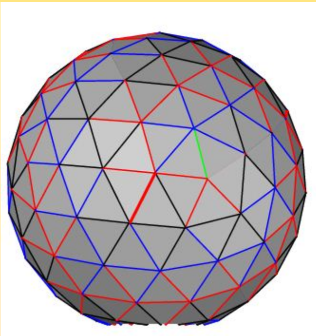
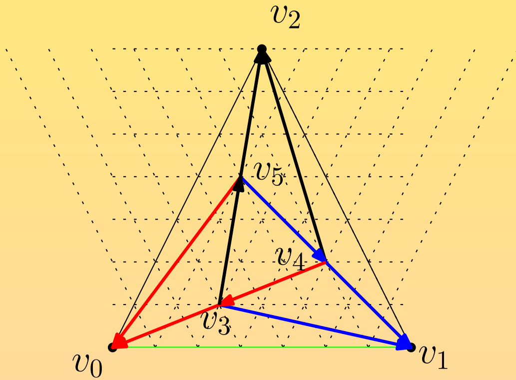
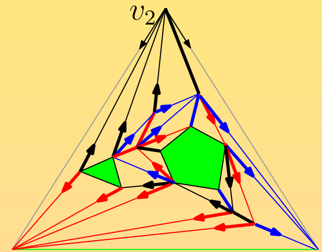
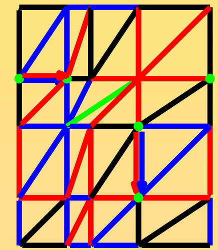
Array-based compact data structures for triangle meshes









Encuentros de Geometría Computacional, EGC2011

Luca Castelli Aleardi

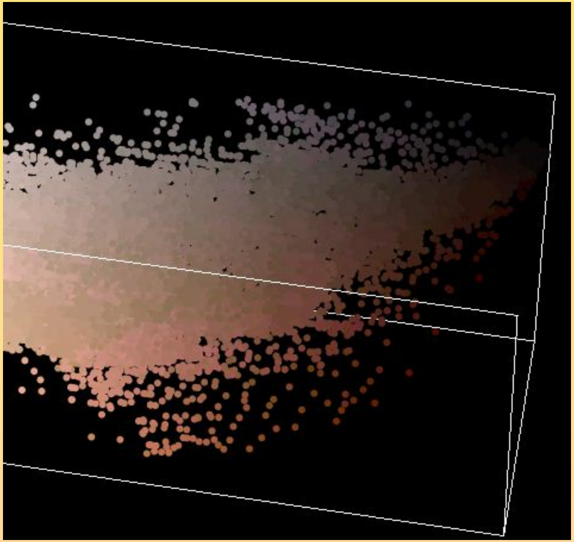
Olivier Devillers

ÉCOLE POLYTECHNIQUE

INRIA SOPHIA ANTIPOLIS

GEOMETRICA

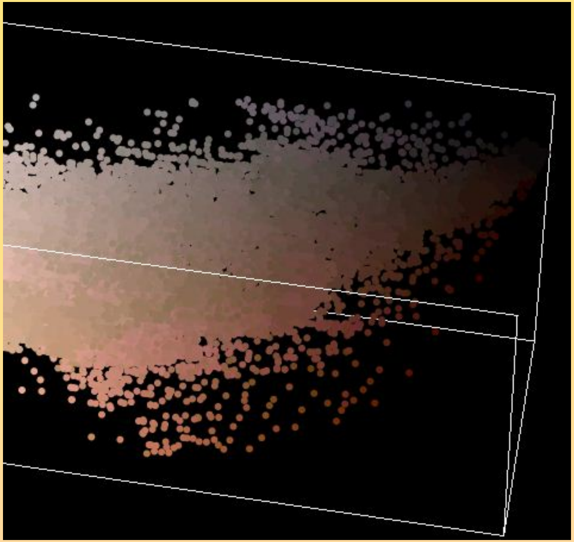
Geometrica - INRIA Sophia



Point Cloud representation of Ferran

Buon Compleanno

Ferran



Point Cloud representation of Ferran

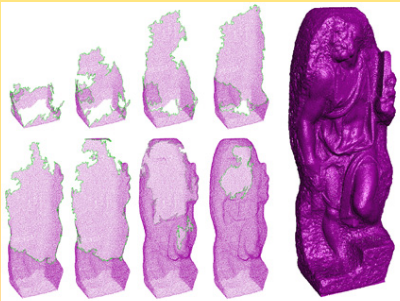
Ferran segmented with MeanShift

Buon Compleanno

Ferran

Compact Data Structures: motivation and goal

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

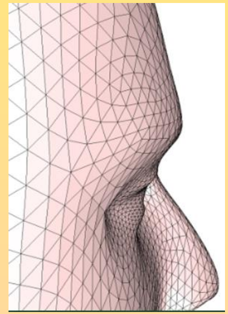


6 Giga bytes (for storing on disk)
186 millions vertices

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

2 billions polygons
32 Giga bytes (without compression)

