Μηδείς άγεωμέτρητος είσίτω μου τήν στέγην (let no one ignorant of geometry come under my roof)

SCARST: Schnyder Compact and Regularity Sensitive Triangulation Data Structure

(when theory meets practice)



june 14, Athens (SoCG 2024)

Luca Castelli Aleardi

(Ecole Polytechnique, Palaiseau)







Olivier Devillers

(Loria, Inria, Nancy)





Main goals of this talk

either you do not know Schnyder woods
 I will make you discover the magic world of
 Schnyder woods

or you already encountered Schnyder woods
 I will convince you that Schnyder woods
 lead to practical and fast implementations
 (a C++ implementation is going to be
 integrated in CGAL... coming soon)







Let us review some major results on planar graphs

Kuratowski theorem (1930) (cfr Wagner's theorem, 1937)

• G contains neither K_5 nor $K_{3,3}$ as minors (or no subdivisions of K_5 nor $K_{3,3}$)





subdivision of $K_{3,3}$

Thm (Koebe-Andreev-Thurston) Every planar graph with n vertices is isomorphic to the intersection graph of n



Thm (Tutte barycentric method, 1963) Every 3-connected planar graph G admits a convex representation in R^2 .



Thm (Colin de Verdière, 1990) Colin de Verdiere invariant (multiplicity of λ_2 eigenvalue of a generalized laplacian) • $\mu(G) \leq 3$



Schnyder woods (Walter Schnyder '89)

- planarity criterion via dimension of partial orders: $dim(G) \leq 3$
- \bullet linear-time grid drawing, with $O(n) \times O(n)$ resolution



Schnyder woods: some (classical) applications

[Felsner, Bonichon et al. '10, ...]



(Chuang, Garg, He, Kao, Lu, Icalp'98) (He, Kao, Lu, 1999)

Graph encoding



Figure 2: A coplanar orthogonal surface with its geodesic emł





Figure 3: (a) TD-Voronoi diagram. (b) $\lambda_1 < \lambda_2 < \lambda_3$ stand for three triangular distances.

Schnyder woods, TD-Delaunay graphs, orthogonal surfaces and Half- Θ_6 -graphs

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



 \Rightarrow optimal encoding ≈ 3.24 bits/vertex

(Schnyder '90) Planar straight-line grid drawing (on a $O(n \times n)$ grid)



Schnyder woods: definition (for genus 0 triangulations)

Schnyder woods genus 0 (plane) triangulations: definition



local Schnyder condition



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

ii) colors and orientations around each inner node must respect the local Schnyder condition iii) inner edges incident to V_i are of color i and oriented toward V_i

Schnyder woods: global spanning property

Theorem [Schnyder '90]

The three sets T_0 , T_1 , T_2 are spanning trees of the inner vertices of \mathcal{T} (each rooted at vertex V_i)



Minimal (maximal) Schnyder woods: no cw (ccw) oriented circuit





minimal Schnyder wood: ccw-oriented faces are forbidden

Thm: [Ossona de Mendez'94, Felsner'03]

The set S(T) of all distinct Schnyder woods of a given triangulation T is a partial order set with respect to the flip operation (a lattice): for every pair of Schnyder woods of T there is an unique supremum and infinum.



The min is the unique $S \in \mathcal{S}(T)$ with **no clockwise circuit**

Data structures for (triangle) surface meshes





return cont;





```
vertexDegree(Flag f) {
    int j=0;
    Flag g=f;
    do {
        ++j;
        g=g.ei().fi();
    } while (g!=f);
    return j;
}
```

Problem: how to represent huge 3D meshes?

Geometric v.s combinatorial information

Geometry



vertex coordinates

between 30 et 96 bits/vertex

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



2 billions polygons32 Giga bytes (without compression)

"Connectivity": combinatorial information underlying triangulation (incidence relations between triangles, vertices, edges)



(19n references)

 $19n\log n$ or 608n bits

Memory efficient mesh representations

(standard) data structures



19n references = 608n bits



Memory efficient mesh representations



D. Knuth (Bordeaux, dec. 2007) "Dear Luca and Jeremy, if you want you that your algorithm and data structures will appear in my books, first please provide an implementation and check its performance." (D. Knuth) (standard) data structures



