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. Ecole Nationale de l'Aviation Civile, Toulouse, France

PART III:
APPLICATIONS &
OPEN RESEARCH QUESTIONS

Speakers: Hans-Jörg Schulz & Christophe Hurter

Applications

1. Visualizing the DBLP dataset
   (914,492 nodes / 3,802,317 edges / 22 time points)
2. Visualizing flight routes
   (34,550 nodes / 17,275 edges)
3. Software engineering Visualization
   (13,856 nodes / 48,591 edges)
4. Visualizing eye tracking data
   (5,000 nodes / 2,500 edges)

1. Visualizing the DBLP dataset
(914,492 nodes / 3,802,317 edges / 22 time points)

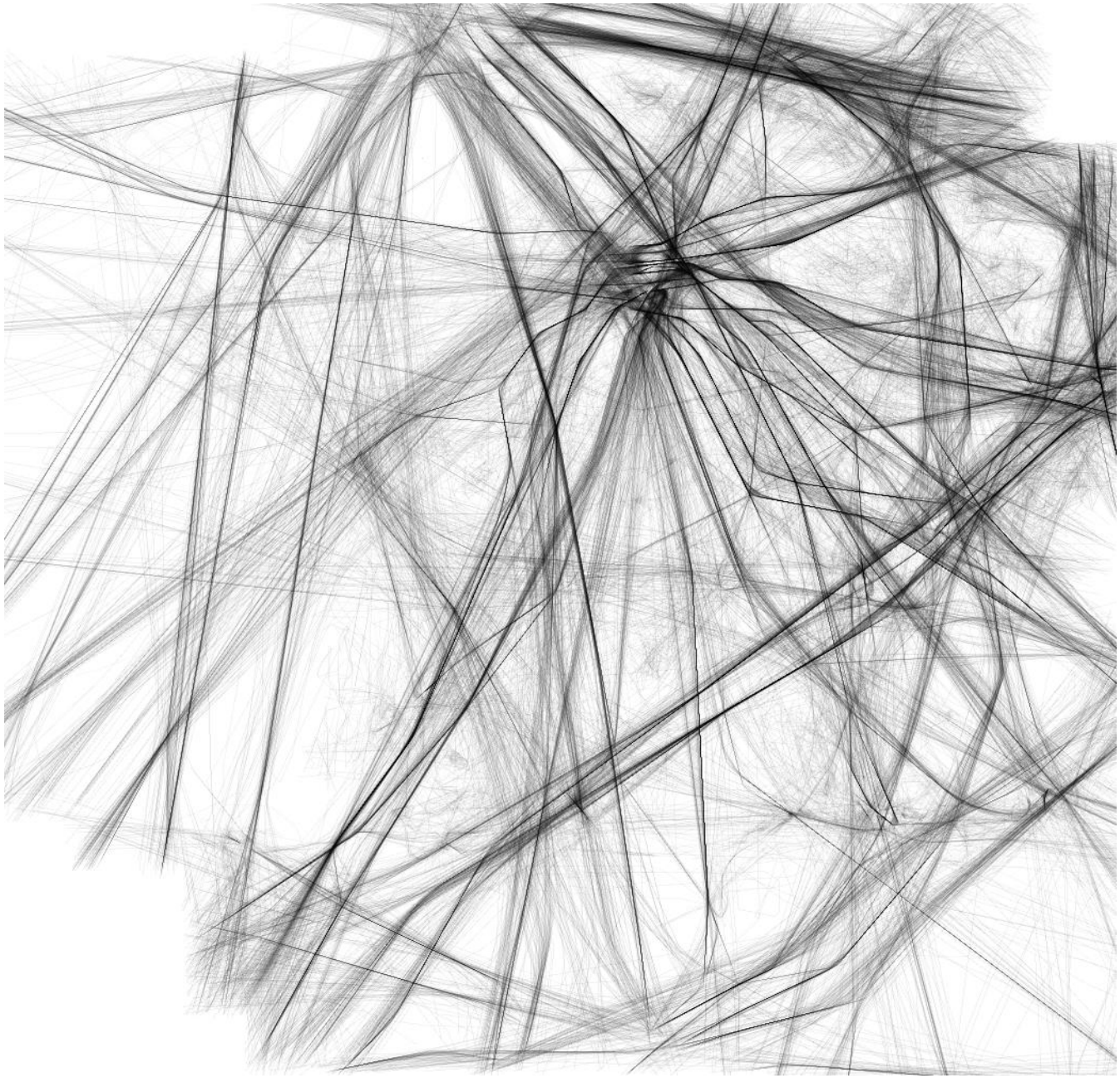
Abello2014 - Modular DOI Function.avi

FromDaDy: Spreading Aircraft Trajectories Across Views to Support Iterative Queries

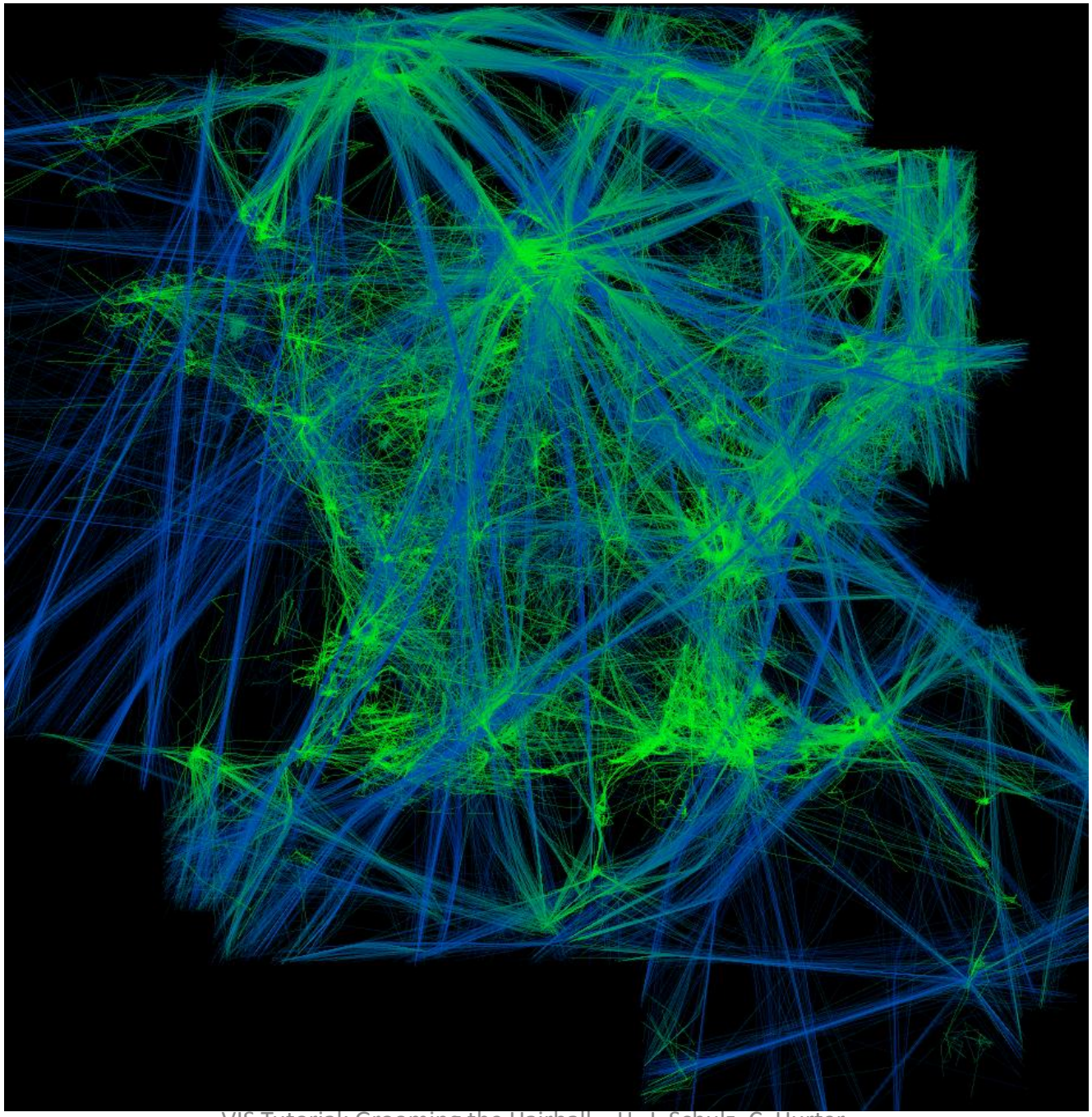
Christophe Hurter, Benjamin Tissoires, Stephane Conversy,

IEEE Transactions on Visualization and Computer Graphics 15, 6 (November 2009), 1017-1024.

Infovis 2009



[Hurter et al. Infovis 2009]

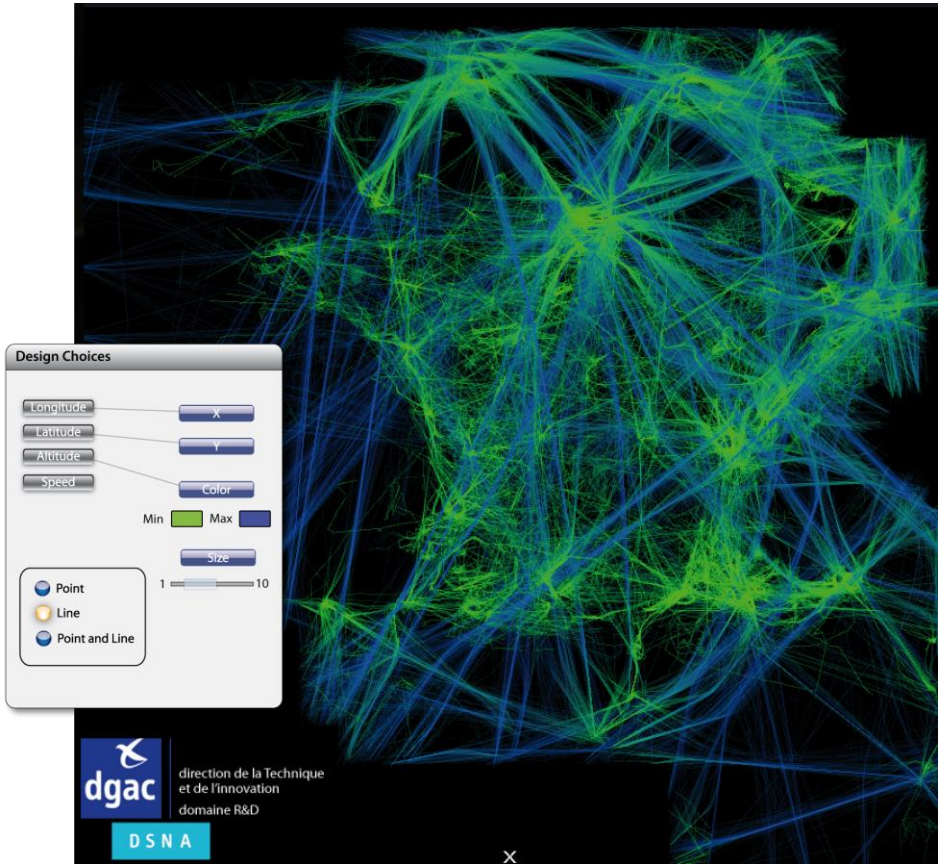


[Hurter et al. Infovis 2009]

