Proportional Contact Representation of Planar Graphs

Stephen Kobourov
University of Arizona
Proportional Contact Representation of Planar Graphs

Stephen Kobourov
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
- ...

[Diagram showing various contact representations]
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 polygons, side-contacts, hole-free
proportional representation: a.k.a. *rectilinear cartogram*
**Goal:** represent the graph with simple polygons
**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)
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 *polygonsal 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
Data Representation

- Redrawing of geographic maps
- Population cartograms
- contact with circles [Koebe, 1936]
- contact with triangles [de Fraysseix et al., 1994]
- contact with 3D cubes [Felsner and Francis, 2011].
Point contact representation of both the primal and the dual graph with triangles [Gonçalves et al. 2010].
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].
Rectilinear Duals: Rectangles are sufficient for 4-connected maximal planar graphs [Ungar 1953, Kozminski and Kinnen 1985, Kant and He, 1997]
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]
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]
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]
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]
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
Ordering of the vertices

$G_i$ (induced on vertices 1, . . . , i) is biconnected
Canonical Order

- Ordering of the vertices
- $G_i$ (induced on vertices 1, ..., $i$) is biconnected
Canonical Order

- Ordering of the vertices
- $G_i$ (induced on vertices 1, \ldots, $i$) is biconnected
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 $C_i$
partition of the internal edges into three spanning trees

vertex rule: ccw order of edges: entering $T_1$, leaving $T_2$, entering $T_3$, leaving $T_1$, entering $T_2$, leaving $T_3$
partition of the internal edges into three spanning trees
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
3 edge-disjoint spanning trees $T_1, T_2, T_3$ cover $G$
3 edge-disjoint spanning trees $T_1$, $T_2$, $T_3$ cover $G$

$T_1$, $T_2$, $T_3$ rooted at external vertices of $G$
When a new vertex is inserted in the canonical order:

1. The leftmost edge is outgoing blue.
2. The rightmost edge is outgoing green.
3. The remaining (0 or more) edges are incoming red.

When the red edge gets closed off, it gets its outgoing red color.

Diagram showing a new vertex (i+1) being inserted into the canonical order, with edges colored blue, green, and red, and vertices labeled 1, 2, and G_i, C_i.
When a new vertex is inserted in the canonical order:

- leftmost edge is **outgoing blue**
When a new vertex is inserted in the canonical order:

- leftmost edge is **outgoing blue**
- rightmost edge is **outgoing green**
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
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”)
Two easy ways:

- counterclockwise preorder traversal of the blue tree
Two easy ways:

- counterclockwise preorder traversal of the blue tree
- topological order of $T_1 \cup T_2^{-1} \cup T_3^{-1}$
Outerplanar graphs are $T_3G$'s [Kobourov et al. GD 2010]
Outerplanar graphs are T3G's\cite{Kobourov et al. GD 2010}.
Outerplanar graphs are T3G’s [Kobourov et al. GD 2010]
Outerplanar graphs are $T_3G$'s $[\text{Kobourov et al. GD 2010}]$.
Outerplanar graphs are T3G's [Kobourov et al. GD 2010]
Outerplanar graphs are T3G’s [Kobourov et al. GD 2010]
Outerplanar graphs are T3G's [Kobourov et al. GD 2010]
Outerplanar: Contact Representation

Outerplanar graphs are $T_3G$'s [Kobourov et al. GD 2010]
Outerplanar graphs are T3G’s.
Outerplanar graphs are T3G’s

[Kobourov et al. GD 2010]
Outerplanar: Proportional Contact Representation

OP graphs are T3Gs; proportional OP graphs are T4Gs (proportional T3G but with $O(n)$ complexity outerface)

[ Alam et al. GD 2011]
Outerplanar: Proportional Contact Representation

OP graphs are T3Gs; proportional OP graphs are T4Gs (proportional T3G but with $O(n)$ complexity outerface) [Alam et al. GD 2011]
Outerplanar: Proportional Contact Representation

OP graphs are T3Gs; proportional OP graphs are T4Gs (proportional T3G but with $O(n)$ complexity outerface)

[Alam et al. GD 2011]
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]
Outerplanar: Proportional Contact Representation

OP graphs are T3Gs; proportional OP graphs are T4Gs (proportional T3G but with $O(n)$ complexity outerface).

