# Proportional Contact Representation of Planar Graphs 

Stephen Kobourov<br>University of Arizona

## Proportional Contact Representation of Planar Graphs

## Stephen Kobourov <br> University of Arizona



CUNY, September 21, 2018

## Contact Representation of Graphs

Different contact flavors:

- circles, segments, triangles, boxes, ...
- point contact vs. side contact
- unweighted vs. weighted
- rectilinear vs. many slopes
- convex regions vs. arbitrary
- with holes vs. without holes
- 2D, 3D
- ...



## Contact Representations



- vertices: polygons
- edges: non-trivial borders
- parameters: complexity, convexity, holes


## Proportional Contact Representations



- vertices: polygons
- edges: non-trivial borders
- parameters: complexity, convexity, holes
- vertex weight $\Rightarrow$ area of polygon


## Rectilinear Contact Representations



- rectilinear polygons, side-contacts, hole-free
- unweighted representation: a.k.a. rectilinear dual


## Rectilinear Proportional Representations



- rectilinear polygons, side-contacts, hole-free
- proportional representation: a.k.a. rectilinear cartogram


## Contact Representation Problem

## Goal: represent the graph with simple polygons



## Contact Representation Problem

## Goal: represent the graph with simple polygons



- Minimize polygonal complexity


## Contact Representation Problem

## Goal: represent the graph with simple polygons



- Minimize polygonal complexity
- Minimize holes (unused areas)


## Proportional Contact Representation Problem

## Goal: represent the graph with simple polygons



- Minimize polygonal complexity
- Minimize holes (unused areas)
- Use convex polygons (when possible)


## Proportional Contact Representation Problem

## Goal: represent the graph with simple polygons



- Minimize polygonal complexity
- Minimize holes (unused areas)
- Use convex polygons (when possible)
- Realize the given weights by areas (value-by-area representation)


## Motivation

## Architectural Floorplanning

- room topology: graph
- pre-specified room areas
- land management



## Motivation cont.

## VLSI Layout

- VLSI modules: polygons
- connections: adjacencies

- module sizes: areas



## Motivation cont.

## Data Representation

- Redrawing of geographic maps
- Population cartograms




## Related Work

- contact with circles [Koebe, 1936]
- contact with triangles [de Fraysseix et al., 1994]
- contact with 3D cubes [Felsner and Francis, 2011].



## Related Work cont.

Point contact representation of both the primal and the dual graph with triangles [ Gonçalves et al. 2010].


## Related Work cont.

Rectilinear Duals: Eight-sided rectilinear polygons are always sufficient and sometimes necessary for maximal planar graphs [Yeap and Sarrafzadeh 1993, He 1999, Liao et al., 2003].


## Related Work cont.

Rectilinear Duals: Rectangles are sufficient for 4-connected maximal planar graphs [Ungar 1953, Kozminski and Kinnen 1985, Kant and He , 1997]


## Related Work cont.

## Rectilinear Cartograms

- Lower bound on the complexity is 8 , even for the unweighted case [Yeap and Sarrafzadeh, 1993]
- Upper bound on the polygonal complexity:
- from the initial 40 [de Berg et al. 2006]
- to 34 [Kawaguchi and Nagamochi, 2007]
- to 12 by [Biedl and Velázquez, 2011]


## Related Work cont.

## Rectilinear Cartograms

- Lower bound on the complexity is 8 , even for the unweighted case [Yeap and Sarrafzadeh, 1993]
- Upper bound on the polygonal complexity:
- from the initial 40 [de Berg et al. 2006]
- to 34 [Kawaguchi and Nagamochi, 2007]
- to 12 by [Biedl and Velázquez, 2011]
- to 10 [Alam et al., ISAAC'11]
- to 8 [Alam et al., SoCG'12]



## Related Work cont.

Connections...

- The edges of any maximal planar graph can be partitioned into 3 edge-disjoint spanning trees [Nash 1961, Tutte 1961]


## Related Work cont.

Connections...

