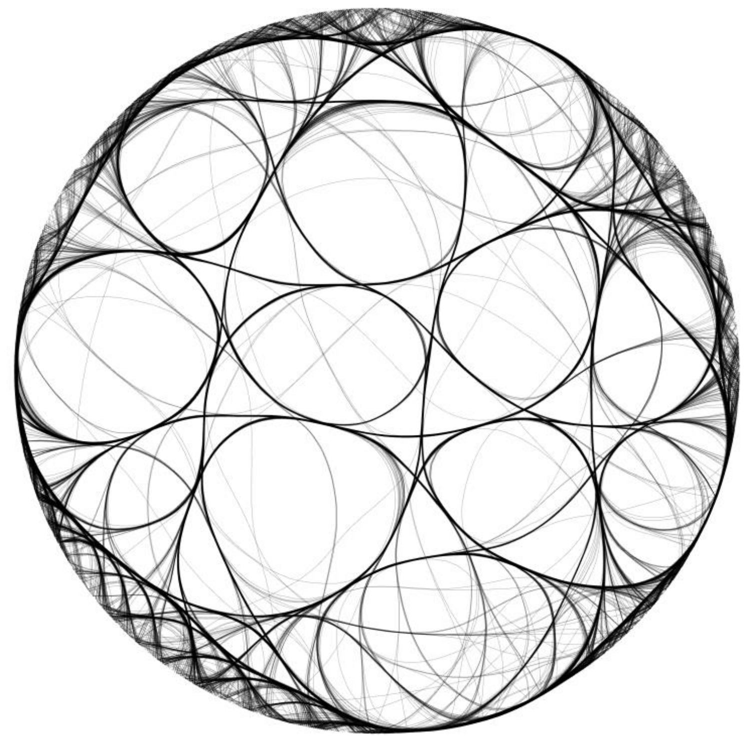
- density map sampling (texture splatting)
- edge sampling (polylines)

After n iterations

Original graph



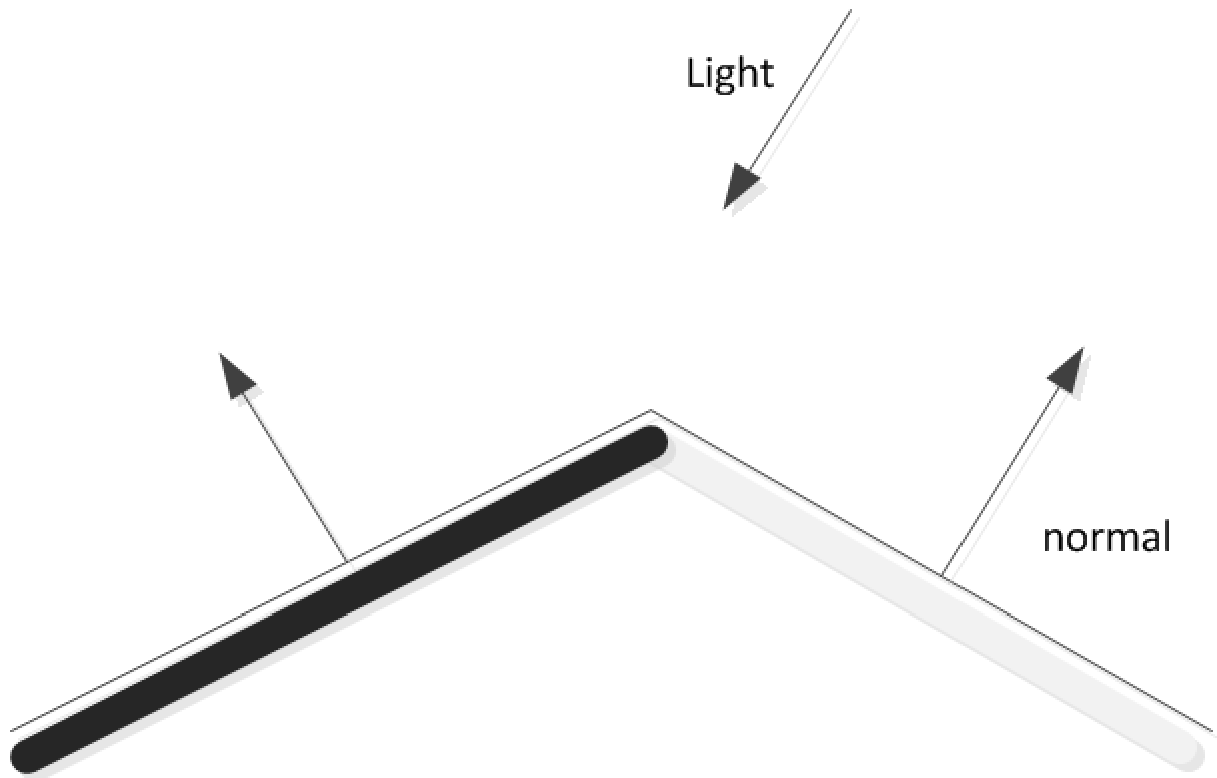
Bundled graph



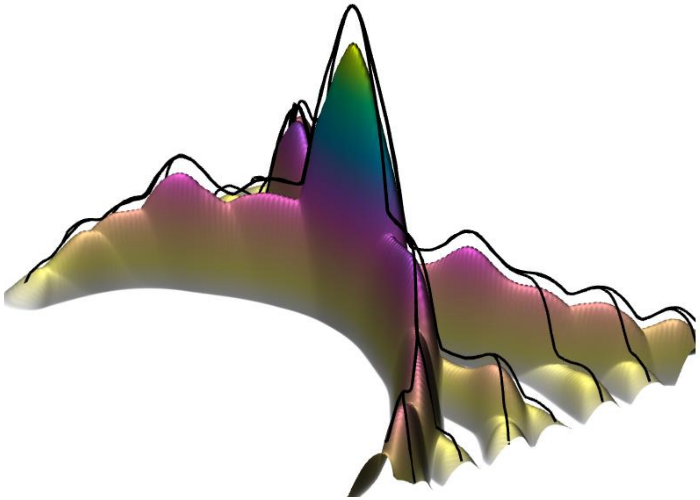
Edge bundles: asymptotically tend to Dirac-like impulses

