This Is Time in Graph Drawing

Giuseppe Di Battista
Università degli Studi Roma Tre

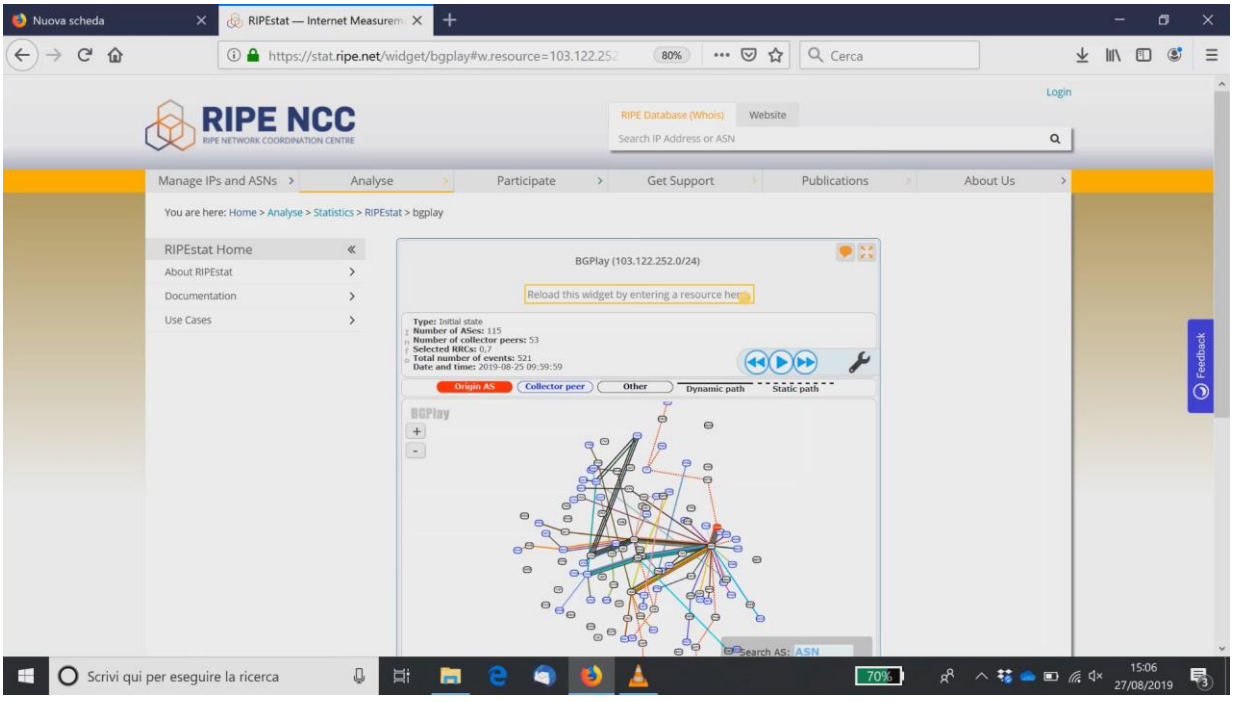
27th International Symposium on Graph Drawing and Network Visualization – GD2019
Průhonice/Prague, September 17-20, 2019

1

BGPlay

- My starting point (time): Bgplay
  - gdb Mariani Patrignani Pizzonia, Bgplay: A System for Visualizing the Interdomain Routing Evolution, GD 2003
  - part of the RIPE Stat service
    - its Web page is visited an average of more than five thousands times per day
  - also available at RouteViews, BGPStream, and Isolario

2



3

The concept of time in Graph Drawing

- One of the main challenges for Graph Drawing is the relationship between drawings and time
  - Show the temporal evolution of the visualized graphs
- About 40 papers on the subject in the GD Conferences
- Main GD fields related to *time*
  - dynamic algorithms, streaming, animation, time reconciliation, storyline, morphs, human perception

4

The point of view of the InfoVis community

- Visualization metaphor
  - Node-link or matrices
- Span of knowledge on data
  - Offline if, at any time instant, future data are known and can be used to compose the drawing
  - Online otherwise
- Representation of time
  - Animation or timeline
- Mental map preservation
- Modeling of transitions

Beck, Burch, Diehl, Weiskopf, The state of the art in visualizing dynamic graphs, Eurovis 2014

5

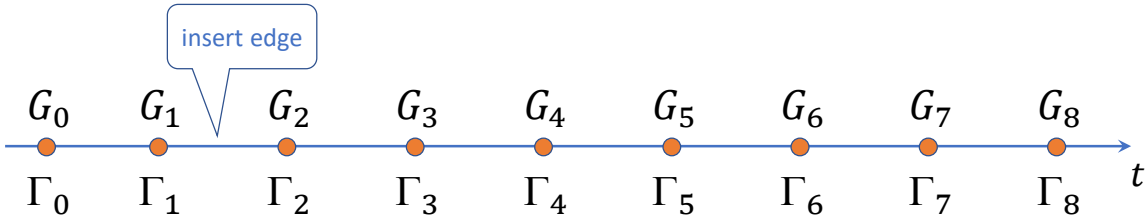
Purpose and limits of this talk

- Assess the maturity level of GD on the topic
- Mainly focusing on
  - Personal experience
  - Combinatorial algorithms
  - Straight-line drawings
  - Networking applications
- General methods
- Open problems

6

Graph stories and drawing stories

- A *graph story* is a sequence $G_0 \cdots G_n$ of graphs, where G_{i+1} is obtained from G_i by applying an *update operation*
- A *drawing story* $\Gamma_0 \cdots \Gamma_n$ of a graph story $G_0 \cdots G_n$ is a sequence of drawings such that Γ_i is a drawing of G_i



7

Type of graph stories and of drawing stories

- A graph story is a *tree story* or a *forest story* or a *planar graph story* or …. if all G_i's are trees or forests or planar graphs or ….
- A drawing story is *planar* or *straight-line* or …. if all its drawings are planar or straight-line or ….

8

Dynamic Algorithms

	Animation	Timeline
Offline		
Online	Dynamic Algorithms	

The drawing story is *not* known in advance

Rely on an *implicit* representation of the drawing

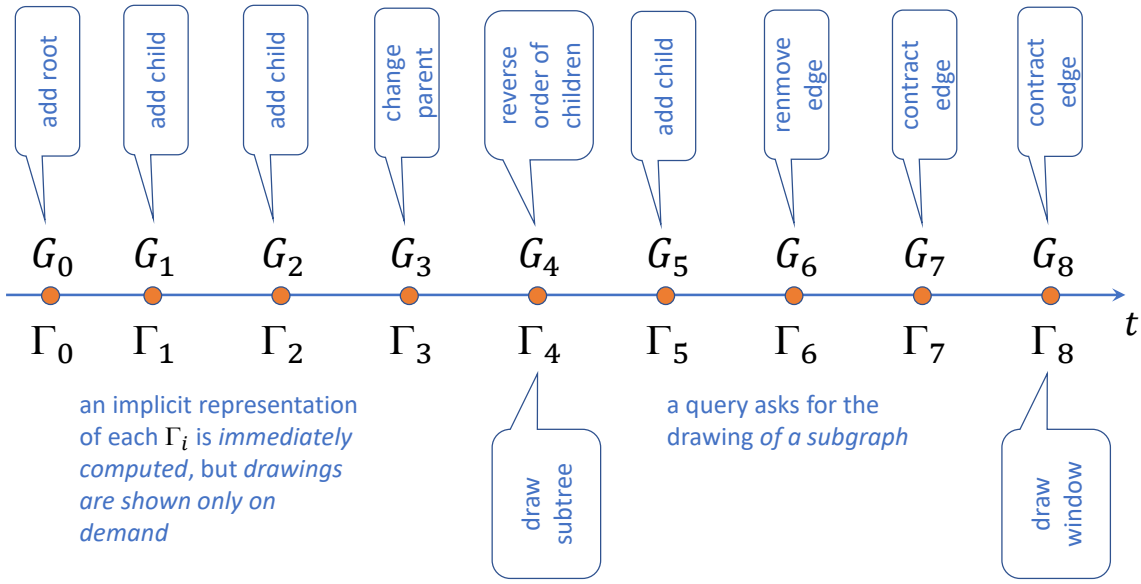
9

A Dynamic GD Problem

- Type of the graph story and type of the drawing story
- Repertory of update operations
  - E.g. insertion of vertices and edges
- Queries on the current drawing
  - E.g. draw subgraph, draw window
- Dynamic drawing predicate
  - Similarity properties of consecutive drawings

10

Dynamic GD – a drawing story of a forest story



11

Selected contributions

- Cohen gdb Tamassia Tollis Bertolazzi, A framework for dynamic graph drawing, SoCG 1992
- Cohen gdb Tamassia Tollis, Dynamic graph drawings: Trees, series-parallel digraphs, and planar st-digraphs, SICOMP 1995
- Papakostas Tollis, Incremental orthogonal graph drawing in three dimensions, GD 1997
- Bachl, Semi-dynamic orthogonal drawings of planar graphs, GD 2002

- Plus several papers that focus on the topology of the drawing, e.g. dynamic planarity testing

12

Some results

Graph story	Drawing story	Memory	Update time	Draw subgraph time	Draw window time	Overall area
Forest of rooted trees	Upward planar straight-line grid	$O(n)$	$O(\log n)$	$O(k + \log n)$ Subtree	$O(k \log n)$	$O(n^2)$
Series-parallel digraphs	Upward planar straight-line grid	$O(n)$	$O(\log n)$	$O(k + \log n)$ Series-parallel subgraph	$O(k \log^2 n)$	$O(n^2)$
Planar st-digraphs	Upward planar *polyline* grid	$O(n)$	$O(\log n)$	$O(k \log n)$	-	$O(n^2)$

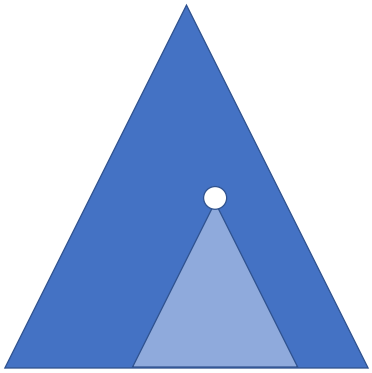
n is the total number of vertices and k is the number of vertices to be drawn

targets: a time that is: (1) sublinear in n for updates and (2) linear in the number of objects to be drawn and sublinear in n for drawing queries

13

Dynamic GD – methods

- Maintain a tree decomposition of the graph equipped with variables representing the features of the drawing
- The decomposition is implemented using dynamic trees



The position of a node in the drawing depends on several features of the subtrees rooted at that node

14

Dynamic GD – problems

- Currently, the tree drawings that are maintained use the x-coordinates of an in-order traversal; is it possible to maintain Rengold-Tilford-type drawings?
- Maintain a straight-line grid drawing of a planar graph (e.g. a Schnyder-wood)

15

Streaming

	Animation	Timeline
Offline		
Online	Streaming	

The drawing story is *not* known in advance

When an *object* (vertex/edge) arrives, it is explicitely drawn, its placement cannot (almost) be altered, objects have a given *persistence*

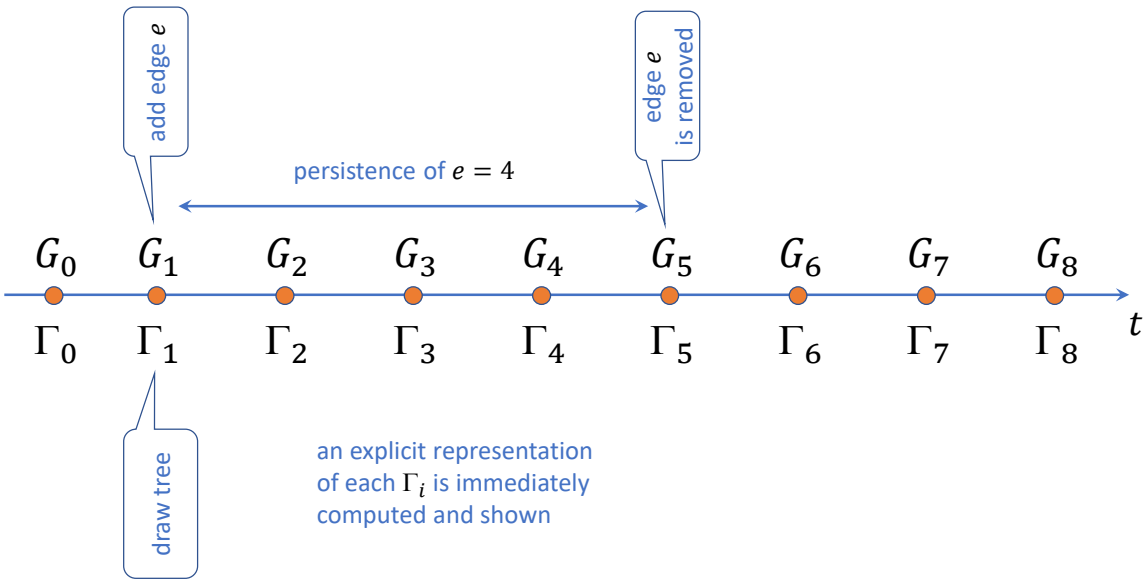
16

A Streaming GD Problem

- A source produces a sequence of n objects (vertices/edges) that are immediately and explicitely drawn
- The *persistence* of an object specifies how much time it remains in the drawing; tipically all objects have the same persistence (also called *window* size) $k \ll n$
- Once an object is drawn, its placement *cannot be altered* until it is removed (k instants after it was produced)
- All drawings should fit an area that is a function of k

17

Streaming GD – a drawing story of a tree story



18

Selected contributions

- Binucci Brandes gdb Didimo Gaertler Palladino Patrignani Symvonis Zweig, Drawing trees in a streaming model, GD 2009
- Nguyen Eades Hong, Streameb: Stream edge bundling, GD 2012
- Goodrich Pszona, Streamed graph drawing and the file maintenance problem, GD 2013
- Crnovrsanin Chu Ma, An Incremental Layout Method for Visualizing Online Dynamic Graphs, GD2015

19

Some Results

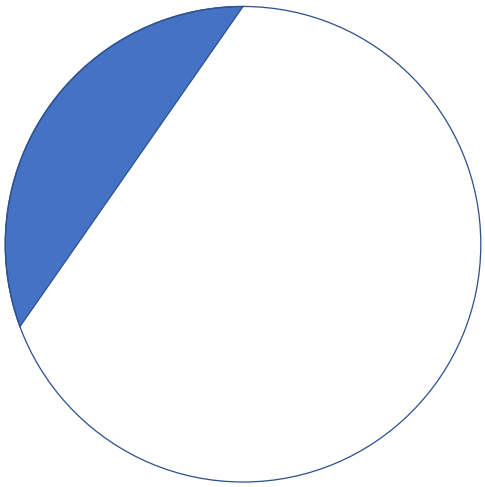
Graph story	Drawing story	Objects	Memory	Update time	Number of points in convex position
Degree d Trees (Eulerian Tour only)	planar straight-line, circular layout	edges	$O(k)$	$O(k)$	$\left\lceil \frac{k}{2} \right\rceil (d-1)+k+1$
Trees (Eulerian Tour only)	planar straight-line, circular layout	edges	$O(k)$	$O(k)$	$2k-1$

k is the persistence (size of the window)