vertex
triangle



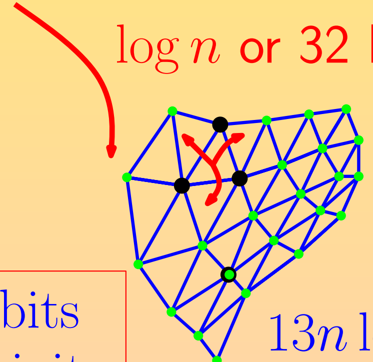
Connectivity information

1 reference to a triangle

3 references to vertices

3 references to triangles

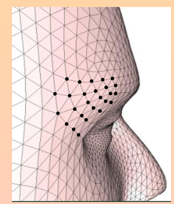
$\log n$ or 32 bits



$13n \log n$

$416n$ bits connectivity

Geometric object

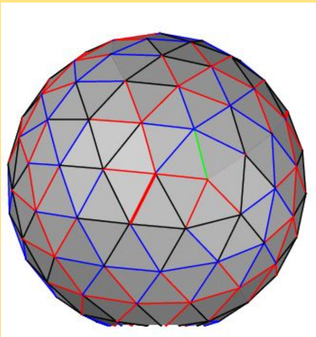


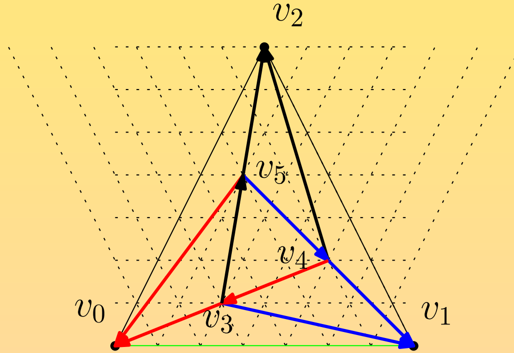
$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

between 30 et 96 bits/vertex

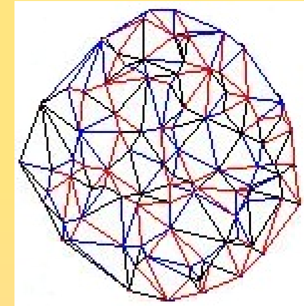
(planar) Triangle meshes

Surface triangle mesh endowed with a Schnyder wood





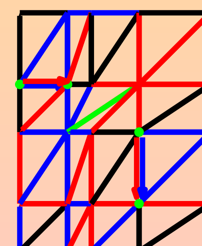
Schnyder drawing of a planar triangulation

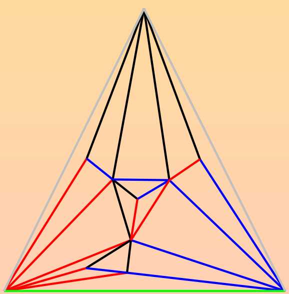


Delaunay triangulation of random points (endowed with a Schnyder wood)

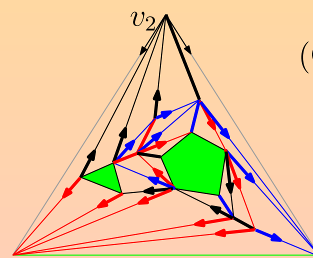
g-Schnyder woods, for genus g triangulations

(Castelli-Fusy-Lewiner, SoCG08)





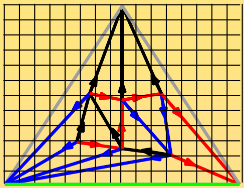
Tutte drawing of a random planar triangulation



Schnyder woods for triangulations with multiple boundaries of arbitrary size

(Castelli-Fusy-Lewiner,CCCG2010)

Graph planarity characterizations

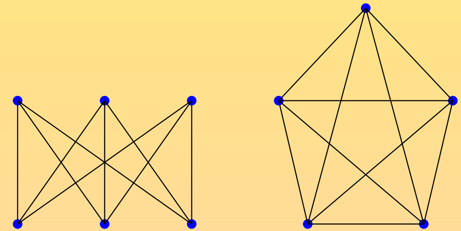


Schnyder woods (via dimension of partial orders)

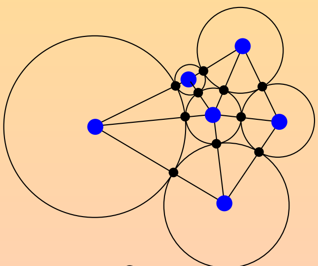
- $dim(G) \leq 3$

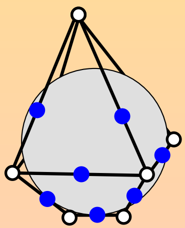
Kuratowski theorem (excluded minors)

- G contains neither K_5 nor $K_{3,3}$ as minors

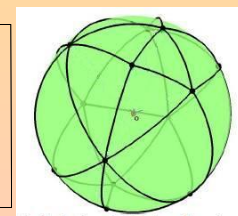


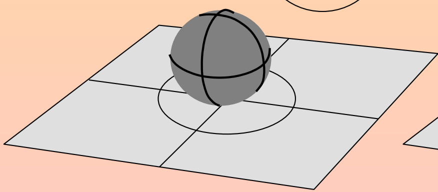
Thm (Koebe-Andreev-Thurston)
Every planar graph with n vertices is isomorphic to the intersection graph of n disks in the plane.

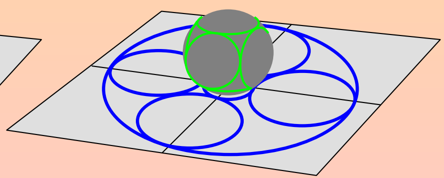




$$M = \begin{bmatrix} -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 \end{bmatrix} \quad \begin{matrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{matrix} \; \xi_x = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad \xi_y = \begin{bmatrix} -1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \xi_z = \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$





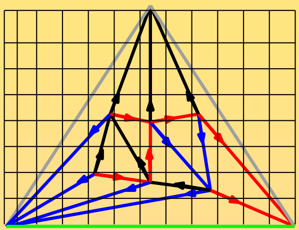


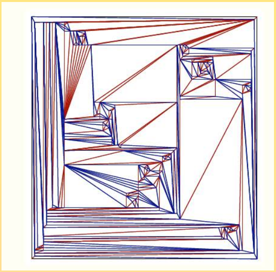
Colin de Verdiere invariant (multiplicity of λ_2 eigenvalue of a generalized laplacian)

- $\mu(G) \leq 3$

Schnyder woods: applications
(and related properties)