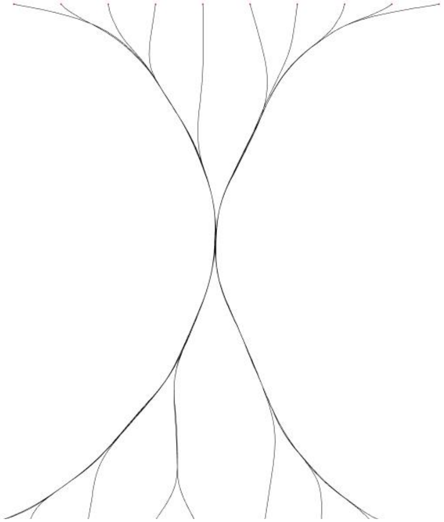
Shading computation

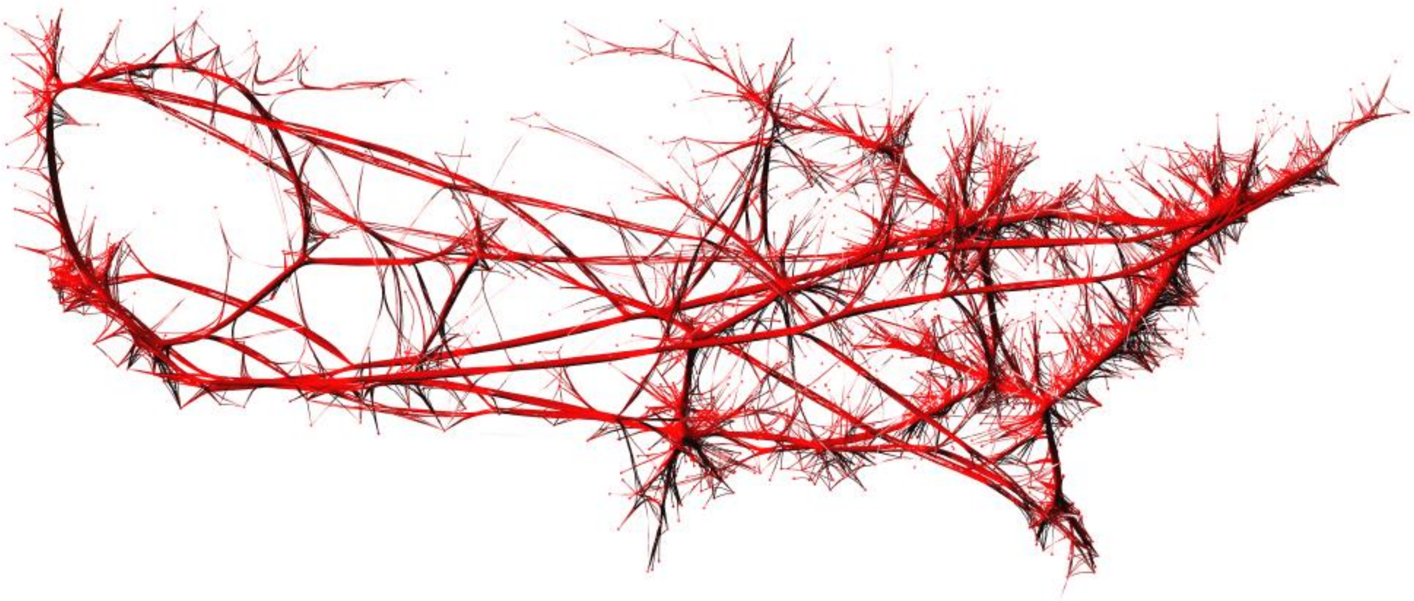
Dot product



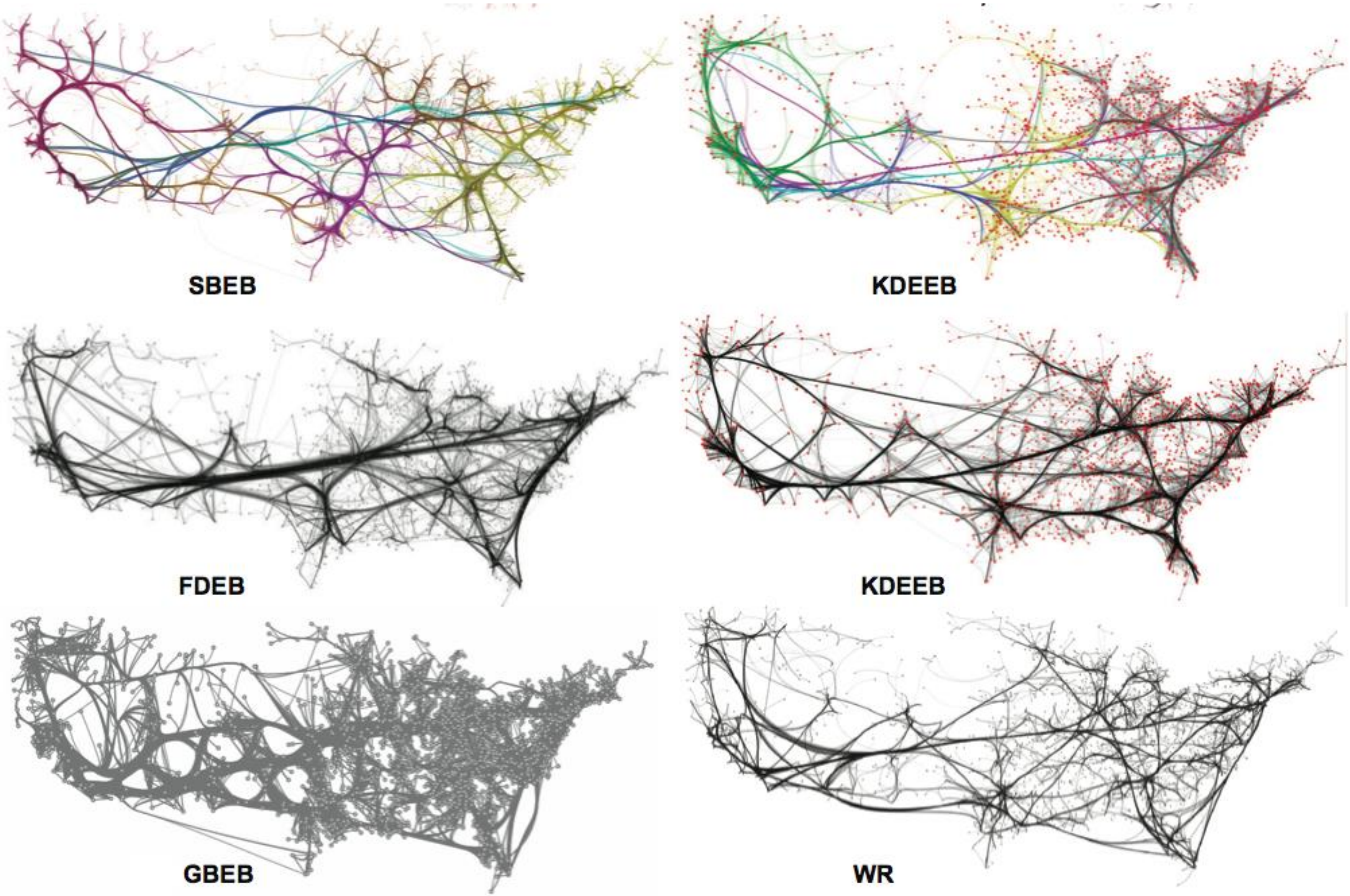
Demo





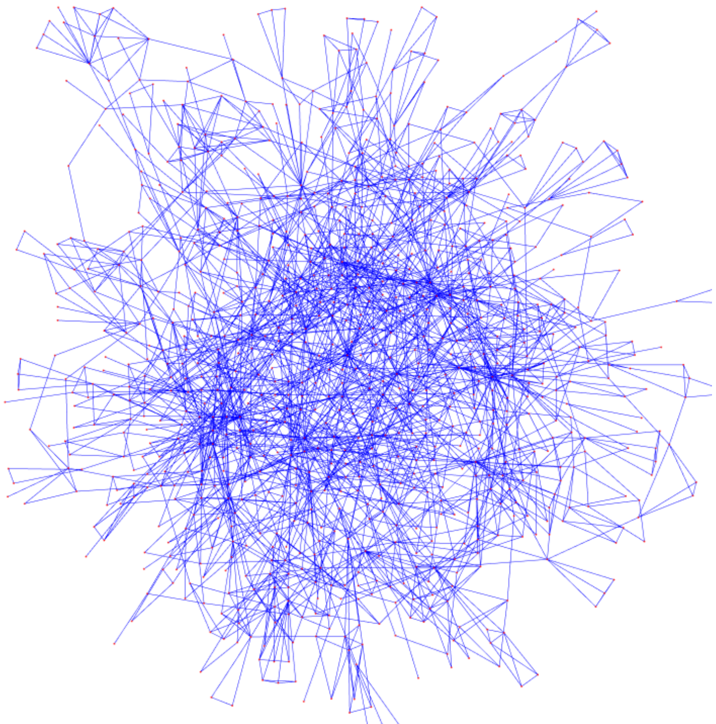
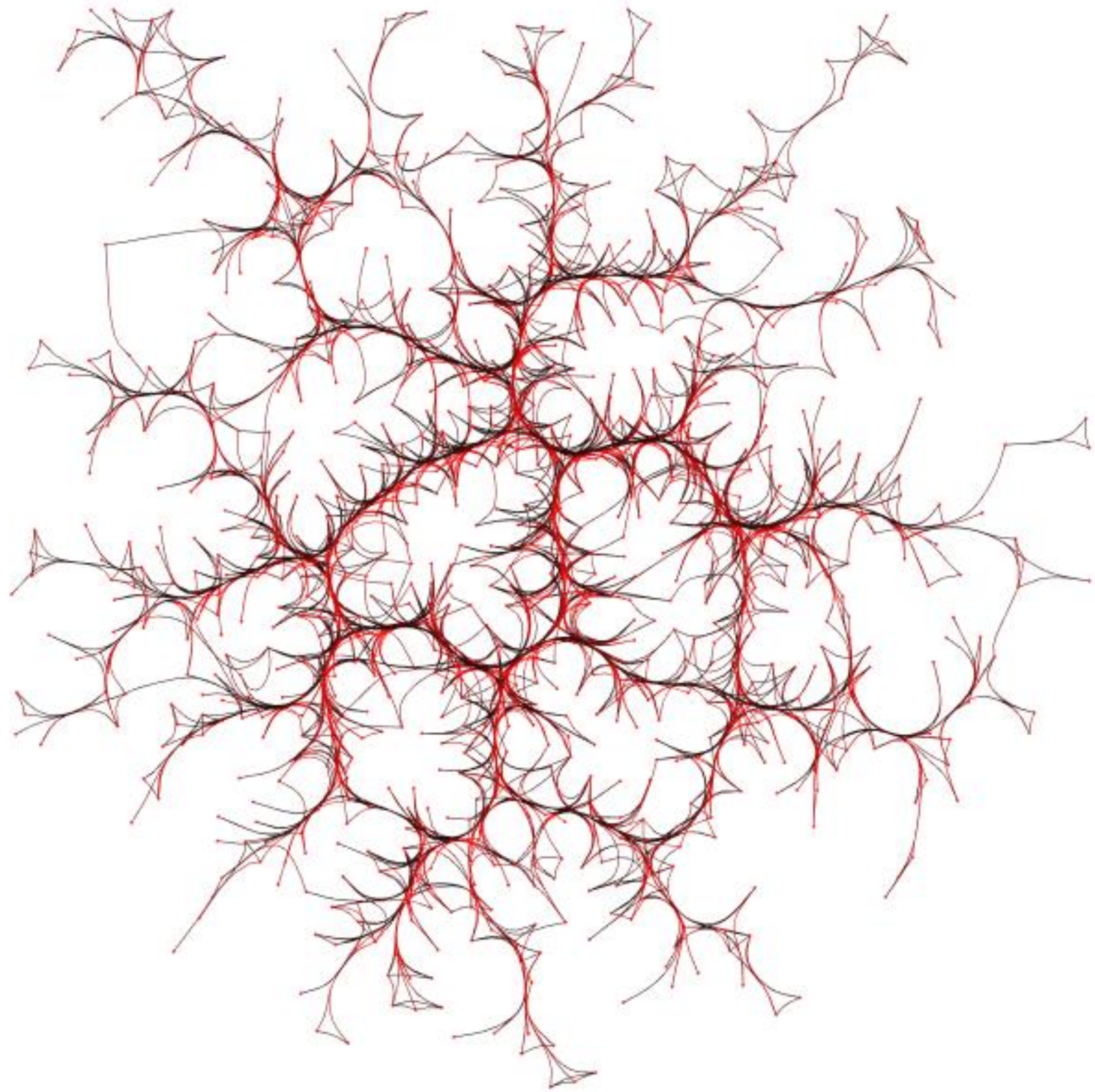