19n references = 608n bits



Practical mesh data structures: related works

		Data	size (references/vertex)			navigation	preserves		
		structure	lower	average	upper	navigation	target	vertex	
_			bound	(regular meshes)	bound	between edges	operator	ordering	
		Half-edge/Winged-edge/Quad-edge	19	19	19	O(1)	O(1)	yes	dynamic
non compact		Triangle DS (CGAL)/Corner Table [Rossignac]	13	13	13	O(1)	O(1)	yes	dynamic
		Directed edge [Campagna et al. '99]	13	13	13	O(1)	O(1)	yes	dynamic
		2D catalogs [Castelli,Devillers,Mebarki '06]	-	-	7.67	O(1)	O(1)	yes	dynamic
		Star vertices [Kallmann et al. '02]	7	7	7	$O(d^{\circ})$	O(1)	yes	
		sorted TRIPOD [Snoeyink, Speckmann, '99]	6	6	6	O(1)	$O(d^{\circ})$	yes	
		SOT [Gurung Rossignac '09]	6	6	6	O(1)	$O(d^{\circ})$	yes	
		ESQ [Castelli,Devillers,Rossignac '12]	4	-	4.8	O(1)	$O(d^{\circ})$	yes	dynamic
		SQUAD [Gurung et al. '10]	4	4.1	6	O(1)	$O(d^{\circ})$	yes	
compact		LR (Laced Ring) [Gurung et al. '11]	2	2.16	?	O(1)	O(1)	no	
		Thm 2 in [Castelli, Devillers '11] (no vertex reordering)	6	6	6	O(1)	$O(d^{\circ})$	yes	
		Thm 4 in [Castelli, Devillers '11] (no vertex reordering)	5	5	5	O(1)	$O(d^{\circ})$	yes	
L	-	Thm 5 in [Castelli, Devillers '11] (with vertex reordering)	4	4	4	O(1)	$O(d^{\circ})$	no	





LR: Laced Ring data structure LR (Gurung et al. 2011)



space performances in practice (results reported from original papers)

triangle classification







Ring-expander

c = s;	<pre>// start at the seed corner s</pre>						
c.n.v.m = c.p.v.m = true; // mark vertices as visited							
do {							
<pre>if (!c.v.m) c.v.m = c.t.m = true; // invade c.t else if (!c.t.m) c = c.o; // go back one triangle</pre>							
c = c.r; // advance	to next ring edge on the right						
<pre>} while (c != s.o);</pre>	<pre>// until back at the beginning</pre>						

$$size(LR) := 2v_r + v_i + 6|T_0| + 3|T_1^i + T_2^i$$

LR: Laced Ring data structure LR (Gurung et al. 2011)



meshes are sorted according to % of degree 6 vertices

triangle classification



 $size(LR) := 2v_r + v_i + 6|T_0| + 3|T_1^i + T_2^i|$ isolated vertices



Egea



input testested datasets: 3d meshes from aim@shape repository (with our implementation)



Are Delaunay triangulations bad for LR?



delaunay of random points (in unit circle)





Are random triangulations the worst case for LR?



Planar random triangulation

(sampled according to uniform distribution) (Poulalhon-Schaeffer sampler)



(FR91 force-directed layout)



LR: Laced Ring data structure





(force-directed layout, picture by J. Bettinelli) random stack triangulations



Let us try to devise better data structures (with better storage bounds for real-world graphs)

Compact data	storage cost (rpv)					navigation	preserves	
structure	best	average	(tested meshes) v		worst	edge/face	target	vertex
	case	3D meshes	Delaunay	random	case	navigation	operator	ordering
2D catalogs [Castelli Devillers Mebarki '06]	-	-	-	-	7.67	O(1)	O(1)	yes
Star vertices[Kallmann, Thalmann 2001]	7	7	7	7	7	$O(d^{\circ})$	O(1)	yes
sorted TRIPOD Snoeyink, Speckmann '99	6	6	6	6	6	O(1)	$O(d^{\circ})$	yes
SOT [Gurung Rossignac '09]	6	6	6	6	6	O(1)	$O(d^{\circ})$	yes
ESQ [Castelli et al. '12]	4	-	-	-	4.8	O(1)	$O(d^{\circ})$	yes
SQUAD [Gurung Laney, Lindstrom, Rossignac '10]	4	4.14	4.31	4.59	6	O(1)	$O(d^{\circ})$	yes
LR[Gurung et al. '11]	2	2.27	3.04	6.15	> 12	O(1)	O(1)	no
Thm 2 in Castelli Devillers '11	6	6	6	6	6	O(1)	$O(d^{\circ})$	yes
Thm 4 in [Castelli Devillers '11]	5	5	5	5	5	O(1)	$O(d^{\circ})$	yes
Thm 5 [Castelli Devillers '11]	4	4	4	4	4	O(1)	$O(d^{\circ})$	no
ScarstOS	3	3	3	3	3	$O(d_M + d^\circ)$	$O(d^{\circ})$	yes
ScarstOT	3	3.34	3.71	3.93	5	O(1)	$O(d^{\circ})$	yes
ScarstRS	2	2	2	2	2	$O(d_M + d^\circ)$	$O(d^{\circ})$	no
ScarstRT	2	2.26	2.54	2.65	3.67	O(1)	$O(d^{\circ})$	no
ScarstWC	3	3.03	3.08	3.04	3.33	O(1)	$O(d^{\circ})$	no