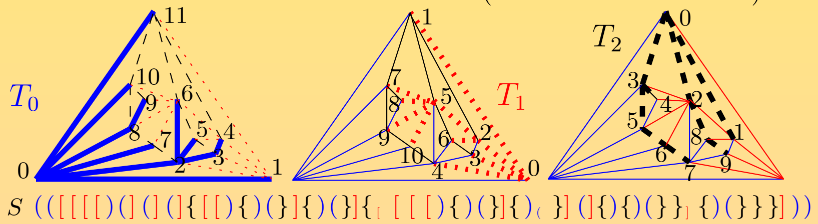
grid drawing





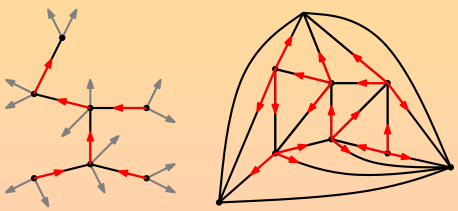
Graph encoding

(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)



graph counting, random generation

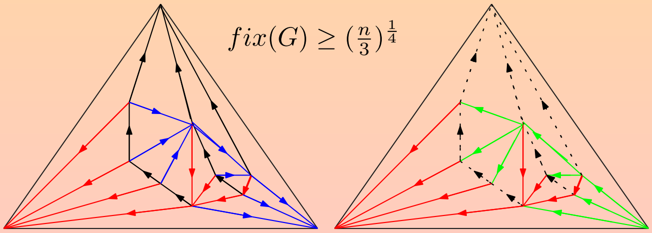
(Poulalhon-Schaeffer, Icalp 03)



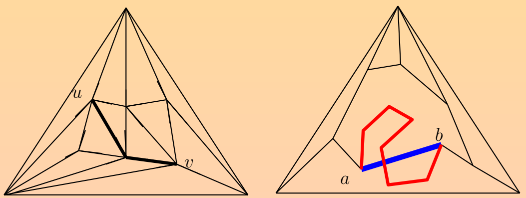
Untangling geometric graphs

Bose, Dujmovic, Hurtado, Langerman, Morin, Wood (DCG 2009)

$$fix(G) \geq (\frac{n}{3})^{\frac{1}{4}}$$



Greedy routing

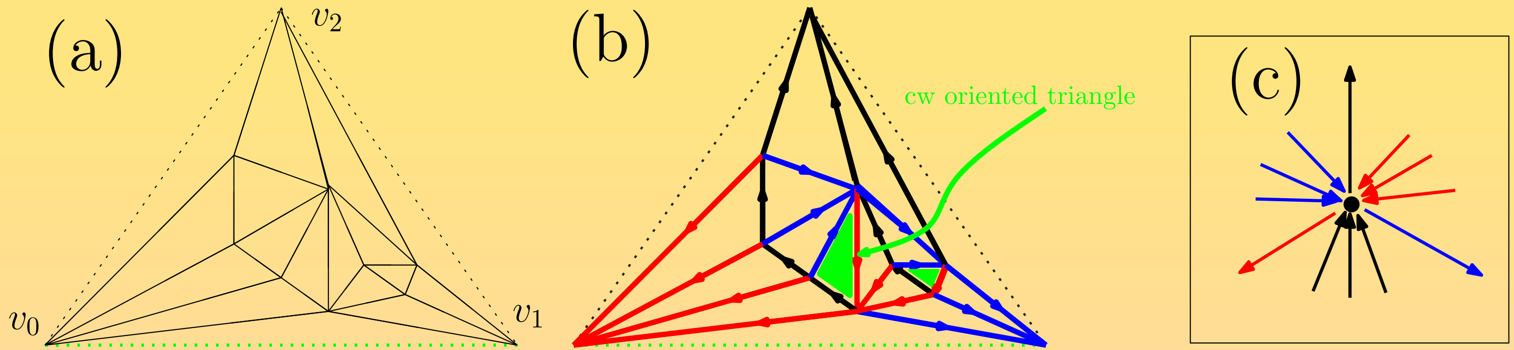


Every planar triangulation admits a *greedy drawing* (Dhandapani, Soda08)

Conjectured by Papadimitriou and Ratajczak for 3-connected planar graphs

Schnyder woods: the definition

v_0 v_1 v_2 outer face



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

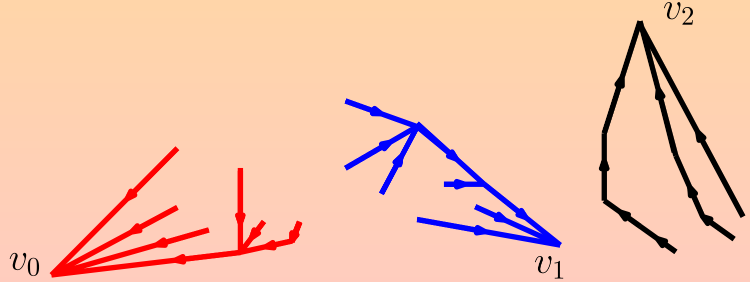
ii) colors and orientations around each inner node must respect the local Schnyder condition

Theorem

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

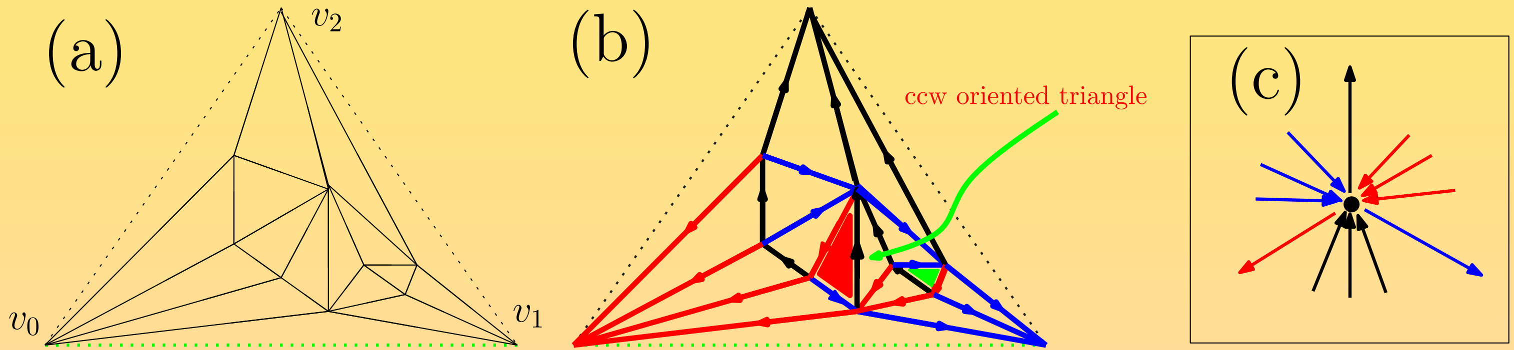
Theorem

The three set T_0, T_1, T_2 are spanning trees of (the inner nodes of) T:



minimal Schnyder woods: the definition (no ccw triangles)

v_0 v_1 v_2 outer face

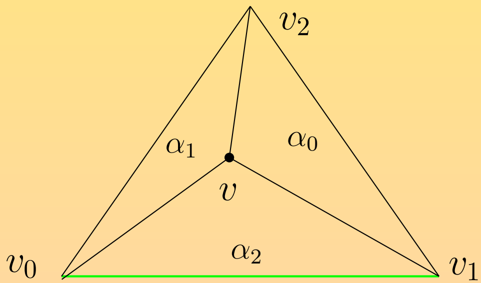


i) edge are colored and oriented in such a way that each inner node has exactly one outgoing edge of each color

ii) colors and orientations around each inner node must respect the local Schnyder condition

Schnyder woods and (planar)
barycentric coordinates

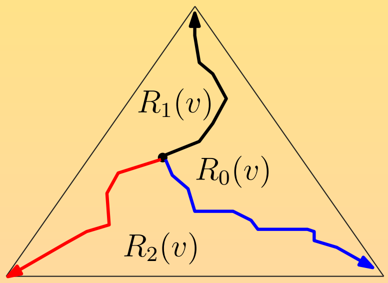
Geometric interpretation of barycentric coordinates



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

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

Combinatorial interpretation



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

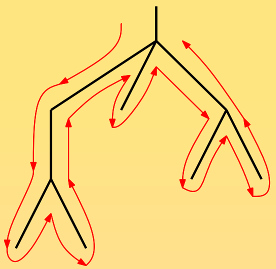
$R_i(v)$:=number of triangles in region i

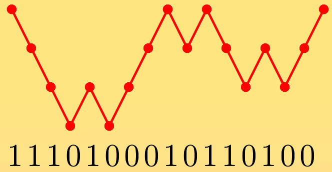
Theorem (Schnyder, Soda '90)

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

compression schemes



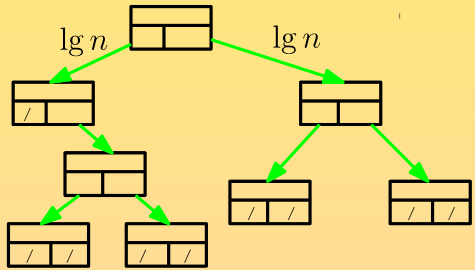


$2n$ bits

$$\|\mathcal{B}_n\| = \frac{1}{n+1}\binom{2n}{n} \approx 2^{2n} n^{-\frac{3}{2}}$$

$$log_2 \|\mathcal{B}_n\| = 2n + O(\lg n)$$

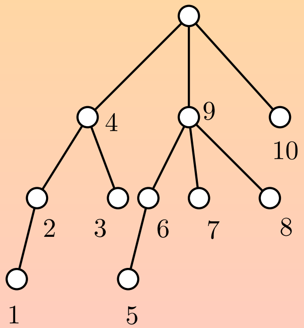
(Explicit) data structures

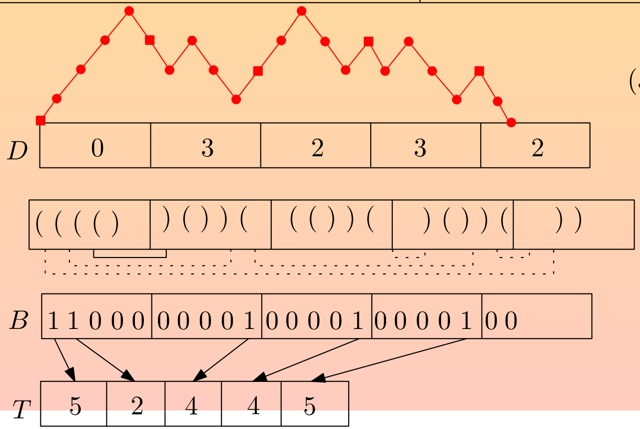


we need $2n$ references, or $\Theta(n \lg n)$ bits

Succinct representations

(Jacobson Focs89, Munro and Raman Focs97)





$2n + o(n)$ bits

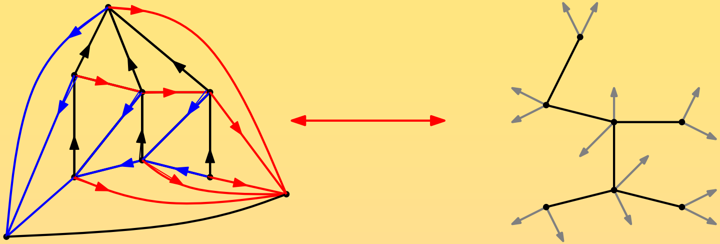
$+$

$O(1)$ time navigation

Several kinds of encodings: triangle meshes

Optimal compression scheme

(Poulalhon Schaeffer, Icalp03)