Comparison: Bundling style

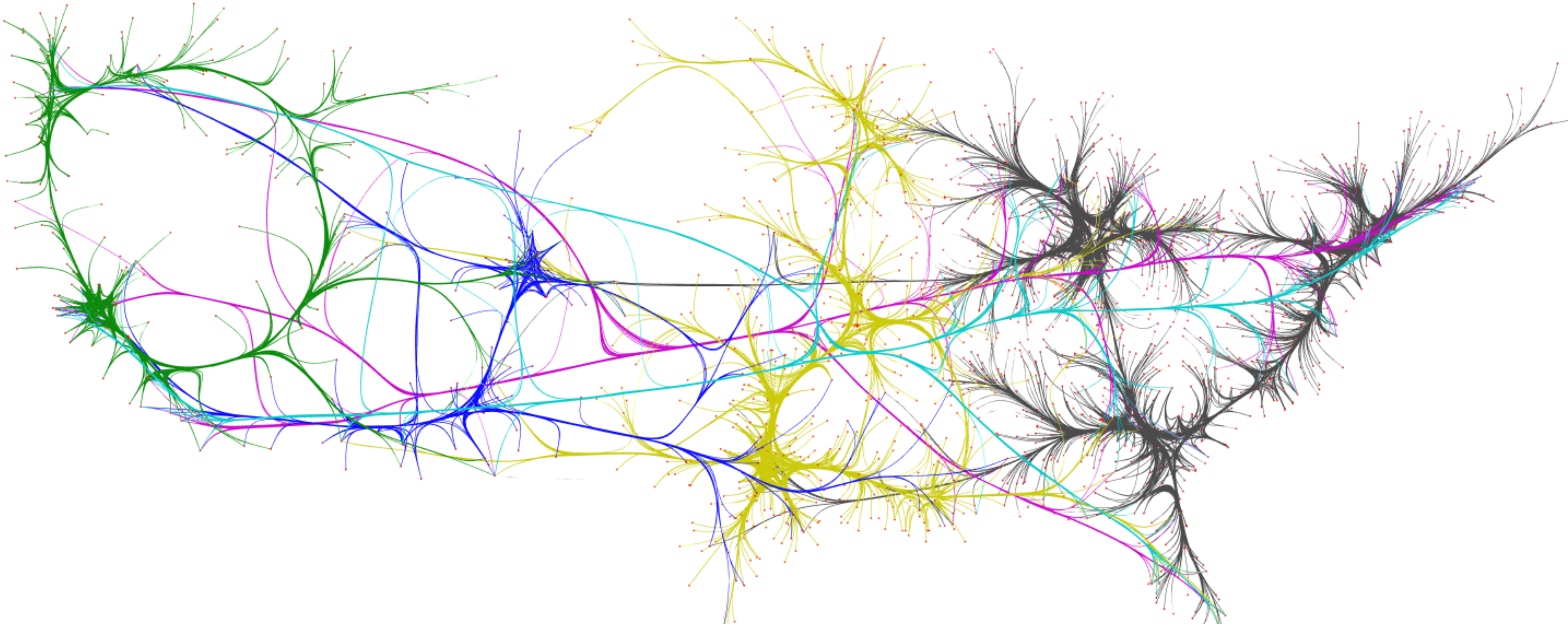


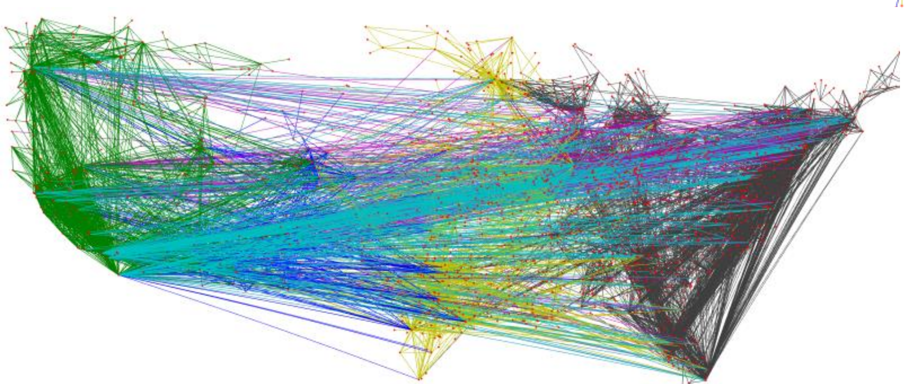
- we achieve same look-and-style as other methods
- but we are much faster

Graph	Nodes	Edges	Edge samples	Bundling time (sec.) 8800 GTX	Bundling time (sec.) GeForce 580
Poker	859	2127	50K	0.8	0.4

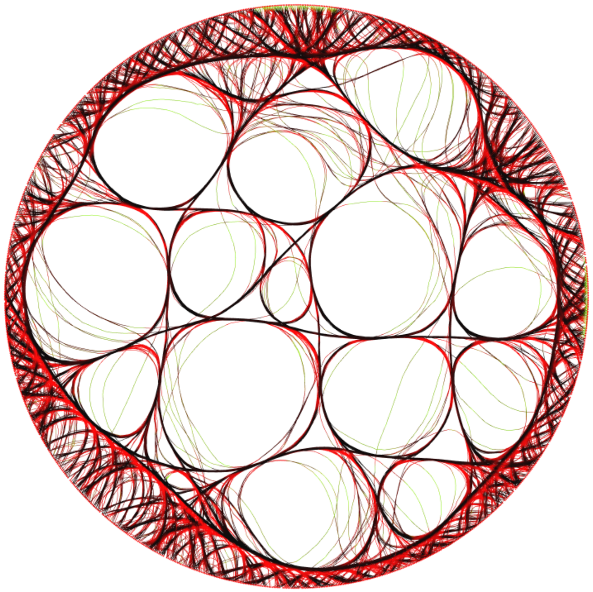



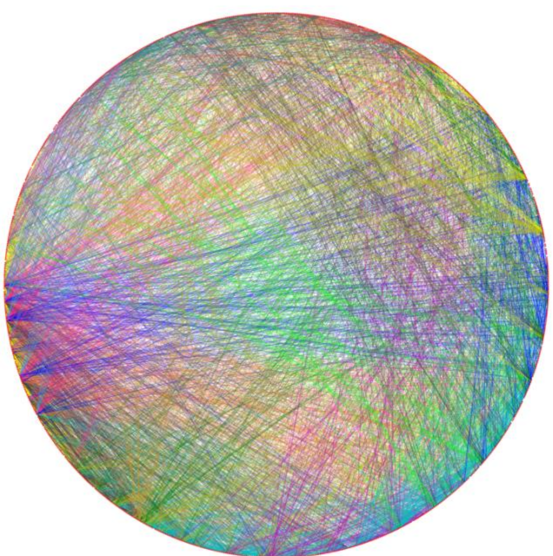
Graph	Nodes	Edges	Edge samples	Bundling time (sec.)	
				8800 GTX	GeForce 580
US airlines	235	2099	86K	1.4	0.5

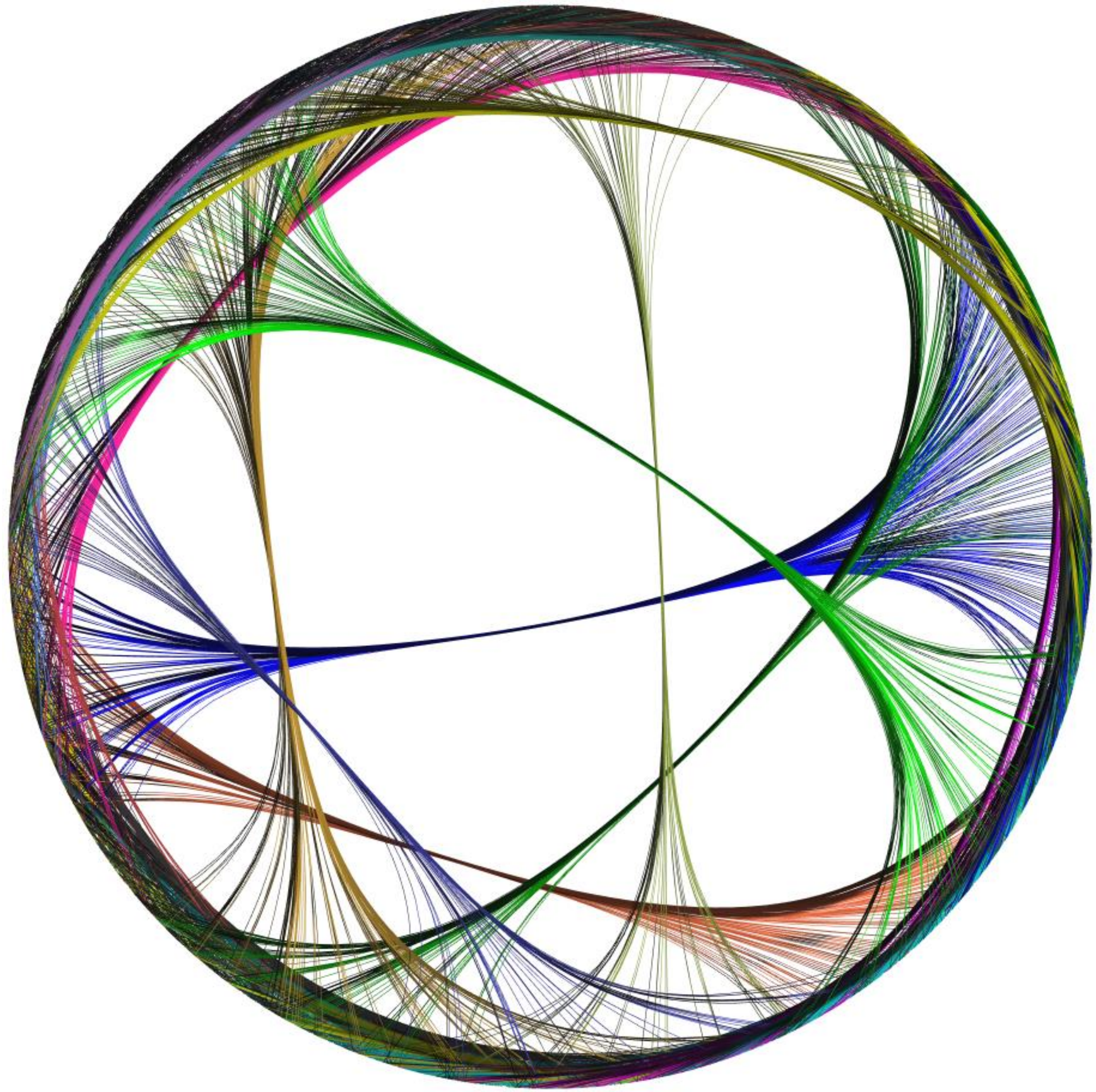




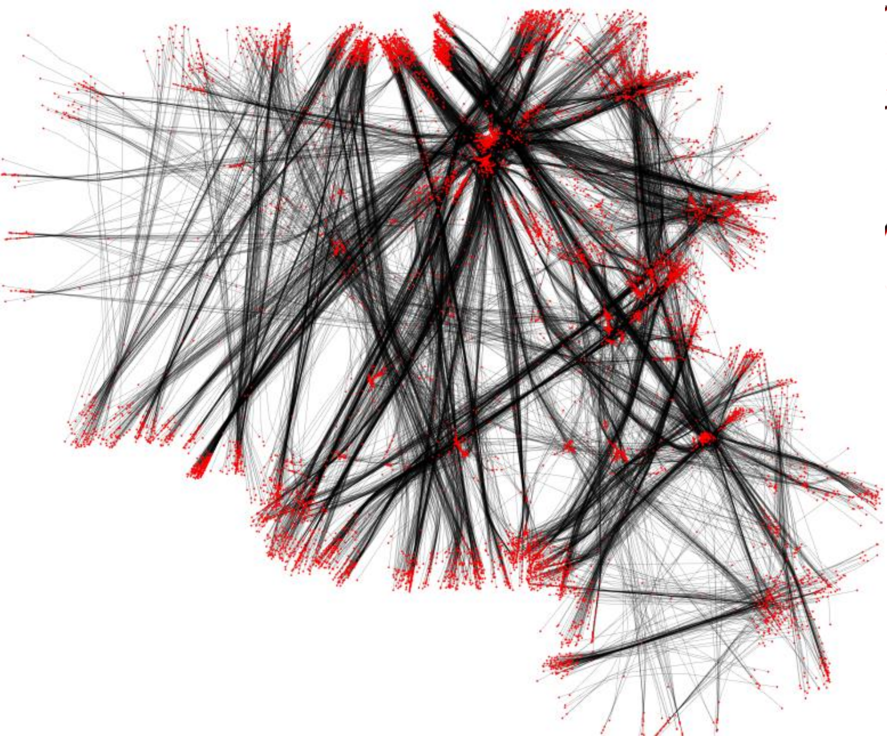
Graph	Nodes	Edges	Edge samples	Bundling time (sec.)	
				8800 GTX	GeForce 580
Radial	1024	4021	290K	4.5	1.5