- The edges of any maximal planar graph can be partitioned into 3 edge-disjoint spanning trees [Nash 1961, Tutte 1961]
- Schnyder realizer was defined and used for straight-line drawing [Schnyder 1990]


## Related Work cont.

Connections...

- The edges of any maximal planar graph can be partitioned into 3 edge-disjoint spanning trees [Nash 1961, Tutte 1961]
- Schnyder realizer was defined and used for straight-line drawing [Schnyder 1990]
- Canonical order defined and used for straight-line drawing [de Fraysseix, Pach and Pollack 1990]


## Related Work cont.

Connections...

- The edges of any maximal planar graph can be partitioned into 3 edge-disjoint spanning trees [Nash 1961, Tutte 1961]
- Schnyder realizer was defined and used for straight-line drawing [Schnyder 1990]
- Canonical order defined and used for straight-line drawing [de Fraysseix, Pach and Pollack 1990]
- Relations between canonical order, Schnyder realizer used to prove various results [de Fraysseix, Kant, He, Felsner, Fusy, Ueckerdt,...]



## Canonical Order



- Ordering of the vertices


## Canonical Order



- Ordering of the vertices
- $G_{i}$ (induced on vertices $1, \ldots, i$ ) is biconnected


## Canonical Order



- Ordering of the vertices
- $G_{i}$ (induced on vertices $1, \ldots, i$ ) is biconnected


## Canonical Order



- Ordering of the vertices
- $G_{i}$ (induced on vertices $1, \ldots, i$ ) is biconnected


## Canonical Order



- Ordering of the vertices
- $G_{i}$ (induced on vertices $1, \ldots, i$ ) is biconnected
- $i+1$ is on outerface of $G_{i+1}$


## Canonical Order



- Ordering of the vertices
- $G_{i}$ (induced on vertices $1, \ldots, i$ ) is biconnected
- $i+1$ is on outerface of $G_{i+1}$
- $i+1$ has 2 or more consecutive neighbors on $C_{i}$


## Schnyder Realizer



## Schnyder Realizer



- partition of the internal edges into three spanning trees


## Schnyder Realizer



- partition of the internal edges into three spanning trees
- every vertex has out-degree exactly one in $T_{1}, T_{2}$ and $T_{3}$


## Schnyder Realizer



- partition of the internal edges into three spanning trees
- every vertex has out-degree exactly one in $T_{1}, T_{2}$ and $T_{3}$
- vertex rule: ccw order of edges: entering $T_{1}$, leaving $T_{2}$, entering $T_{3}$, leaving $T_{1}$, entering $T_{2}$, leaving $T_{3}$


## Schnyder Realizer cont.



- 3 edge-disjoint spanning trees $T_{1}, T_{2}, T_{3}$ cover $G$


## Schnyder Realizer cont.



- 3 edge-disjoint spanning trees $T_{1}, T_{2}, T_{3}$ cover $G$
- $T_{1}, T_{2}, T_{3}$ rooted at external vertices of $G$


## From Canonical Order to Schnyder Realizer



When a new vertex is inserted in the canonical order:

## From Canonical Order to Schnyder Realizer



When a new vertex is inserted in the canonical order:

- leftmost edge is outgoing blue


## From Canonical Order to Schnyder Realizer



When a new vertex is inserted in the canonical order:

- leftmost edge is outgoing blue
- rightmost edge is outgoing green


## From Canonical Order to Schnyder Realizer



When a new vertex is inserted in the canonical order:

- leftmost edge is outgoing blue
- rightmost edge is outgoing green
- remaining (0 or more edges) incoming red


## From Canonical Order to Schnyder Realizer



When a new vertex is inserted in the canonical order:

- leftmost edge is outgoing blue
- rightmost edge is outgoing green
- remaining (0 or more edges) incoming red
- (it gets its outgoing red when it is "closed off")


## From Schnyder Realizer to Canonical Order



Two easy ways:

- counterclockwise preorder traversal of the blue tree


## From Schnyder Realizer to Canonical Order



Two easy ways:

- counterclockwise preorder traversal of the blue tree
- topological order of $T_{1} \cup T_{2}^{-1} \cup T_{3}^{-1}$


## Outerplanar: Contact Representation

(1)-2

## Outerplanar: Contact Representation

(1)-(2)


## Outerplanar: Contact Representation



## Outerplanar: Contact Representation

(1)-(2)



## Outerplanar: Contact Representation



## Outerplanar: Contact Representation



## Outerplanar: Contact Representation

(1)-(2)


(4)


## Outerplanar: Contact Representation



## Outerplanar: Contact Representation



- Outerplanar graphs are T3G's


## Outerplanar: Contact Representation



- Outerplanar graphs are T3G's
- [Kobourov et al. GD 2010]


## Outerplanar: Proportional Contact Representation



## Outerplanar: Proportional Contact Representation



## Outerplanar: Proportional Contact Representation



## Outerplanar: Proportional Contact Representation



## Outerplanar: Proportional Contact Representation



## Outerplanar: Proportional Contact Representation



## Outerplanar: Proportional Contact Representation



## Outerplanar: Proportional Contact Representation



## Outerplanar: Proportional Contact Representation



- OP graphs are T3G's; proportional OP graphs are T4Gs


## Outerplanar: Proportional Contact Representation



- OP graphs are T3G's; proportional OP graphs are T4Gs
- (proportional T3G but with $O(n)$ complexity outerface)


## Outerplanar: Proportional Contact Representation



- OP graphs are T3G's; proportional OP graphs are T4Gs
- (proportional T3G but with $O(n)$ complexity outerface)
- [Alam et al. GD 2011]


## Contact Representation of Maximal Planar Graphs



## Contact Representation of Maximal Planar Graphs



## Contact Representation of Maximal Planar Graphs



## Contact Representation of Maximal Planar Graphs



## Contact Representation of Maximal Planar Graphs



- General planar graphs are T6G’s


## Contact Representation of Maximal Planar Graphs



- General planar graphs are T6G’s
- Convex, max complexity 6 , no holes


## Contact Representation of Maximal Planar Graphs



- General planar graphs are T6G’s
- Convex, max complexity 6, no holes
- [Duncan et al. Algorithmica 2011]


## Lower Bound

We showed how to do it with 6 sides; 6 is also necessary:


## Lower Bound

We showed how to do it with 6 sides; 6 is also necessary:


## Lower Bound

We showed how to do it with 6 sides; 6 is also necessary:


- add a vertex in each internal face and triangulate


## Lower Bound

We showed how to do it with 6 sides; 6 is also necessary:


- add a vertex in each internal face and triangulate
- "how can it be otherwise" argument follows...


## Cartograms for Maximal Planar Graphs



- compute a canonical order and Schnyder realizer


## Cartograms for Maximal Planar Graphs



- compute a canonical order and Schnyder realizer
- represent each vertex by a canonical 8-sided polygon


## Cartograms for Maximal Planar Graphs



- Union of 4 rectangles: base, stump, left box, right box.
- red outgoing adjacency through top of stump blue outgoing adjacency through left of base green outgoing adjacency through right of base


## Cartograms for Maximal Planar Graphs



## Cartograms for Maximal Planar Graphs



## Cartograms for Maximal Planar Graphs



## Cartograms for Maximal Planar Graphs



## Cartograms for Maximal Planar Graphs



## Cartograms for Maximal Planar Graphs



## Cartograms for Maximal Planar Graphs



## Cartograms for Maximal Planar Graphs



## Cartograms for Maximal Planar Graphs



## Cartograms for Maximal Planar Graphs



## Cartograms for Maximal Planar Graphs



- How can we realize the areas?