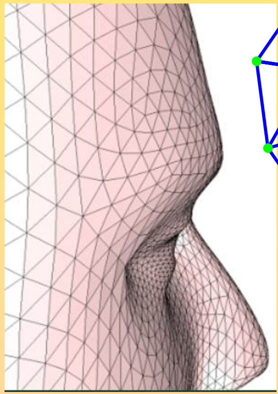


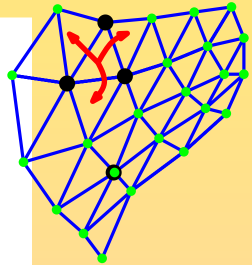
$$\Psi_n = \frac{2(4n+1)!}{(3n+2)!(n+1)!} \approx \frac{16}{27}\sqrt{\frac{3}{2\pi}}n^{-5/2}(\frac{256}{27})^n$$

$$\frac{1}{n}\log_2 \Psi_n \approx \log_2(\frac{256}{27}) \approx 3.2451 \text{ bits/vertex}$$

(Explicit) Geometric data structures

$\beta n + O(1)$ references (pointers)





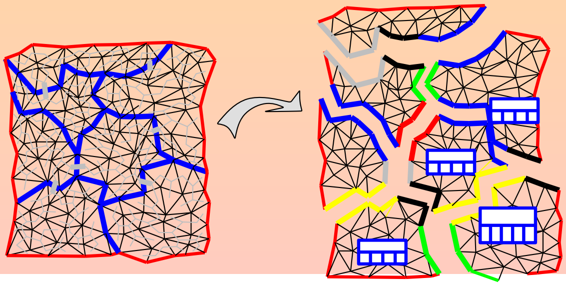
$2 \times n \times 6 \times \log n$

$+$

$n \times 1 \times \log n$

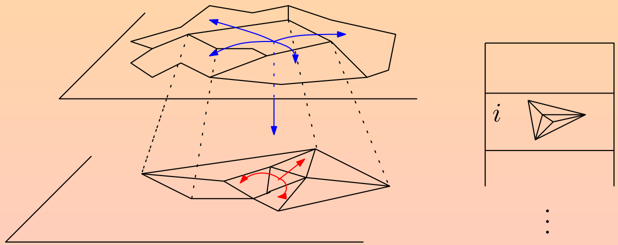
$13n$ references *or* $13n \log n$ bits

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

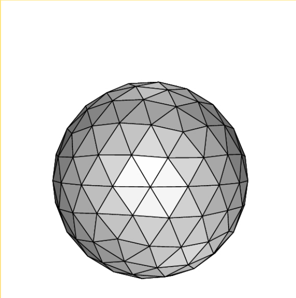


Succinct representations

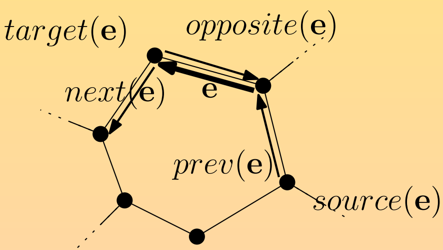
(Castelli Aleardi- Devillers-Schaeffer, WADS05, SoCG06)



Popular mesh data structures



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



Half-edge

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

geometry

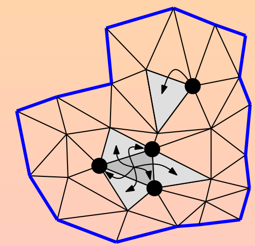
```
class Halfedge{
  Halfedge  prev, next, opposite;
  Vertex source, target;
  Face f;
}
 class Vertex{
  Halfedge  e;
  Point p;
}
```

connectivity

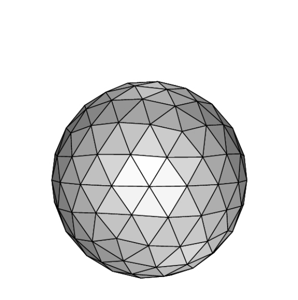
Triangle-based

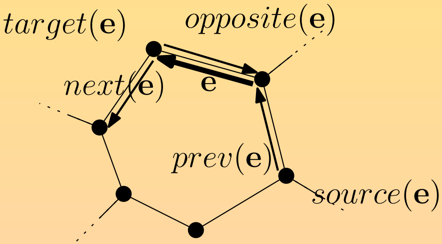
```
class Triangle{
  Triangle t1, t2, t3;
  Vertex v1, v2, v3;
}
 class Vertex{
  Triangle  root;
  Point p;
}
```

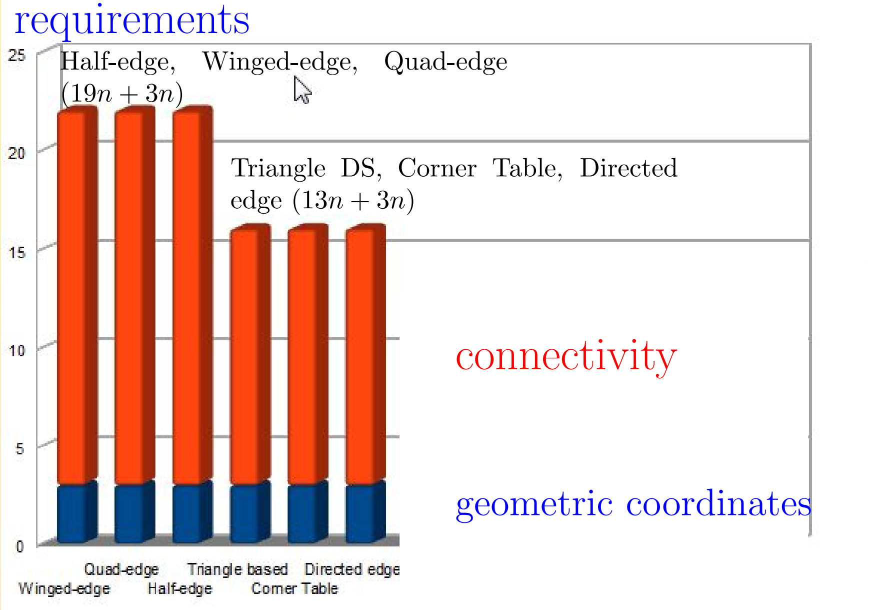
```
class Point{
  float x;
  float y;
  float z;
}
```