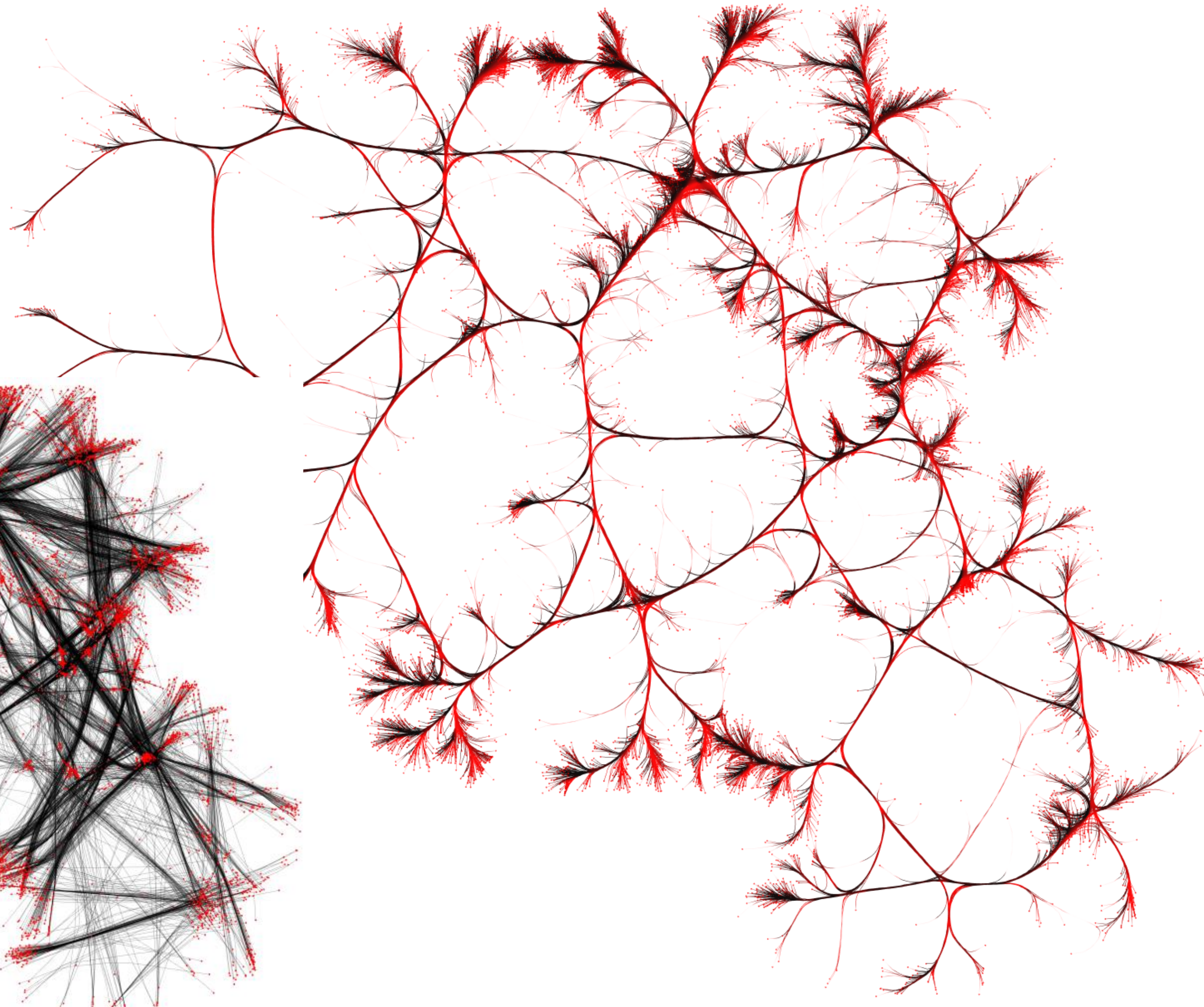




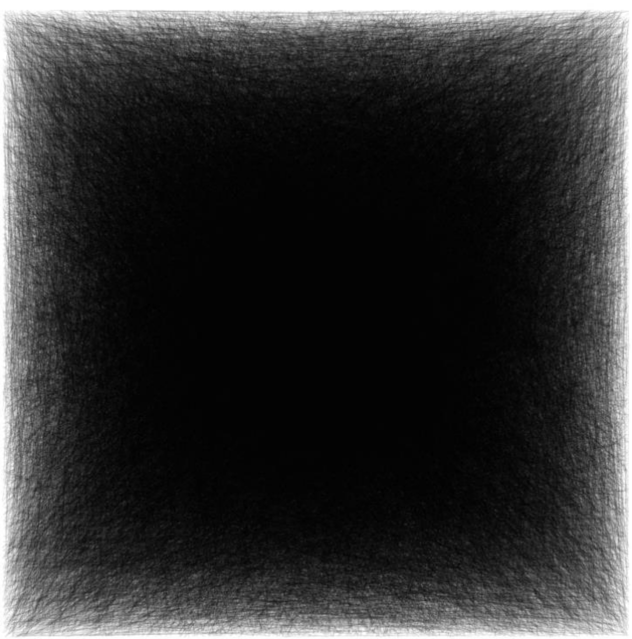


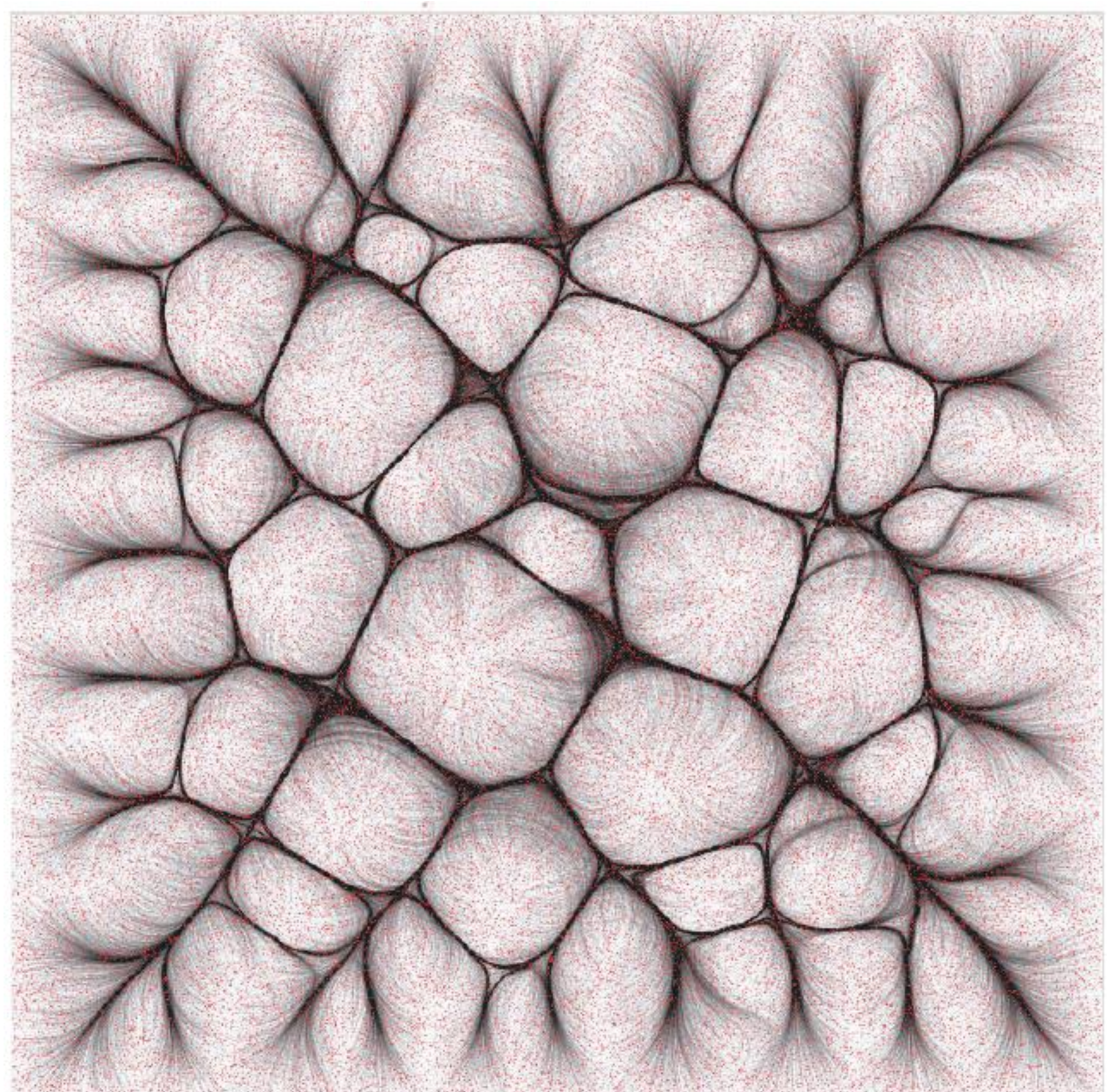
Graph	Nodes	Edges	Edge samples	Bundling time (sec.)	
				8800 GTX	GeForce 580
France air	34550	17275	330K	3.8	1.8





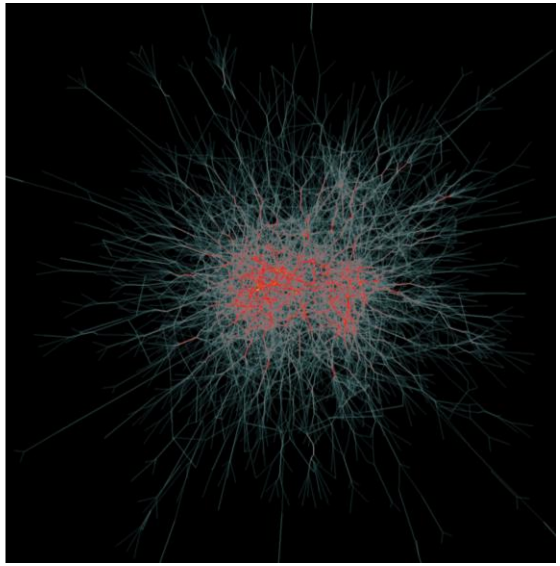
Graph	Nodes	Edges	Edge samples	Bundling time (sec.)	
				8800 GTX	GeForce 580
Random	200K	100K	4.8M	43	18