target: *reuse* the area

20

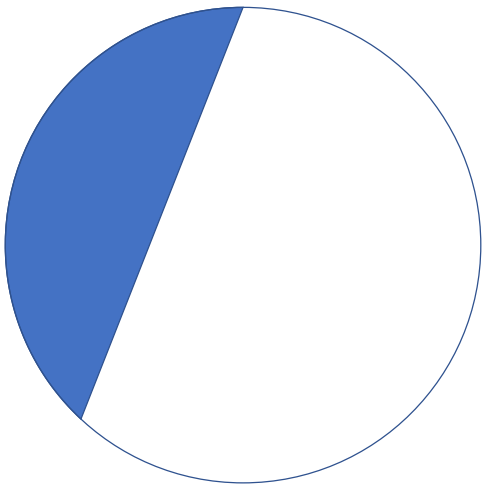
Methods – the art of reusing the space



Evolution of the drawing during the Eulerian tour of the tree

21

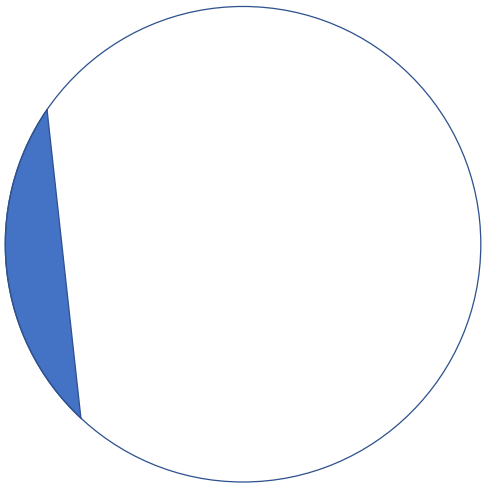
Methods – the art of reusing the space



Evolution of the drawing during the Eulerian tour of the tree

22

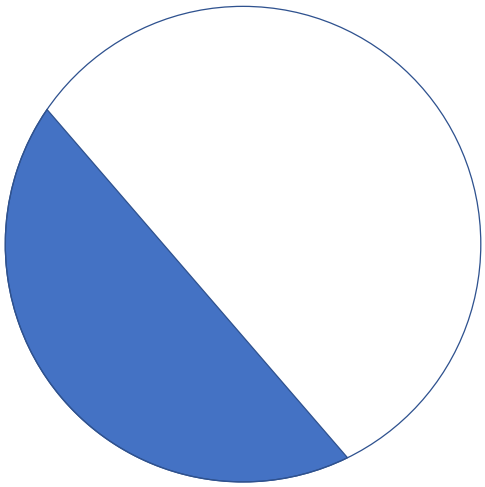
Methods – the art of reusing the space



Evolution of the drawing during the Eulerian tour of the tree

23

Methods – the art of reusing the space



Evolution of the drawing during the Eulerian tour of the tree

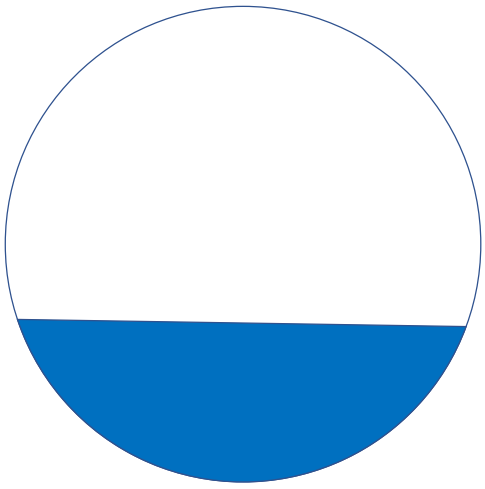
24

Methods – the art of reusing the space

Evolution of the drawing during the Eulerian tour of the tree

25

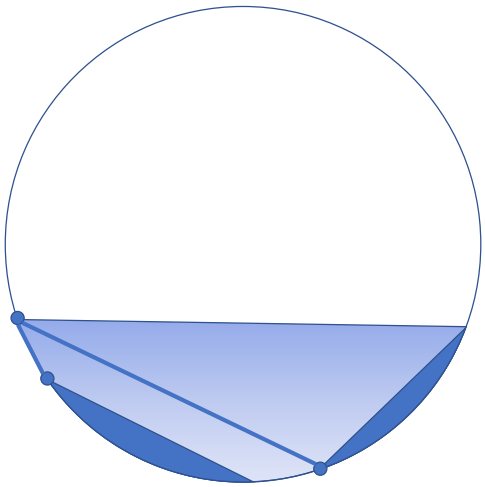
Methods – the art of reusing the space



Evolution of the drawing during the Eulerian tour of the tree

26

Methods – the art of reusing the space

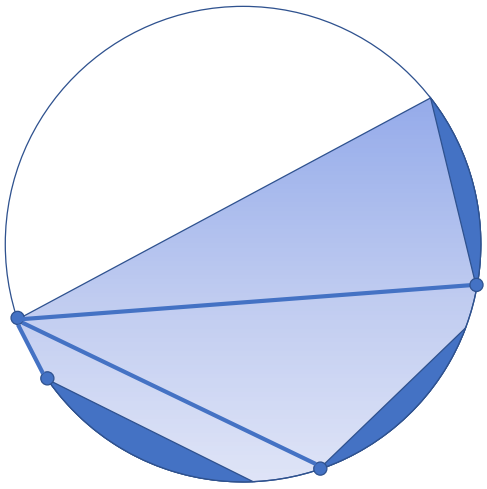


Evolution of the drawing during the Eulerian tour of the tree

Eulerian tour = DFS + explicit backtrack edges

27

Methods – the art of reusing the space

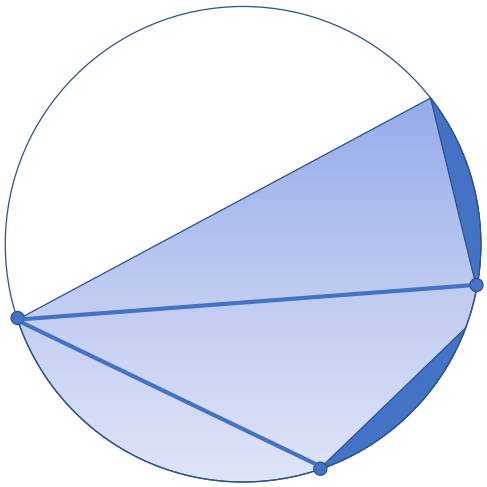


Evolution of the drawing during the Eulerian tour of the tree

Eulerian tour = DFS + explicit backtrack edges

28

Methods – the art of reusing the space

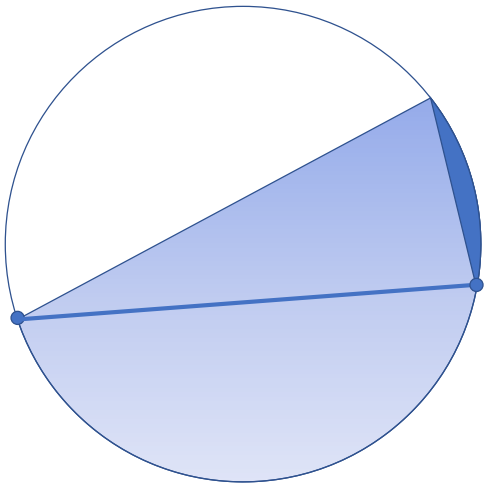


Evolution of the drawing during the Eulerian tour of the tree

Eulerian tour = DFS + explicit backtrack edges

29

Methods – the art of reusing the space

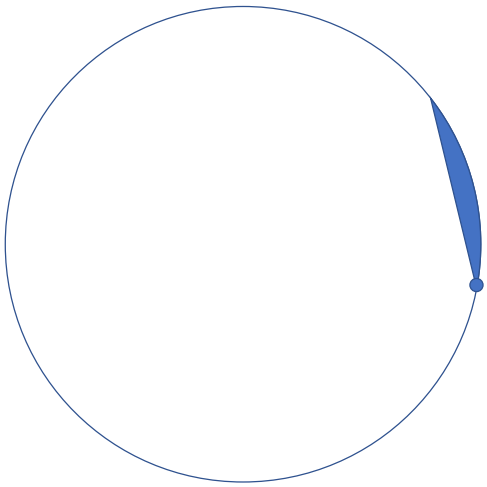


Evolution of the drawing during the Eulerian tour of the tree

Eulerian tour = DFS + explicit backtrack edges

30

Methods – the art of reusing the space



Evolution of the drawing during the Eulerian tour of the tree

Eulerian tour = DFS + explicit backtrack edges

31

If we use dynamic graph drawing for streaming

Graph story	Drawing story	Objects	Memory	Update time	Number of points in convex position (streaming)	Area
Trees (Eulerian Tour)	planar straight-line, circular layout	edges	$O(k)$ $O(k)$	$O(k)$ $O(k)$	$2k-1$	$O(k^3)$ $O(k^2)$

k is the persistence (size of the window)
Dynamic graph drawing bound

So what? In dynamic graph drawing the objects *move*, even of a linear amount of space for just one update

32

Streaming – problems

- Non-tree stories
- Non-Eulerian-tour visit
- Explore the area-movement tradeoff, following the developments in Goodrich Pszona GD2013

33

Offline - animation

	Animation	Timeline
Offline		
Online		

All the updates are known in advance

The graphs of the resulting sequence are drawn one-after-the-other so that the mental map is preserved

34

Selected contributions

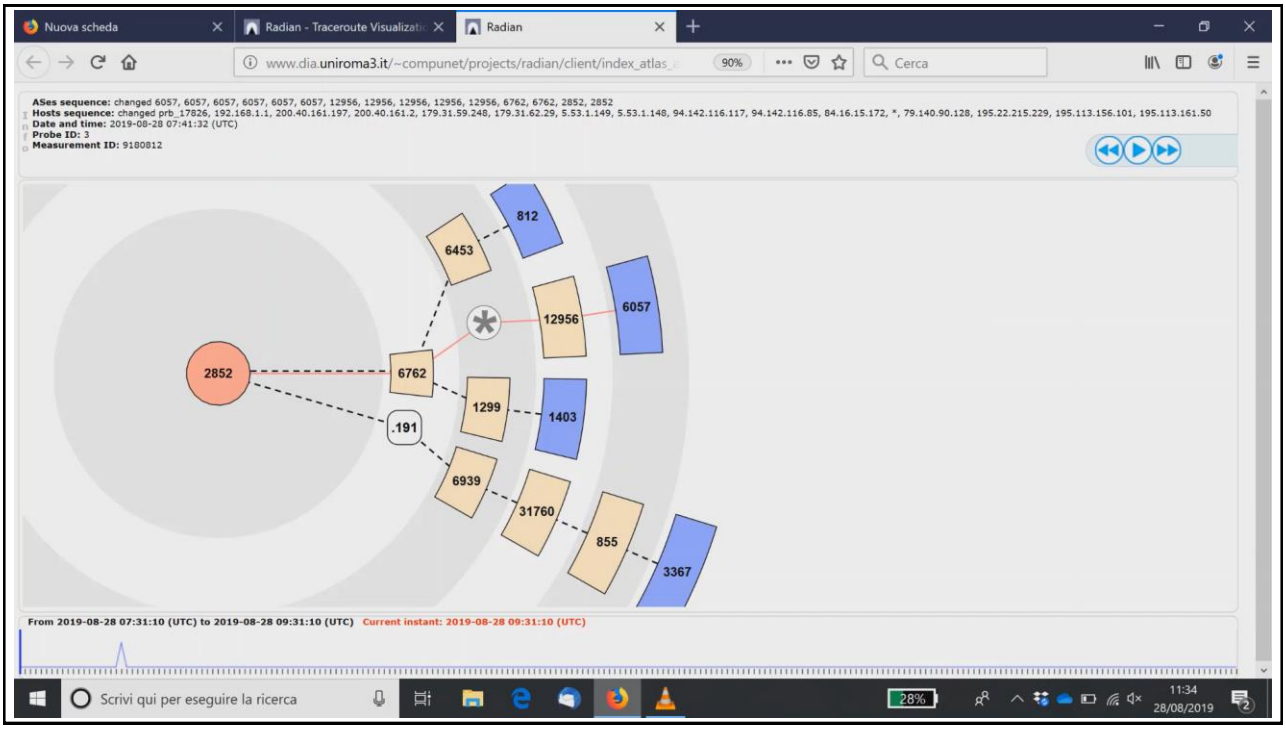
- Brandes Wagner, Dynamic grid embedding with few bends and changes, ISAAC 1998
- Diehl Görg, Graphs, they are changing, GD 2002
- Erten Harding Kobourov Wampler Yee, Graphael: Graph animations with evolving layouts, GD 2003
- Brandes Fleischer Puppe, Dynamic spectral layout with an application to small worlds, JGAA 2007
- Da Lozzo Rutter, Planarity of streamed graphs, CIAC 2015
- Skambath Tantau, Offline drawing of dynamic trees: Algorithmics and document integration, GD 2016
- *Borrazzo Da Lozzo Frati Patrignani, Graph Stories in Small Area, GD 2019*

35

Radian

- Candela Di Bartolomeo gdb Squarcella, Radian: Visual Exploration of Traceroutes, TVCG 2018
- Application of clustered planarity techniques
- available at http://www.dia.uniroma3.it/~compunet/projects/radian/

36



37

Radian Methods

- Augment clusters
  - Add vertices and edges into clusters so that every path traversing a cluster c has the same number of vertices in c
- Assign layers
  - Compute a layering for all vertices
- PQ-tree layout
  - Use a PQ-tree to order vertices along the layers so that (1) vertices on a layer and belonging to the same cluster are consecutive and (2) the number of edge crossings is low
  - The PQ-tree is initialized with a spanning tree of the graph and is incrementally updated with the remaining edges, which induce ordering constraints

38

Offline animation – problems

- In some problems it is difficult or impossible to maintain certain invariants of the drawing
  - *Temporal cuts* can be introduced (see Skambath Tantau 2016) when some invariants are violated and the drawing is "reset"
  - Look for drawing stories with few temporal cuts

39

Storyline & C.

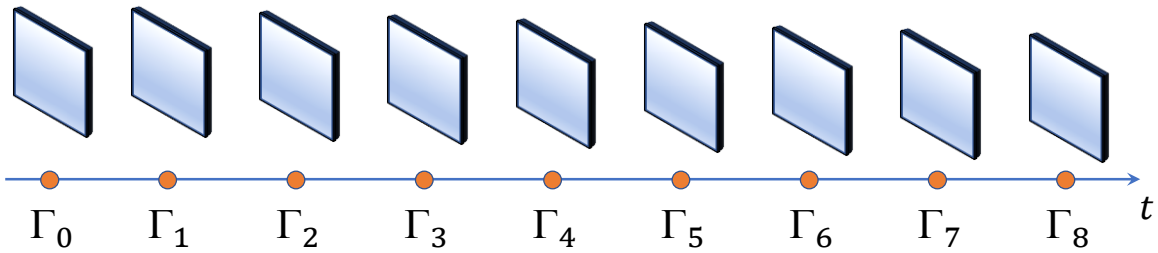
	Animation	Timeline
Offline		Storyline & C.
Online		

One Cartesian coordinate represents time

40

Timeline representation – an example

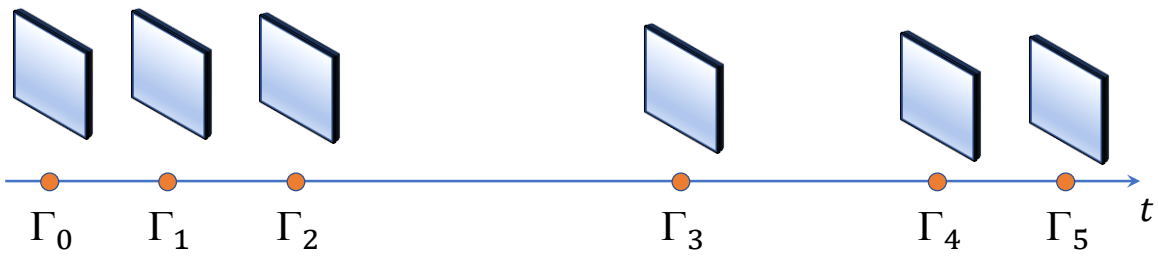
- All the Γ_i are *together* in the drawing



41

Timeline representation – deformation of time

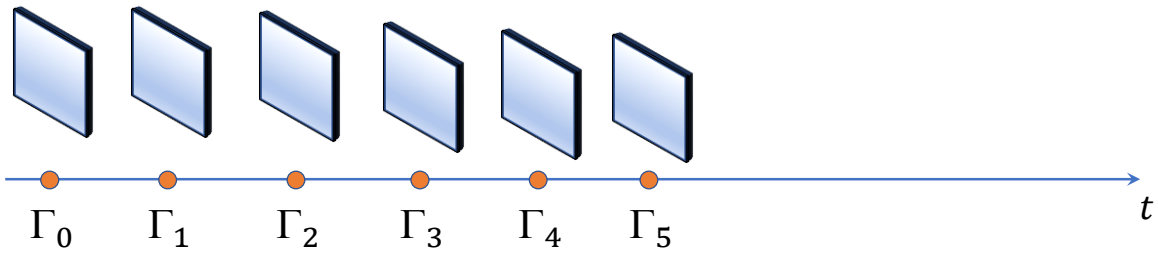
Wang Archambault Haleem Moeller Wu Qu, Nonuniform Timeslicing of Dynamic Graphs Based on Visual Complexity, VIS 2019



42

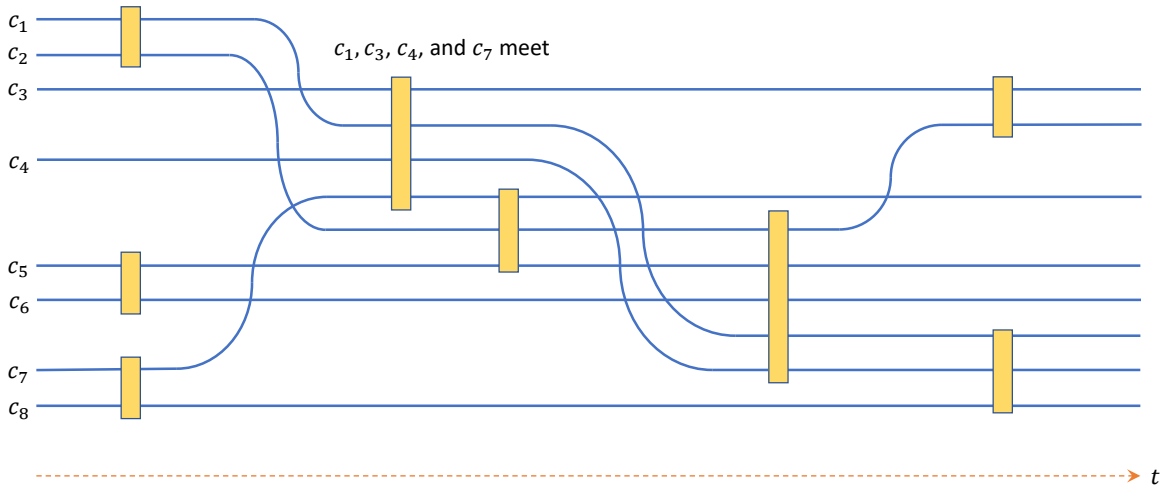
Timeline representation – deformation of time

Wang Archambault Haleem Moeller Wu Qu, Nonuniform Timeslicing of Dynamic Graphs Based on Visual Complexity, VIS 2019