Popular mesh data structures: space requirements







Half-edge

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

geometry

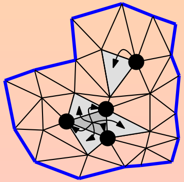
```
class Halfedge{
  Halfedge  prev, next, opposite;
  Vertex source, target;
  Face f;
}
 class Vertex{
  Halfedge  e;
  Point p;
}
```

connectivity

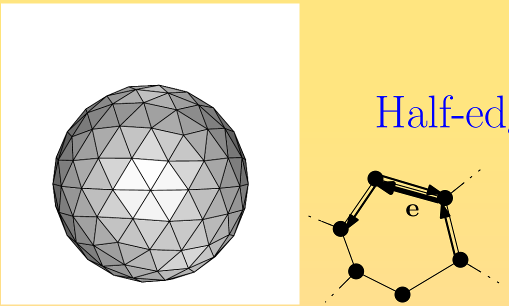
Triangle-based

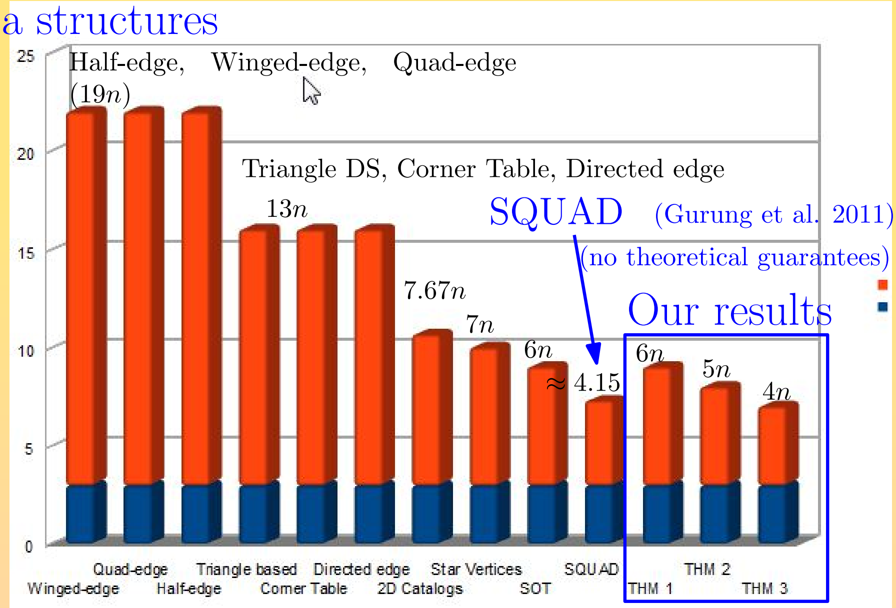
```
class Triangle{
  Triangle t1, t2, t3;
  Vertex v1, v2, v3;
}
 class Vertex{
  Triangle  root;
  Point p;
}
```

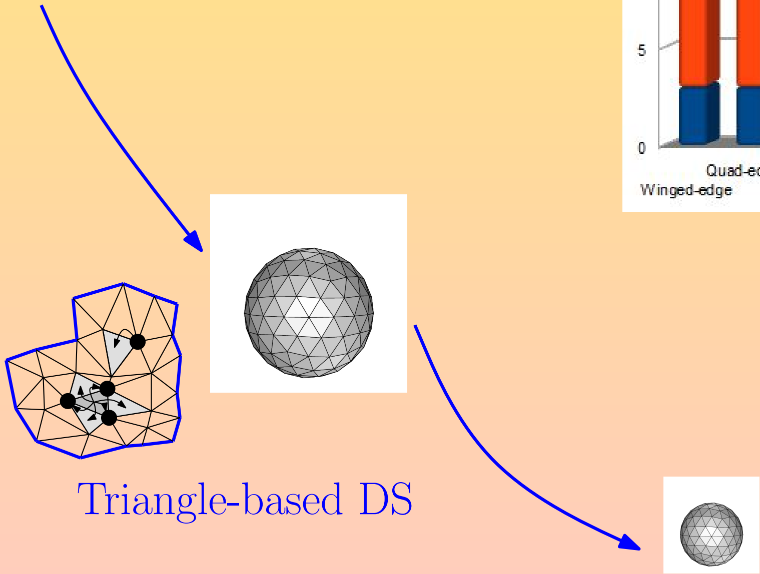
```
class Point{
  float x;
  float y;
  float z;
}
```

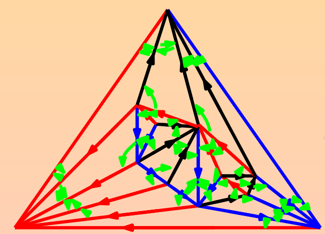


Non compact vs. compact mesh data structures









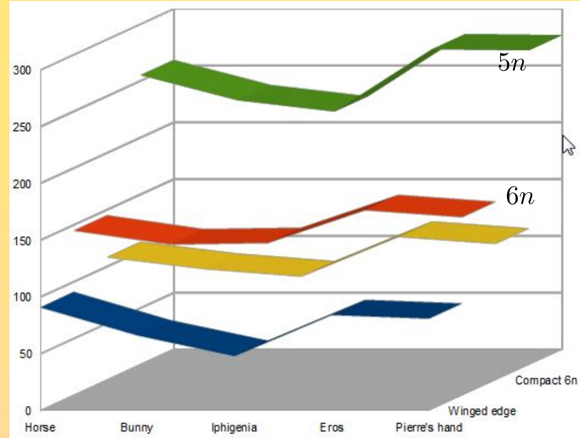
Experimental comparison

(timings are expressed in nanoseconds/vertex)