[Alam et al. GD 2011]
Outerplanar: Proportional Contact Representation

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

General planar graphs are T6G's. Convex, max complexity 6, no holes [Duncan et al. Algorithmica 2011]
General planar graphs are T6G's Convex, max complexity 6, no holes [Duncan et al. Algorithmica 2011]
General planar graphs are T6G's. Convex, max complexity 6, no holes [Duncan et al. Algorithmica 2011]
Contact Representation of Maximal Planar Graphs

General planar graphs are $T_6$G's, convex, max complexity 6, no holes.

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

[Duncan et al. Algorithmica 2011]
We showed how to do it with 6 sides; 6 is also necessary:
We showed how to do it with 6 sides; 6 is also necessary:
We showed how to do it with 6 sides; 6 is also necessary:

- add a vertex in each internal face and triangulate
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...
compute a *canonical order* and *Schnyder realizer*
compute a *canonical order* and *Schnyder realizer*
represent each vertex by a canonical 8-sided polygon
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

How can we realize the areas?
How can we realize the areas?
Cartograms for Maximal Planar Graphs

[Diagram of a maximal planar graph with vertices labeled 1 to 9 and segments labeled T1 to T9]
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]
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.
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!
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

- Stack blocks for vertices in order of the Hamiltonian cycle
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
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
Complexity 8 is sometimes necessary:
Complexity 8 is sometimes necessary:
Complexity 8 is sometimes necessary:

Optimal polygonal complexity 8, optimal $O(n)$ time computation
[Alam et al. SoCG 2012]
Planar 3-trees: either a 3-cycle or a planar graph $G$ with vertex $v$, s.t., $\text{deg}(v) = 3$ and $G - v$ is a planar 3-tree

- proportional rectilinear 8-sided representation
Algorithm for Planar 3-Trees

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

- proportional rectilinear 8-sided representation
- linear time exact computation for max planar 3-trees
Planar 3-trees: either a 3-cycle or a planar graph $G$ with vertex $v$, s.t., $\text{deg}(v) = 3$ and $G - v$ is a planar 3-tree

- proportional rectilinear 8-sided representation
- linear time exact computation for max planar 3-trees
Planar 3-trees: either a 3-cyle or a planar graph $G$ with vertex $v$, s.t., $\text{deg}(v) = 3$ 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-cycle or a planar graph $G$ with vertex $v$, s.t., $\text{deg}(v) = 3$ and $G - v$ is a planar 3-tree

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

![Diagram of a planar 3-tree](image)
Algorithm for Planar 3-Trees

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

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

[Image of a graph and a 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., $\text{deg}(v) = 3$ and $G - v$ is a planar 3-tree

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

[Diagram of a planar 3-tree with labeled vertices and a proportional rectilinear 8-sided representation]
Algorithm for Planar 3-Trees

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

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

[Diagram of a planar 3-tree]
Planar 3-trees: either a 3-cycle or a planar graph $G$ with vertex $v$, s.t., $\text{deg}(v) = 3$ 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., $\text{deg}(v) = 3$ 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-cycle or a planar graph \( G \) with vertex \( v \), s.t., \( \text{deg}(v) = 3 \) 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

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!
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!
For rectilinear proportional side contact representation:

<table>
<thead>
<tr>
<th>Class of Graphs</th>
<th>LB</th>
<th>UB</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximal Planar Graphs</td>
<td>[8]</td>
<td>[10]</td>
<td>*O(n)</td>
</tr>
<tr>
<td>Planar 3-Trees</td>
<td>8</td>
<td>8</td>
<td>*O(n)</td>
</tr>
<tr>
<td>Hamiltonian Max-Planar</td>
<td>8</td>
<td>8</td>
<td>*O(n)</td>
</tr>
<tr>
<td>Graphs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximal Outerplanar</td>
<td>4/6</td>
<td>4/6</td>
<td>*O(n)</td>
</tr>
</tbody>
</table>

Existential proof for 8, linear-time algorithm for 10. \*Complexity 6 if representation fits into a rectangle; else 4.
Summary of Results: Rectilinear Representations

For rectilinear proportional side contact representation:

<table>
<thead>
<tr>
<th>Class of Graphs</th>
<th>LB</th>
<th>UB</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximal Planar Graphs</td>
<td>[8]</td>
<td>8*</td>
<td>?</td>
</tr>
<tr>
<td>Maximal Planar Graphs</td>
<td>[8]</td>
<td>10*</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Planar 3-Trees</td>
<td>8</td>
<td>8</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Hamiltonian Max-Planar Graphs</td>
<td>8</td>
<td>8</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Maximal Outerplanar Graphs</td>
<td>4/6*</td>
<td>4/6*</td>
<td>$O(n)$</td>
</tr>
</tbody>
</table>

* 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]
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]
Theorem

(Thomassen) Planar graphs have touching boxes contact representation.
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., $\text{deg}(v) = 3$ and $G - v$ is a planar 3-tree
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., $\deg(v) = 3$ and $G - v$ is a planar 3-tree.
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., $\text{deg}(v) = 3$ and $G - v$ is a planar 3-tree.
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., $deg(v) = 3$ 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., $\text{deg}(v) = 3$ 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., $\deg(v) = 3$ and $G - v$ is a planar 3-tree.
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., $\text{deg}(v) = 3$ and $G - v$ is a planar 3-tree.
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]
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.*
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 $W(v) = \prod_{i=1}^{3} [W(v_i) + \sqrt[3]{w(v)}]$
- 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, Georgia 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]

![Diagram of segment contact graphs with labeled vertices and edges.](image)
Laman graphs: $n$-vertex connected graph with $2n - 3$ edges and every $k$-vertex subgraph has at most $2k - 3$ 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]
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 operations (aka, Henneberg construction):

- let \( x, y \in G_{i-1} \): add \( v_i \) together with the edges \((v_i, x)\) and \((v_i, y)\).
- 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 \((v_i, x), (v_i, y),\) and \((v_i, z)\).
recall, maximally planar graphs can be decomposed into 3 edge-disjoint spanning trees ($|V| = n, |E| = 3n - 6$)

planar Laman graphs can be decomposed into 2 edge-disjoint spanning trees ($|V| = n, |E| = 2n - 3$)
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.
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 $O(n^2)$ time.

**Proof.**

Compute **angle graph** $A_G$ (vertices and faces become vertices, edges b/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 $(v_1, v_2)$ is a set $T$ of edges of $A_G$ such that:

- **Vertex rule:** Every vertex $v \neq \{v_1, v_2\}$ 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 $O(n^2)$ 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 $G$. 
Theorem

A red-blue edge labeling \((E_r, E_b)\) of a 2-connected plane graph \(G\) has the following two properties:

1. The graph \(E_r \cup E_b^{-1} (E_b \cup E_r^{-1})\) is acyclic;
2. The graph \(E_r (E_b)\) is a spanning tree of \(G \setminus \{v_2\}\) \((G \setminus \{v_1\})\) with all edges directed towards \(v_1\) \((v_2)\).

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 $L(v)$ for each $v \in V$, such that $L(u)$ and $L(v)$ 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
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.
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 $\mathcal{O}(n^2)$ time, where $n$ is the number of vertices of $G$.

[Kobourov et al., SODA’13]