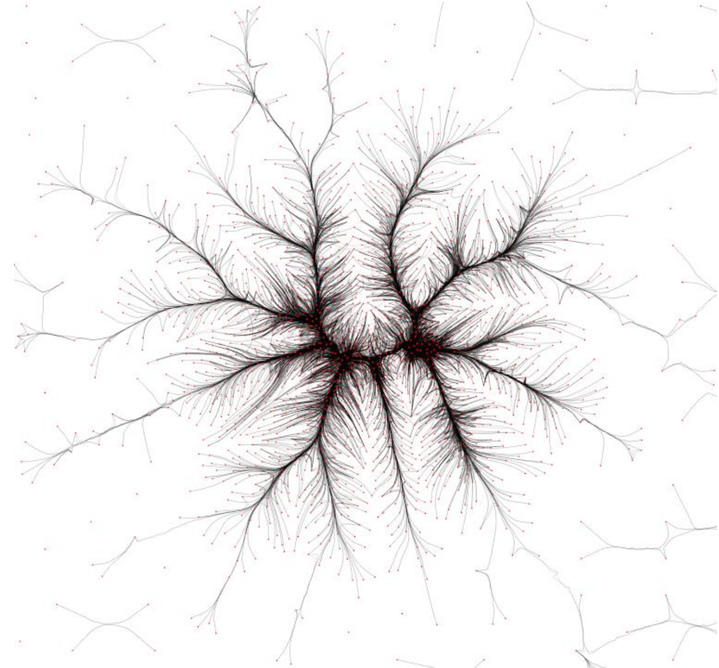




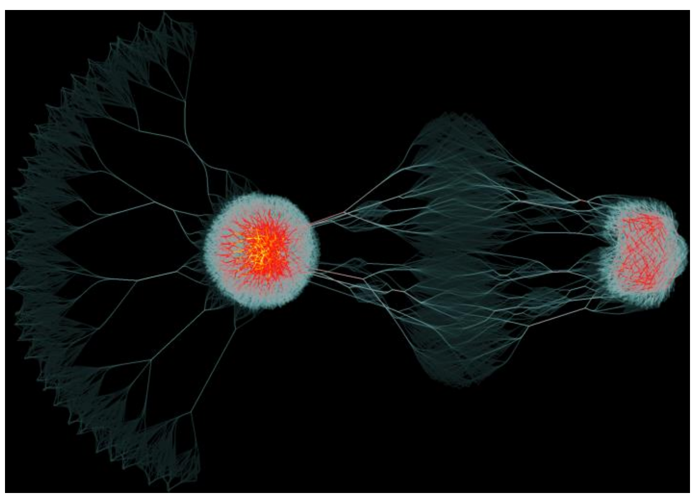
Comparison: MINGLE[1]

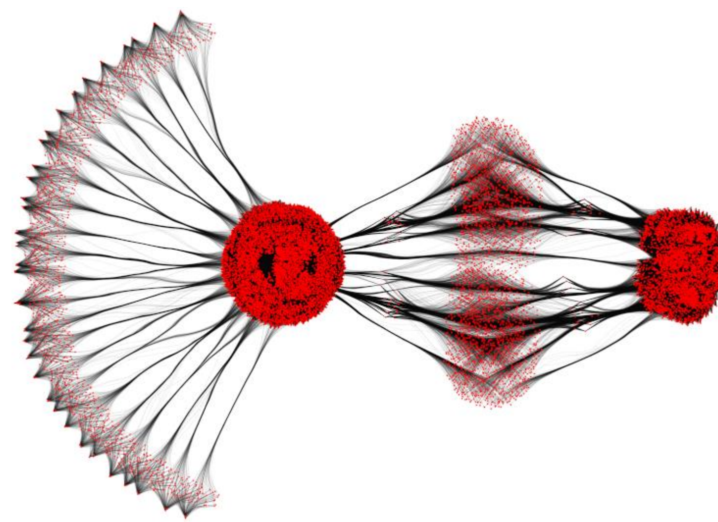
Yeast graph
6 646 edges





Net-50 graph
464 440 edges

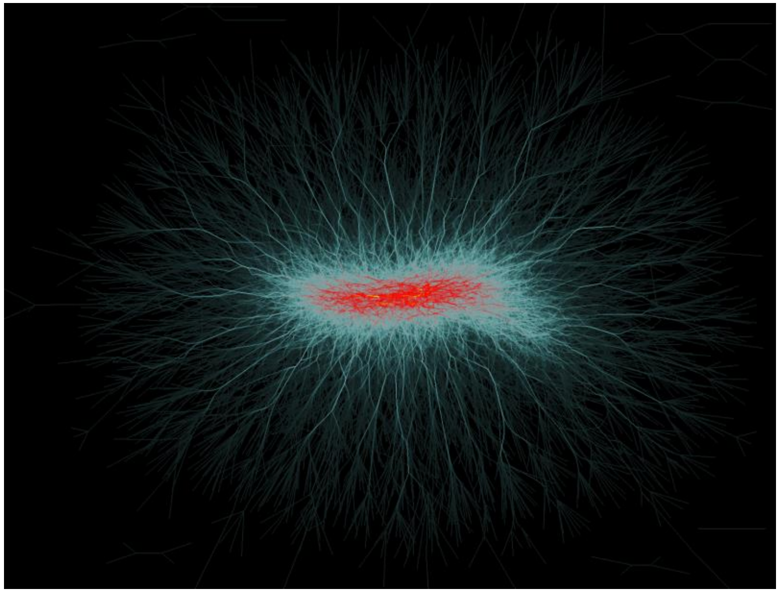


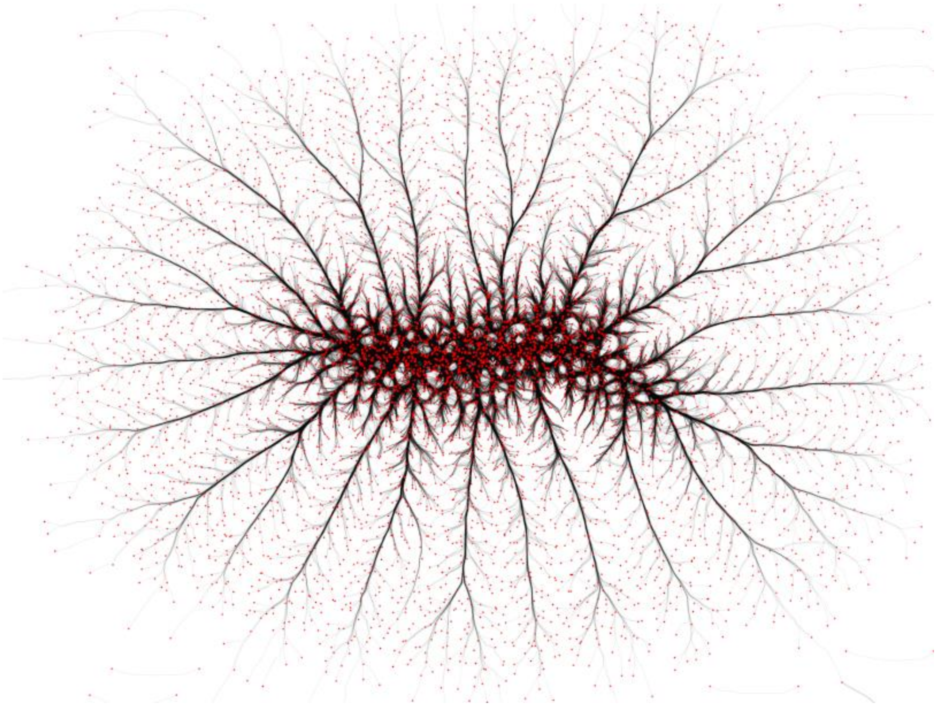


[1] GANSNER E., HU Y., NORTH S., SCHEIDEGGER, C.: Multilevel agglomerative edge bundling for visualizing large graphs. In Proc. PacificVis (2011),

Comparison: MINGLE

Wiki graph 100 762 edges



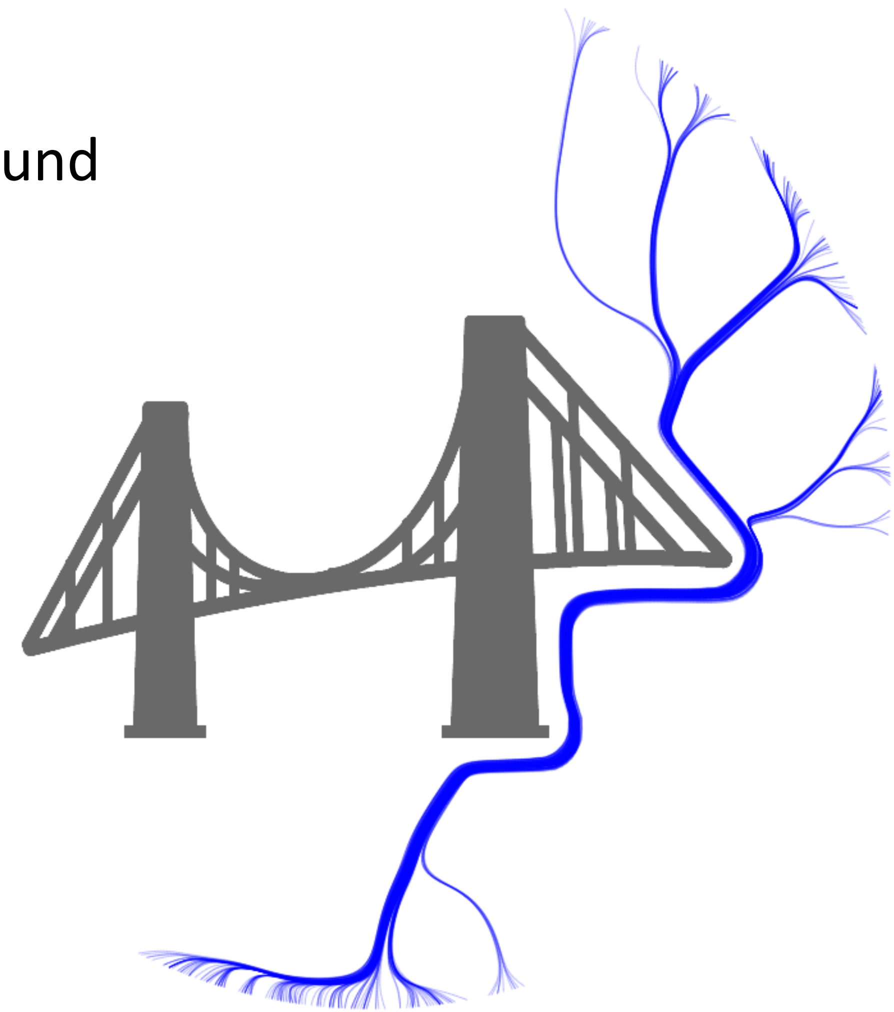


- we produce **less clutter** than MINGLE
- we achieve roughly the **same speed** for small dataset, but **three time faster** with big datasets

Obstacle-constrained bundles

Why: avoid regions (labels, icons)