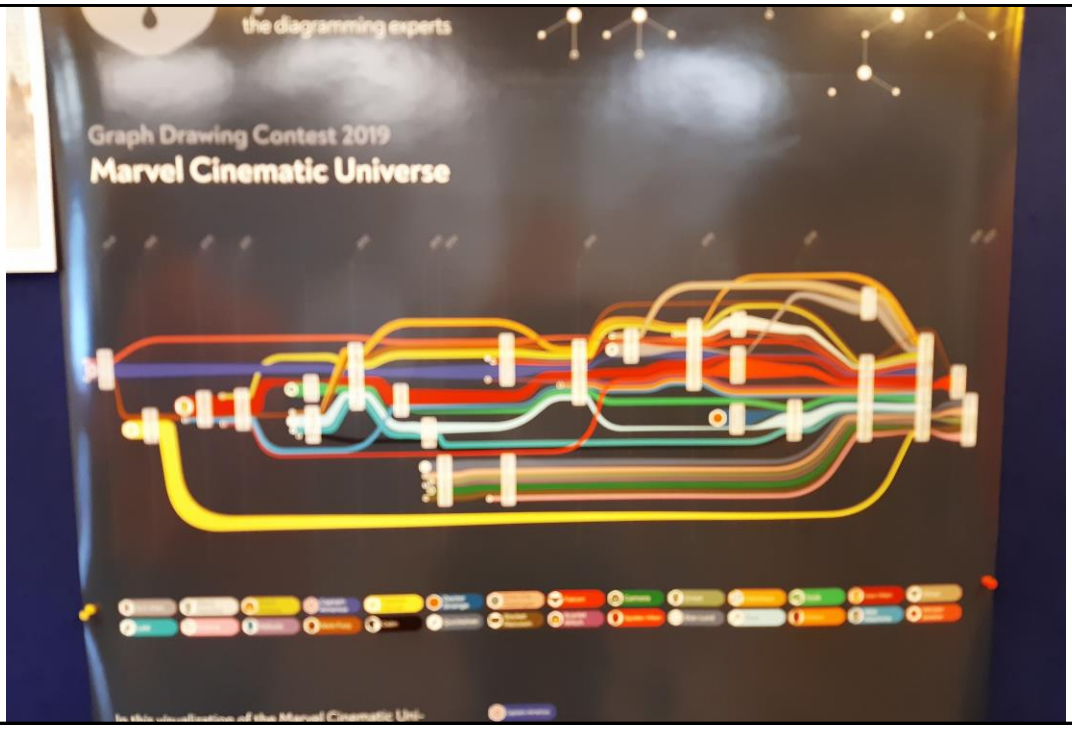


43

Storyline - abstraction of the structure of a narrative



44



45

Goals and Methods

- Minimize crossings between characters
- Minimize crossings between bundles of characters (block crossings)

- Exact algorithms
- Integer Linear Programming
- SAT solvers

46

Selected contributions

- Tanahashi Ma, Design Considerations for Optimizing Storyline Visualizations, TVCG 2012
- Kostitsyna Nöllenburg Polishchuk Schulz Strash, On minimizing crossings in storyline visualizations, GD 2015
- van Dijk Fink Fischer Lipp Markfelder Ravsky Suri Wolff, Block crossings in storyline visualizations, GD 2016
- Gronemann Jünger Liers Mambelli, Crossing minimization in storyline visualization, GD 2016
- van Dijk Lipp Markfelder Wolff, Computing storyline visualizations with few block crossings, GD 2017

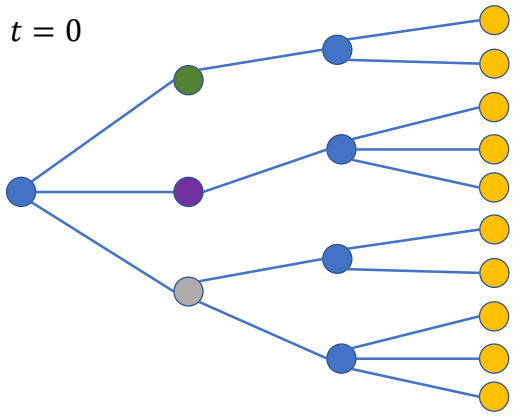
47

Upstream visibility

- Candela gdb Marzialetti, Upstream Visibility: A Multi-View Routing Visualization, VINCI 2018