Warmup: a simple compact DS (size 6n)

 $\begin{array}{lll} \texttt{color} \in \{0,1,2\} & 0 \leq u \leq n-1 \\ & 0 \leq \mathsf{e} < \mathtt{3n} \end{array}$ **Challenge:** simulate Idea of the solution: e = (u, v)Winged-edge, while storing LeftFront • use Schnyder woods Implementation 1: Implementation 2 (faster): only 2 references per edge six tables: $T_{red}^{left}[.] T_{red}^{right}[.]$ • re-order edges one table T (of size 6n) e := (u, v)according to the $T_{\text{blue}}^{\text{left}}[.] \quad T_{\text{blue}}^{\text{right}}[.] \quad T_{\text{black}}^{\text{left}}[.] \quad T_{\text{black}}^{\text{right}}[.]$ $e := 3 \cdot u + color$ (input) vertex e := (u, color)u = e/3ordering RightBack (w)color(e) = e%3u=source(e) $if(C_{color}^{left}[u] == true) c_e = color$ (w,v)=T[2e]else $c_e = (color + 1)\%3$ $C_{\rm rod}^{\rm left}[j] = true$ (z,v) = T[2e+1] $(w, u) = (T_{\text{color}}^{\text{left}}[u], c_e)$ $\bigcap C_{\rm rod}^{\rm right}[j] = {\tt false}$ Main goal: retrieve missing neighboring edges (w, u) and (u, z) (in O(1) time) $I_{\mathbf{red}}[j] = \mathtt{false}$ $T_{\rm red}^{\rm left}[j]$ $T_{\rm red}^{\rm right}[j]$ 2 6 1 2 W (z)(z) \bigcirc 6 4 0 0 3 0 1 $\mathbf{2}$ 3 7 7 0 8 2 1 8 2 1 3 solution: simple case analysis 7 9 6 7 9 9 (e+1)%3case 1 6 * n entries 5(6n integers on 32 bits) $(w,u) := \langle T[e] + 2) \% 3$ case 24 1 0 4 1 1 $\mathbf{6}$ 7T[T[e]]case 3 8 3 9 3 3 9 3

(Sorted) TRIPOD (Snoeyink, Speckmann, '99)

(Thm 2 in Castelli Aleardi, Devillers 2011)

Adaptive compact data structures

Goal: devise a mesh data structures with provable storage bounds that takes into account graph regularity

Can we exploit the regularity of the triangulation to improve the previous existing bounds of 4 rpv?













Idea 2: use only one reference for low degree vertices. Use (skip) extra references for high degree references, for red, blue and black edges. **Remark:** in a (planar) graph the number of forbidden cases (no ccw triangles) high degree vertices is small access extra references Low degree vertices Navigation for (black) high degree vertices via address indirection $1\ {\rm reference}$ to the right for every black edge indegree(v) < 41 **extra** reference to the left for $\lfloor \frac{d}{k} \rfloor$ black edges use only 1 reference indegree(v) = 9(navigate as before) u_3 u_6 u_8 u_0 u_5 u_1 u_7 u_2 u_0 $\overline{u_4}$ u_3 u_6 u_5 1)

SCART-OT (memory cost: between 3n and 5n references)

Towards worst-case O(1) time navigation

Idea 1: use minimal Schnyder woods

For k = 3 the number of extra reference (in each color) is at most $\frac{n}{3}$

use minimal Schnyder woods

(still preserving original vertex ordering)

SCARST-RT (memory cost at most 3.67n in worst-case)

Idea 1 (as before): use minimal Schnyder woods

Idea 2 (as before): use only one reference for low degree vertices. Use extra references only for high degree references, for red, blue and black edges.