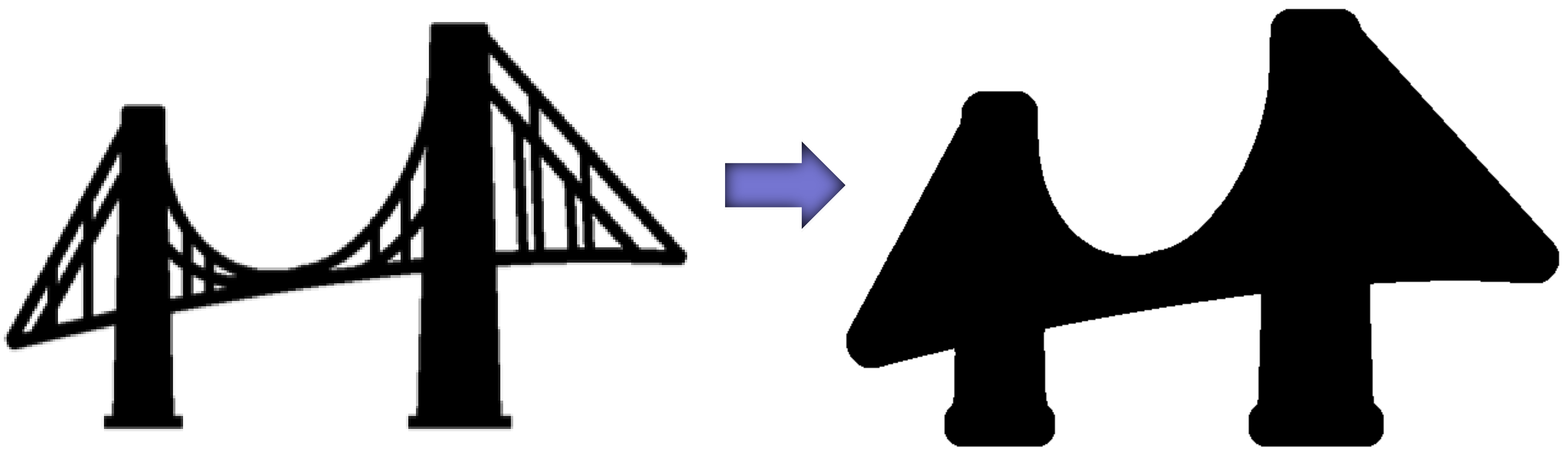
How: smoothly route bundles around obstacles of arbitrary geometry



Obstacle-constrained bundles

Technique

Inflate the obstacle Ω

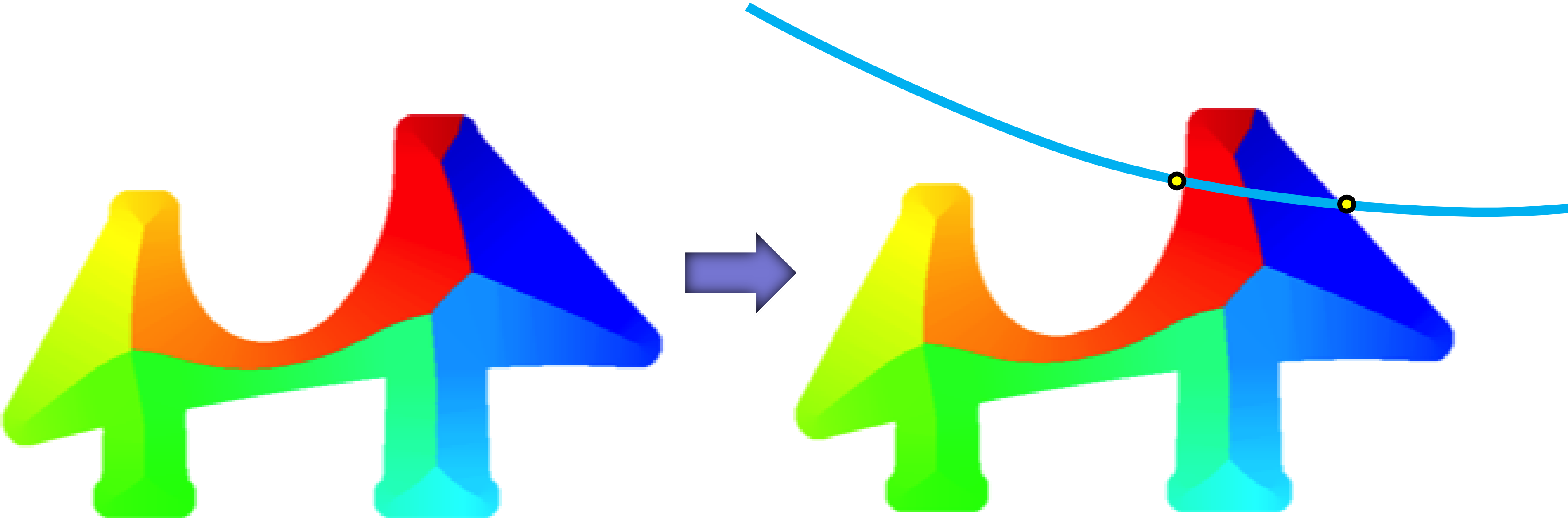


Obstacle-constrained bundles

Technique

Compute the feature transform FT_{Ω}

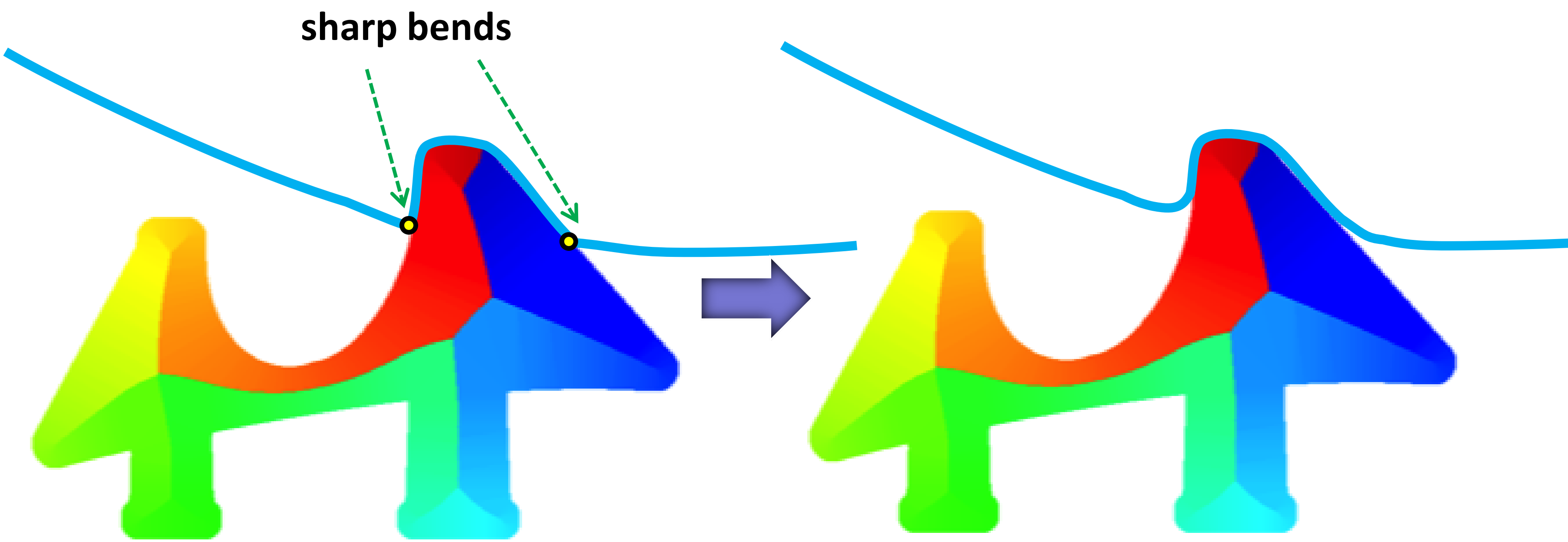
Find where the edges enter and exit the obstacle