Winged edge vs. Our Compact DS

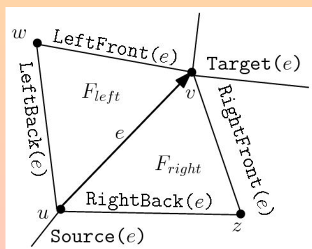
Our Compact DS

vertex degree (only topological navigation)



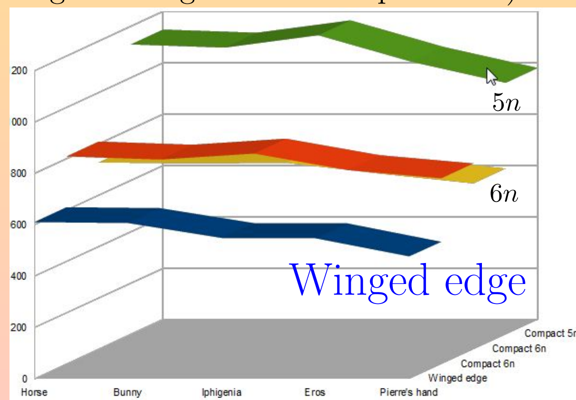
1.2 - 1.9 times slower

Tested on 3D models and random planar triangulations



vertex normals (navigation + geometric computations)

1.19 - 1.35 times slower



Winged edge

Thm1: Compact DS (6n references)

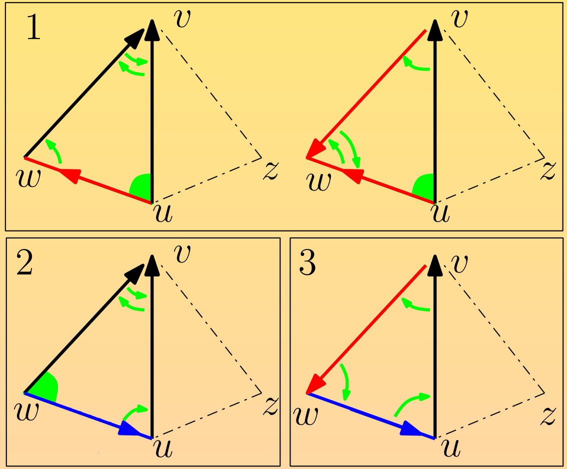
$e := (u, v)$ $\quad 0 \leq v \leq n-1$

$0 \leq e \leq 3n$

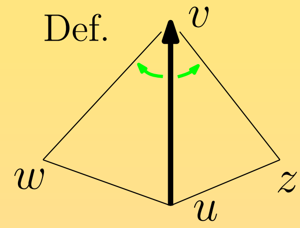
Simple solution: redundant and "simple to implement"

"sorted variant" of TRIPOD DS

(Snoeyink-Speckmann, 1999)

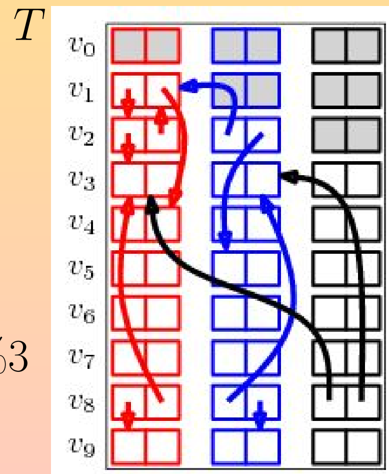


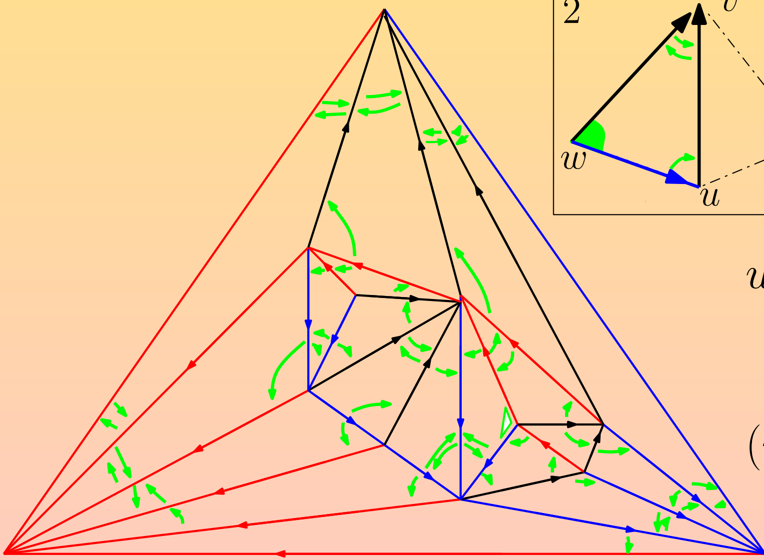
Def.



(w,v):=T[2e]