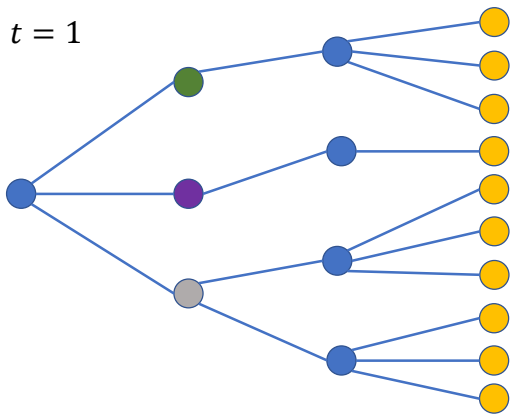
48

A streamgraph paradigm for routing visualization



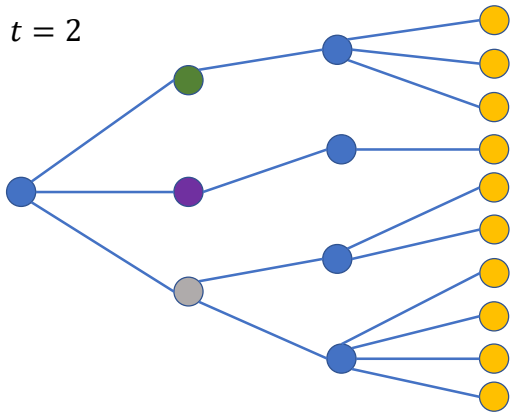
49

A streamgraph paradigm for routing visualization



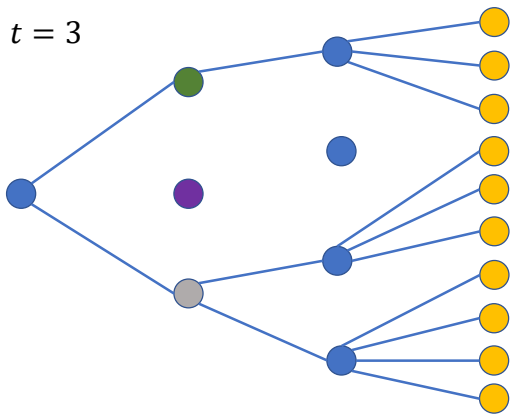
50

A streamgraph paradigm for routing visualization



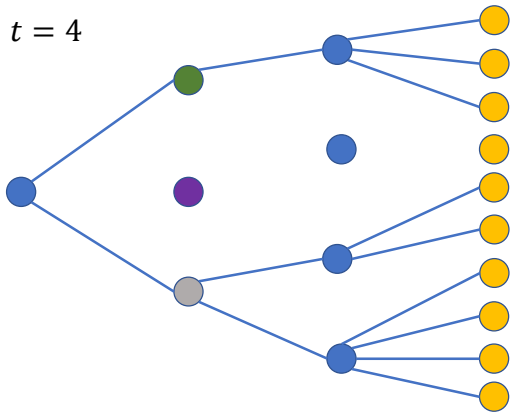
51

A streamgraph paradigm for routing visualization



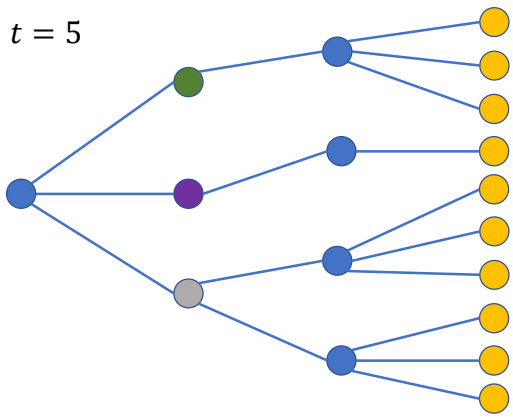
52

A streamgraph paradigm for routing visualization



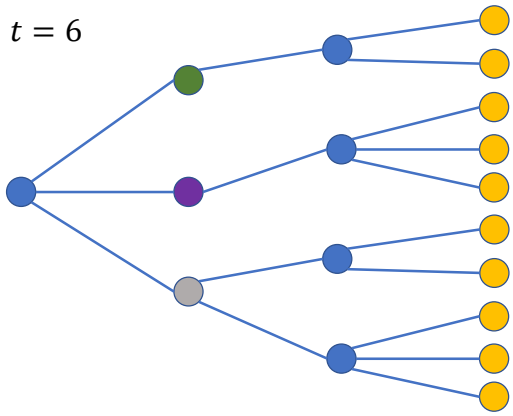
53

A streamgraph paradigm for routing visualization



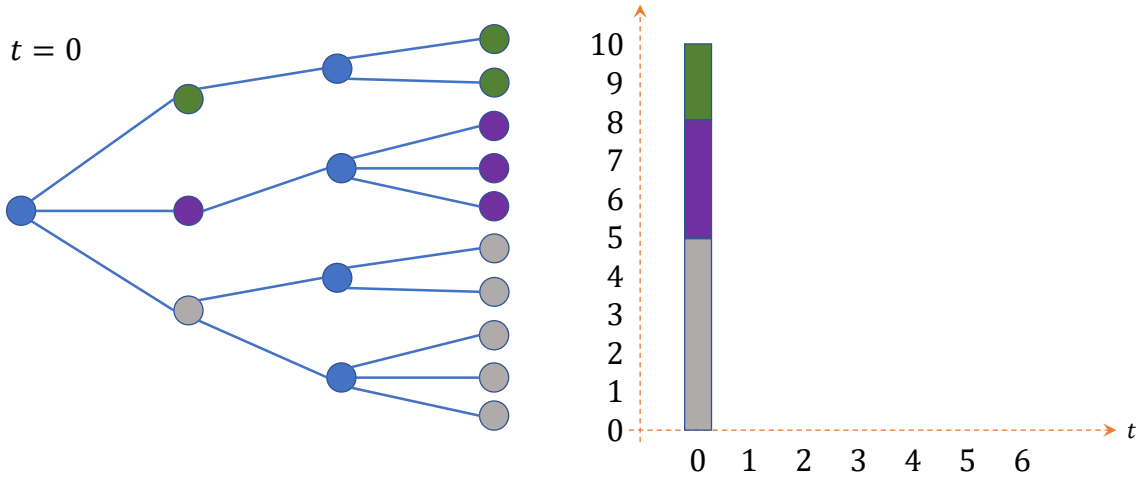
54

A streamgraph paradigm for routing visualization



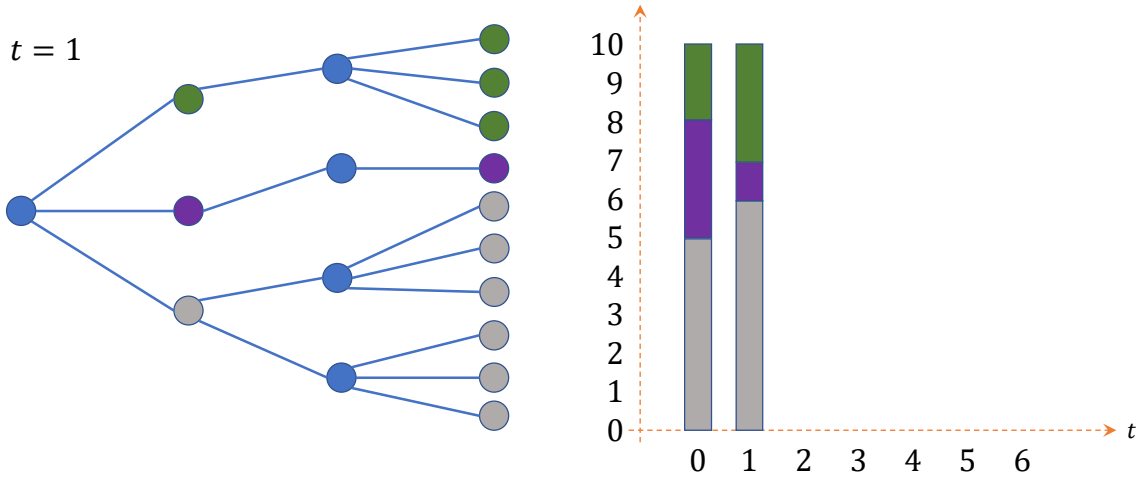
55

A streamgraph paradigm for routing visualization



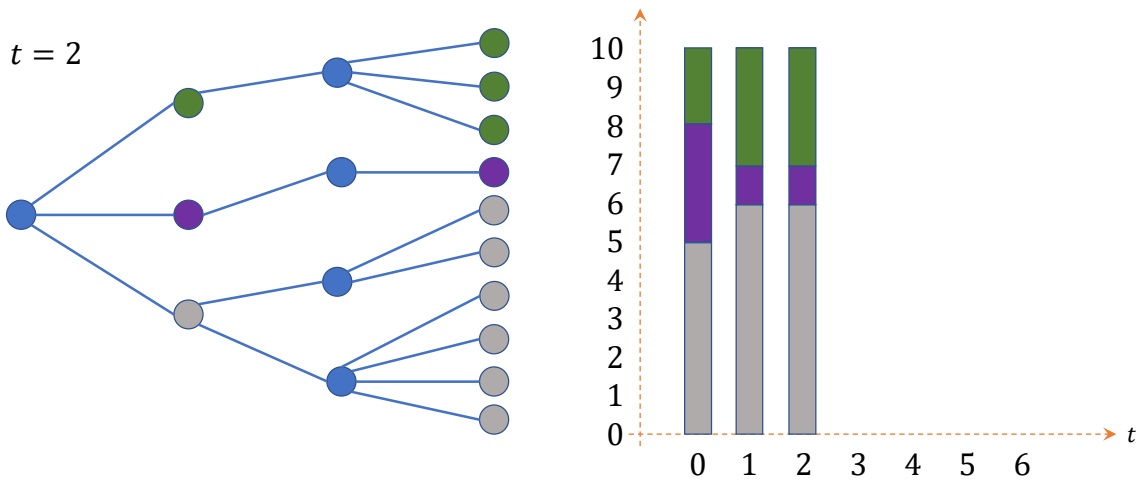
56

A streamgraph paradigm for routing visualization



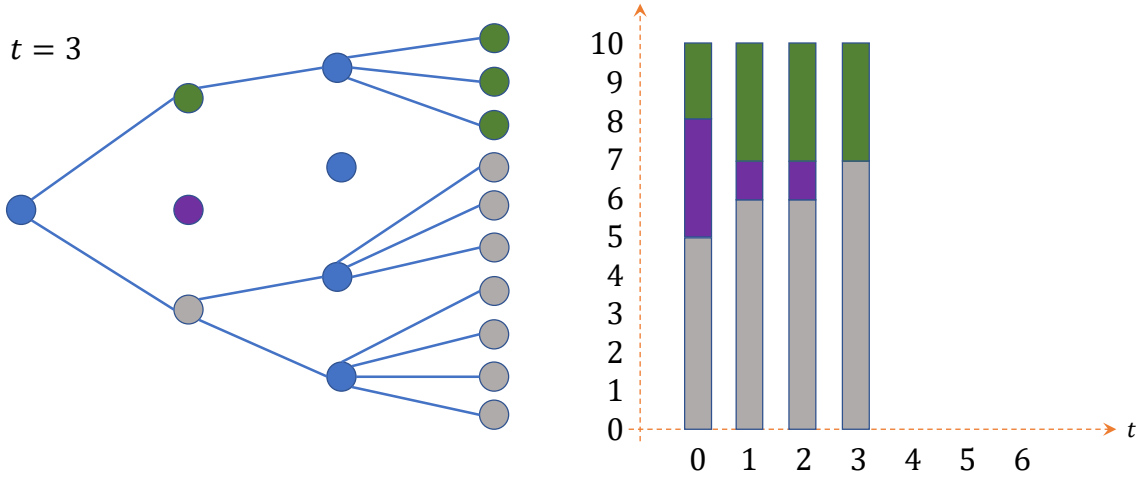
57

A streamgraph paradigm for routing visualization



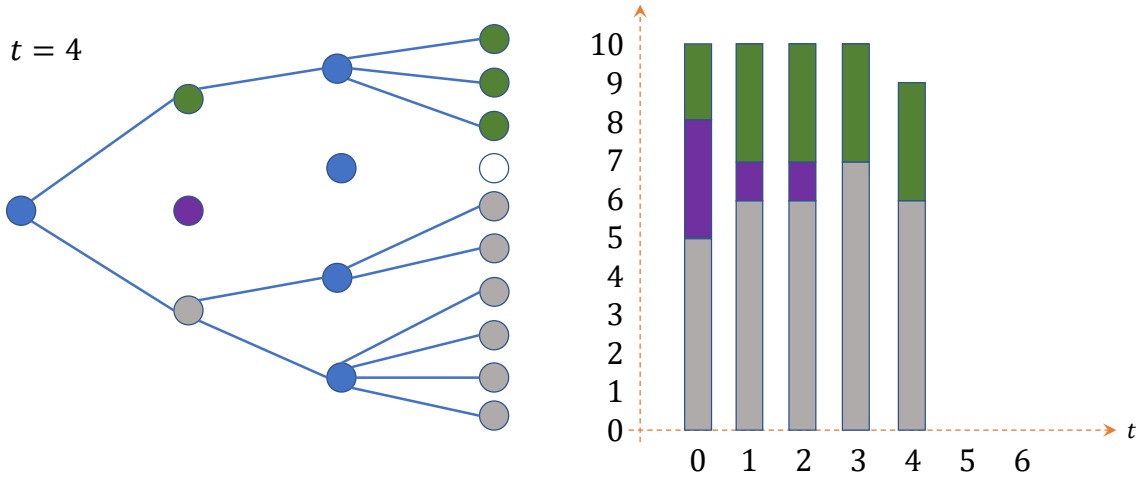
58

A streamgraph paradigm for routing visualization



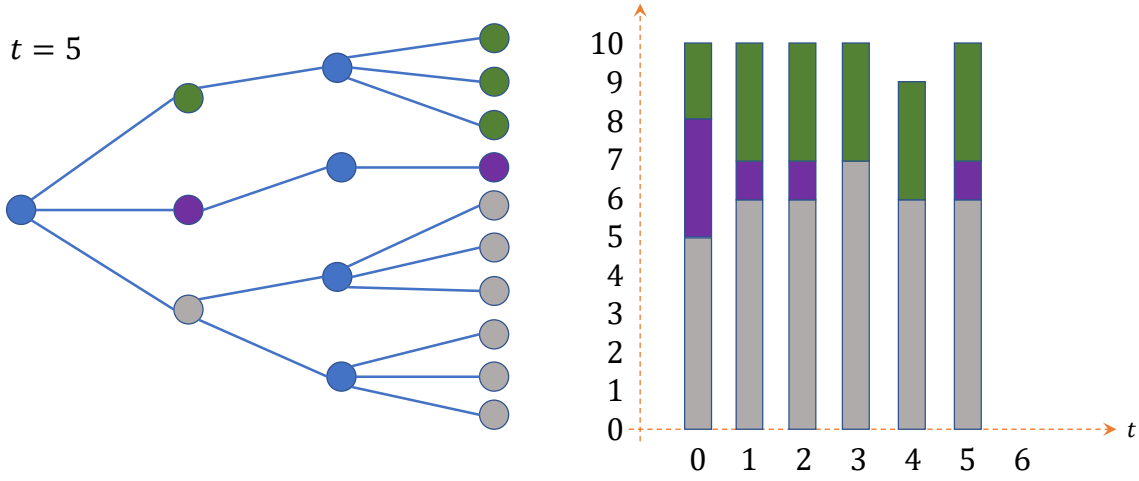
59

A streamgraph paradigm for routing visualization



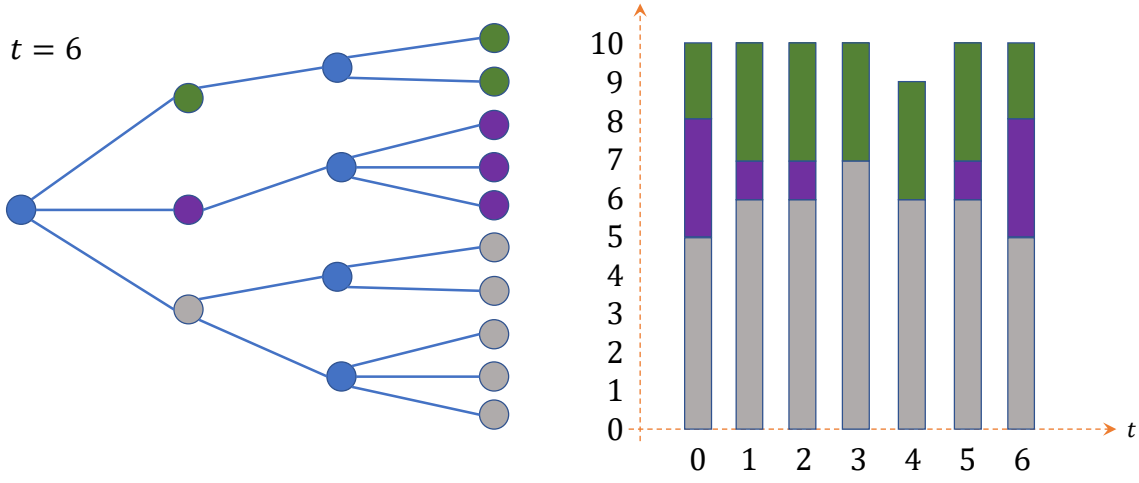
60

A streamgraph paradigm for routing visualization



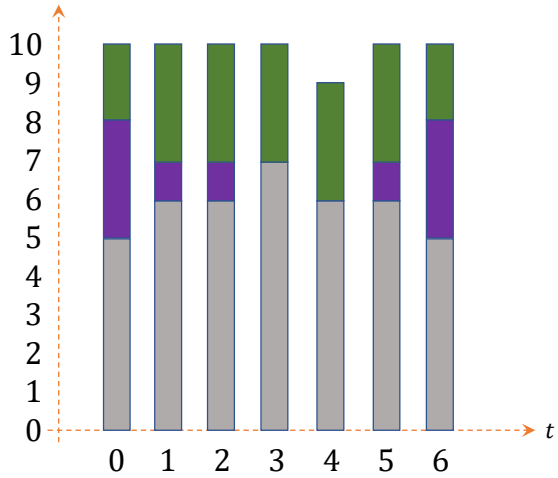
61

A streamgraph paradigm for routing visualization



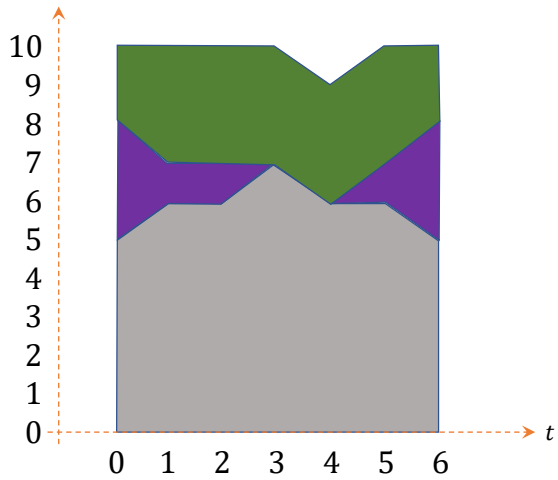
62

A streamgraph paradigm for routing visualization



63

A routing streamgraph

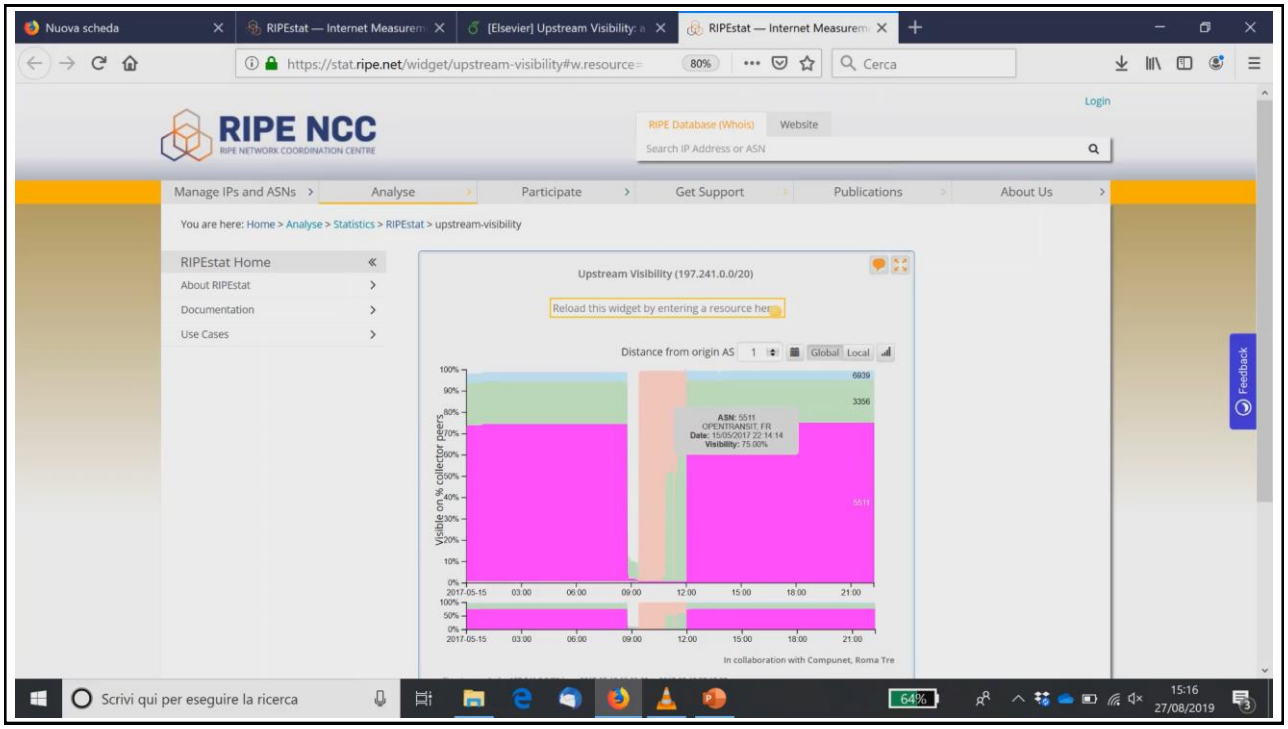


64

Upstream Visibility **clip**

- service available at RIPE Stat

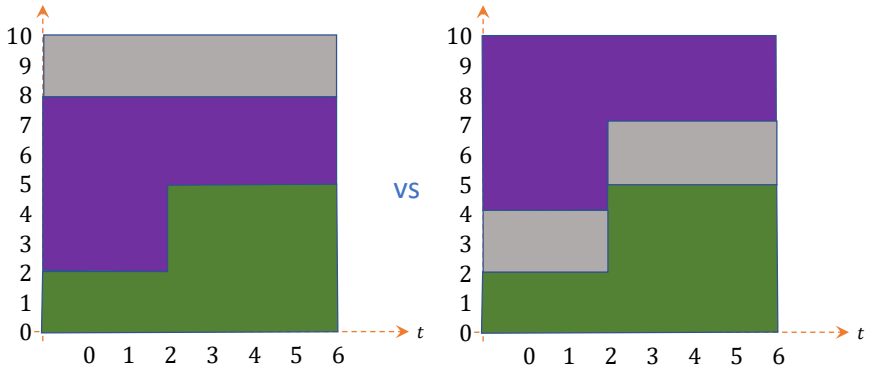
65



66

Streamgraph – problems

- Find an ordering of the curves so that each colored region is «smooth»



See also: Di Bartolomeo Hu, There is more to streamgraphs than movies: Better aesthetics via ordering and lassoing, Computer Graphics Forum 2016

67

Timeline representations – problems

- How to visualize a graph story according two or more clocks?
  - Physical quantity clock
  - Event clock
  - ….
- Lotker, The tale of two clocks, IEEE/ACM ASONAM 2016
  - using the drift between clocks is useful to understand the dynamics in social networks

68

Morphs

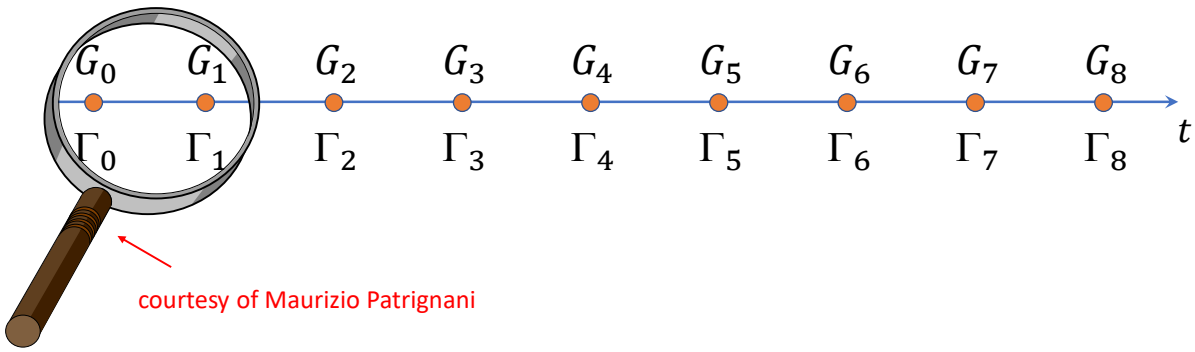
	Animation	Timeline
Offline	Morphs	
Online		

How to transform Γ_i into Γ_{i+1}

69

A closer look at one of the steps

- Focus on Γ_0 and Γ_1
- In general they are different
- How to transform (morph) one into the other?



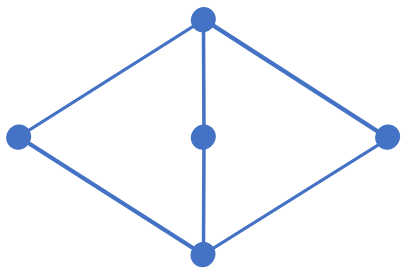
70

The problem of morphing graph drawings

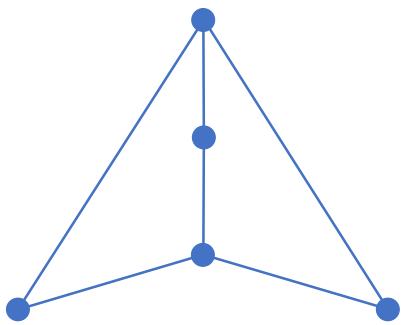
- input: two drawings Γ_0 and Γ_1 of the same graph G
- A *morph* between Γ_0 and Γ_1 is a continuously changing family of drawings of G indexed by time $t \in [0,1]$, such that the drawing at time $t = 0$ is Γ_0 and the drawing at time $t = 1$ is Γ_1

