Array-based compact data structures
for triangle meshes

Encuentros de Geometría Computacional, EGC2011

Luca Castelli Aleardi

Olivier Devillers
Buon Compleanno
Ferran
Buon Compleanno
Ferran
Compact Data Structures: motivation and goal

St. Matthew (Stanford’s Digital Michelangelo Project, 2000)

6 Giga bytes (for storing on disk)
186 millions vertices

David statue (Stanford’s Digital Michelangelo Project, 2000)

2 billions polygons
32 Giga bytes (without compression)

Connectivity information
1 reference to a triangle
3 references to vertices
3 references to triangles
log n or 32 bits

Geometric object

\[
\begin{pmatrix}
x \\
y \\
z
\end{pmatrix}
\]

between 30 et 96 bits/vertex
Surface triangle mesh endowed with a Schnyder wood

(Planar) Triangle meshes

Delaunay triangulation of random points (endowed with a Schnyder wood)

Tutte drawing of a random planar triangulation

Schnyder drawing of a planar triangulation

Schnyder woods for triangulations with multiple boundaries of arbitrary size
(Castelli-Fusy-Lewiner, CCCG2010)

g-Schnyder woods, for genus $g$ triangulations
(Castelli-Fusy-Lewiner, SoCG08)
Graph planarity characterizations

**Schnyder woods** (via dimension of partial orders)
- $\text{dim}(G) \leq 3$

**Kuratowski theorem** (excluded minors)
- $G$ contains neither $K_5$ nor $K_{3,3}$ as minors

**Thm (Koebe-Andreev-Thurston)**
*Every planar graph with $n$ vertices is isomorphic to the intersection graph of $n$ disks in the plane.*

**Colin de Verdière invariant** (multiplicity of $\lambda_2$ eigenvalue of a generalized laplacian)
- $\mu(G) \leq 3$
Schnyder woods: applications
(and related properties)

grid drawing

graph counting, random generation
(Poulalhon-Schaeffer, Icalp 03)

graph encoding
(Chuang-Garg-He-Kao-Lu Icalp ’98)
(He-Kao-Lu ’99)
(Chiang et al. Soda’01)
(Barbay-Castelli Aleardi-He-Munro Isaac’07)
(Castelli Aleardi-Fusy-Lewiner SoCG08)
(Castelli Aleardi-Fusy-Lewiner CCCG’10)
(Yamanaka-Nakano ’08)

Greedy routing

Untangling geometric graphs
Bose, Dujmovic, Hurtado, Langerman, Morin, Wood (DCG 2009)

fix(G) \geq \left( \frac{2}{3} \right)^{\frac{1}{2}}

Every planar triangulation admits a greedy drawing (Dhandapani, Soda08)
Conjectured by Papadimitriou and Ratajczak for 3-connected planar graphs
Schnyder woods: the definition

\begin{itemize}
\item[(a)] Outer face
\item[(b)] CW oriented triangle
\item[(c)]
\end{itemize}

\begin{enumerate}
\item[i)] Edges are colored and oriented in such a way that each inner node has exactly one outgoing edge of each color.
\item[ii)] Colors and orientations must respect the local Schnyder condition.
\end{enumerate}

**Theorem**
Every planar triangulation admits a Schnyder wood, which can be computed in linear time.

**Theorem**
The three set $T_0$, $T_1$, $T_2$ are spanning trees of (the inner nodes of) $T$: $v_0$, $v_1$, $v_2$. 
minimal Schnyder woods: the definition (no ccw triangles)

i) edge are colored and oriented in such a way that each inner node has exactly one outgoing edge of each color

ii) colors and orientations around each inner node must respect the local Schnyder condition
Schnyder woods and (planar) barycentric coordinates

Geometric interpretation of barycentric coordinates

$$v = \alpha_0 v_0 + \alpha_1 v_1 + \alpha_2 v_2$$

$$v = \frac{\text{area}(v,v_1,v_2)v_0 + \text{area}(v_0,v,v_2)v_1 + \text{area}(v_0,v_1,v)v_2}{\text{area}(v_0,v_1,v_2)}$$