## Cartograms for Maximal Planar Graphs



## Cartograms for Maximal Planar Graphs



- rectangular Layout and each segment is "one-sided"


## Cartograms for Maximal Planar Graphs



- rectangular Layout and each segment is "one-sided"
- can realize any specified set of areas for the rectangles


## Cartograms for Maximal Planar Graphs



- rectangular Layout and each segment is "one-sided"
- can realize any specified set of areas for the rectangles


## Cartograms for Maximal Planar Graphs



- rectangular Layout and each segment is "one-sided"
- can realize any specified set of areas for the rectangles
- [Alam et al. SoCG 2012]


## Matching Lower Bound

- Lower bound on the complexity for a maximal planar graph is 8 [Yeap and Sarrafzadeh, 1993].



## Matching Lower Bound

- Lower bound on the complexity for a maximal planar graph is 8 [Yeap and Sarrafzadeh, 1993].

- So our 8-sided result is optimal in terms of complexity


## Matching Lower Bound

- Lower bound on the complexity for a maximal planar graph is 8 [Yeap and Sarrafzadeh, 1993].

- So our 8-sided result is optimal in terms of complexity
- However, we don't have a polynomial time algorithm!


## Matching Lower Bound

- Lower bound on the complexity for a maximal planar graph is 8 [Yeap and Sarrafzadeh, 1993].

- So our 8-sided result is optimal in terms of complexity
- However, we don't have a polynomial time algorithm!
- (We do have a linear-time Algorithm for complexity 10)


## Algorithm for Hamiltonian Graphs

- proportional rectilinear 8-sided representation


## Algorithm for Hamiltonian Graphs

- proportional rectilinear 8-sided representation
- linear time exact computation for Hamiltonian max planar


## Algorithm for Hamiltonian Graphs

- proportional rectilinear 8-sided representation
- linear time exact computation for Hamiltonian max planar
- use different canonical 8-sided polygon:


## Algorithm for Hamiltonian Graphs

- proportional rectilinear 8-sided representation
- linear time exact computation for Hamiltonian max planar
- use different canonical 8-sided polygon:



## Algorithm for Hamiltonian Graphs



## Algorithm for Hamiltonian Graphs



## Algorithm for Hamiltonian Graphs



## Algorithm for Hamiltonian Graphs



- Stack blocks for vertices in order of the Hamiltonian cycle


## Algorithm for Hamiltonian Graphs



| 10 |
| :---: |
| 9 |
| 8 |
| 7 |
| 6 |
| 5 |
| 4 |
| 3 |
| 2 |
| 1 |

- Stack blocks for vertices in order of the Hamiltonian cycle
- Extend "arms" left and right to reach neighbors


## Algorithm for Hamiltonian Graphs



- Stack blocks for vertices in order of the Hamiltonian cycle
- Extend "arms" left and right to reach neighbors


## Algorithm for Hamiltonian Graphs



- Stack blocks for vertices in order of the Hamiltonian cycle
- Extend "arms" left and right to reach neighbors


## Algorithm for Hamiltonian Graphs



- Stack blocks for vertices in order of the Hamiltonian cycle
- Extend "arms" left and right to reach neighbors


## Algorithm for Hamiltonian Graphs



- Stack blocks for vertices in order of the Hamiltonian cycle
- Extend "arms" left and right to reach neighbors


## Algorithm for Hamiltonian Graphs



- Stack blocks for vertices in order of the Hamiltonian cycle
- Extend "arms" left and right to reach neighbors


## Algorithm for Hamiltonian Graphs



- Stack blocks for vertices in order of the Hamiltonian cycle
- Extend "arms" left and right to reach neighbors


## Algorithm for Hamiltonian Graphs



- Stack blocks for vertices in order of the Hamiltonian cycle
- Extend "arms" left and right to reach neighbors


## Algorithm for Hamiltonian Graphs



- Stack blocks for vertices in order of the Hamiltonian cycle
- Extend "arms" left and right to reach neighbors
- Horizontal sweep line pass to realize correct areas


## Lower Bound for Hamiltonian Graphs

Complexity 8 is sometimes necessary:

## Lower Bound for Hamiltonian Graphs

Complexity 8 is sometimes necessary:


## Lower Bound for Hamiltonian Graphs

Complexity 8 is sometimes necessary:


Optimal polygonal complexity 8 , optimal $O(n)$ time computation [Alam et al. SoCG 2012]

## Algorithm for Planar 3-Trees

Planar 3-trees: either a 3-cyle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3-tree

- proportional rectilinear 8-sided representation


## Algorithm for Planar 3-Trees

Planar 3-trees: either a 3-cyle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3 -tree

- proportional rectilinear 8-sided representation
- linear time exact computation for max planar 3-trees



## Algorithm for Planar 3-Trees

Planar 3-trees: either a 3-cyle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3 -tree

- proportional rectilinear 8-sided representation
- linear time exact computation for max planar 3-trees



## Algorithm for Planar 3-Trees

Planar 3-trees: either a 3-cyle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3 -tree

- proportional rectilinear 8-sided representation
- linear time exact computation for max planar 3-trees



## Algorithm for Planar 3-Trees

Planar 3-trees: either a 3-cyle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3 -tree

- proportional rectilinear 8-sided representation
- linear time exact computation for max planar 3-trees



## Algorithm for Planar 3-Trees

Planar 3-trees: either a 3-cyle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3 -tree

- proportional rectilinear 8-sided representation
- linear time exact computation for max planar 3-trees



## Algorithm for Planar 3-Trees

Planar 3-trees: either a 3-cyle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3 -tree

- proportional rectilinear 8 -sided representation
- linear time exact computation for max planar 3-trees



## Algorithm for Planar 3-Trees

Planar 3-trees: either a 3-cyle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3 -tree

- proportional rectilinear 8-sided representation
- linear time exact computation for max planar 3-trees



## Algorithm for Planar 3-Trees

Planar 3-trees: either a 3-cyle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3 -tree

- proportional rectilinear 8 -sided representation
- linear time exact computation for max planar 3-trees



## Algorithm for Planar 3-Trees

Planar 3-trees: either a 3-cyle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3 -tree

- proportional rectilinear 8 -sided representation
- linear time exact computation for max planar 3-trees



## Algorithm for Planar 3-Trees

Planar 3-trees: either a 3-cyle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3 -tree

- proportional rectilinear 8 -sided representation
- linear time exact computation for max planar 3-trees

- Matches lower bound complexity of 8


## Algorithm for Planar 3-Trees

Planar 3-trees: either a 3-cyle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3 -tree

- proportional rectilinear 8 -sided representation
- linear time exact computation for max planar 3-trees

- Matches lower bound complexity of 8
- Optimal polygonal complexity 8, optimal $O(n)$ time computation [Alam et al. ISAAC 2011]


## Summary of Results: Contact Representations

For side contact representation, without weights


## Summary of Results: Contact Representations

For side contact representation, without weights


- T3G: outerplanar graphs


## Summary of Results: Contact Representations

For side contact representation, without weights


- T3G: outerplanar graphs
- T4G: planar graphs without filled triangle [Kant 1996]


## Summary of Results: Contact Representations

For side contact representation, without weights


- T3G: outerplanar graphs
- T4G: planar graphs without filled triangle [Kant 1996]
- T5G: Hamiltonian planar graphs [Ueckerdt 2011]


## Summary of Results: Contact Representations

For side contact representation, without weights


- T3G: outerplanar graphs
- T4G: planar graphs without filled triangle [Kant 1996]
- T5G: Hamiltonian planar graphs [Ueckerdt 2011]
- T6G: all planar graphs


## Summary of Results: Contact Representations

For side contact representation, without weights


- T3G: outerplanar graphs
- T4G: planar graphs without filled triangle [Kant 1996]
- T5G: Hamiltonian planar graphs [Ueckerdt 2011]
- T6G: all planar graphs

However, only two are complete characterizations!