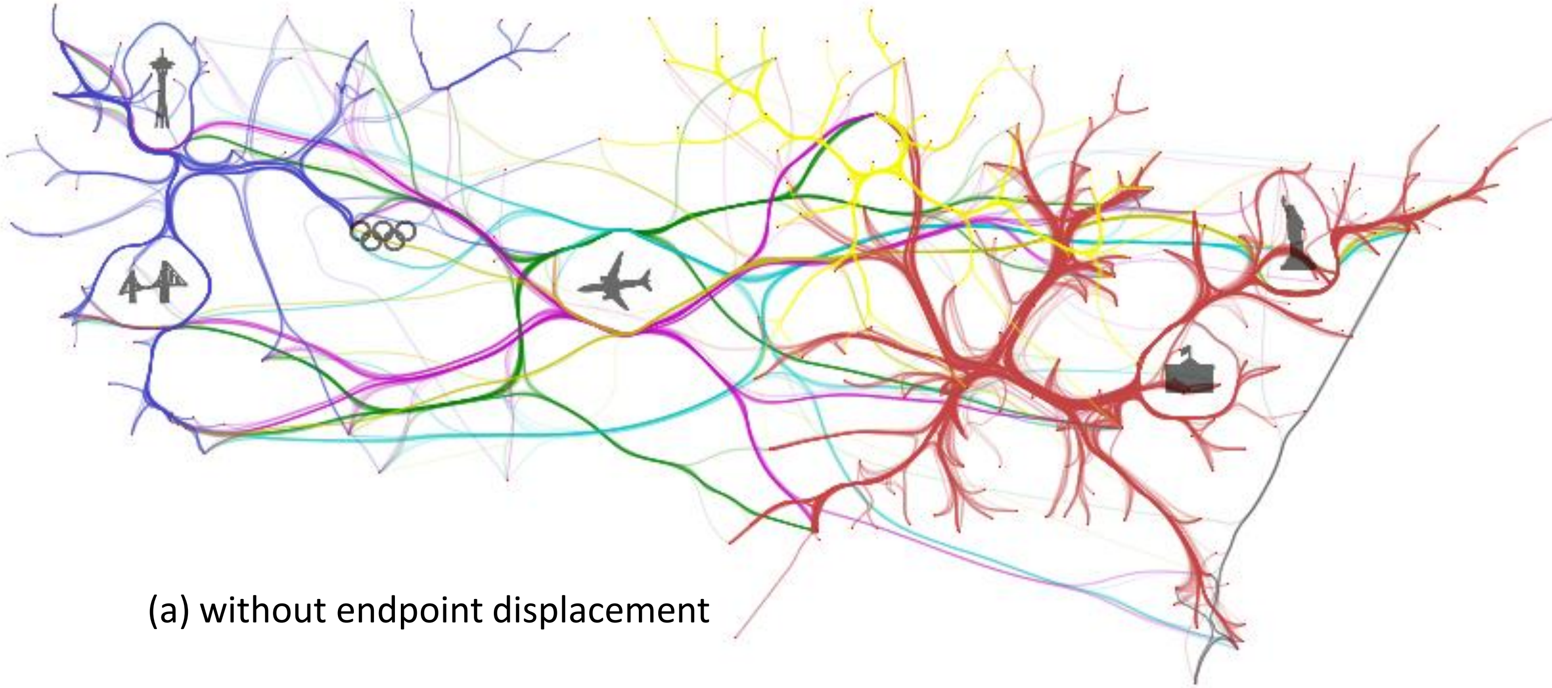
Obstacle-constrained bundles

Technique

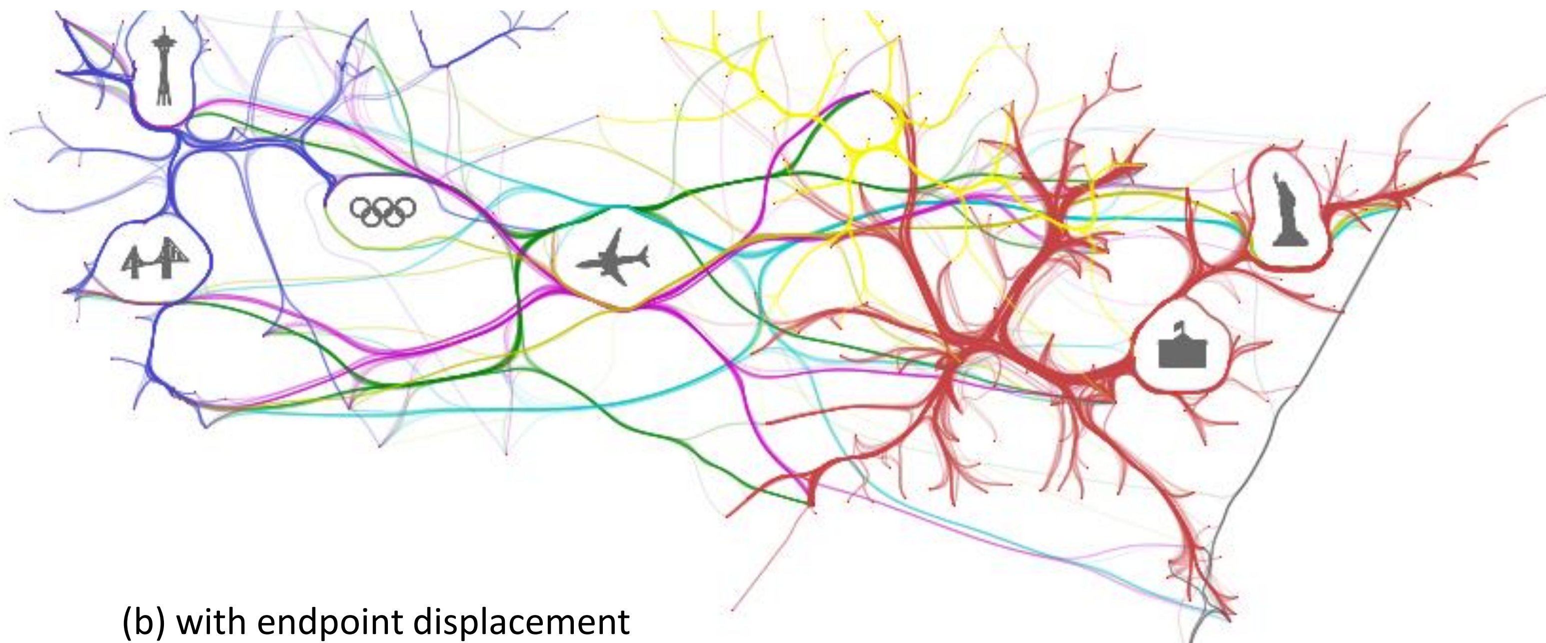
- Replace edge segment with shortest path along boundary of Ω
- Smooth resulting edges (constraining this outside Ω)



US Migrations Dataset



(a) without endpoint displacement



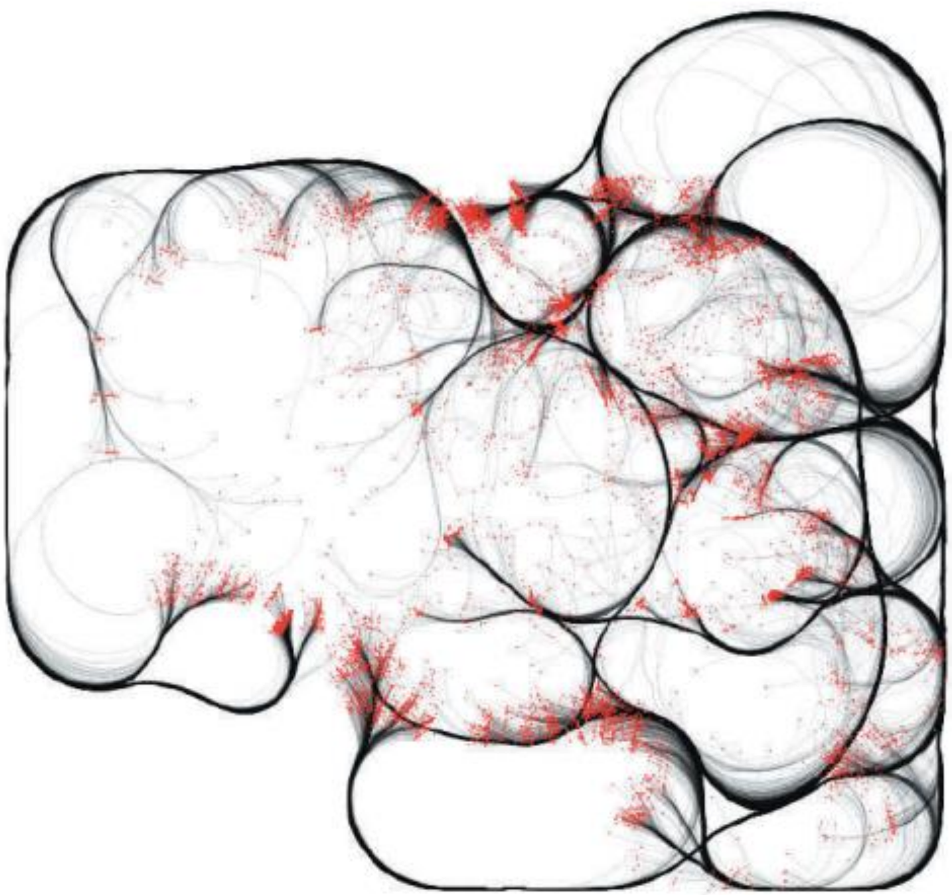
(b) with endpoint displacement

Graph Bundling by Kernel Density Estimation

Simple: image-based techniques (~500 LOC)

Fast: 5..20 faster than comparable methods

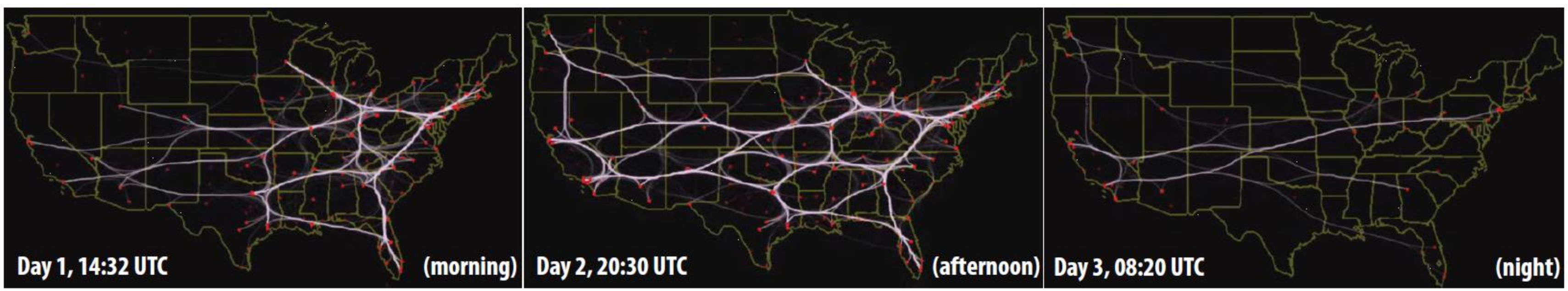
Additions: simple/generic obstacle avoidance

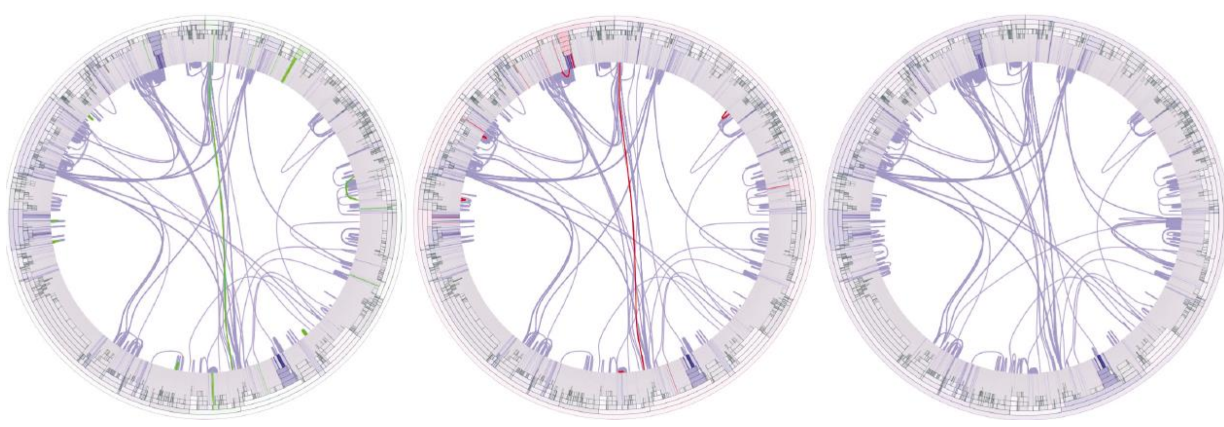


Smooth Bundling of Large Streaming and Sequence Graphs

C. Hurter, O.Ersoy, A.Telea

PacificVis 2013





Bundling methods remains mainly used for static graph

Dynamic graph visualization challenges:

Larger data volumes than static graph

Show changes in the overall graph structure

Avoid abrupt changes in the graph layout

Design guide lines:

- Use of graph interpolation
- Use of KDEEB (Kernel Density Estimation Edge Bundling)

KDEEB Assets

(Graph Bundling by Kernel Density Estimation)

Continuity (iteration = bundling strength)