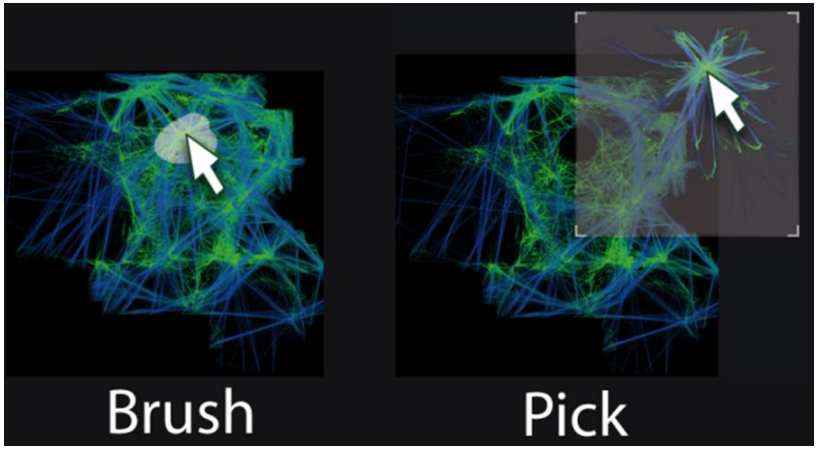
Demo

Exploration of large datasets

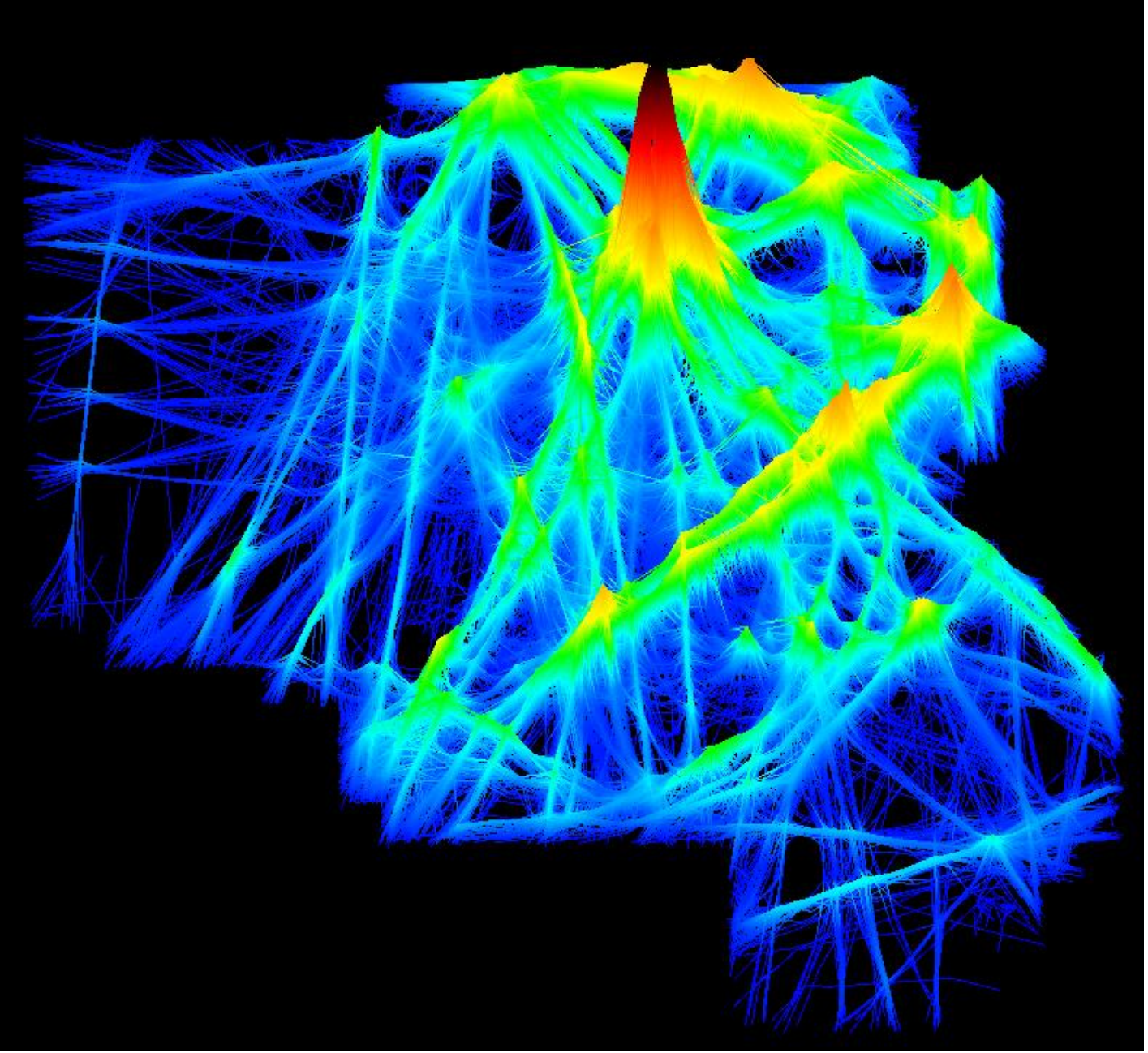
Design exploration



Interaction paradigms



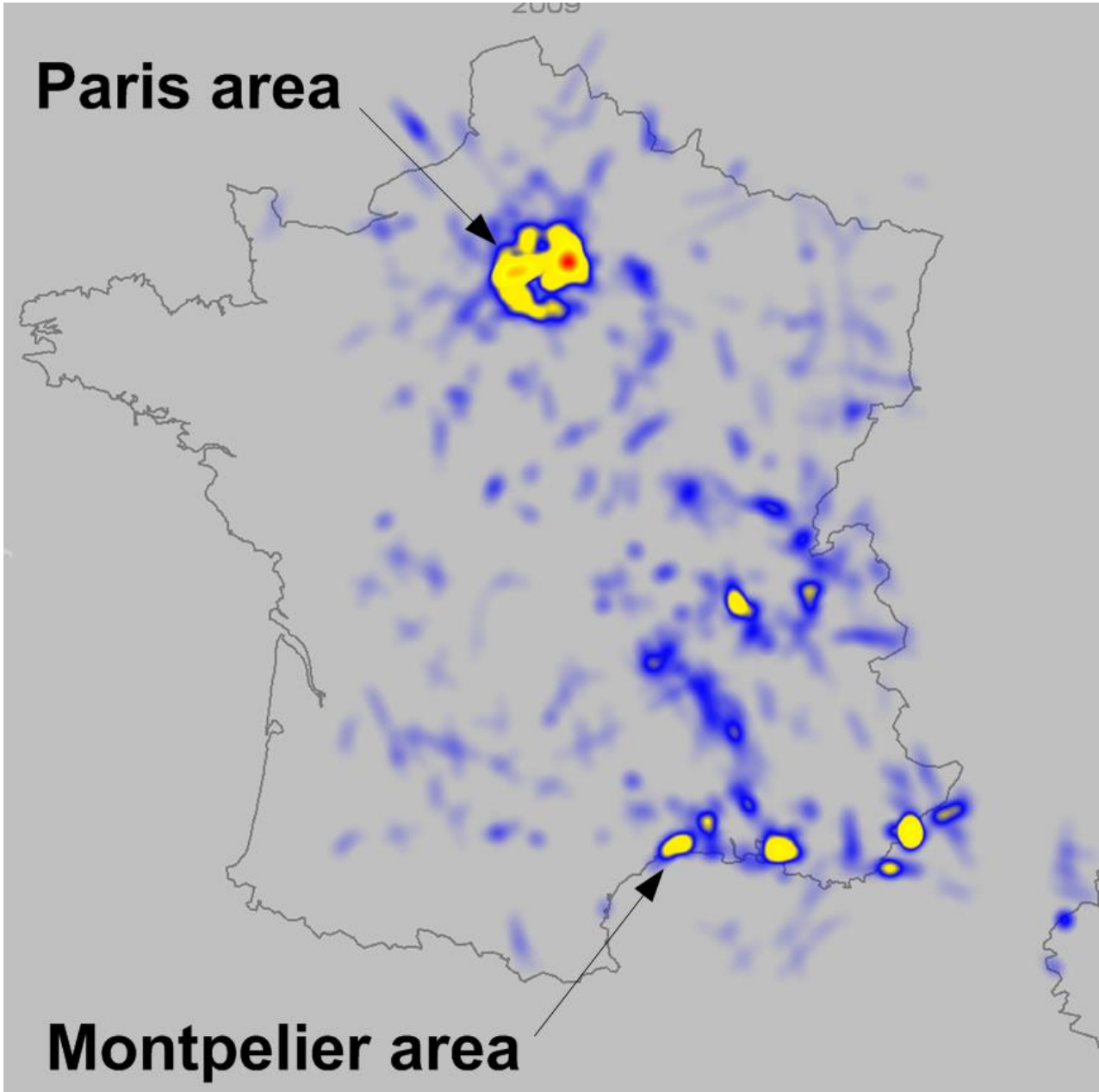
[Hurter et al. Infovis 2009]



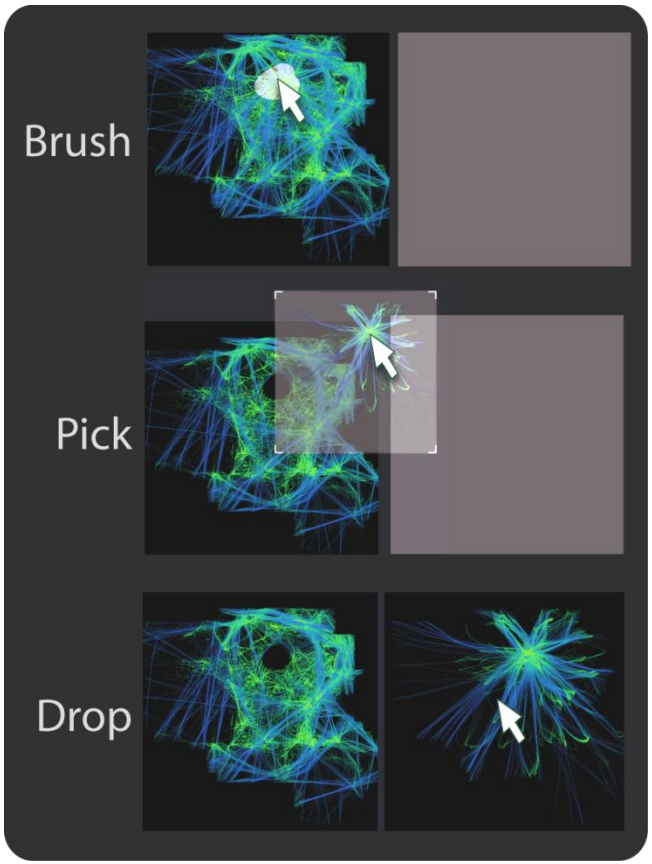
[Hurter et al. Infovis 2009]

Improving safety

Accumulation map of aircraft proximity alarm (TCAS).

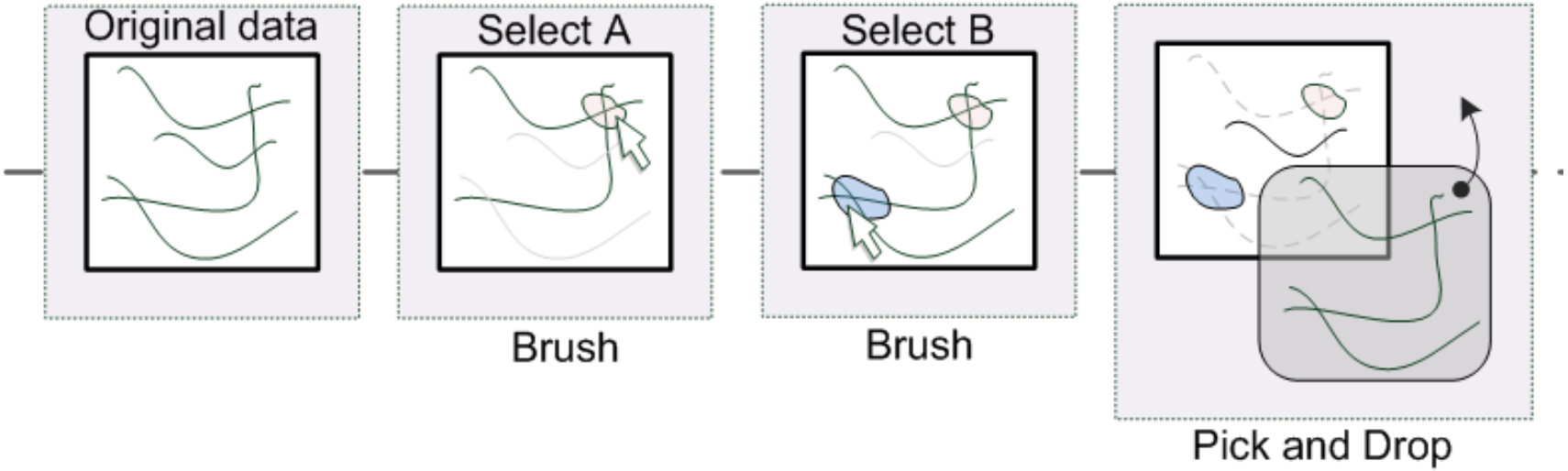


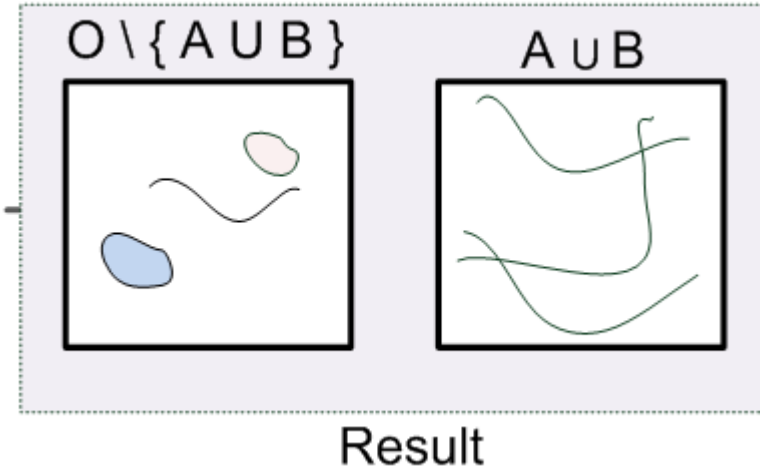
FromDaDy features



- Visual configuration tool
- Brushing and incremental selection
- Pick and drop
- A matrix of juxtaposed views
- Spreading trajectories across views