## Summary of Results: Rectilinear Representations

For rectilinear proportional side contact representation:


## Summary of Results: Rectilinear Representations

For rectilinear proportional side contact representation:


| Class of Graphs | LB | UB | Time |
| :---: | :---: | :---: | :---: |
| Maximal Planar Graphs | $[8]$ | $8^{*}$ | $?$ |
| Maximal Planar Graphs | $[8]$ | $10^{*}$ | $O(n)$ |
| Planar 3-Trees | 8 | 8 | $O(n)$ |
| Hamiltonian Max-Planar Graphs | 8 | 8 | $O(n)$ |
| Maximal Outerplanar Graphs | $4 / 6^{* *}$ | $4 / 6^{* *}$ | $O(n)$ |

* Existential proof for 8, linear-time algorithm for 10.
** Complexity 6 if representation fits into a rectangle; else 4.


## Contacts in 3D

- A planar graph has a representation using axis-parallel boxes in 3D, where two boxes have a non-empty intersection iff their corresponding vertices are adjacent. It holds with contacts rather than intersections. [Thomassen 1986]
- A planar graph has a representation using axis-parallel cubes in 3D, where two boxes touch iff their corresponding vertices are adjacent.
[Felsner and Francis 2011]



## More 3D Results

We study proper contact representations by boxes: generalization from 2D side contact to 3D face contact where touching cubes have non-trivial-area face overlap


- Deciding unit cube proper contact is NP-Complete
- Planar 3-trees have proper cube contact representation
- Two new proofs of Thomassen's proper box contact
[Bremner, Evans, Frati, Heyer, K., Lenhart, Liotta, Rappaport, Whitesides, GD'2012]


## Box Representation via FPP

## Theorem

(Thomassen) Planar graphs have touching boxes contact representation.

(a)

(b)



## Cube Representation for Planar 3-Trees

## Theorem

Every (partial) planar 3-tree has a proper contact representation by cubes.

Recall planar 3-trees: either a 3-cycle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3-tree


## Cube Representation for Planar 3-Trees

## Theorem

Every (partial) planar 3-tree has a proper contact representation by cubes.

Recall planar 3-trees: either a 3-cycle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3-tree


## Cube Representation for Planar 3-Trees

## Theorem

Every (partial) planar 3-tree has a proper contact representation by cubes.

Recall planar 3-trees: either a 3-cycle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3-tree


## Cube Representation for Planar 3-Trees

## Theorem

Every (partial) planar 3-tree has a proper contact representation by cubes.

Recall planar 3-trees: either a 3-cycle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3-tree


## Cube Representation for Planar 3-Trees

## Theorem

Every (partial) planar 3-tree has a proper contact representation by cubes.

Recall planar 3-trees: either a 3-cycle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3-tree


## Cube Representation for Planar 3-Trees

## Theorem

Every (partial) planar 3-tree has a proper contact representation by cubes.

Recall planar 3-trees: either a 3-cycle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3-tree


## Cube Representation for Planar 3-Trees

## Theorem

Every (partial) planar 3-tree has a proper contact representation by cubes.

Recall planar 3-trees: either a 3-cycle or a planar graph $G$ with vertex $v$, s.t., $\operatorname{deg}(v)=3$ and $G-v$ is a planar 3-tree


## Unit Cube Graphs

- Grid graphs can be represented by unit cubes: square, triangle, pentagonal, hexagonal, parabolic
- Not clear for subgraphs thereof...
- Deciding unit cube proper contact is NP-Complete: logic engine reduction [Eades and Whitesides 1996]



## Proportional Box Representation

Two straightforward theorems:

## Theorem

Every internally triangulated 4-connected planar graph has a proper proportional contact representation with boxes.

Use 2D rectangle contact rep.; "grow" in 3D to get volumes.

## Theorem

Every (partial) planar 3-tree has a proper proportional contact representation with boxes.


## Proportional Box Representation

## Theorem

Every (partial) planar 3-tree has a proper proportional contact representation with boxes.