Stability : mean shift[1] converges

Simplicity and speed

Implicit design

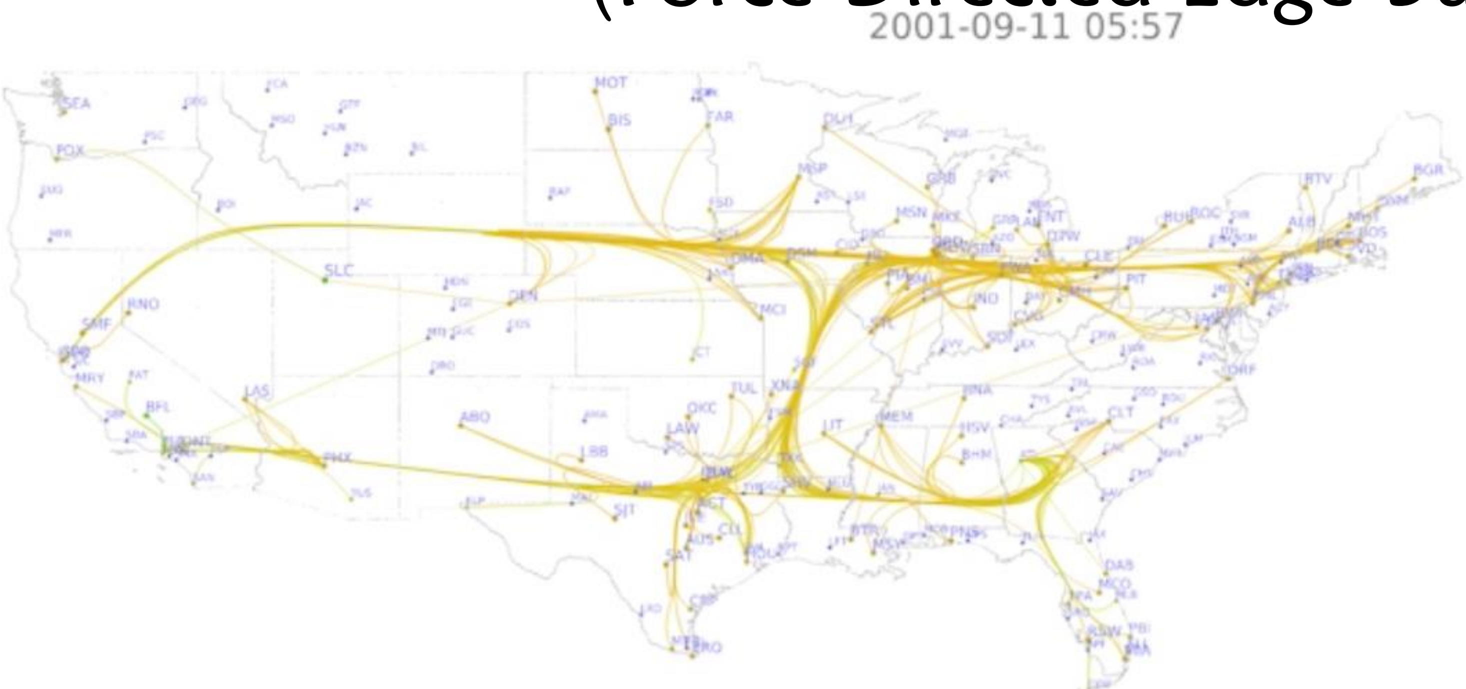
[1] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE TPAMI, 24(5):603–619, 2002.

Conclusion:

KDEEB is ideal for bundling changing graphs

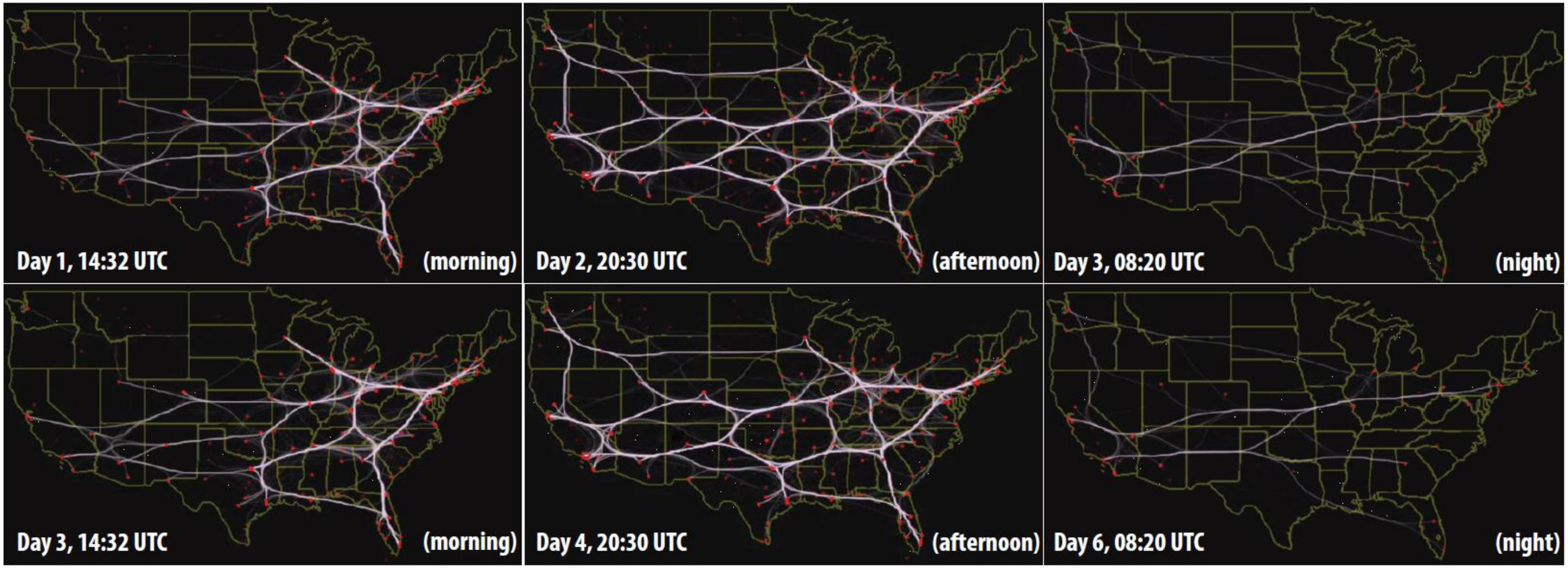
- Mouse bundling demo -

Animation with FDEB
(Force Directed Edge Bundling)



(Video)

[Q. Nguyen, P. Edges, and S.-H. Hong. StreamEB: Stream edge bundling. In Proc. Graph Drawing, pages 324–332, 2012]



[Hurter et al. PacificVis 2013]

Mouse trail Demo

Streaming and sequence graph

Streaming graph: temporally ordered, unstructured edge sequences (start, end, life time).

Sequence graph: discreet set of graphs with edge correspondence over time.

Discussion

streaming graphs

Fits naturally to the KDEEB method

Set bundling time = physical time

Cannot be (easily) done with other bundling methods

sequence graphs

Need correspondence information

Can use other bundling methods than KDEEB

Main KDEEB asset: speed

Complexity

streaming graph visualization: $O(E)$ per animation frame, where E is the average number of edges

sequence graph visualization: $O(BN)$ for a sequence of N graphs, and a bundling algorithm of complexity B

Performance

2.3 GHz PC with 8 GB RAM, NVidia GT 480 card

Implementation C# and OpenGL 1.1, 2.3 GHz

US dataset (streaming graph): **0.05 seconds/frame** (2K edges on average)

France dataset (streaming graph): **0.17 seconds/frame** (15K edges on average)

we are roughly 10 times faster than the original KDEEB which is expected, as we do only one iteration per frame. the total time needed for a stream depends on the stream's length.

In contrast, if we were to use FDEB (1.7 GHz PC), we would

US dataset (streaming graph): **19 seconds/frame** (2K edges on average)

Small graph: **6 seconds/frame** (900 edges)

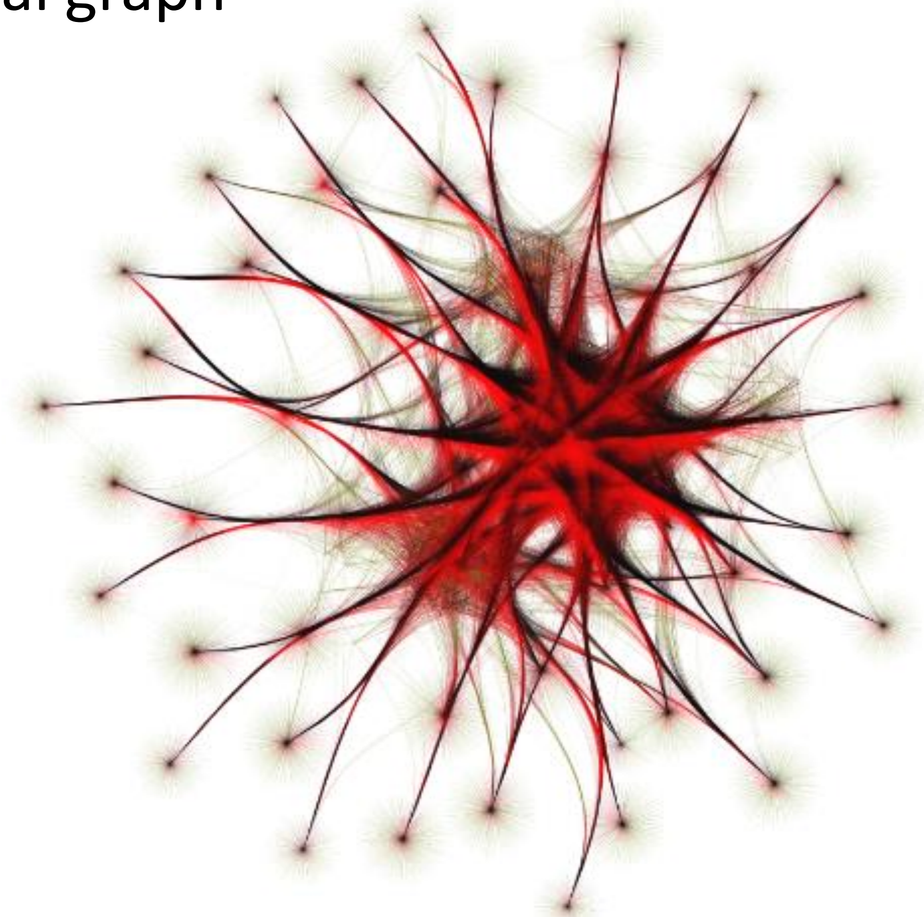
Summary of contributions

Real time EB: image-based techniques

Simple: few parameters

Flexible: applicable on general graph

Scalable



Edge Bundling papers