Boolean operations

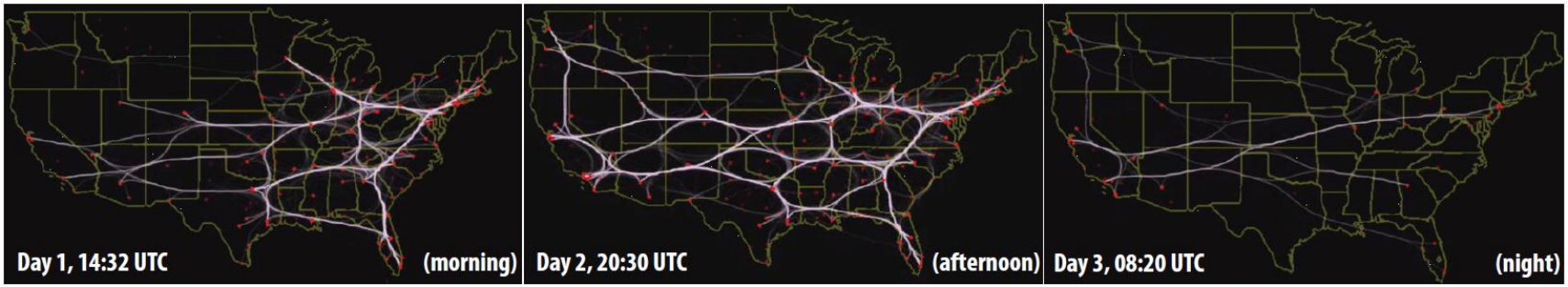


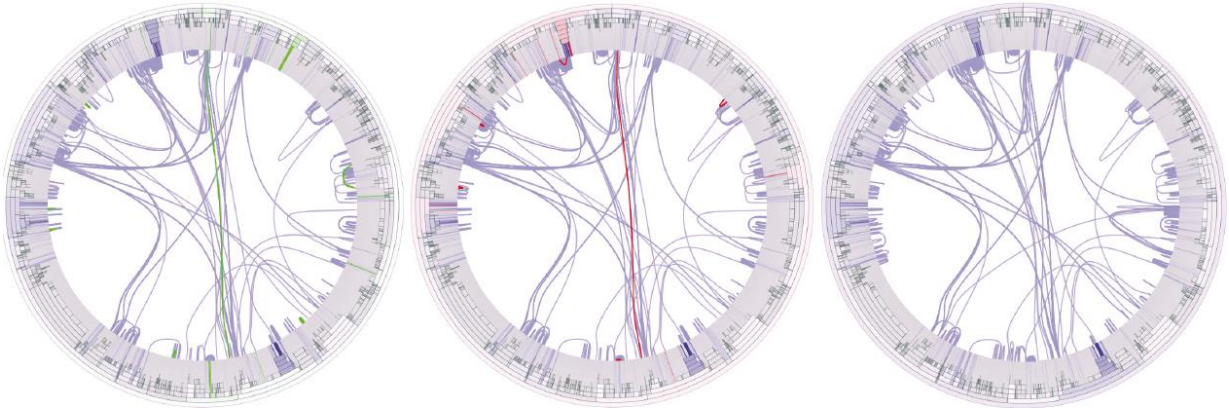


Bundled Visualization of Dynamic Graph and Trail Data

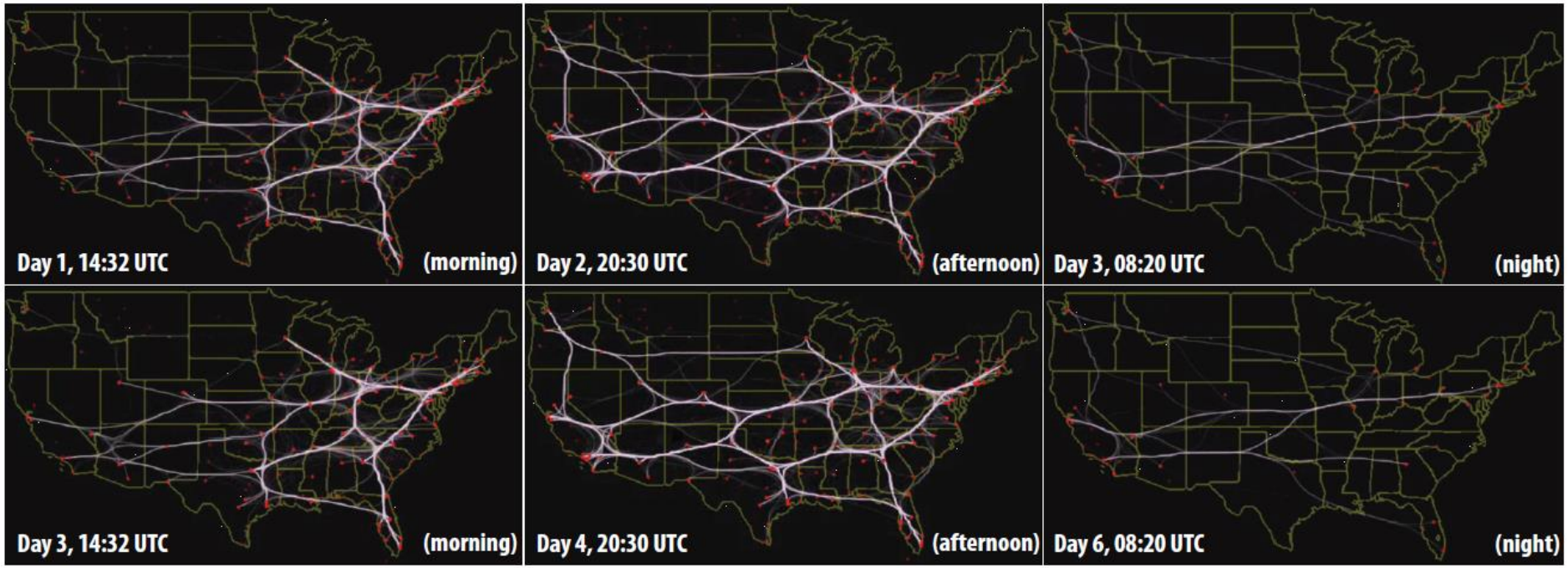
C. Hurter, O.Ersoy, S.I. Fabrikant, T.R. Klein, A. C. Telea