71

A morph



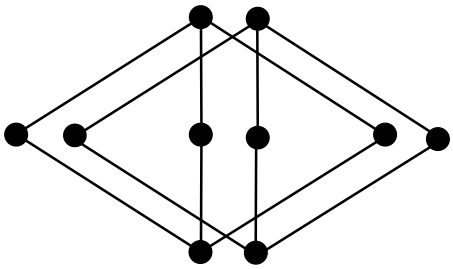
Γ_0



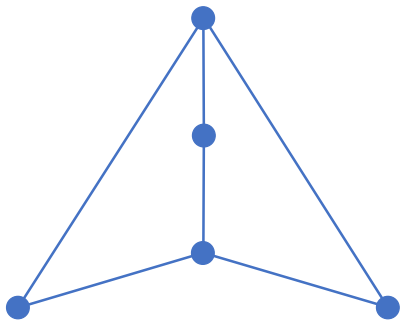
Γ_1

72

A morph



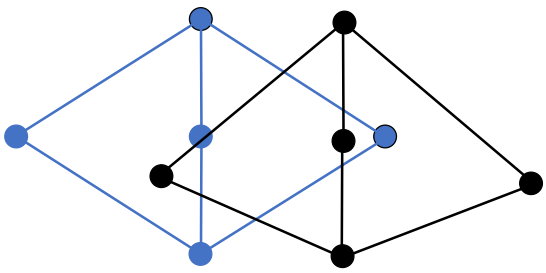
Γ_0



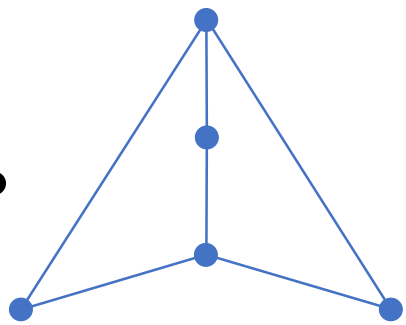
Γ_1

73

A morph



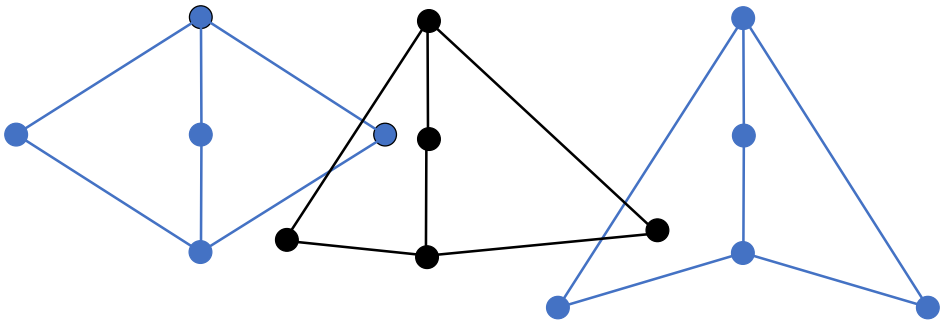
Γ_0



Γ_1

74

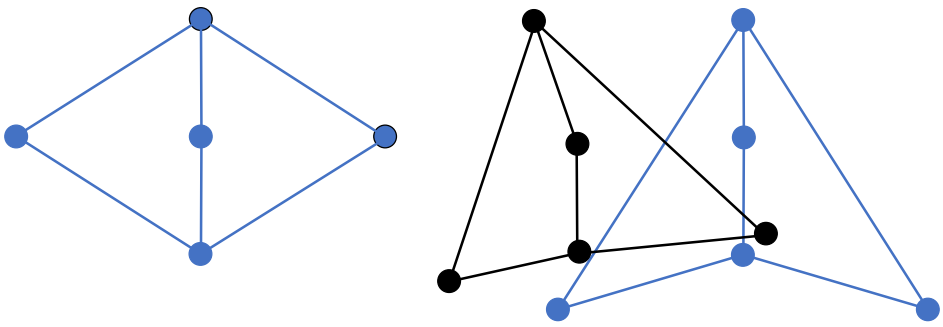
A morph



Γ_0 Γ_1

75

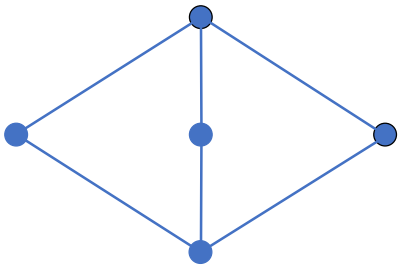
A morph



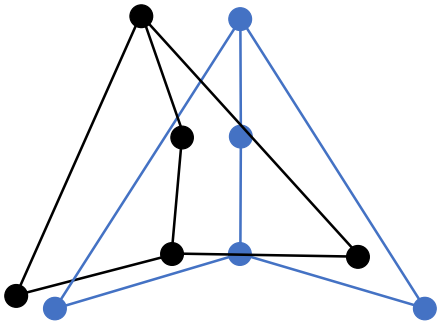
Γ_0 Γ_1

76

A morph



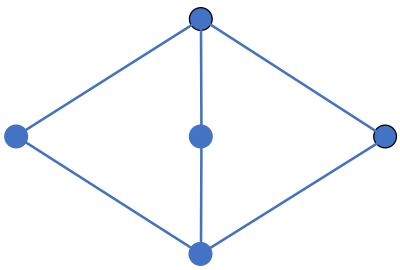
Γ_0



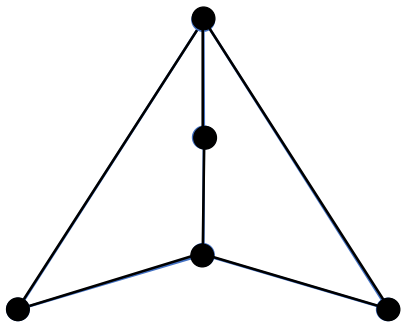
Γ_1

77

A morph



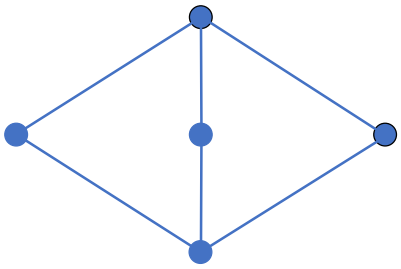
Γ_0



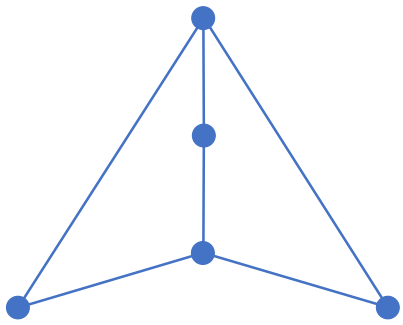
Γ_1

78

A morph



Γ_0



Γ_1

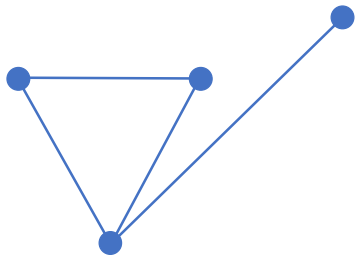
79

Morphing preserving one or more properties

- suppose that both Γ_0 and Γ_1 have a certain geometric property, e.g.
  - they are planar drawings
  - they are straight-line drawings
  - their edges are polygonal lines composed of horizontal and vertical segments
  - their faces are covex polygons
  - ….
- it is interesting that all the drawings of the morph *preserve* that property