Combinatorial interpretation

$$v = \frac{R_0}{2n-5}v_0 + \frac{R_1}{2n-5}v_1 + \frac{R_2}{2n-5}v_2$$

$$R_i(v) := \text{number of triangles in region } i$$

**Theorem (Schnyder, Soda '90)**

For a triangulation $\mathcal{T}$ having $n$ vertices, we can draw it on a grid of size $(2n-5) \times (2n-5)$, by setting $v_0 = (2n-5,0)$, $v_1 = (0,0)$ and $v_2 = (0,2n-5)$.
Several kinds of encodings: plane trees

\[ \|B_n\| = \frac{1}{n+1} \binom{2n}{n} \approx 2^{2n} n^{-\frac{3}{2}} \]

\[ \log_2 \|B_n\| = 2n + O(\lg n) \]

(Explicit) data structures

we need \( 2n \) references, or \( \Theta(n \lg n) \) bits

Succinct representations

\( 2n + o(n) \) bits

\( O(1) \) time navigation

(Jacobson Focs89, Munro and Raman Focs97)
Several kinds of encodings: triangle meshes

Optimal compression scheme
(Poulalhon Schaeffer, Icalp03)

\[ \Psi_n = \frac{2(4n+1)!}{(3n+2)!(n+1)!} \approx \frac{16}{27} \sqrt{\frac{3}{2\pi}} n^{-5/2} \left( \frac{256}{27} \right)^n \]

\[ \frac{1}{n} \log_2 \Psi_n \approx \log_2 \left( \frac{256}{27} \right) \approx 3.2451 \text{ bits/vertex} \]

\[ 3.2451n + O(n \frac{n \log \log n}{\log n}) = 3.2451n + o(n) \text{ bits} \]

(Explicit) Geometric data structures
\[ \beta n + O(1) \text{ references (pointers)} \]

\[ 2 \times n \times 6 \times \log n + n \times 1 \times \log n \]

Succinct representations
(Castelli Aleardi-Devillers-Schaeffer, WADS05, SoCG06)

\[ 13n \text{ references or } 13n \log n \text{ bits} \]
Popular mesh data structures

<table>
<thead>
<tr>
<th>Data structure</th>
<th>references</th>
<th>navigation</th>
<th>vertex access</th>
<th>dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge-based data structures [17, 3, 4]</td>
<td>18n + n</td>
<td>O(1)</td>
<td>O(1)</td>
<td>yes</td>
</tr>
<tr>
<td>triangle based [7] / Corner Table</td>
<td>12n + n</td>
<td>O(1)</td>
<td>O(1)</td>
<td>yes</td>
</tr>
<tr>
<td>Directed edge [9]</td>
<td>12n + n</td>
<td>O(1)</td>
<td>O(1)</td>
<td>yes</td>
</tr>
<tr>
<td>2D catalogs [10]</td>
<td>7.67n</td>
<td>O(1)</td>
<td>O(1)</td>
<td>yes</td>
</tr>
<tr>
<td>Star vertices [20]</td>
<td>7n</td>
<td>O(d)</td>
<td>O(1)</td>
<td>no</td>
</tr>
<tr>
<td>TRIPOD [25] + reordering / Thm 1</td>
<td>6n</td>
<td>O(1)</td>
<td>O(d)</td>
<td>no</td>
</tr>
<tr>
<td>SOT data structure [19]</td>
<td>6n</td>
<td>O(1)</td>
<td>O(d)</td>
<td>no</td>
</tr>
<tr>
<td>SQUAD data structure [18]</td>
<td>(4 + c)n</td>
<td>O(1)</td>
<td>O(d)</td>
<td>no</td>
</tr>
<tr>
<td>(no vertex reordering) Thm 2</td>
<td>5n</td>
<td>O(1)</td>
<td>O(d)</td>
<td>no</td>
</tr>
<tr>
<td>(with vertex reordering) Thm 3</td>
<td>4n</td>
<td>O(1)</td>
<td>O(d)</td>
<td>no</td>
</tr>
<tr>
<td>(with vertex reordering) Cor 3</td>
<td>6n</td>
<td>O(1)</td>
<td>O(d)</td>
<td>no</td>
</tr>
</tbody>
</table>