TVCG 2013



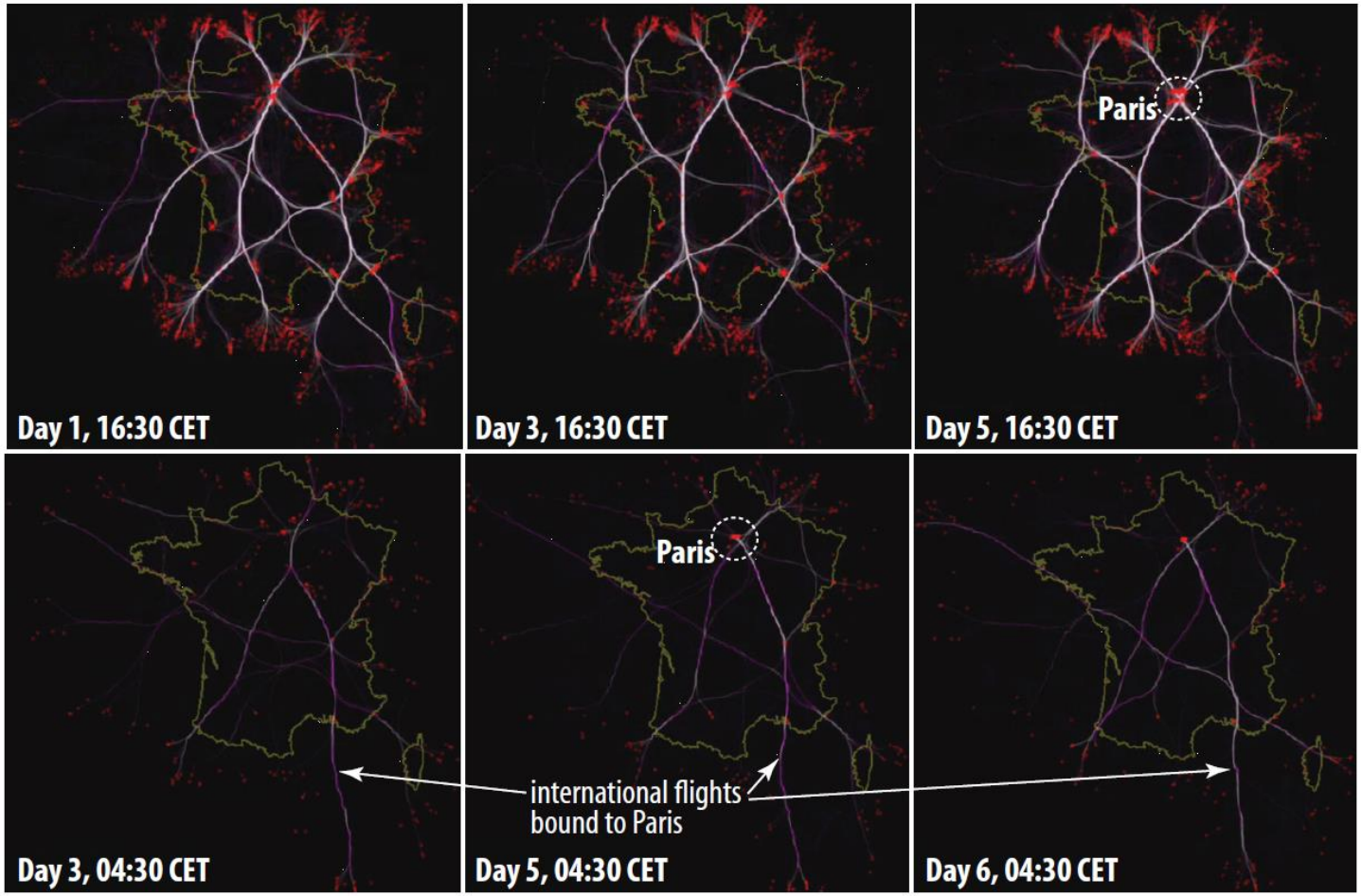


Streaming graph 6 weeks US flight



Streaming visualization for 6-days US airline flight dataset

Streaming graph
One week France aircraft



Software engineering
sequence graph

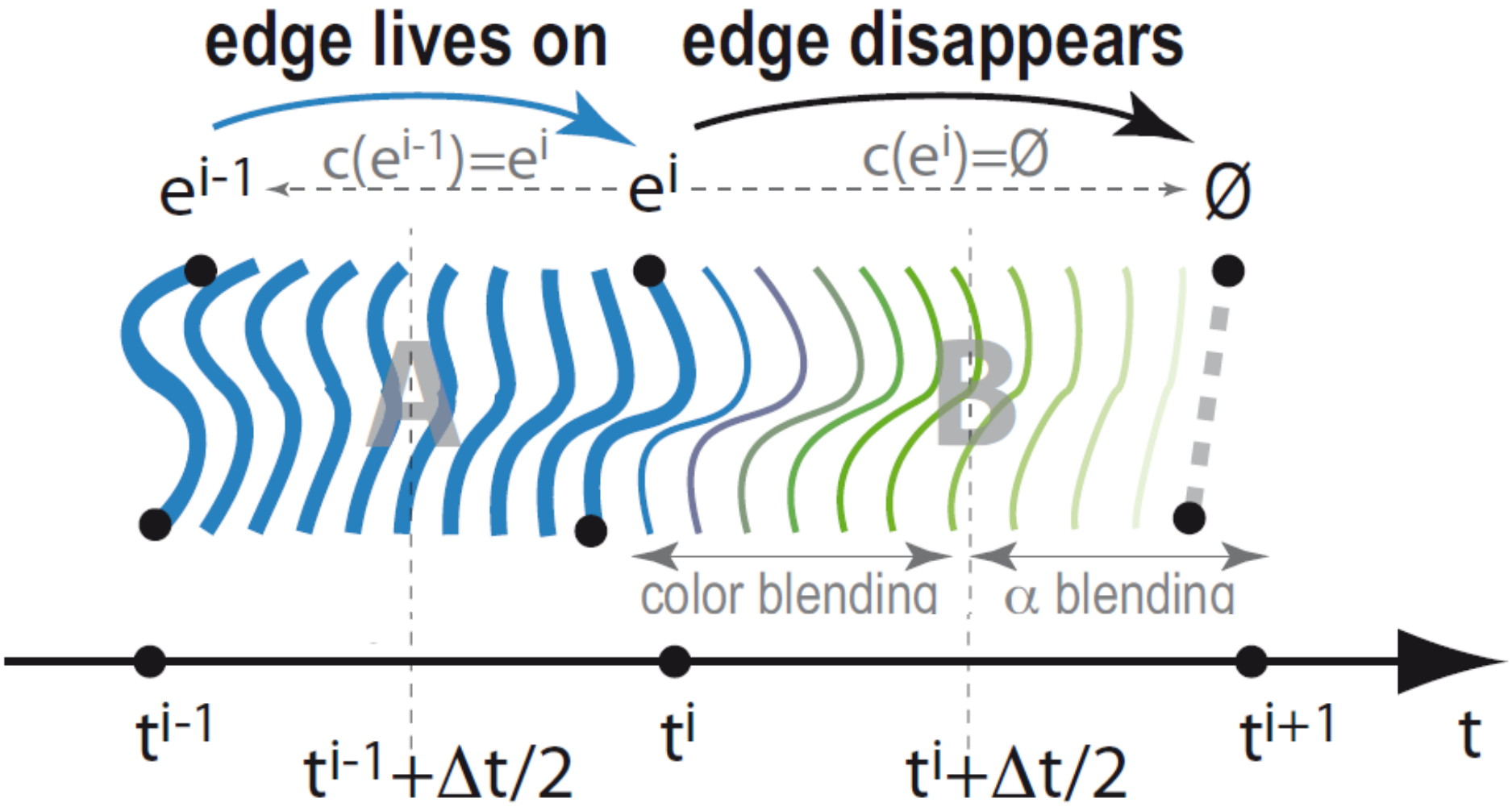
The dataset contains **22 releases** of Mozilla-Firefox[1]

For each revision, we extracted the code hierarchy (folders and files), and **code duplicates**, using the clone detector SolidSDD[2]

[1] Mozilla. Firefox repository, 2012. http://www.mozilla.org/en-US/firefox/fx
[2] SolidSource IT. SolidSDD Clone Detector, 2012. http://www.solidsourceit.com