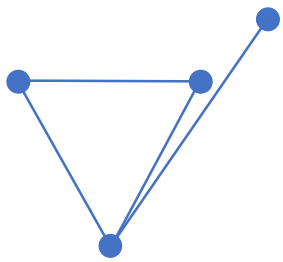
80

A morph that does not preserve planarity



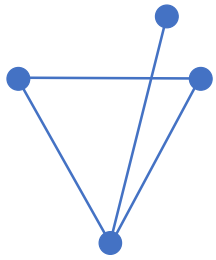
81

A morph that does not preserve planarity



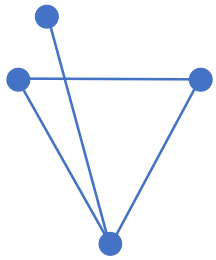
82

A morph that does not preserve planarity



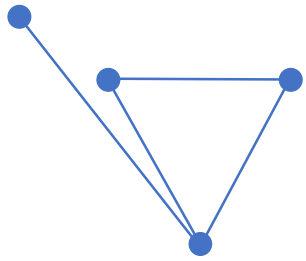
83

A morph that does not preserve planarity



84

A morph that does not preserve planarity

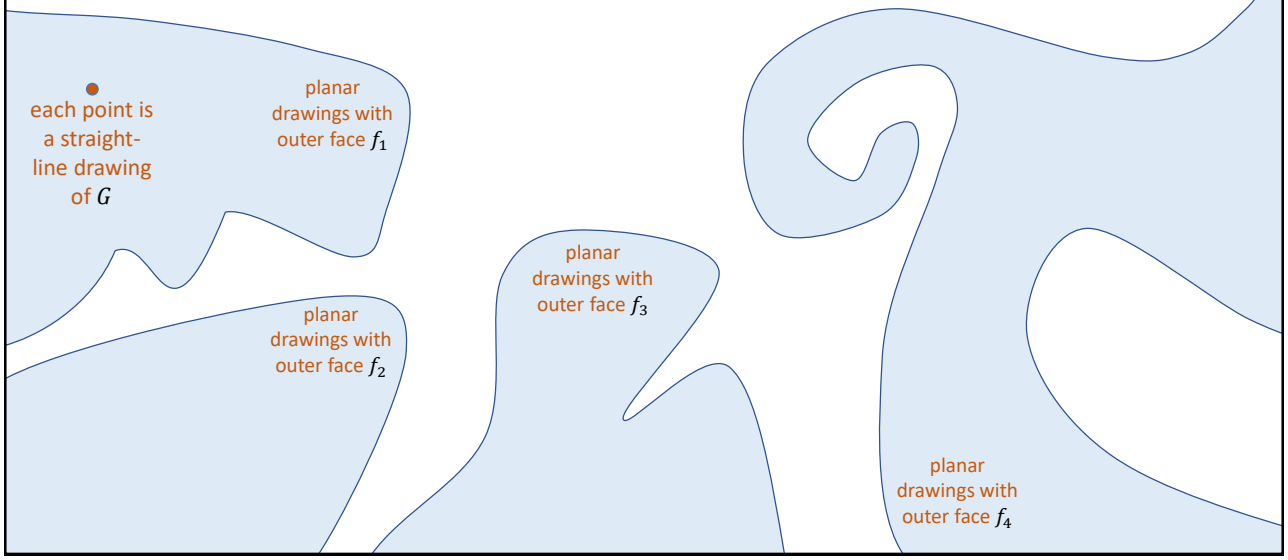


85

morphs
straight-line drawings of triangulations

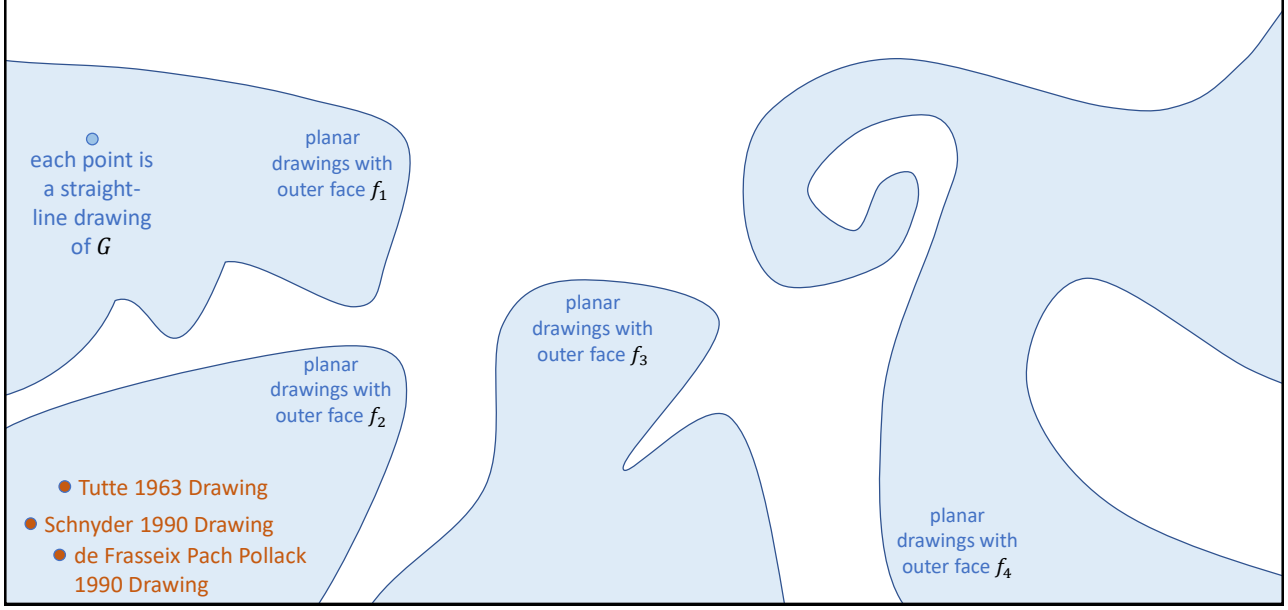
86

Straight-line drawings of a triangulation G with n vertices in R^{2n}



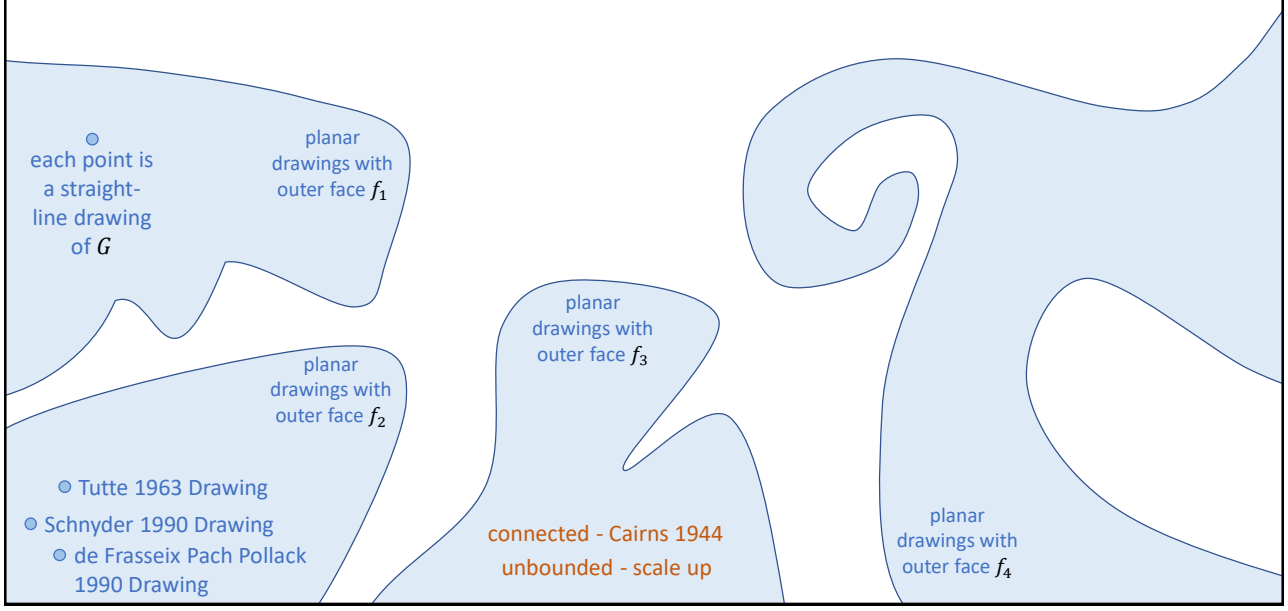
87

Straight-line drawings of a triangulation G with n vertices in R^{2n}



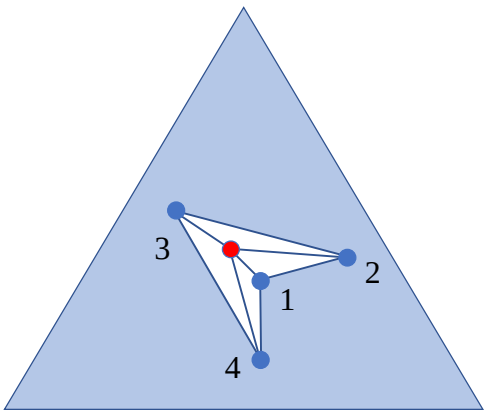
88

Straight-line drawings of a triangulation G with n vertices in R^{2n}

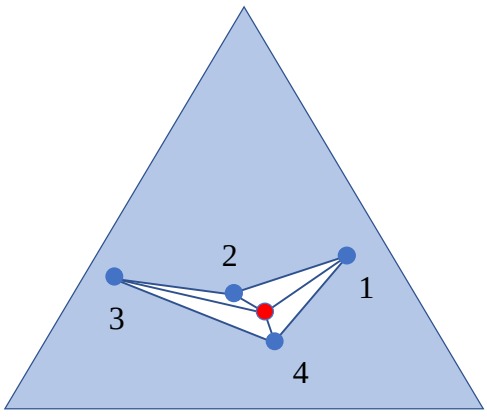


89

Cairn's double recursion approach



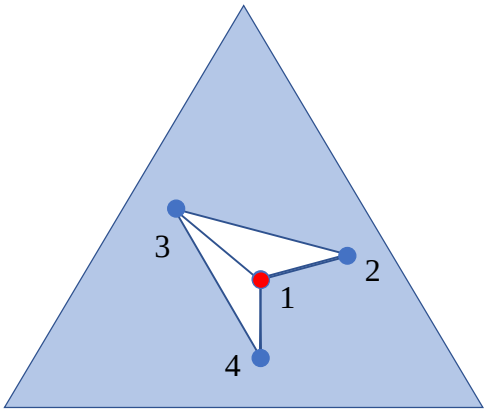
Γ_0



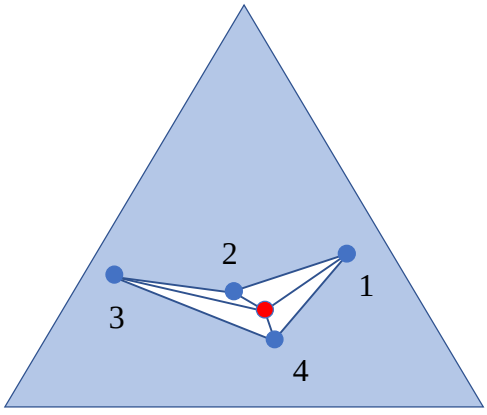
Γ_1

90

Cairn's double recursion approach



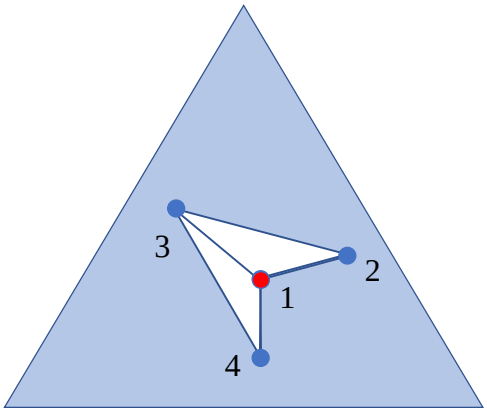
Γ_0



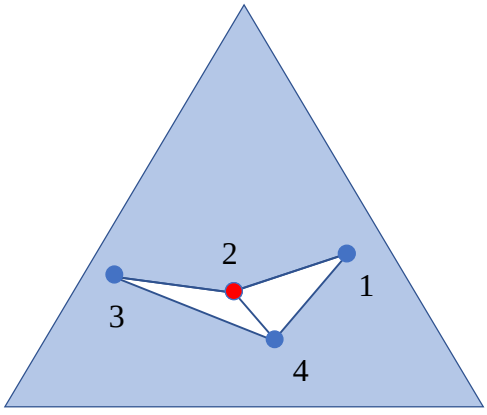
Γ_1

91

Cairn's double recursion approach



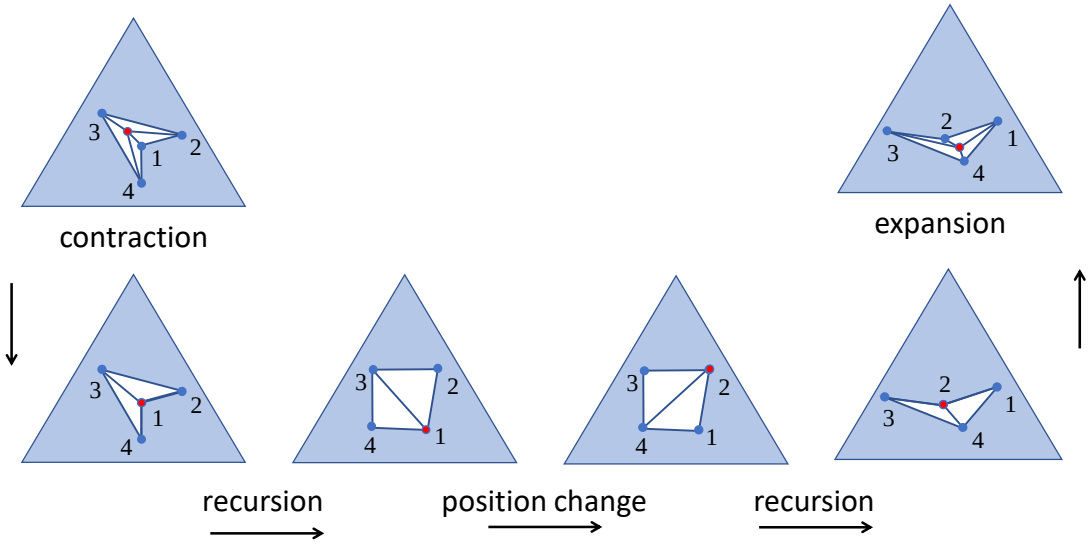
Γ_0



Γ_1

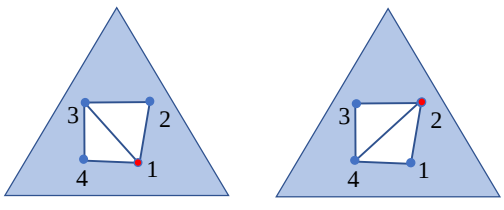
92

Cairn's double recursion approach



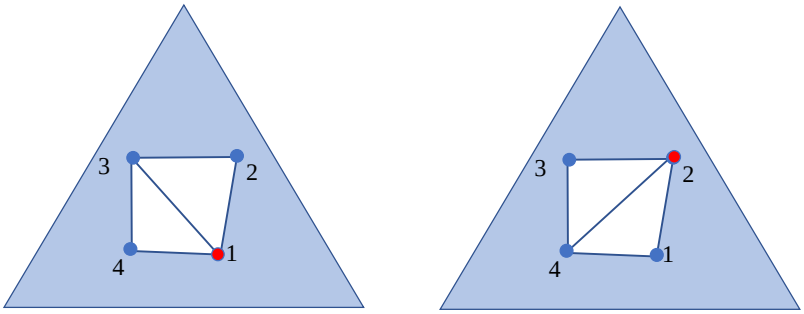
93

A closer look at the position change



94

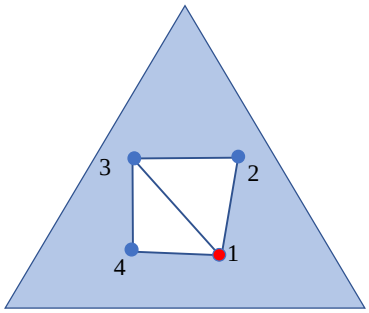
A closer look at the position change



position change

95

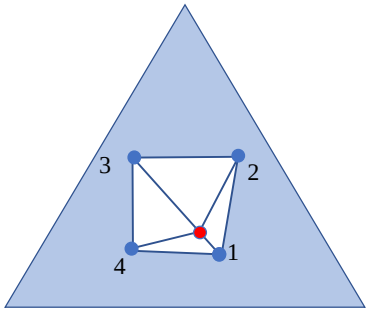
A closer look at the position change



position change

96

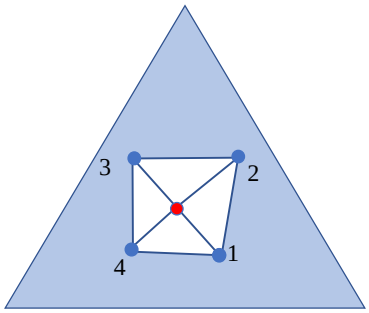
A closer look at the position change



position change

97

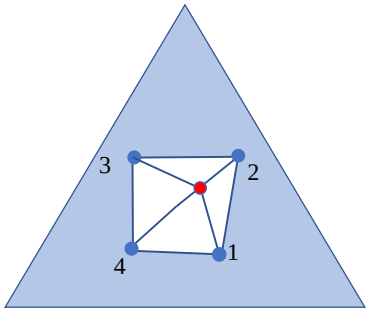
A closer look at the position change



position change

98

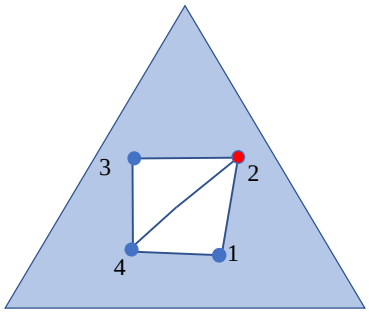
A closer look at the position change



position change

99

A closer look at the position change



position change

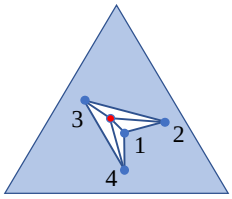
100

Cairn's double recursion approach

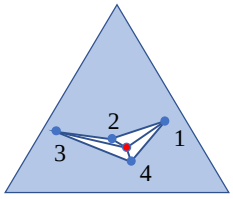
number $S(n)$ of morphing steps

$$S(n) = 2S(n-1) + O(1)$$

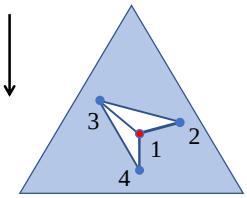
$$S(n) \in O(2^n)$$



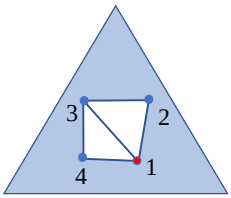
contraction $O(1)$



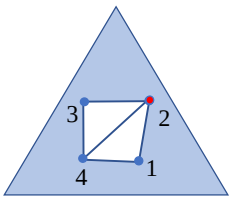
expansion $O(1)$



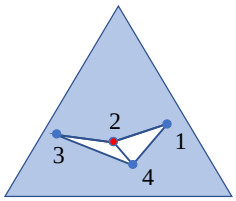
recursion $S(n-1)$



position change $O(1)$

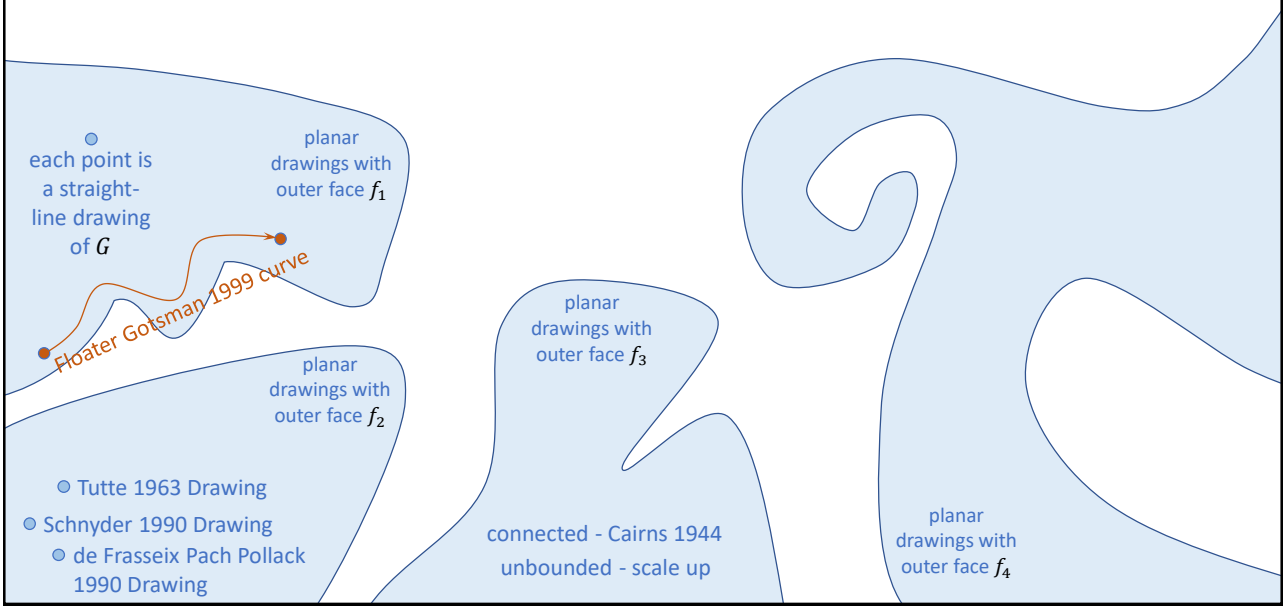


recursion $S(n-1)$



101

Straight-line drawings of a triangulation G with n vertices in R^{2n}



102

Floater Gotsman Curve – outer face

Suppose the outer face has the same drawing in Γ_0 and in Γ_1

103

Floater Gotsman Curve – drawing Γ_0

Determine coefficients $\lambda_{ij}(0)$ such that Γ_0 is a weighted barycentric drawing with weights $\lambda_{ij}(0)$

- Let x_i^0 be the coordinates of vertex i in Γ_0
- Compute positive values $\lambda_{ij}(0)$ so that

$$\sum_{j\in N(i)} \lambda_{ij}(0)\, x_j^0 = x_i^0 \quad \text{and} \quad \sum_{j\in N(i)} \lambda_{ij}(0) = 1$$

104

Floater Gotsman Curve – drawing Γ_1

Determine coefficients $\lambda_{ij}(1)$ such that Γ_1 is a weighted barycentric drawing with weights $\lambda_{ij}(1)$

- Let x_i^1 be the coordinates of vertex i in Γ_1
- Compute positive values $\lambda_{ij}(1)$ so that

$$\sum_{j\in N(i)} \lambda_{ij}(1)\, x_j^1 = x_i^1 \quad \text{and} \quad \sum_{j\in N(i)} \lambda_{ij}(1) = 1$$

105

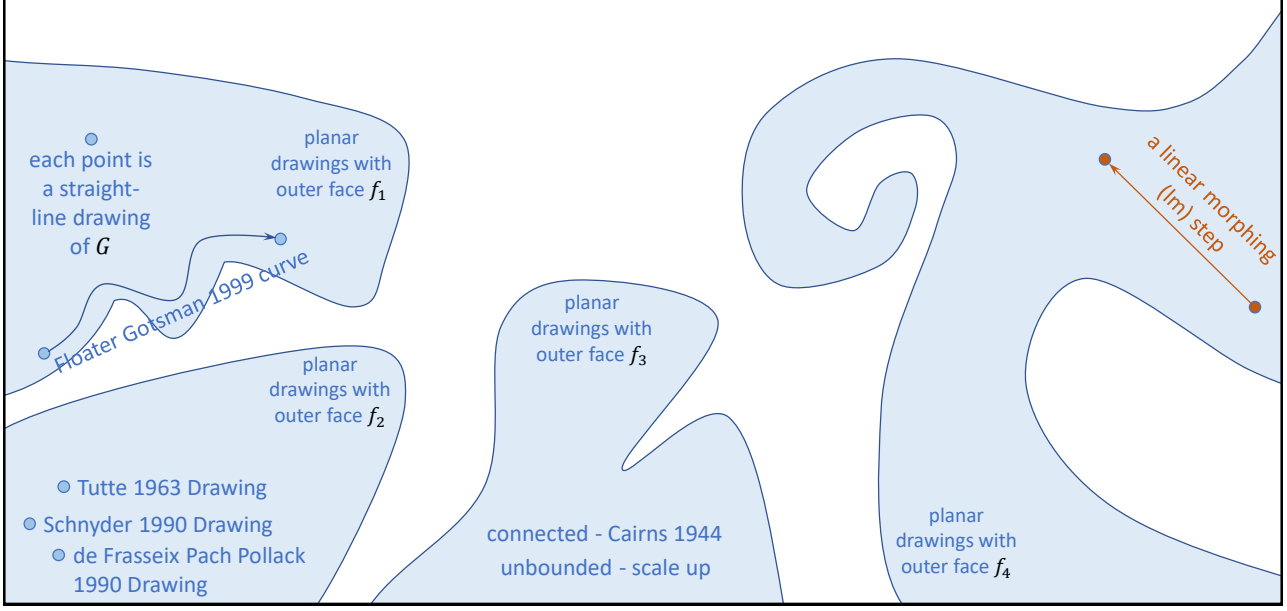
Floater Gotsman Curve

- At each time t, Γ_t is a weighted barycentric (planar) drawing with weights

$$\lambda_{ij}(t) = \lambda_{ij}(0)(1-t) + \lambda_{ij}(1)t$$

106

Straight-line drawings of a triangulation G with n vertices in R^{2n}



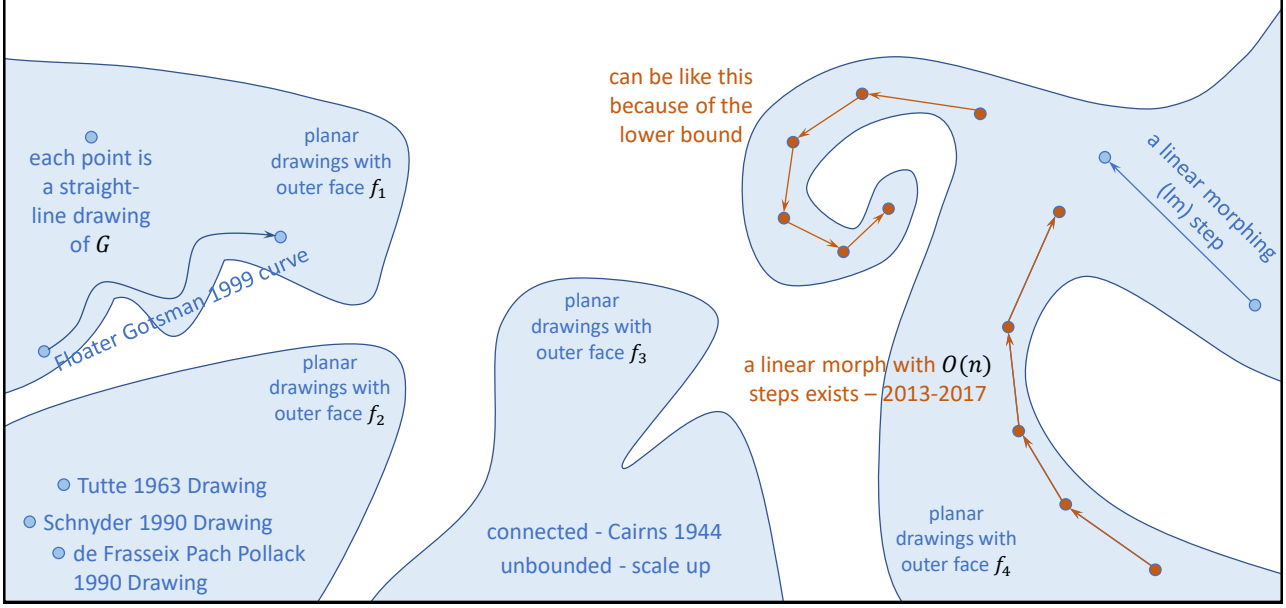
107

Linear morphs and morphing steps

- In a *linear morphing step* every vertex moves along a straight-line segment at uniform speed
  - Vertices may move at different speeds, and some vertices may remain stationary
- A *linear morph* consists only of linear morphing steps

108

Straight-line drawings of a triangulation G with n vertices in R^{2n}



109

a morph with $O(n)$ lm steps exists – 2013-2017

- S. Alamdari, P. Angelini, T.M. Chan, gdb, F. Frati, A. Lubiw, M. Patrignani, V. Roselli, S. Singla, and B. T. Wilkinson, Morphing planar graph drawings with a polynomial number of steps, SODA 2013
- P. Angelini, F. Frati, M. Patrignani, and V. Roselli, Morphing planar graph drawings Efficiently, GD 2013
- P. Angelini, G. Da Lozzo, gdb, F. Frati, M. Patrignani, and V. Roselli, Morphing planar graph drawings optimally, ICALP 2014
- S. Alamdari, P. Angelini, F. Barrera-Cruz, T.M. Chan, G. Da Lozzo, gdb, F. Frati, P. Haxell, A. Lubiw, M. Patrignani, V. Roselli, S. Singla, B.T. Wilkinson. How to morph planar graph drawings, SICOMP 2017

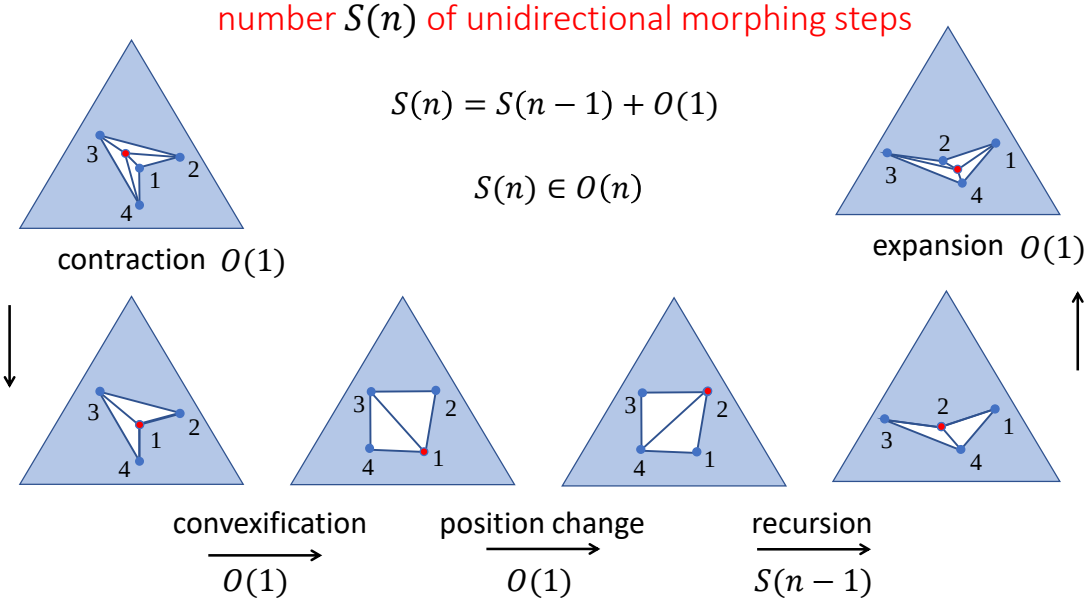
110

Morphing triangulations with a few steps

number $S(n)$ of unidirectional morphing steps

$S(n) = S(n-1) + O(1)$

$S(n) \in O(n)$



111

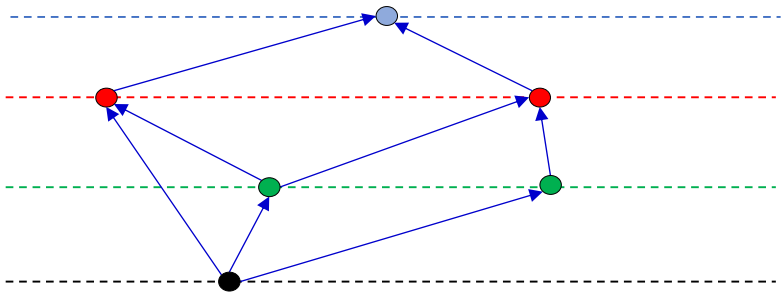
A convexification tool

- Given:
  - a triconnected plane graph $G = (V, E)$
  - a set L of parallel lines
  - a mapping of the vertices of V to lines of L such that orienting the edges of E according to the order of the lines in L yields an st-orientation of G
- Then G admits a convex drawing (all faces are convex polygons) in which each vertex of V lies on the line of L it is mapped to

