A Schnyder-type drawing algorithm for 5-connected triangulations

Olivier Bernardi\textsuperscript{1}, Éric Fusy\textsuperscript{2} and Shizhe Liang\textsuperscript{1}

1. Dept. of Math, Brandeis University
2. LIGM/CNRS, Université Gustave Eiffel
Triangulations

Triangulation = graph embedded in the plane, all faces of degree 3

= embedded maximal planar graph
Triangulations

Triangulation = graph embedded in the plane, all faces of degree 3

= embedded maximal planar graph

For any graph (def):
  $k$-connected: deleting any subset of $< k$ vertices does not disconnect
**Triangulations**

Triangulation = graph embedded in the plane, all faces of degree 3

\[\begin{array}{c}
\text{3-connected} \\ \Leftrightarrow \\ \text{simple}
\end{array} \quad \text{and} \quad \begin{array}{c}
\text{4-connected} \\ \Leftrightarrow \\ \text{no separating 3-cycle}
\end{array} \quad \begin{array}{c}
\text{5-connected} \\ \Leftrightarrow \\ \text{no separating 4-cycle}
\end{array}\]

For any graph (def):

* \(k\)-connected: deleting any subset of \(< k\) vertices does not disconnect

For triangulations:

\[\begin{array}{c}
\text{embedded maximal planar graph}
\end{array}\]
**Triangulations**

Triangulation = graph embedded in the plane, all faces of degree 3

= embedded maximal planar graph

4-connected

not 5-connected

For any graph (def):

$k$-connected: deleting any subset of $< k$ vertices does not disconnect

For triangulations:

3-connected $\iff$ simple

4-connected $\iff$ no separating 3-cycle

5-connected $\iff$ no separating 4-cycle
3-connected case
Schnyder structures on simple triangulations

Any triangulation admits a **labeling** of corners by \(\{1, 2, 3\}\) satisfying
Schnyder structures on simple triangulations

[Yields 3 spanning trees $T_1, T_2, T_3$ (Schnyder wood)]

3 types of edges

3 types of edges

inner vertex
Schnyder’s face-counting algorithm [Schnyder’90]
Schnyder’s face-counting algorithm

Tree-paths from $v$ partition inner faces into 3 regions $R_1(v)$, $R_2(v)$, $R_3(v)$
Schnyder’s face-counting algorithm

Tree-paths from $v$ partition inner faces into
3 regions $R_1(v)$, $R_2(v)$, $R_3(v)$

In $R_1(v)$:
- 3 faces

In $R_2(v)$:
- 4 faces

In $R_3(v)$:
- 2 faces
Schnyder’s face-counting algorithm

Tree-paths from $v$ partition inner faces into
3 regions $R_1(v)$, $R_2(v)$, $R_3(v)$

2 faces in $R_3(v)$
3 faces in $R_1(v)$
4 faces in $R_2(v)$

Place $v$ at
\[ \frac{3}{9}v_1 + \frac{4}{9}v_2 + \frac{2}{9}v_3 \]
Schnyder’s face-counting algorithm [Schnyder’90]
Schnyder’s face-counting algorithm

planar straight-line drawing
Schnyder’s face-counting algorithm

[\text{Schnyder’90}]

planar straight-line drawing

cone-property
(implies planarity)
Schnyder’s face-counting algorithm

Planar straight-line drawing

Cone-property (implies planarity)

Grid-drawing

Sheer
4-connected case
Triangulations of the 4-gon, irreducibility

A triangulation of the 4-gon is **irreducible** if all 3-cycles bound faces.

The left graph shows a non-irreducible triangulation, while the right graph illustrates an irreducible one.
Triangulations of the 4-gon, irreducibility

A triangulation of the 4-gon is **irreducible** if all 3-cycles bound faces.