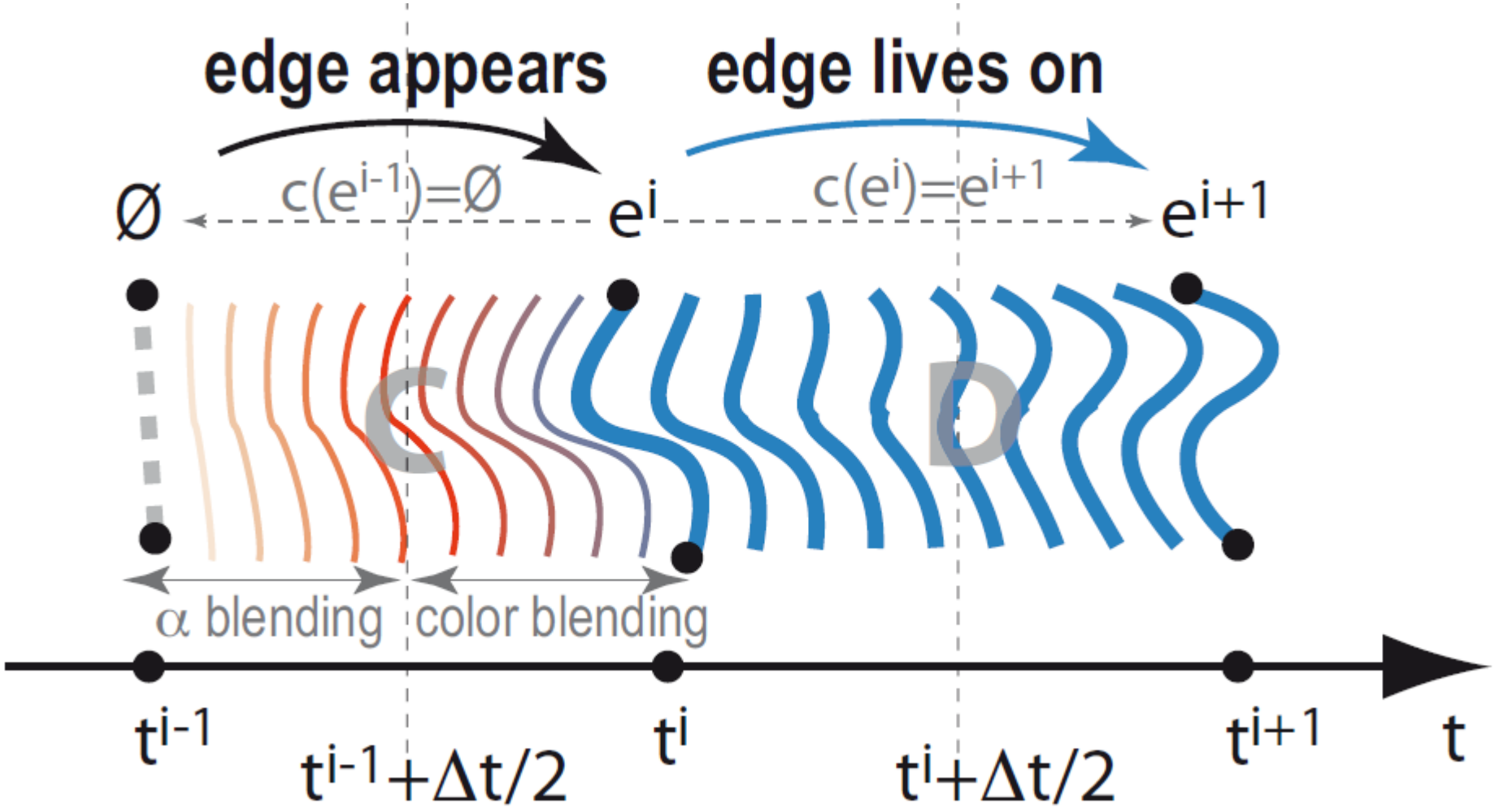
Disappearing edge

sequence graph

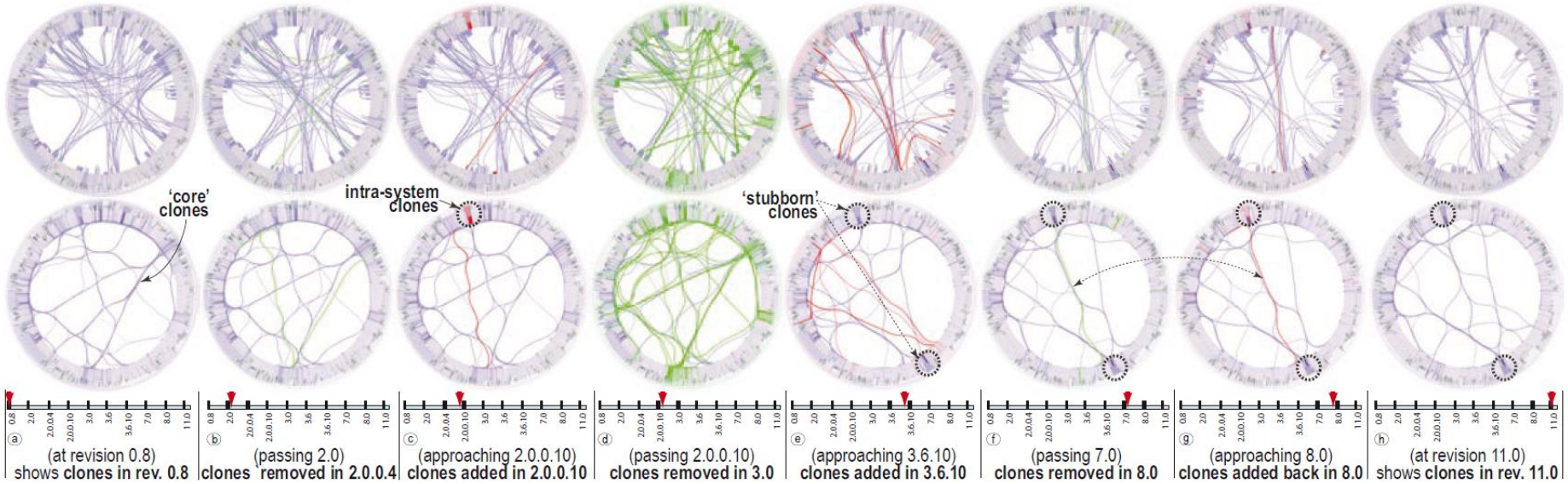


Edge appearing

sequence graph

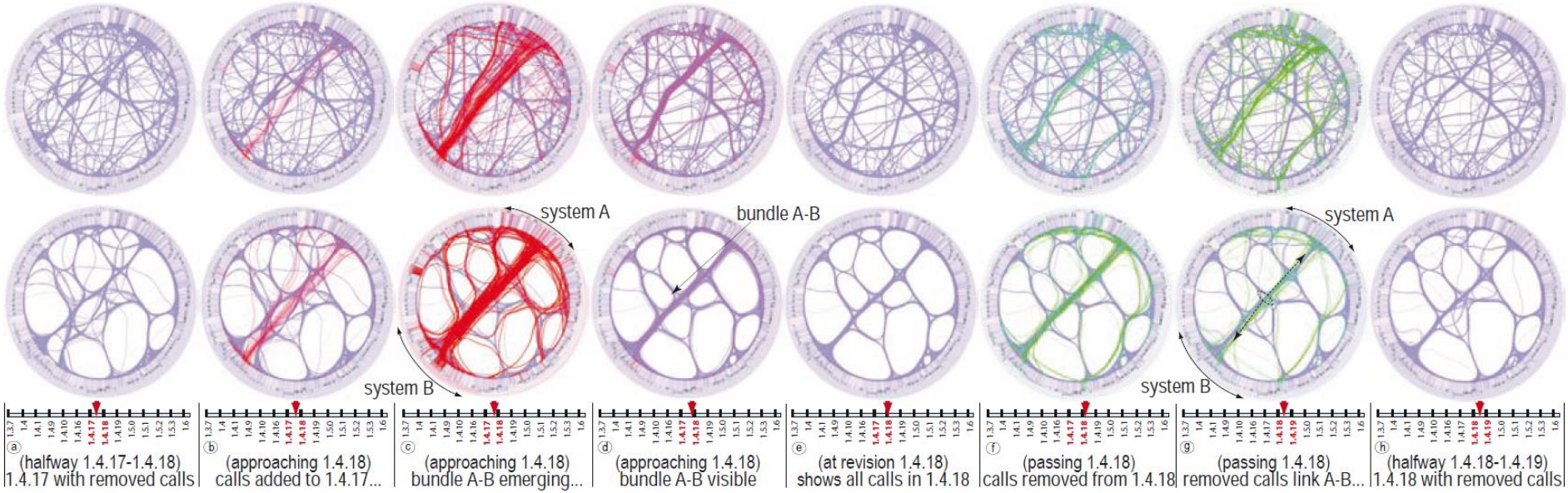


Sequence-based visualization for clones in Firefox



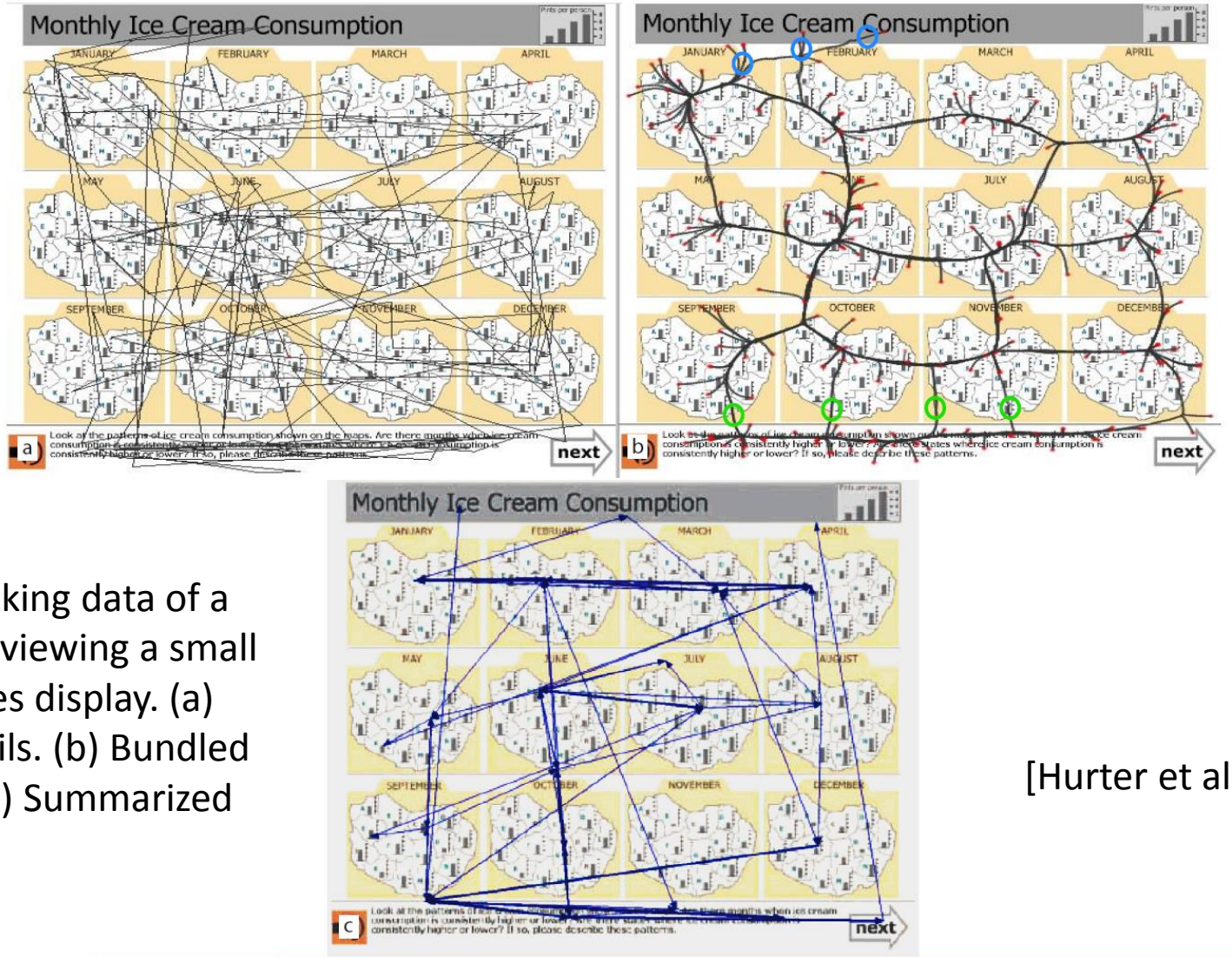
Top row: HEB bundling. Bottom row: KDEEB bundling

Sequence-based visualization for Wicket call graphs



Top: SBEB bundling. Bottom: KDEEB bundling

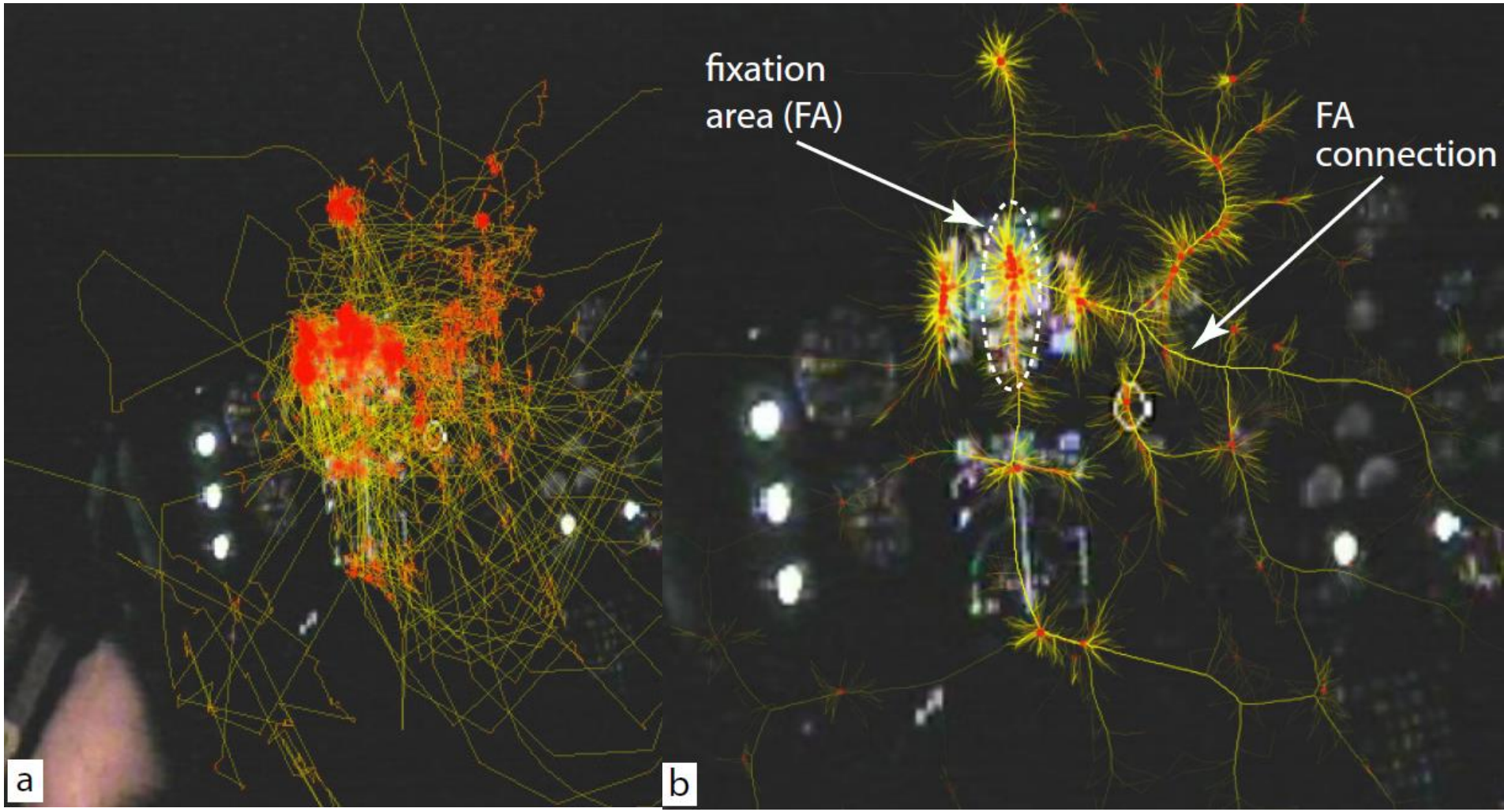
Eye tracking data



Eye tracking data of a subject viewing a small multiples display. (a) Raw trails. (b) Bundled trails. (c) Summarized trails

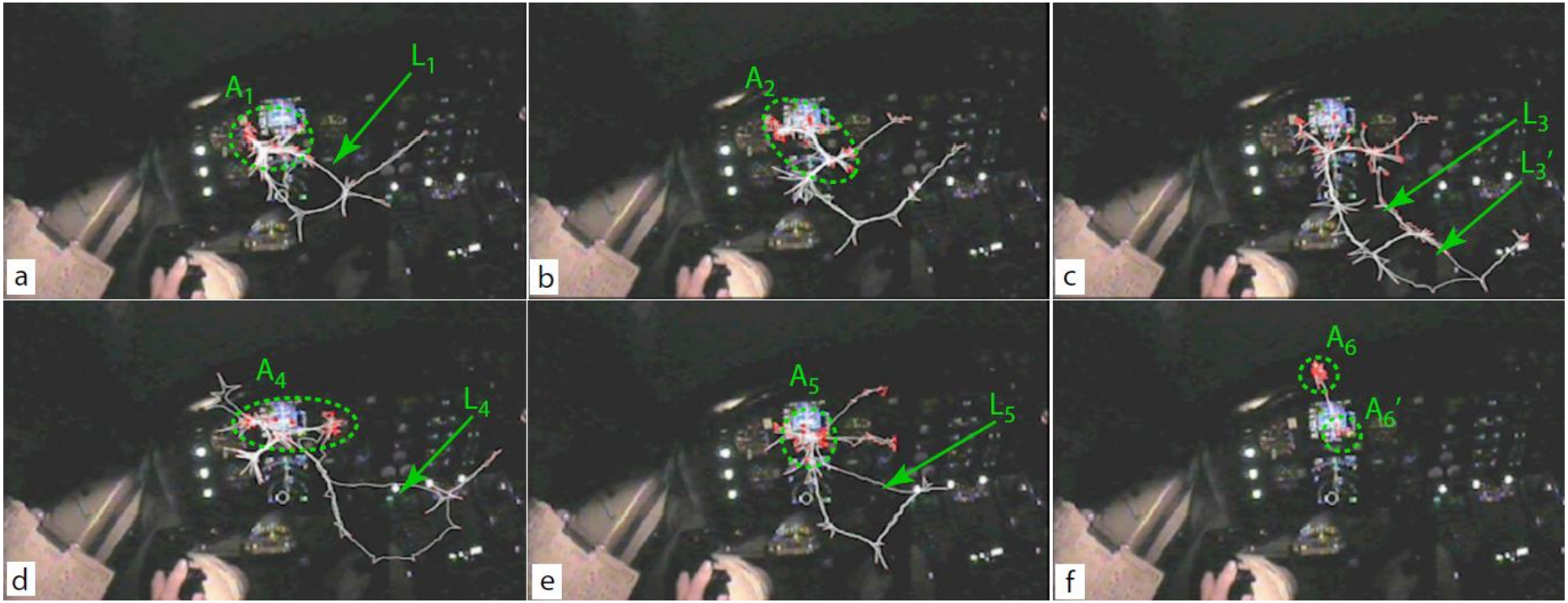
[Hurter et al. TCVG 2013]

Eye tracking data



[Hurter et al. TCVG 2013]

Streaming data eye tracking



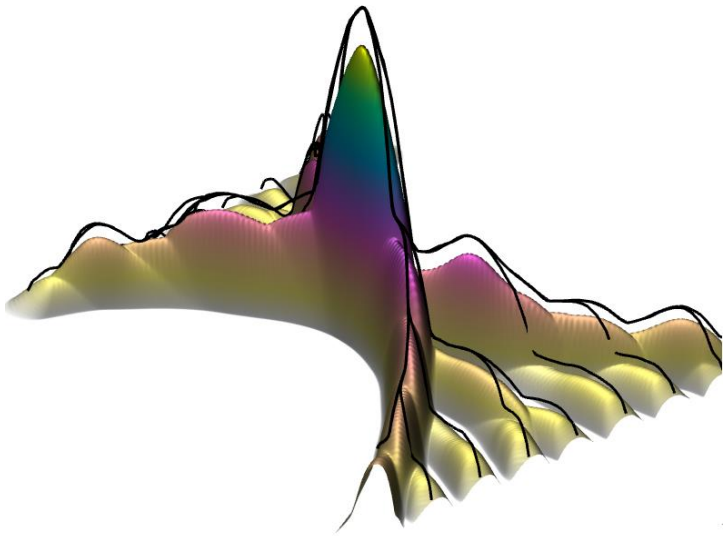
Visual analysis of pilot eye-tracking data, with dynamic bundling. Six salient eye-movement patterns are shown