Idea 3: re-order vertices according to a BFS traversal of the red tree T_0



$$indegree(v) \ge 4$$

(worst-case O(1) time navigation)

$$indegree(v) \ge$$

1 reference to the right for every black edge

 $e := (u_5, v)$ $\texttt{LeftFront}(e) = 3(u_5 + 1) + 0$ $\texttt{RightFront}(e) = 3(u_5 - 1) + 0$

ccw triangles are forbidden:

edge $e = (u_8, w)$ must be blue $LeftFront(e) = RightFront(3u_8 + 1)$

or equivalently $LeftFront(e) = RightFront(LeftBack(u_8))$



Low degree vertices

indegree(v) < 4

DO NOT use



Navigation for (black) high degree vertices

SCARST-RT (memory cost at most 3.67n in worst-case) (worst-case O(1) time navigation)

Idea: use only one reference for low degree vertices. Use extra references only for high degree references, for red, blue and black edges. Remark: re-order vertices according to a BFS traversal **Good news:** the number of **bad** situations is "small"



SCARST-RT (memory cost at most 3.67n in worst-case) (worst-case O(1) time navigation)





L(T):= number of leaves in a rooted tree T $N_{\geq d}(T):=$ number of nodes with indegree at least d

LEMMA 2. For all positive integer d, we have

$$L(T) > (d-1)N_{\geq d}(T).$$

Proof. The total indegree must be the same as the total outdegree. In a rooted tree the outdegree is 1 for all nodes except for the root (outdegree 0). Since the number of nodes of the tree is $\sum_{i=0}^{\infty} N_{=i}(T)$ we have:

$$\begin{split} \Big(\sum_{i=0}^{\infty} N_{=i}(T)\Big) - 1 &= \sum_{i=0}^{\infty} i \cdot N_{=i}(T) \\ 0 &= 1 + \sum_{i=0}^{\infty} (i-1) \cdot N_{=i}(T) \\ N_{i=0}(T) &= 1 + \sum_{i=2}^{d-1} (i-1) \cdot N_{=i}(T) + \sum_{i=d}^{\infty} (i-1) \cdot N_{=i}(T) \\ L(T) &> 0 + 0 + (d-1) \cdot N_{\geq d}(T) \\ \Box \end{split}$$

Theorem [Castelli, Devillers '24] The above data structure supports O(1) time navigation and uses at most 3.67n references (take d = 3, see below) $3.67n = 2n + 2 \cdot \left(\frac{n}{3} + \frac{n}{3} + \frac{n}{2d}\right)$

Idea: choose T_2 (black tree) with the minimal number of leaves (permute the colors if needed)

let
$$K_{\mathbf{r}}^{d}$$
:= number of vertices w such that $indegree_{\mathbf{r}}(w) \ge d+1$

at least d children are not a leaf of T_2

LEMMA 4. Given a Schnyder word on n vertices, such that the black tree has more leaves than the red tree, we have $K_r^{(d)} \leq \frac{1}{2d}n$.

Proof. A vertex to be counted can be associated with d internal nodes of the green tree, as a consequence T_2 has at least $dK_r^{(d)}$ internal nodes. Thus

$$\begin{array}{rcl} dK_{r}^{(d)} &\leq & N_{\geq 1}(T_{2}) \\ n - dK_{r}^{(d)} &\geq & n - N_{\geq 1}(T_{2}) \\ &\geq & L(T_{2}) & \text{total size of } T_{g} \text{ is } n \\ &\geq & L(T_{r}) & T_{2} \text{ minimizes } \sharp \text{leaves} \\ &\geq & dN_{\geq d+1}(T_{r}) & \text{by Lemma } 2 \\ &\geq & dK_{r}^{(d)} & & \Box \end{array}$$

Adaptive behaviour: storage cost evaluation



Lampan Hack

Egea

sphere

Fast navigation (betweem 1.5 and 3.8 times slower than Winged-edge)



Concluding remarks

Topological extensions

Toroidal Schnyder woods [Goncalves Lévêque, DCG'14]

3-orientation + Schnyder local rule valid at each vertex







toroidal Schnyder wood

Efficient decompression from compressed format (with no additional memory)



