Hong Nagamochi, Convex drawings of hierarchical planar graphs and clustered planar graphs, JDA 2010

112

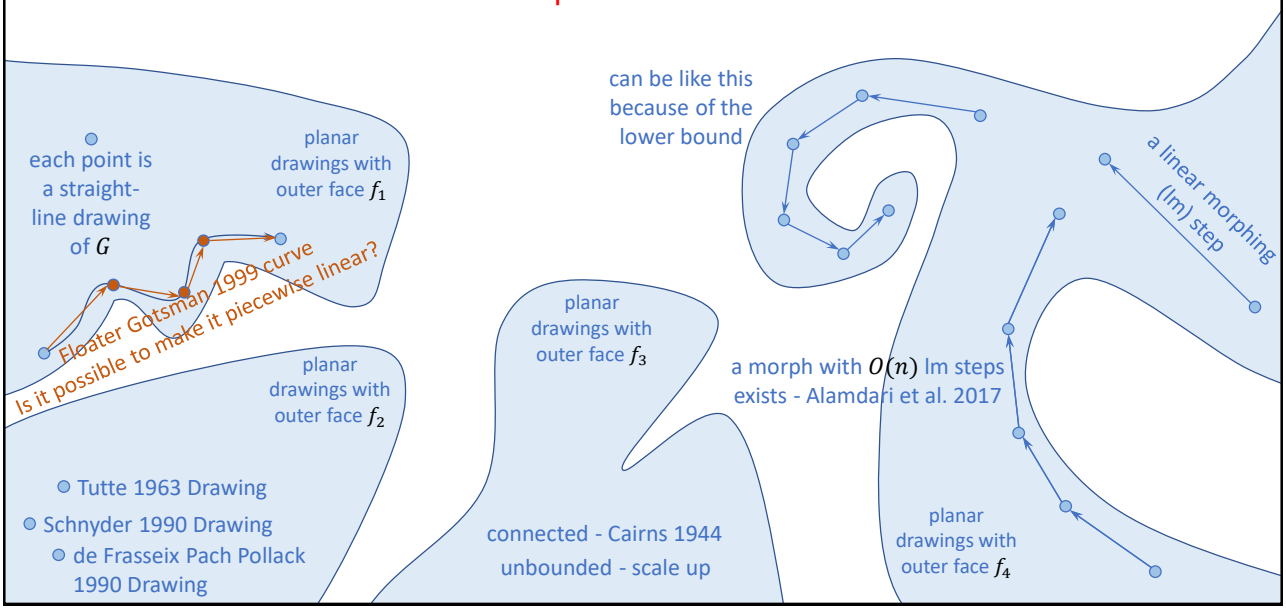
A convexification tool



This slide: Courtesy of Patrizio Angelini

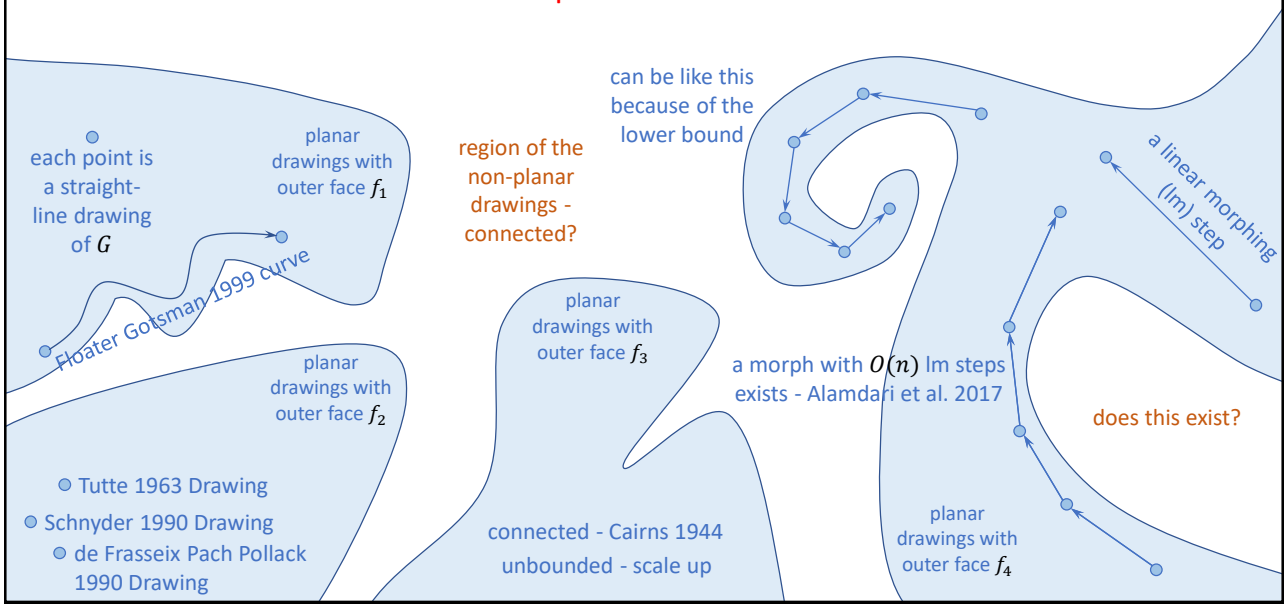
113

Straight-line drawings of a triangulation G with n vertices in R^{2n} problems



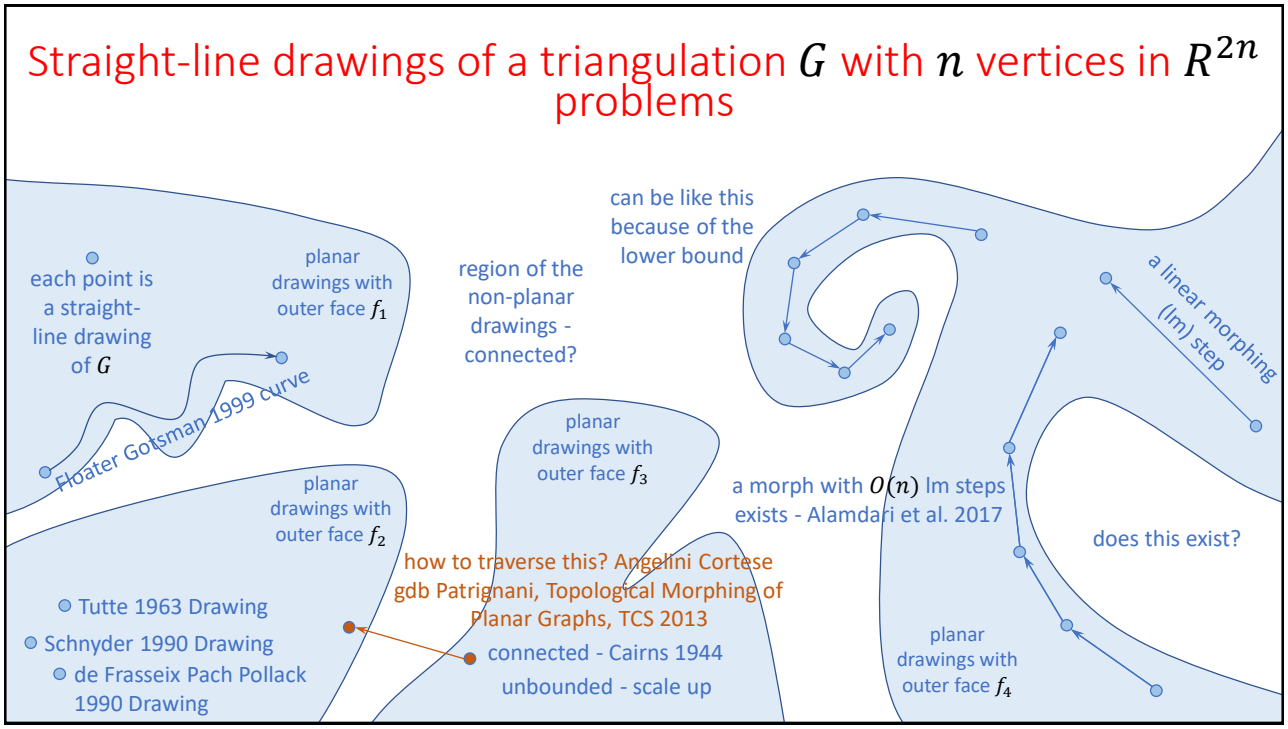
114

Straight-line drawings of a triangulation G with n vertices in R^{2n} problems



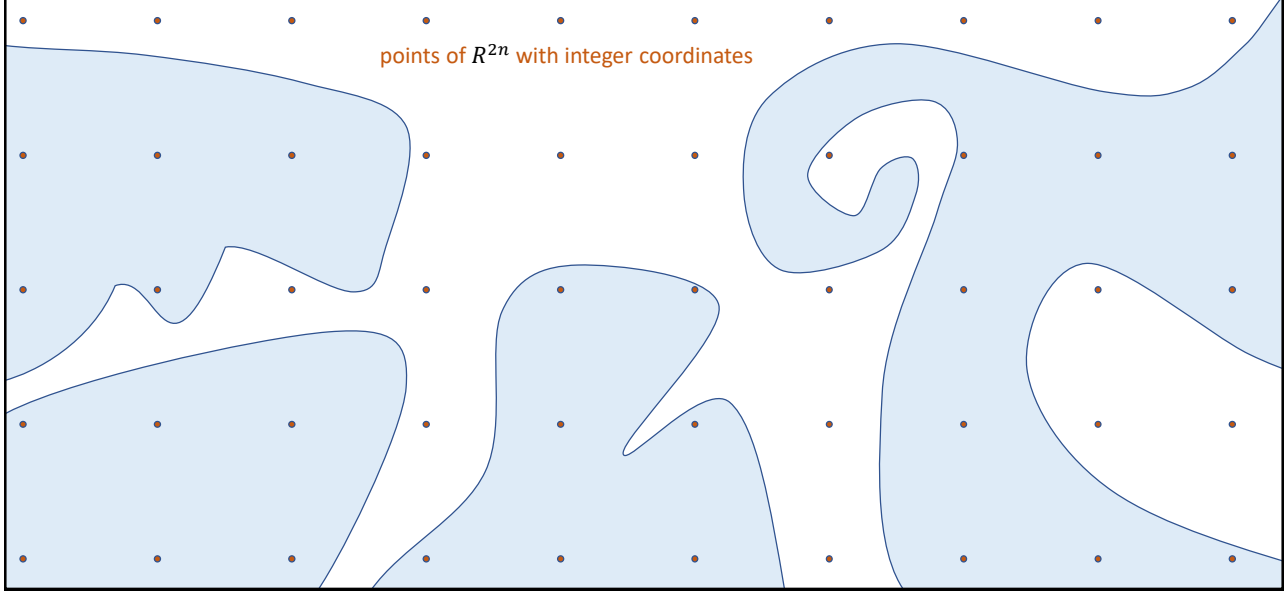
115

Straight-line drawings of a triangulation G with n vertices in R^{2n} problems



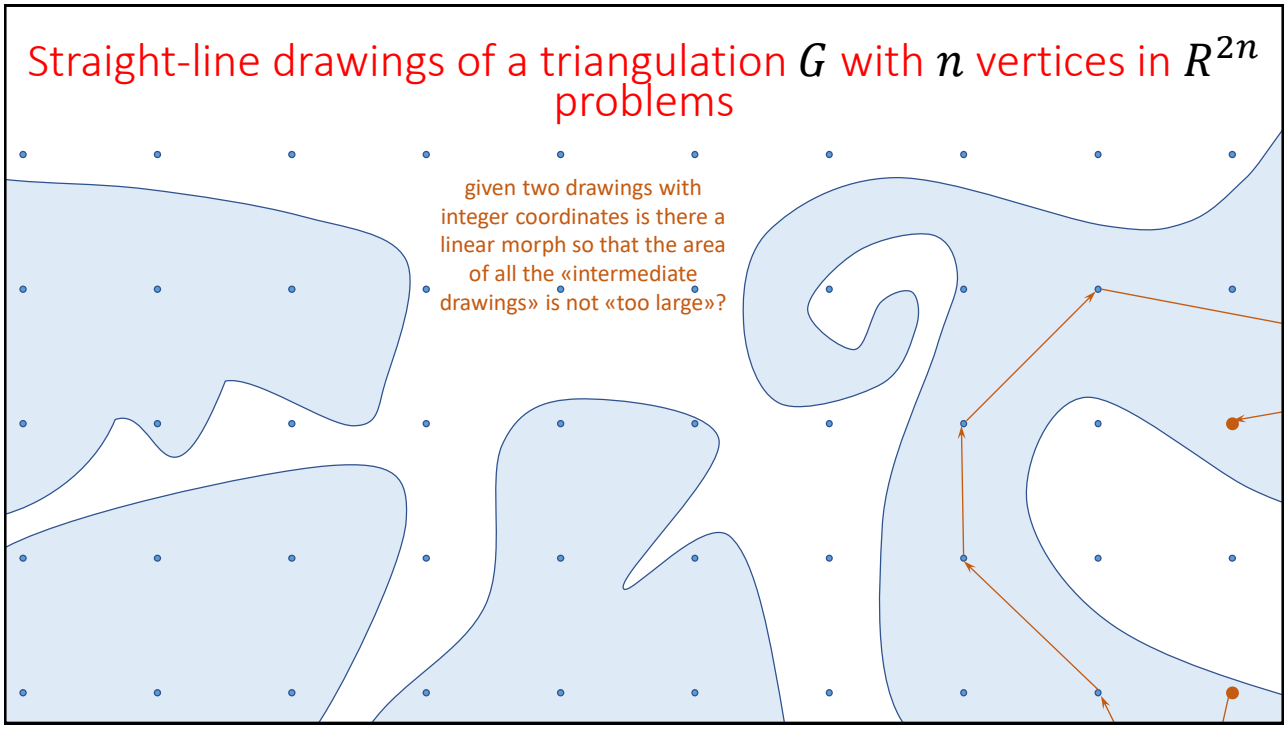
116

Straight-line drawings of a triangulation G with n vertices in R^{2n}



117

Straight-line drawings of a triangulation G with n vertices in R^{2n} problems

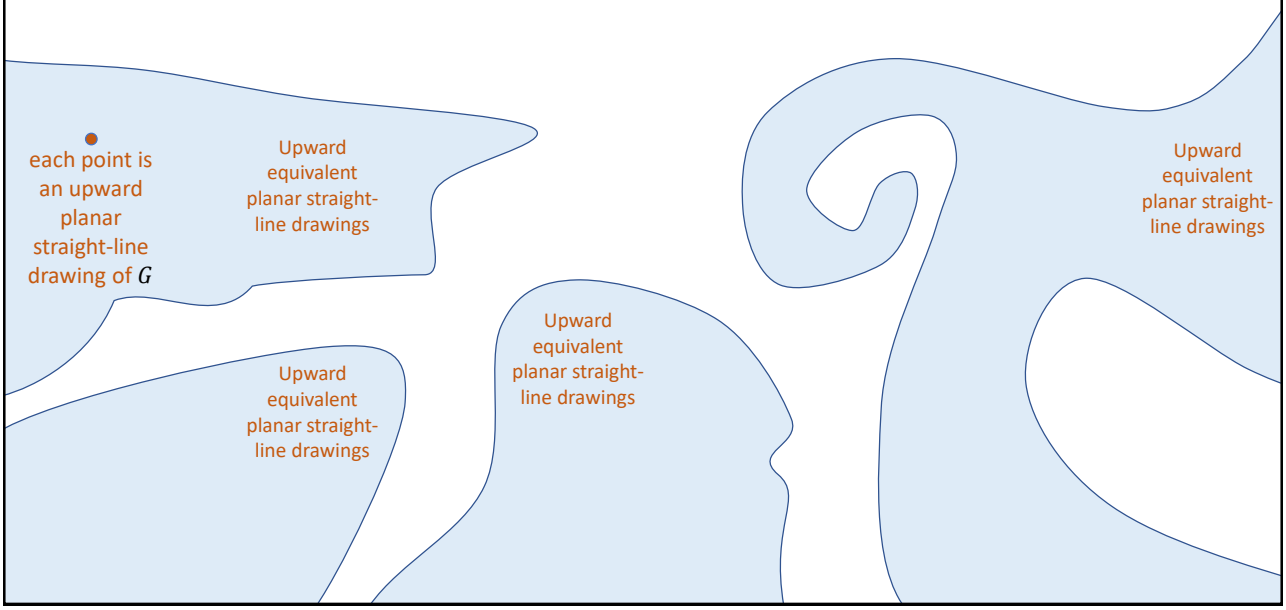


118

morphs
upward planar straight-line drawings of upward-planar graphs

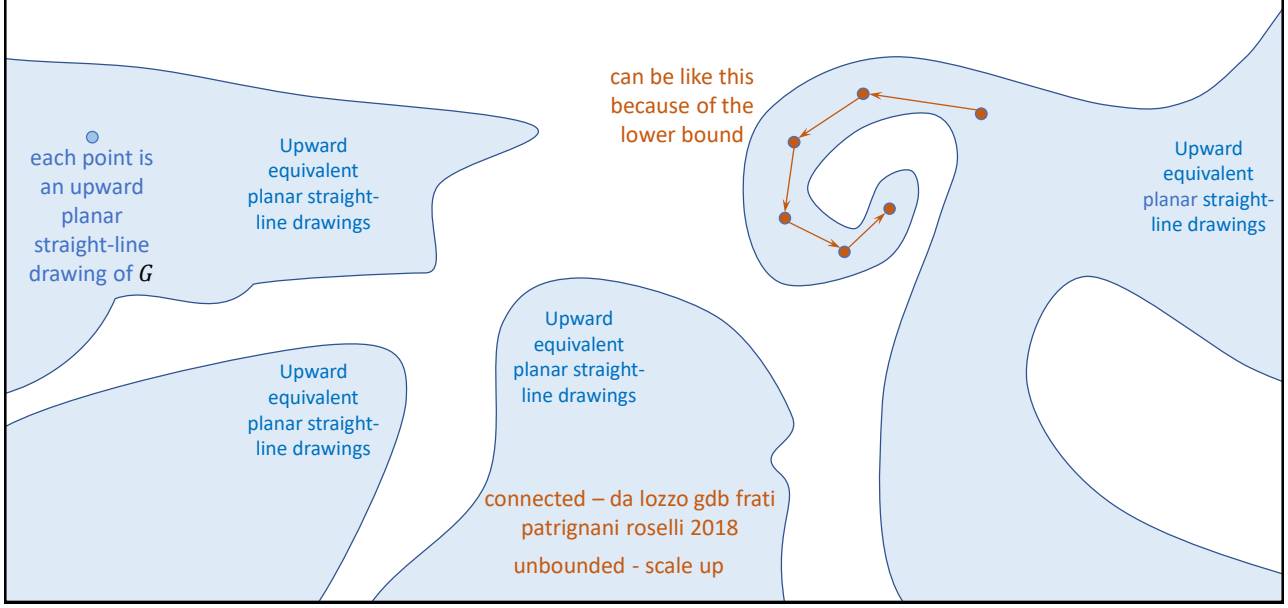
119

Upward straight-line drawings of an upward planar graph G with n vertices in R^{2n}