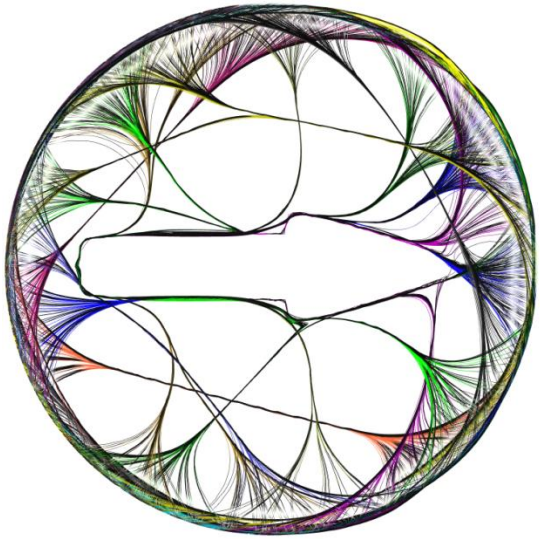
[Hurter et al. TCVG 2013]

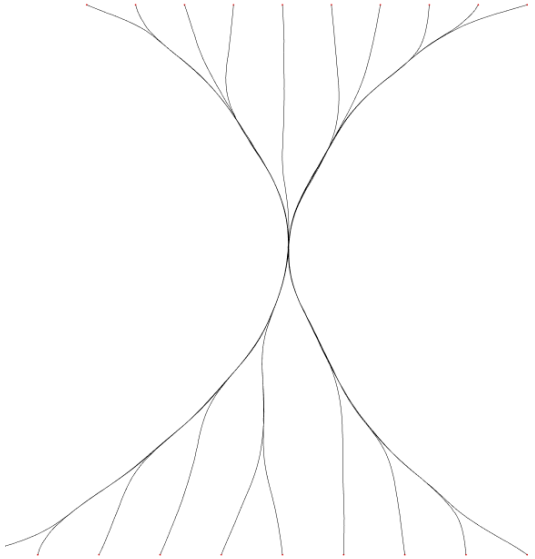
Graph Bundling by Kernel Density Estimation

C. Hurter, O.Ersoy, A.Telea

Eurovis 2012

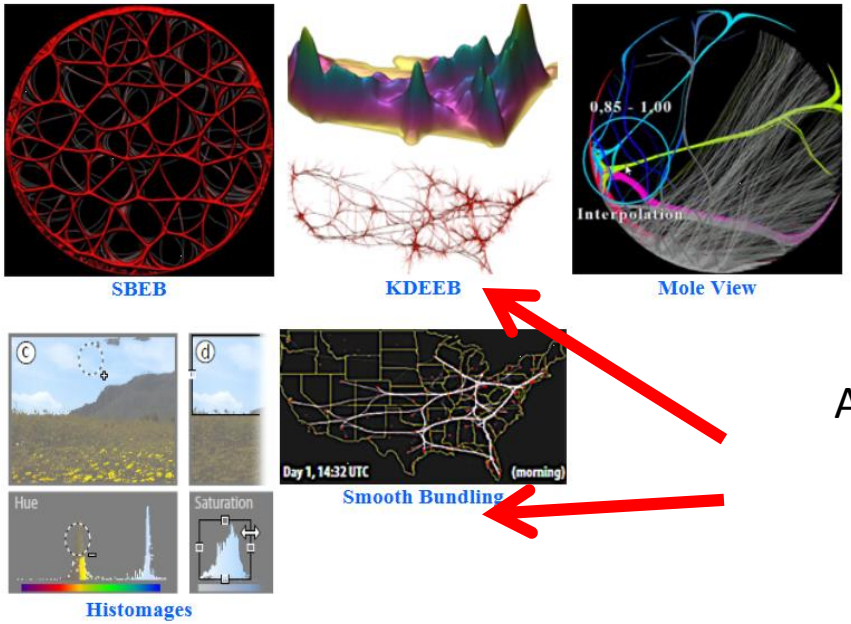






Code

http://www.recherche.enac.fr/~hurter/

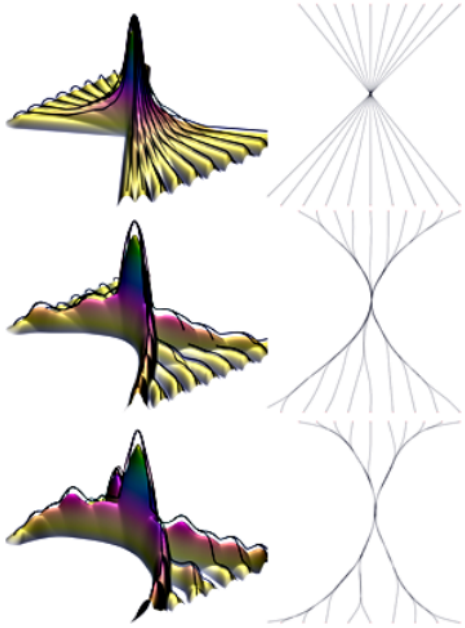


Available code

Kernel Density Estimation-based Edge Bundling

Christophe Hurter, Alexandru Telea, and Ozan Ersoy. pdf video slides
Graph Bundling by Kernel Density Estimation.
EuroVis 2012. Computer Graphics Forum journal.

new Visual Studio C# code instance (GPU version)
Visual Studio C# code instance (CPU version), thanks to Antoine LHUILLIER



Edge bundling is a recent, increasingly promising, technique which generates graph layouts of limited clutter. Bundled layouts can be used to get insight into the coarse-scale structure of networks, geographical maps, and software systems.

For general graphs, many bundling methods have been proposed in the last few years. However, the following requirements are still challenging

- bundle graphs of tens..hundreds of thousands of edges *efficiently* (near-real-time)
- *declutter* graphs with many overlapping edges and nodes
- intuitively control the *look and feel* of the bundling (e.g. produce smooth or ramified bundles)
- easy *implementation* (no complex parameter settings or algorithms)

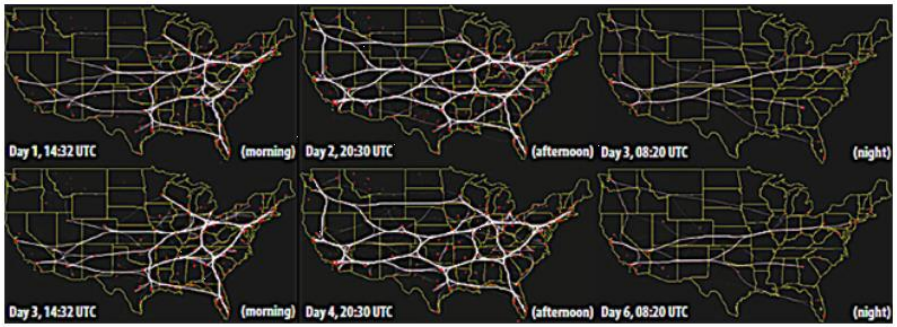
Smooth Bundling of Large Streaming and Sequence Graphs

(Realtime Edge Bundling)

Christophe Hurter, Ozan Ersoy, Alexandru Telea pdf Video

Exe and source code available (CPU version, thanks to Oussama Sabra)
CPU. C# code (Visual Studio)

Dynamic graphs are increasingly pervasive in modern information systems. However, understanding how a graph changes in time is difficult. We present here two techniques for simplified visualization of dynamic graphs using edge bundles. The first technique uses a recent image-based graph bundling method to create smooth changing
bundles from of streaming graphs. The second technique incorporates additional edge-correspondence data and is thereby suited to visualize discrete graph sequences. We illustrate our methods with examples from real-world large dynamic graph datasets.



KDEEB

```
// Resampling
Resample(splitDistance, removeDistance);

// Splatting
ComputeSplatting();

// Apply Gradient
ApplyGradient(GradientW, attractionFactor, SplattedAccMap);

// Smooth trajectories
for (int i = 0; i < smoothValue; i++)
{
    SmoothTrajectories();
}
```

Resampling

```
private void Resample(double splitDist, double removeDist)
{
    Vertex[][] tempVertex = dataSet.getCurrentTraj();
    for (int i = 0; i < tempVertex.Length; i++)
    {
        List<Vertex> tmpVertexList = new List<Vertex>();
        tmpVertexList.Add(tempVertex[i][0]);
        // Sampling on each lines (= trajectory)
        for (int j = 0; j < tempVertex[i].Length - 1; j++)
        {
            Vertex currentVert = tempVertex[i][j];
            Vertex nextVert = tempVertex[i][j + 1];
            double dist = Math.Sqrt((currentVert.getX()
                - nextVert.getX()) * (currentVert.getX() - nextVert.getX())
                + (currentVert.getY() - nextVert.getY())
                * (currentVert.getY() - nextVert.getY()));
            // Test if the next point is too far or too close
            if (dist > splitDist)
            {
                tmpVertexList.Add(new Vertex((currentVert.getX() + nextVert.getX())
                    / 2.0, (currentVert.getY() + nextVert.getY()) / 2.0,
                    (currentVert.getZ() + nextVert.getZ()) / 2.0, currentVert.getColor()));
            }
            if (!(dist < removeDist) || j == tempVertex[i].Length - 2)
            {
                tmpVertexList.Add(nextVert);
            }
        }
        tempVertex[i] = tmpVertexList.ToArray();
    }
    dataSet.CurrentTraj = tempVertex;
}
```

Splatting

```
private void ComputeSplatting()
{
    float[] AccMap = new float[AccResolution * AccResolution];
    foreach (var line in dataSet.CurrentTraj)
    {
        foreach (var point in line)
        {
            for (int i = 0; i < Kernel.Length; i += KernelSize)
            {
                for (int j = 0; j < KernelSize; j++)
                {
                    if (Convert.ToInt32(point.getX() * (double)AccResolution)
                        * AccResolution + (i / KernelSize - KernelSize / 2) * AccResolution > 0
                        && Convert.ToInt32(point.getX() * (double)AccResolution) * AccResolution
                        + (i / KernelSize - KernelSize / 2) * AccResolution < AccMap.Length
                        && Convert.ToInt32(point.getY() * (double)AccResolution) + (j - KernelSize / 2) > 0
                        && Convert.ToInt32(point.getY() * (double)AccResolution) + (j - KernelSize / 2) < AccResolution)
                    {
                        AccMap[Convert.ToInt32(point.getX() * (double)AccResolution) * AccResolution +
                            Convert.ToInt32(point.getY() * (double)AccResolution) + (i / KernelSize - KernelSize / 2)
                            * AccResolution + (j - KernelSize / 2)] += Kernel[i + j];
                    }
                }
            }
        }
    }
    SplattedAccMap = AccMap;
}
```

Apply Gradient

```
private void ApplyGradient(int gradientW, double attracFactor, float[] AccMap)
{
    // Apply the Gradient for each points
    foreach (var lines in dataSet.CurrentTraj)
    {
        for (int j = 1; j < lines.Length - 1; j++)
        {
            // Get the gradient value at the current point
            int pointIndex = Convert.ToInt32(lines[j].getX()
                * (double)AccResolution) * AccResolution
                + Convert.ToInt32(lines[j].getY() * (double)AccResolution);
            Vector2 localGradient = GetLocalGradient(gradientW, AccMap, lines[j], pointIndex);

            // Normalizing the localGradient.
            if (localGradient.X != 0 || localGradient.Y != 0)
            {
                localGradient.Normalize();
            }

            // Move each point by the gradient vector associated within
            lines[j].setX(lines[j].getX() + attracFactor * localGradient.X / (double)AccResolution);
            lines[j].setY(lines[j].getY() + attracFactor * localGradient.Y / (double)AccResolution);
        }
    }
}
```

Smooth Trajectories

```
private void SmoothTrajectories()
{
    foreach (var lines in dataSet.CurrentTraj)
    {
        for (int i = 1; i < lines.Length - 1; i++)
        {
            lines[i].setX((lines[i - 1].getX() + lines[i].getX() + lines[i + 1].getX()) / 3.0);
            lines[i].setY((lines[i - 1].getY() + lines[i].getY() + lines[i + 1].getY()) / 3.0);
        }
    }
}
```

Open Research Questions

1. Lack of **benchmarks** to assess and compare the efficiency of existing and new algorithm
2. Lack of **task-based taxonomy**. “What is the best technique to solve my problem?”
3. Lack of **quality metrics**. “What is a good result? How much better is the result from method A compared to method B?”