## Proof.

- compute representative tree $T$ for $G$ (aka 4-block tree)
- $U_{v}$ : the set of the descendants of $v$ in $T_{G}$ including $v$.
- predecessors of $v$ are $N_{G}(v)$ that are not in $U_{v}$
- scale weights so that $w(v) \geq 1 \forall v \in G$
- let $v_{1}, v_{2}$ and $v_{3}$ be the three children of $v$ in $T_{G}$
- define $W(v)=\Pi_{i=1}^{3}\left[W\left(v_{i}\right)+\sqrt[3]{W(v)}\right]$
- compute bottom up


## Future Work and Open Problems

- Contact Representations
- recognition algorithms
- characterizations
- 3D proper cube contact
- Proportional contact representations
- polytime T8G algorithm?
- 4-connected planar graphs: T6G or T8G?
- 3D proper proportional box contact
- Tradeoffs
- complexity vs convexity
- complexity vs holes
- overall vs max complexity



## Acknowledgments

- Colleagues
- Stefan Felsner, Torsten Ueckerdt, TU Berlin
- Michael Kaufmann, Andreas Gerasch, Tübingen
- Yifan Hu, Emden Gansner, AT\&T
- Therese Biedl, Univ. Waterloo
- Jawaherul Alam, Univ. Arizona

- Daisuke Mashima, Georgria Tech
- Workshops
- Dagstuhl
- Bertinoro
- Barbados

- Funding
- NSF
- ONR
- Humboldt

D. Knuth

Graph drawing is the best possible field I can think of. It merges aesthetics, mathematical beauty and wonderful algorithms. It therefore provides a harmonic balance between the left and right brain parts.


## Segment Contact Graphs

- Planar bipartite graphs (axis-aligned segments) [de Fraysseix, de Mendez, Pach 1991]
- Four-connected 3-colorable planar graphs [de Fraysseix, de Mendez 2007]
- Triangle-free planar graphs (only three slopes) [Castro et al. 2002]
- Planar Laman graphs (arbitrary number of slopes) [Alam, Biedl, Felsner, Kaufmann, K., GD'11]



## Laman Graphs

Laman graphs: n-vertex connected graph with $2 n-3$ edges and every $k$-vertex subgraph has at most $2 k-3$ edges.

- minimally rigid; not all planar
- mechanics, robotics, chemistry



## Planar Laman graphs

- series-parallel graphs, outer-planar graphs, planar 2-trees
- graphs that can be drawn as pointed pseudotriangulations [Rote et al. 2005]


## Henneberg Construction

Laman and planar Laman graphs can be labeled $v_{1}, v_{2}, \ldots v_{n}$ such that $G_{3}$ is a triangle and from $G_{i-1}$ we obtain $G_{i}$ via two operation (aka, Henneberg construction):

- let $x, y \in G_{i-1}$ : add $v_{i}$ together with the edges $\left(v_{i}, x\right)$ and $\left(v_{i}, y\right)$.
- let $(x, y) \in G_{i-1}$ and $z \in G_{i-1}$ : remove $(x, y)$ and add $v_{i}$ together with the three edges $\left(v_{i}, x\right),\left(v_{i}, y\right)$, and $\left(v_{i}, z\right)$.



## Laman Graphs and Tree Cover

- recall, maximally planar graphs can be decomposed into 3 edge-disjoint spanning trees $(|V|=n,|E|=3 n-6)$
- planar Laman graphs can be decomposed into 2 edge-disjoint spanning trees $(|V|=n,|E|=2 n-3)$



## Angle Labeling

Planar Laman Graphs have an angle labeling such that:
Vertex rule: Around $v \neq v_{1}, v_{2}$ we have: exactly one angle labeled 3 , zero or more angles labeled 2 , exactly one angle labeled 4, zero or more angles labeled 1. All angles at $v_{1}$ are labeled 1 , all angles at $v_{2}$ are labeled 2 .