A triangulation augmented by $v_\infty$ is 4-connected.
Transversal structures

aka regular edge-labelings
(structures dual to rectangular tilings)

A 4-triangulation admits a transversal structure iff it is irreducible

inner vertices

outer vertices

local conditions

[He'93]
Transversal structures

aka regular edge-labelings
(structures dual to rectangular tilings)

inner vertices

outer vertices

local conditions

A 4-triangulation admits a transversal structure iff it is irreducible

yields two bipolar orientations:
Face-counting algorithm
Face-counting algorithm
Face-counting algorithm

10 faces in red map

9 faces in blue map

10 × 9 grid
Face-counting algorithm

[F’05]
Face-counting algorithm

leftmost outgoing red path
Face-counting algorithm

leftmost outgoing red path
+ rightmost ingoing red path
Face-counting algorithm

leftmost outgoing red path
+ rightmost ingoing red path
Face-counting algorithm

leftmost outgoing red path
+ rightmost ingoing red path

leftmost outgoing blue path
Face-counting algorithm

leftmost outgoing red path + rightmost ingoing red path

leftmost outgoing blue path + rightmost ingoing blue path
Face-counting algorithm
Face-counting algorithm

leftmost outgoing red path
+ rightmost ingoing red path

leftmost outgoing blue path
+ rightmost ingoing blue path
Face-counting algorithm
Face-counting algorithm

planar straight-line drawing
Face-counting algorithm

planar straight-line drawing

cone property (implies planarity)
Face-counting algorithm on square grid
Face-counting algorithm on square grid

18 inner faces

18 × 18 grid
Face-counting algorithm on square grid

6 faces on the left

12 faces below

$y = 12$

$x = 6$
Face-counting algorithm on square grid
Face-counting algorithm on square grid
4-wood associated to transversal structure
4-wood associated to transversal structure

yields 4 regions for each vertex $v$

$R_4(v)$

$R_3(v)$

$R_2(v)$

$R_1(v)$

right incoming red edges

right incoming blue edges

left outgoing blue edges

left outgoing red edges
4-wood associated to transversal structure

yields 4 regions for each vertex $v$

$R_4(v)$

$R_3(v)$

$R_2(v)$

$R_1(v)$

square-grid algo $\updownarrow$ barycentric placement

$(\text{place } v \text{ at } \frac{4}{28} v_1 + \frac{8}{28} v_2 + \frac{4}{28} v_3 + \frac{2}{28} v_4)$

right incoming red edges

$T_1$

$T_4$

right incoming blue edges

$T_2$

left outgoing blue edges

$T_3$

left outgoing red edges
4-labeling associated to transversal structure
4-labeling associated to transversal structure
4-labeling associated to transversal structure
4-labeling associated to transversal structure

vertices

edges

faces

\( j = i + 2 \)

\( i \to j \to i+1 \to i-1 \)
4-labeling associated to transversal structure

vertices

edges

faces

\[ j = i + 2 \]
4-labeling associated to transversal structure

Vertices: $v_1, v_2, v_3, v_4$

Edges: $e \in T_1, T_2, T_3, T_4$

Faces: $f \in \{i, i+1, i+2\}$

$\forall i \in \{1, 2, 3, 4\}: j = i + 2$
5-connected case
5c-triangulations

5c-triangulation = triangulation of 5-gon such that every cycle with at least one vertex inside has length $\geq 5$.

![Diagram]

- not 5c-triangulation
- 5c-triangulation
**5c-triangulations**

5c-triangulation = triangulation of 5-gon such that every cycle with at least one vertex inside has length $\geq 5$

---

**not 5c-triangulation**

---

**5c-triangulation**

\[ \uparrow \approx \]

triangulation augmented by $v_\infty$

is 5-connected
Any 5c-triangulation has a **labeling** of corners by \( \{1, 2, 3, 4, 5\} \) so that...
$5c$-woods

[ Bernardi, F, Liang ’23 ]

$i + 2 \in T_i$

$i + 3$
5c-woods
[Bernardi,F,Liang’23]
Other 5-woods (less restrictive) associated to pentagon-contact representations.