Edge Bundling Open Questions

Many techniques, no task based taxonomy

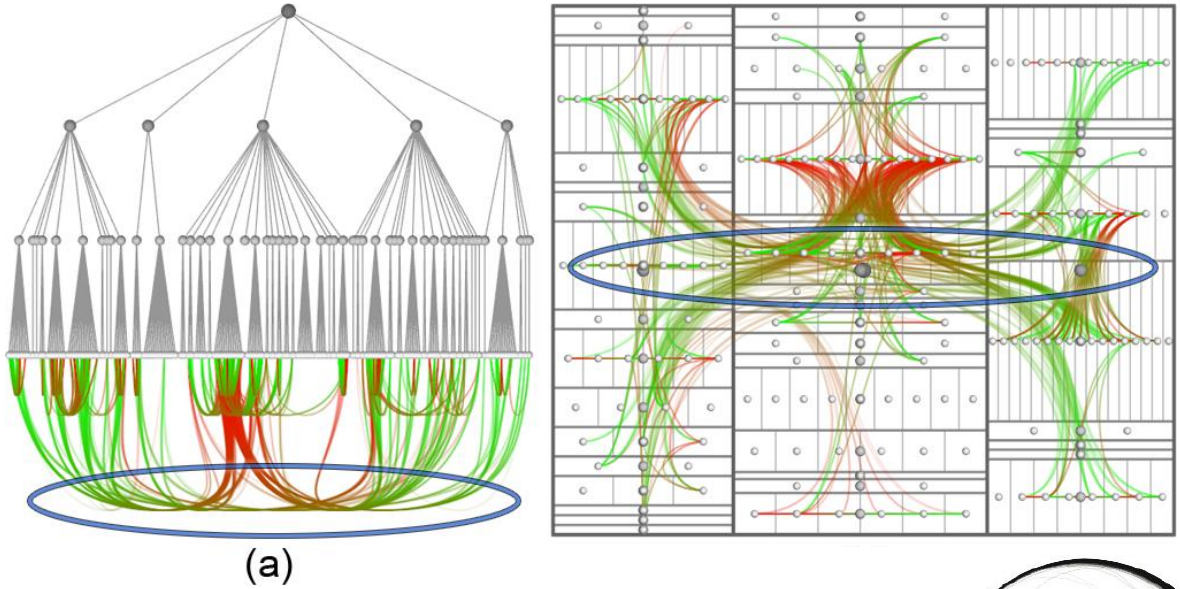
No dataset to perform computation benchmark, and task validation

Some algorithm are complex and it is hard to forecast effects of EB parameterization

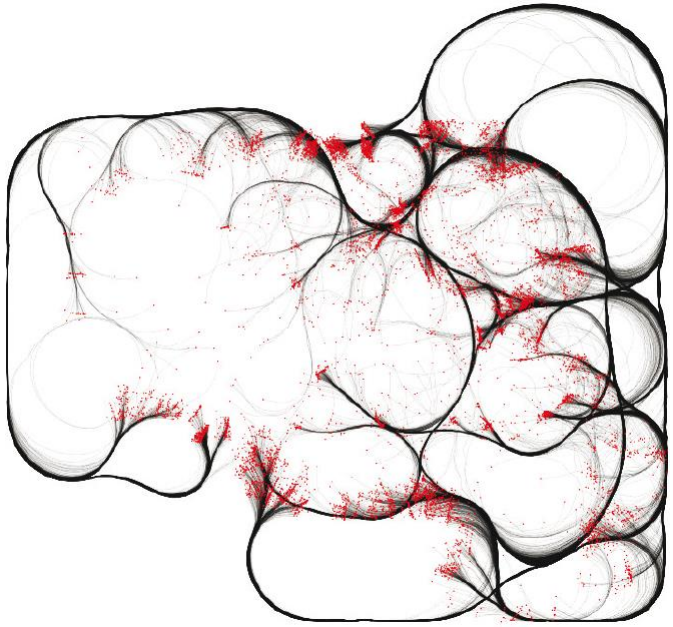
Edge Bundling produces lot of overlapping with Collinearity issues. No general solution to solve it.

Little work addressed directed edge visualisation

Collinearity problems



[Holten Infovis 2006]



[Hurter et al. Eurovis 2012]

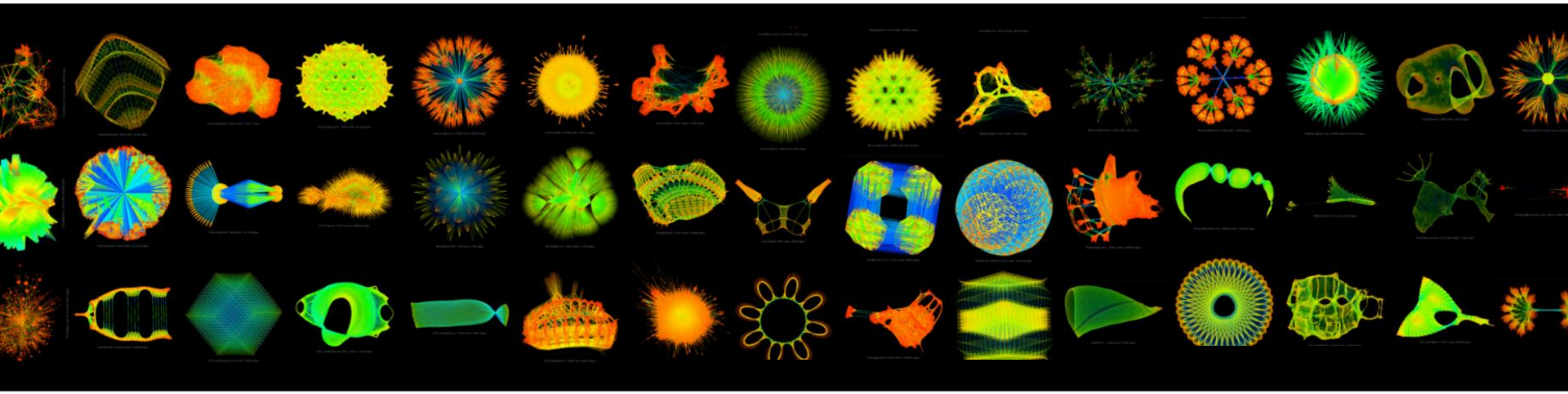
Existing solutions: Benchmarks

Benchmark Data Set:

University of Florida Sparse Matrix Collection

http://www.cise.ufl.edu/research/sparse/matrices/

[Davis+Hu 2011]



Existing solutions: Task Taxonomies

Task Taxonomy for Graph Visualization [Lee et al. 2006]

- Topology-based Tasks (adjacency, accessibility, common connection, connectivity)
- Attribute-based Tasks (on nodes, on links)
- Browsing Tasks (follow path, revisit)

Task Taxonomy for Network Evolution Analysis [Ahn et al. 2012]

- Entities (node/link vs. group vs. network)
- Properties (structural properties vs. domain attributes)
- Temporal Features (individual events vs. aggregate events)

Existing solutions: Quality Metrics

[Tufte 1983]:

- Data Density – points to display / available Pixels
- Data-Ink-Ratio – points to display / used Pixels

[Ellis+Dix 2006]:

- overplotted% – %age of Pixels with >1 point
- overcrowded% – %age of points in overplotted Pixels
- hidden% – %age of points that are hidden from view

TIME FOR QUESTIONS

Make sure to download a copy of the slide deck @ http://tinyurl.com/tutorial2013

PART II:
EDGE SET SIMPLIFICATION

EXISTING SOFTWARE

Gephi
makes graphs handy

Download Blog Store Wiki Forum Support Bug tracker

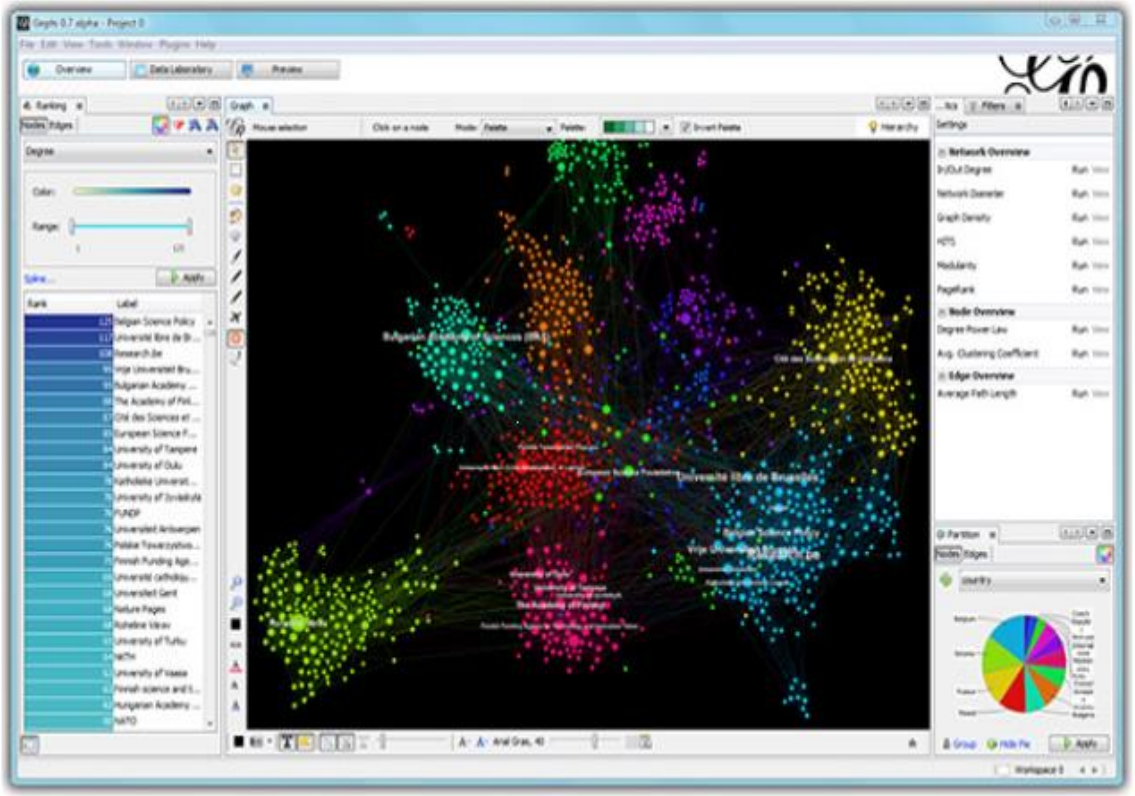
Home Features Learn Develop Plugins Services Consortium

The Open Graph Viz Platform

Gephi is an interactive visualization and exploration platform for all kinds of networks and complex systems, dynamic and hierarchical graphs.

Runs on Windows, Linux and Mac OS X. Gephi is open-source and free.

Learn More on Gephi Platform »