Face rule: Around every face we have exactly one angle labeled 1 , zero or more angles labeled 3, exactly one angle labeled 2 , zero or more angles labeled 4.

## Edge Labeling

Planar Laman Graphs have an edge labeling such that:
Vertex rule: Around $v \neq v_{1}, v_{2}$, we have: exactly one outgoing red edge, zero or more incoming blue edges, zero or more incoming red edges, exactly one outgoing blue edge, zero or more incoming red edges, and zero or more incoming blue edges. At $v_{1}$ : incoming and red; at $v_{2}$ incoming and blue.


Face rule: For every inner face $f$ there exist red sink $r$ and a blue sink $b$ : every red edge on $f$ is directed from $b$ towards $r$, and every blue edge is directed from $r$ towards $b$.

## Computing the Edge Labeling

## Theorem

Given n-vertex planar Laman graph G, a red-blue edge labeling can be computed in $O\left(n^{2}\right)$ time.

## Proof.

Compute angle graph $A_{G}$ (vertices and faces become vertices, edges $\mathrm{b} / \mathrm{n}$ adjacent face-vertex pairs). Then extract an angular tree $T$ from $A_{G}$.
An angular tree of a 2-connected plane graph $G$ with special edge $\left(v_{1}, v_{2}\right)$ is a set $T$ of edges of $A_{G}$ such that:
Vertex rule: Every vertex $v \neq\left\{v_{1}, v_{2}\right\}$ of $G$ has exactly 2 incident edges in $T$.
Face rule: Every face of $G$ has exactly 2 incident edges not in $T$.

## Computing the Edge Labeling, cont.

## Proof.

Build Laman graph $G$ and angular tree $T$ simultaneously, following the Henneberg construction. The Laman construction requires $O\left(n^{2}\right)$ using [Bereg, SoCG 2005] and angular tree can also be constructed without adding much to the complexity. Use angular tree $T$ to compute angle labeling and red-blue edge labeling for $G$.


## Red-Blue Laman Realizer

## Theorem

A red-blue edge labeling $\left(E_{r}, E_{b}\right)$ of a 2-connected plane graph $G$ has the following two properties:
(1) The graph $E_{r} \cup E_{b}^{-1}\left(E_{b} \cup E_{r}^{-1}\right)$ is acyclic;
(2) The graph $E_{r}\left(E_{b}\right)$ is a spanning tree of $G \backslash\left\{v_{2}\right\}\left(G \backslash\left\{v_{1}\right\}\right)$ with all edges directed towards $v_{1}\left(v_{2}\right)$.

Use combinatorial structures to do some geometry...


## L-Contact Graphs

## Definition

A graph $G$ is an $L$-contact graph if there exist non-crossing L-shapes $\mathcal{L}(v)$ for each $v \in V$, such that $\mathcal{L}(u)$ and $\mathcal{L}(v)$ make contact if and only if $(u, v) \in E$.

- match edges of L-contact graphs to endpoints of L-shapes.
- extreme endpoints (N,E,S,W) cannot correspond to edges.

(a) L types
(b) valid contacts
(c) invalid contacts


## L-Contact Graphs

## Definition

An L-contact representation is maximal if every non-extreme endpoint makes a contact, and there are at most three endpoints that do not make a contact.
A maximal L-contact representation is proper if every inner face contains the right angle of exactly one $\mathcal{L}$. An L-contact graph is proper if it has a proper L-contact representation.


## Characterization of L-Contact Graphs

## Theorem

Plane Laman graphs are precisely proper L-contact graphs.

## Proof.

- construct the angular tree
- use it to construct angle labeling
- use that to construct edge labeling
- assign type to each vertex (I, II, III, IV)



## Creating L-Contact Graphs

## Theorem

An L-contact representation of a n-vertex planar Laman graph $G$ can be computed on an $n \times n$ grid in $\mathcal{O}\left(n^{2}\right)$ time, where $n$ is the number of vertices of $G$.

[Kobourov et al., SODA'13]