class Halfedge{
    Halfedge prev, next, opposite;
    Vertex source, target;
    Face f;
}

class Vertex{
    Halfedge e;
    Point p;
}

class Triangle{
    Triangle t1, t2, t3;
    Vertex v1, v2, v3;
}

class Point{
    float x;
    float y;
    float z;
}

Half-edge

Triangle-based
Popular mesh data structures: space requirements

**Half-edge**

```java
class Halfedge{
    Halfedge prev, next, opposite;
    Vertex source, target;
    Face f;
}
class Vertex{
    Halfedge e;
    Point p;
}
```

**Geometry**

- `target(e)`
- `opposite(e)`
- `prev(e)`
- `source(e)`

**Triangle-based**

```java
class Triangle{
    Triangle t1, t2, t3;
    Vertex v1, v2, v3;
}
class Vertex{
    Triangle root;
    Point p;
}
```

**Connectivity**

- `connectivity`

**Geometric coordinates**

- `geometry`

---

<table>
<thead>
<tr>
<th>Data Structure</th>
<th>Connectivity</th>
<th>Geometry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Half-edge</td>
<td>(19n + 3n)</td>
<td></td>
</tr>
<tr>
<td>Winged-edge</td>
<td>(13n + 3n)</td>
<td></td>
</tr>
<tr>
<td>Quad-edge</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Triangle DS, Corner Table</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Directed edge</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

- `class Point{
  float x;
  float y;
  float z;
  }
- `class Halfedge{
  Halfedge prev, next, opposite;
  Vertex source, target;
  Face f;
}
- `class Vertex{
  Halfedge e;
  Point p;
}
- `class Triangle{
  Triangle t1, t2, t3;
  Vertex v1, v2, v3;
}
- `class Vertex{
  Triangle root;
  Point p;
}`
Non compact vs. compact mesh data structures

- Half-edge, Winged-edge, Quad-edge
  \(19n\)

- Triangle DS, Corner Table, Directed edge
  \(7n, 67n, 6n\)

\(\approx 4.15\)

- Our results

- SQUAD (Garung et al. 2011) (no theoretical guarantees)

- Triangle-based DS

- Our compact DS with minimal Schnyder woods
Experimental comparison

Winged edge vs. Our Compact DS

1.2 - 1.9 times slower

Our Compact DS

1.19 - 1.35 times slower

Tested on 3D models and random planar triangulations

(vertex normals (navigation + geometric computations))

(timings are expressed in nanoseconds/vertex)

Winged edge
Thm1: Compact DS (6n references)  

Simple solution: redundant and "simple to implement"  

"sorted variant" of TRIPOD DS  
(Snoeyink-Speckmann, 1999)  

\[ e := (u, v) \quad 0 \leq v \leq n - 1 \]
\[ 0 \leq e \leq 3n \]

\[ (w, v) := T[e] \]
\[ (z, v) := T[2e + 1] \]

\[ u := \text{source}(e) = e / 3 \]
\[ (w, u) := \begin{cases} 
(e + 1) \% 3 \\
(T[e] + 2) \% 3 \\
T[T[e]] 
\end{cases} \]
Thm2: More compact DS (5n references)
more compact solution: less redundant and "more difficult to implement"
Thm2: More compact DS (5n references)
more compact solution: less redundant and ”more difficult
to implement”

long and tedious case analysis
for the the black edges
(similar case analysis for the other two colors)
Thm3: our most compact DS (4n references)
still more compact with vertex reordering
use the DFUDS (Depth First Unary Degree Sequence) order on $T_0$