Download FREE
Gephi 0.8.2-beta

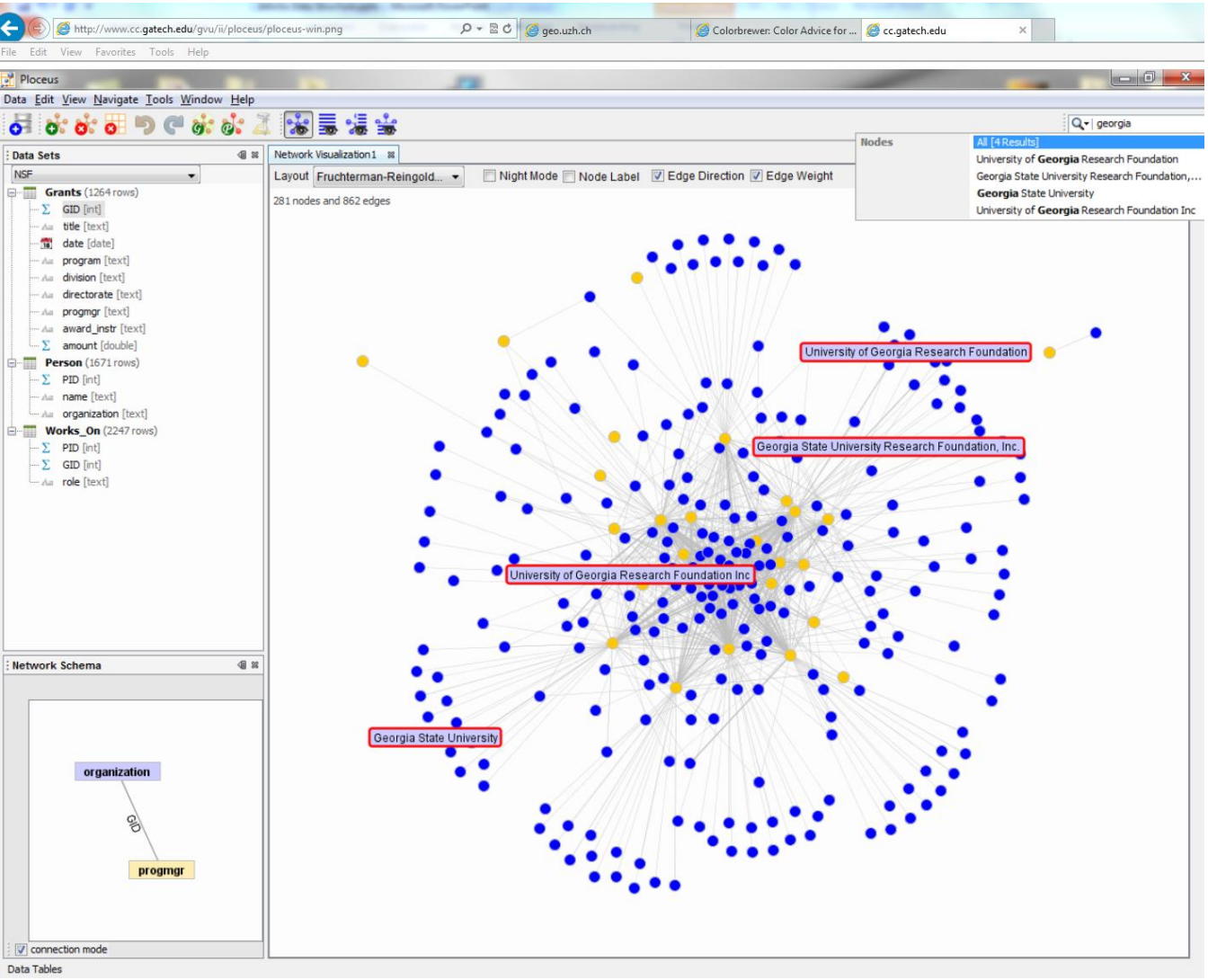
Release Notes | System Requirements

- Features
- Quick start
- Screenshots
- Videos

https://gephi.org/

Ploceus
Network-based Visual Analysis of Tabular Data

http://www.cc.gatech.edu/gvu/ii/ploceus/

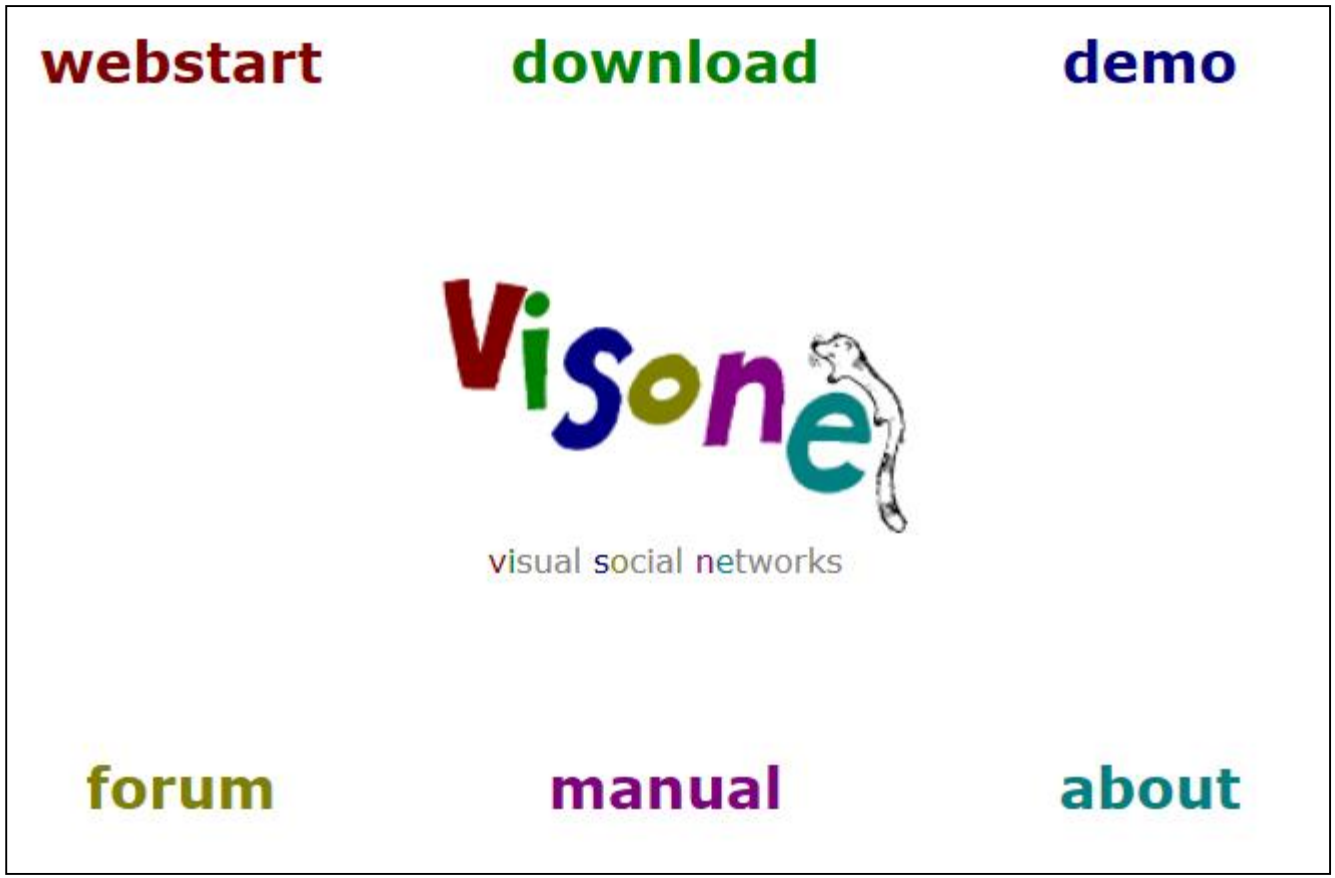


Even when a given table design conveys some static network semantics, analysts may want to look at multiple networks from different perspectives, at different levels of abstraction, and with different edge semantics.

Zhicheng Liu, John Stasko, VAST 2011

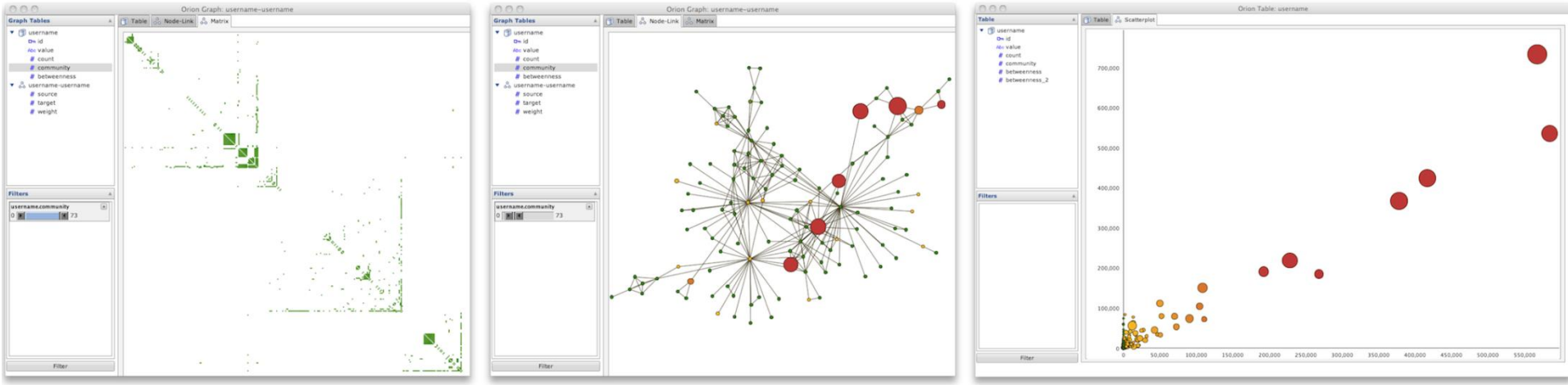
http://visone.info/

Visone



Orion: A System for Modeling, Transformation and Visualization of Multidimensional Heterogeneous Networks

http://vis.stanford.edu/papers/orion

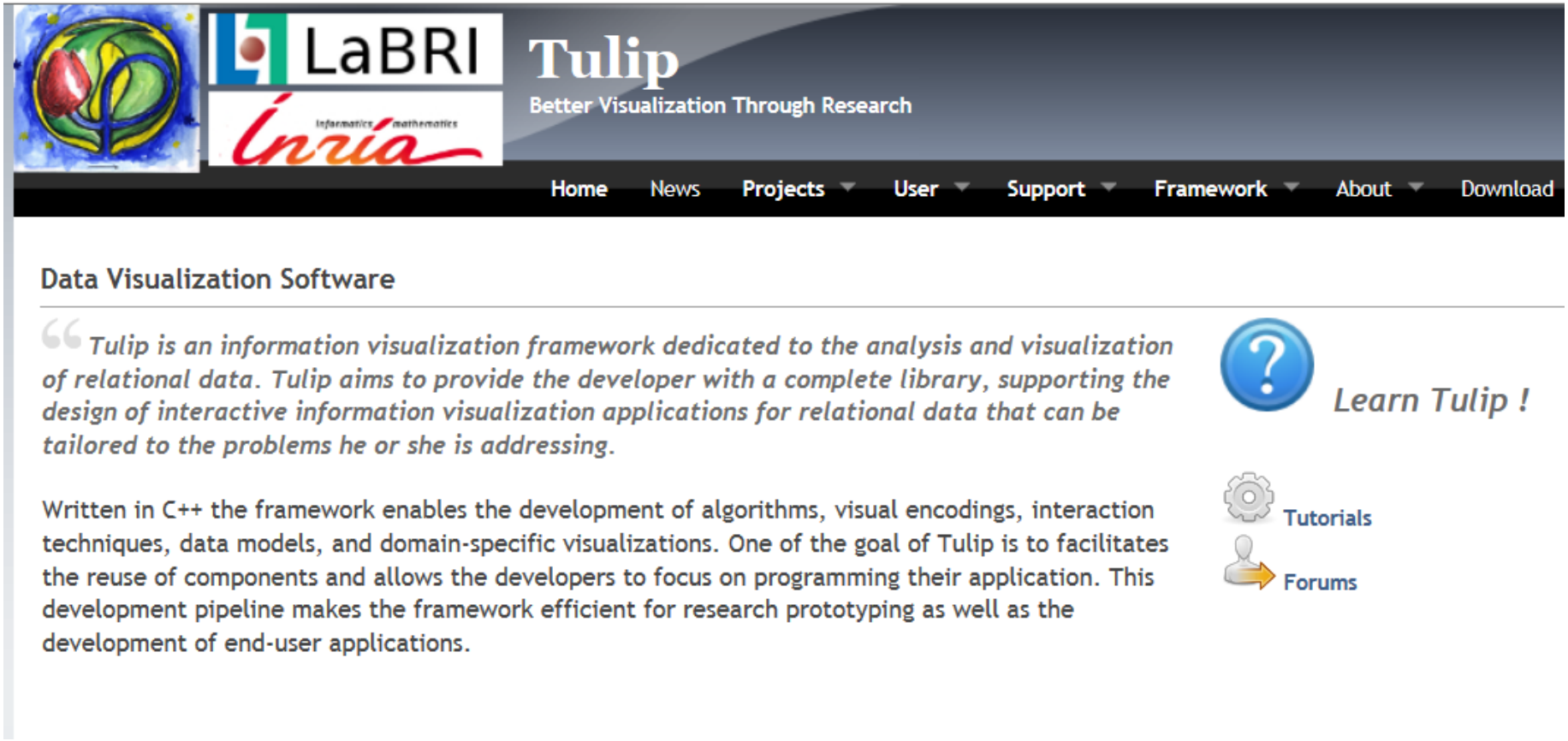


Orion's interface enables the rapid manipulation of large graphs—including the specification of complex linking relationships— using simple drag-and-drop operations with desired node types.

Jeffrey Heer, Adam Perer, (VAST), 2011

Tulip

http://tulip.labri.fr/



Auber, D. Tulip : A Huge Graph Visualisation FrameWork. *Graph Drawing Software.* Springer- Verlag, 2003, 105-126.

Commercial solution: Palantir

http://www.palantir.com/

They build software that allows organizations to make sense of massive amounts of disparate data. We solve the technical problems, so they can solve the human ones. Combating terrorism. Prosecuting crimes. Fighting fraud. Eliminating waste. From Silicon Valley to your doorstep, we deploy our data fusion platforms against the hardest problems we can find, wherever we are needed most.