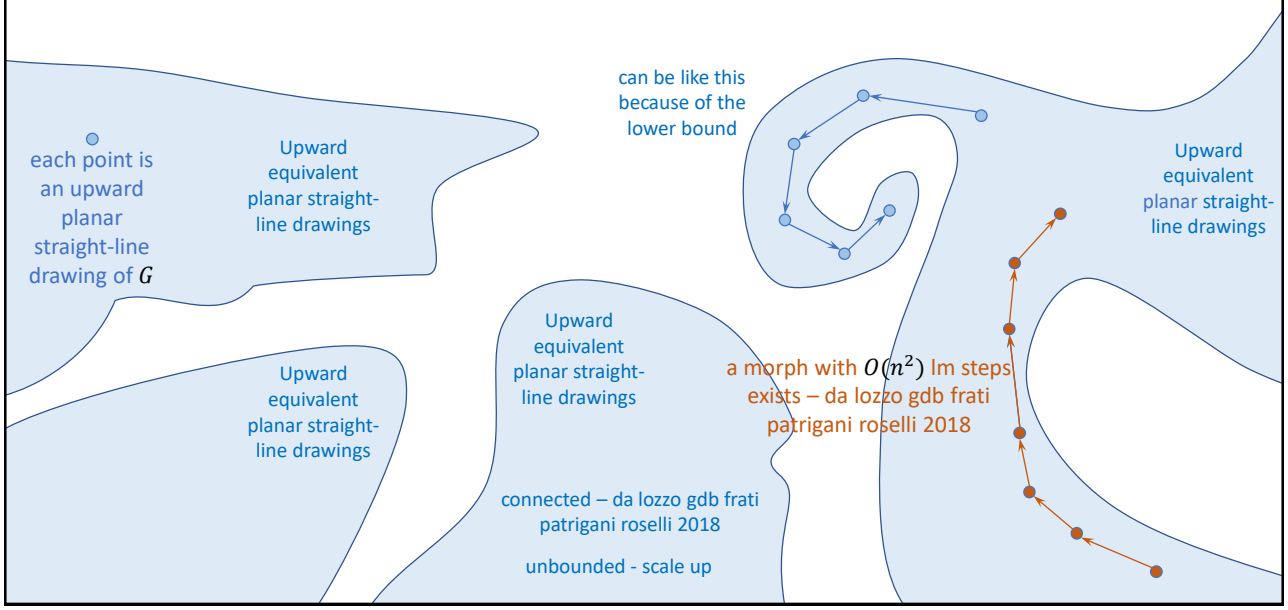
120

Upward straight-line drawings of an upward planar graph G with n vertices in R^{2n}



121

Upward straight-line drawings of an upward planar graph G with n vertices in R^{2n}

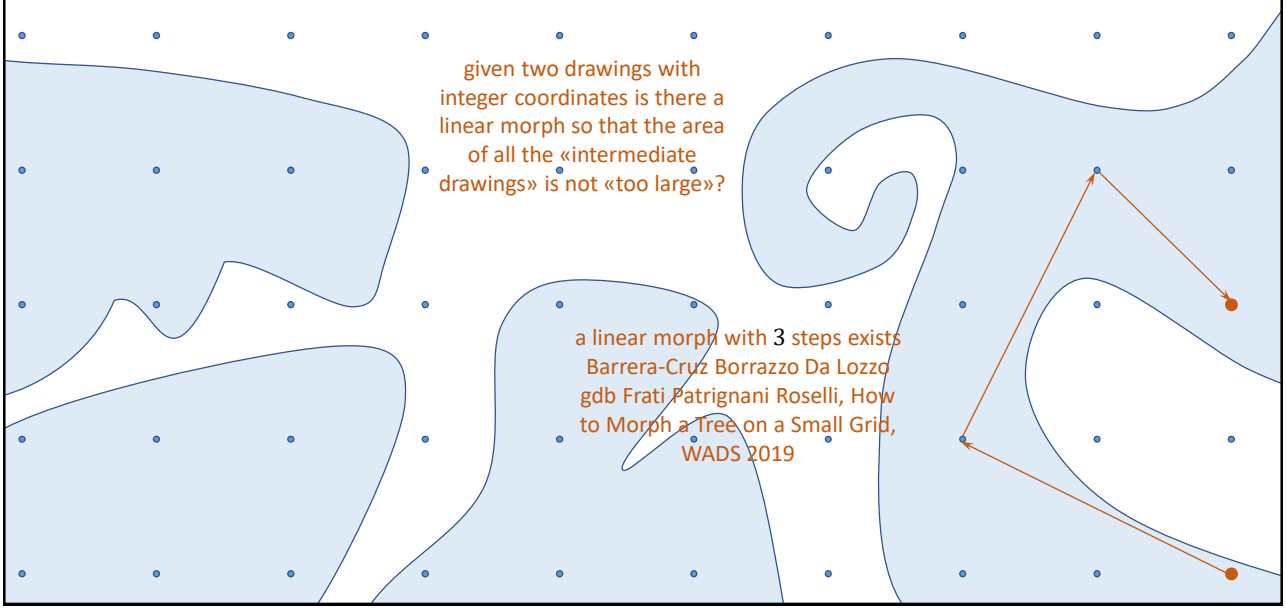


122

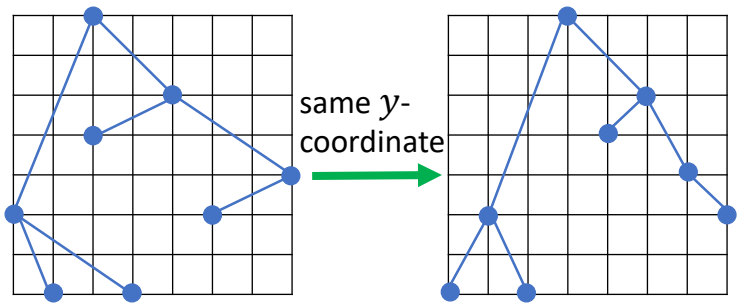
morphs
upward planar straight-line drawings of rooted trees

123

Upward straight-line drawings of a rooted binary tree T with n vertices in R^{2n}



124

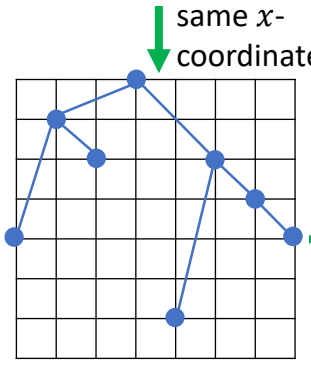


Γ_0

This Slide: Courtesy of Fabrizio Frati

Uses ideas from:

Da Lozzo gdb Frati, Patrignani Roselli, Upward Planar Morphs, GD 2018



Γ_1

Employs proof techniques of:

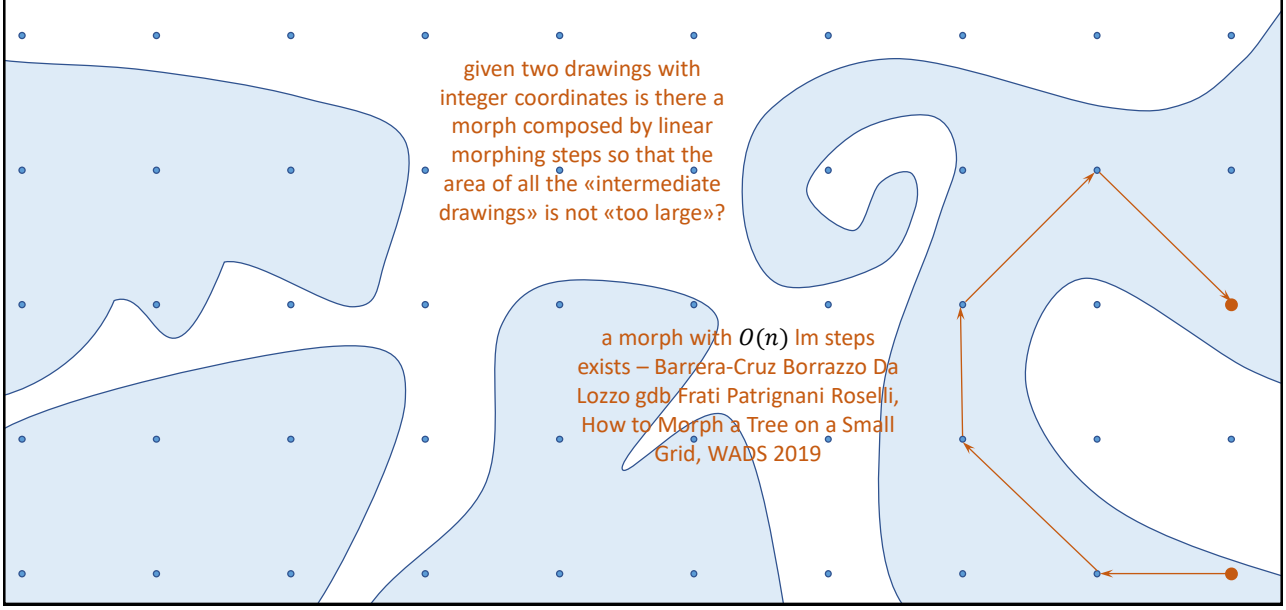
Alamdari Angelini Barrera-Cruz Chan Da Lozzo gdb Frati Haxell Lubiw Patrignani Roselli Singla Wilkinson. How to Morph Planar Graph Drawings, SICOMP 2017

125

Upward straight-line drawings of a rooted tree T with n vertices in R^{2n}

given two drawings with integer coordinates is there a morph composed by linear morphing steps so that the area of all the «intermediate drawings» is not «too large»?

a morph with $O(n)$ lm steps exists – Barrera-Cruz Borrazzo Da Lozzo gdb Frati Patrignani Roselli, How to Morph a Tree on a Small Grid, WADS 2019

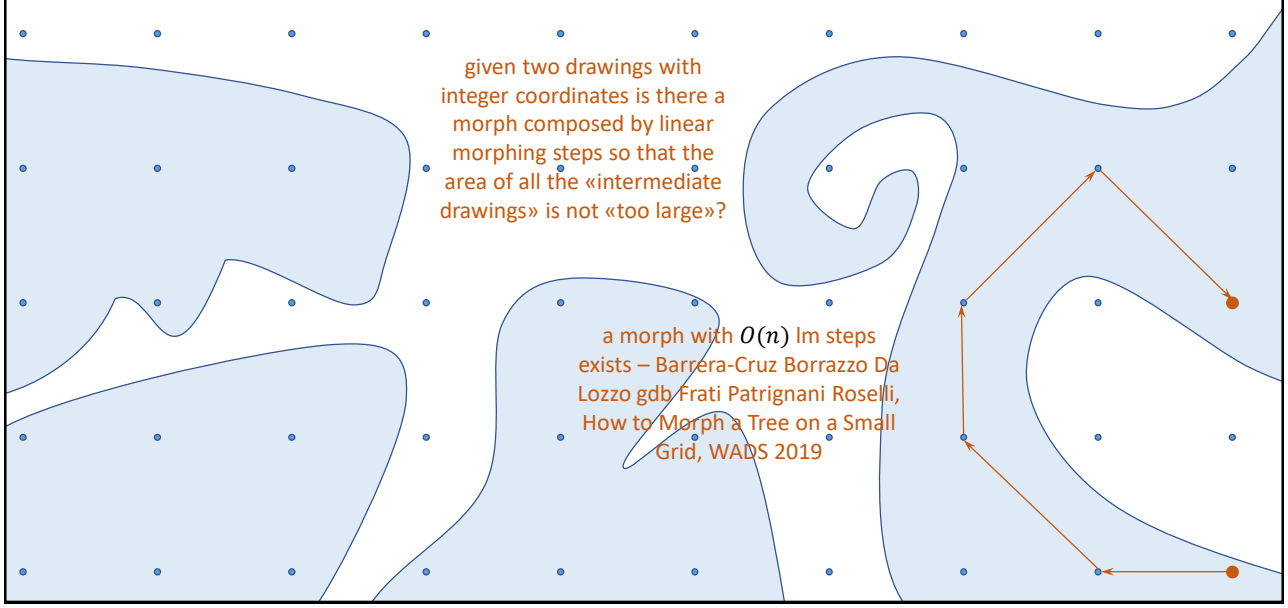


126

Straight-line drawings of a tree T with n vertices in R^{2n}

given two drawings with integer coordinates is there a morph composed by linear morphing steps so that the area of all the «intermediate drawings» is not «too large»?

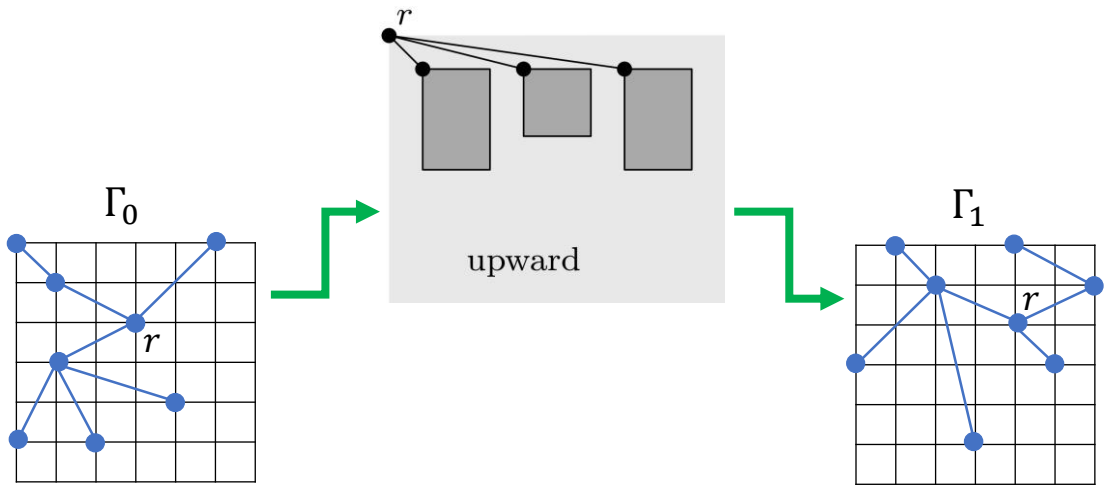
a morph with $O(n)$ lm steps exists – Barrera-Cruz Borrazzo Da Lozzo gdb Frati Patrignani Roselli, How to Morph a Tree on a Small Grid, WADS 2019



127

General strategy

This Slide: Courtesy of Fabrizio Frati



128

More GD topics that deal with time

- Simultaneous embedding with fixed edges (SEFE)
  - Time is in layers
- Phylogenetic Networks
  - Represent the evolution over time
- Greedy drawings
  - Time is in routing
- Constraints
  - Objects that do not change in the graph do not change in the drawing
- Drawing extensions
  - The graph grows during time

129

More generally ….

- All Graph Drawing Papers consider time
- Some of them focus only on instant $t = 0$

130

This Is Time for Graph Drawing

Giuseppe Di Battista
Università degli Studi Roma Tre

27th International Symposium on Graph Drawing and Network Visualization – GD2019
Průhonice/Prague, September 17-20, 2019

131