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



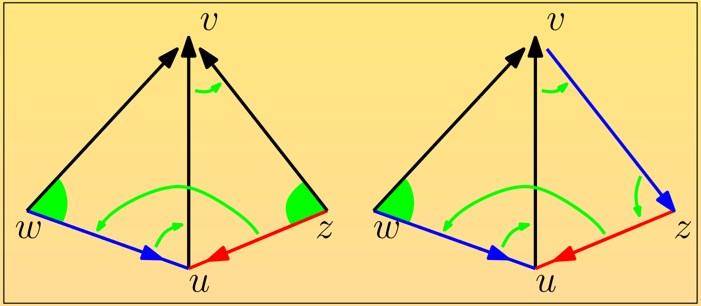


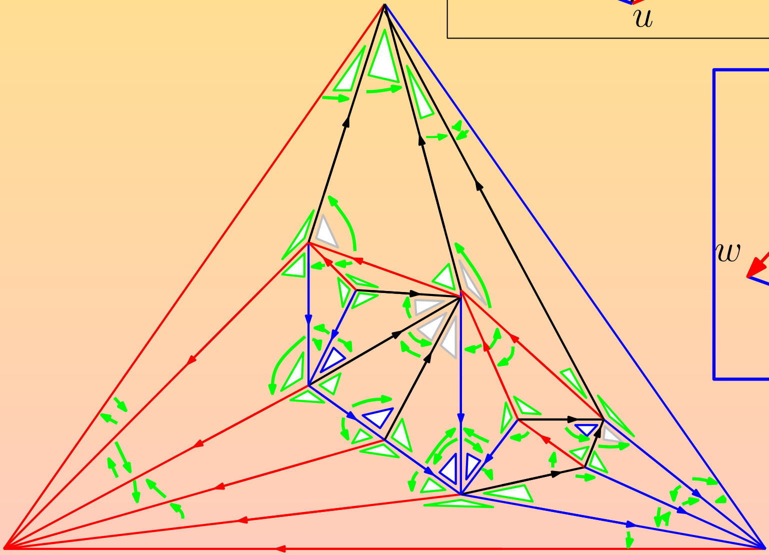
$u := \texttt{source}(e) = e/3$

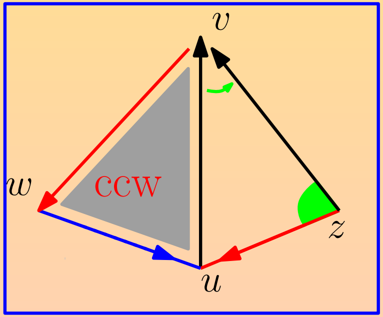
$$(w, u) := \begin{cases} (e+1)\%3 \\ (T[e]+2)\%3 \\ T[T[e]] \end{cases}$$

Thm2: More compact DS (5n references)

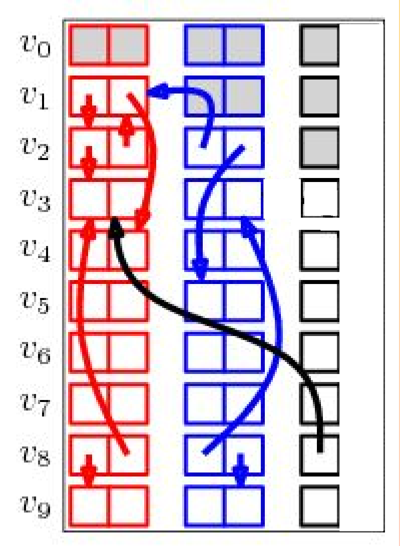
more compact solution: less redundant and "more difficult to implement"





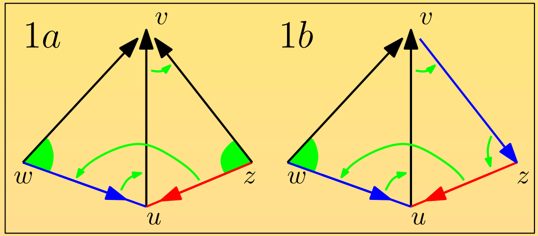


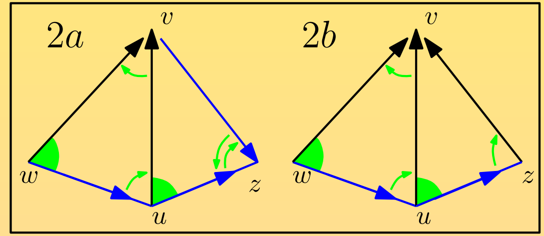
forbidden case

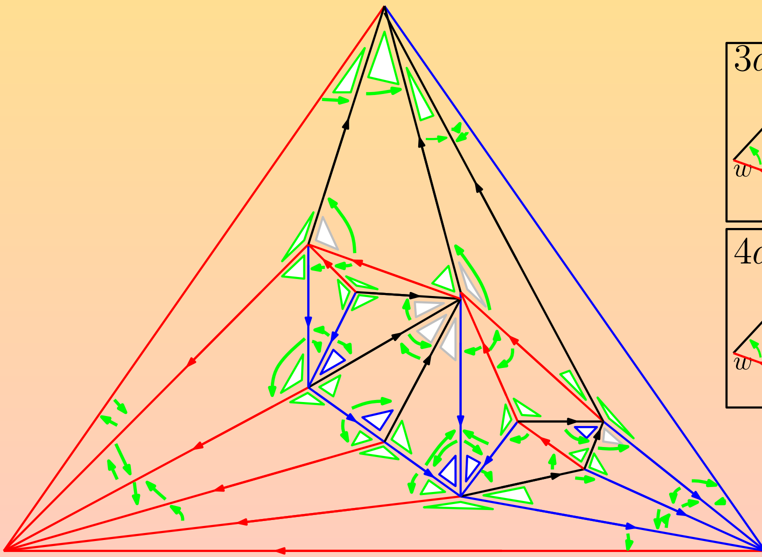


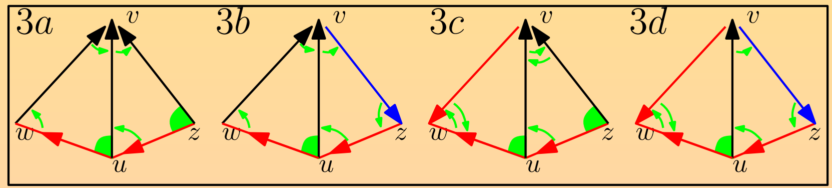
Thm2: More compact DS (5n references)

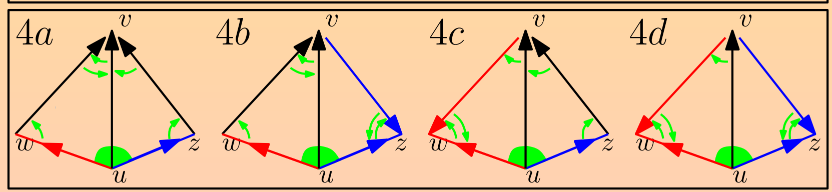
more compact solution: less redundant and "more difficult to implement"











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 $\overline{T}_0$