[Bernardi, F, Liang’23]

[Felsner, Schrezenmaier, Steiner’20]
Face-counting algorithm

[Bernardi, F, Liang’23]
Face-counting algorithm

[Bernardi, F, Liang'23]
Face-counting algorithm

[ Bernardi, F, Liang’23 ]

\[ R_5(v) \]
\[ R_4(v) \]
\[ R_3(v) \]
\[ R_2(v) \]
\[ R_1(v) \]
Face-counting algorithm

\[ R_5(v) \] 12 faces
\[ R_1(v) \] 4 faces
\[ R_2(v) \] 4 faces
\[ R_3(v) \] 7 faces
\[ R_4(v) \] 18 faces

45 inner faces in total

\[
\frac{4}{45}v_1 + \frac{4}{45}v_2 + \frac{18}{45}v_3 + \frac{7}{45}v_4 + \frac{12}{45}v_5
\]

place \( v \) at
Face-counting algorithm

[Bernardi, F, Liang’23]
Face-counting algorithm

\[ v_1, v_2, v_3, v_4, v_5 \]

cone property

1, 2, 3, 4, 5
Face-counting algorithm

[Bernardi,F,Liang’23]

cone property implies planarity

each face drawn as and not

implies planarity

each face drawn as and not
Face-counting algorithm

Bernardi, F., Liang’23

[v1, v2, v3, v4, v5]

cone property

Rk: Not a grid drawing

coordinates in $\mathbb{Q}(\sqrt{5})$
Properties and variations

- Linear time complexity
Properties and variations

- Linear time complexity
- Displays rotational symmetries

A: (2, 6, 4, 2, 1)
B: (1, 2, 6, 4, 2)
C: (2, 1, 2, 6, 4)
D: (4, 2, 1, 2, 6)
E: (6, 4, 2, 1, 2)
F: (3, 3, 3, 3, 3)
Properties and variations

- Linear time complexity
- Displays rotational symmetries
- Variations: weighted faces, vertex-counting

A: (2,6,4,2,1)
B: (1,2,6,4,2)
C: (2,1,2,6,4)
D: (4,2,1,2,6)
E: (6,4,2,1,2)
F: (3,3,3,3,3)
Properties and variations

- Linear time complexity
- Displays rotational symmetries
- Variations: weighted faces, vertex-counting
- Vertex resolution better than in the 3- or 4-connected drawings

![Diagram showing vertex resolutions and smallest distance between vertices](drawing_normalized_to_have_outer_k-gon_inscribed_in_circle_of_radius_1)

A: (2,6,4,2,1)
B: (1,2,6,4,2)
C: (2,1,2,6,4)
D: (4,2,1,2,6)
E: (6,4,2,1,2)
F: (3,3,3,3,3)

(drawing normalized to have outer $k$-gon inscribed in circle of radius 1)
Strategy for proof of existence
new proof of existence for 4-connected (from 3-connected)
Strategy for proof of existence
new proof of existence for 4-connected (from 3-connected)
Strategy for proof of existence

new proof of existence for 4-connected (from 3-connected)
Strategy for proof of existence

new proof of existence for 4-connected (from 3-connected)
Strategy for proof of existence
new proof of existence for 4-connected (from 3-connected)

Schnyder orientation

outdegree 3

outdegree 4

outdegree 1
Strategy for proof of existence

new proof of existence for 4-connected (from 3-connected)

Schnyder orientation

orientation is “co-accessible”

(∃ co-accessibility spanning tree)

no separating triangle
Strategy for proof of existence

new proof of existence for 4-connected (from 3-connected)

Schnyder orientation

local rule for orientation

outdegree 4

outdegree 1

outdegree 3

outdegree 3
Strategy for proof of existence

new proof of existence for 4-connected (from 3-connected)

Schnyder orientation

outdegree 3

local rule for orientation

outdegree 4

outdegree 1

similar proof of existence for 5-connected (from 4-connected)