Compact representations of geometric data structures

24th January 2007, University of Waterloo

Luca Castelli Aleardi

Olivier Devillers  Gilles Schaeffer

Geometrica - INRIA Sophia  LIX École Polytechnique
Domain and motivations
Geometric data

Triangulations and graphs  3D meshes  Tetrahedral Volume Meshes

Applications

Surface reconstruction  GIS Technology  Geometric modeling
Very large geometric data

St. Matthew (Stanford’s Digital Michelangelo Project, 2000)
186 millions vertices
6 Giga bytes (for storing on disk)
tens of minutes (for loading the model from disk)

David statue (Stanford’s Digital Michelangelo Project, 2000)
2 billions polygons
32 Giga bytes (without compression)

No existing algorithm nor data structure for dealing with the entire model
Mesh compression

Transmission

disk storage

Geometric data structures

Compact representations of geometric data structures
Compact representations
An example: plane trees

ordered tree with $n$ edges

balanced parenthesis word

$1110100010110100$

$\Rightarrow 2n$ bits for encoding a tree with $n$ edges

Enumeration of plane trees with $n$ edges

$$\|B_n\| = \frac{1}{n+1} \binom{2n}{n} \approx 2^{2n} n^{-\frac{3}{2}}$$
An example: plane trees

ordered tree with \( n \) edges

balanced parenthesis word

\[ 1110100010110100 \]

\( \Rightarrow 2n \) bits for encoding a tree with \( n \) edges

Asymptotic optimal encoding

- the cost of an object matches asymptotically the entropy

\[
\log_2 \| \mathcal{B}_n \| = 2n + O(\lg n)
\]
An example: plane trees

ordered tree with $n$ edges

balanced parenthesis word

1110100010110100

$\Rightarrow 2n$ bits for encoding a tree with $n$ edges

No efficient implementation of local adjacency queries
An example: plane trees

Explicit pointers based representation

adjacency queries between vertices in $O(1)$ time

not optimal encoding: we need $\Theta(n \lg n)$ bits
Could we do better?

a compact encoding (asymptotically optimal)

testing adjacency queries efficiently (in $O(1)$ time)
An example: binary and ordered trees

(Jacobson, Focs89, Munro et Raman Focs97)

For trees and parenthesis words the answer is... YES

it is possible to test adjacency between vertices in $O(1)$ time with the guarantee the the encoding is still asymptotically optimal

$2n + o(n)$ bits are sufficient
Geometric data

Triangulations with a boundary

Triangulations of the sphere

3-connected planar maps (polygonal meshes)
Which information?

Geometry and connectivity

Geometric object

Combinatorial object

Geometric information
Geometric information

Geometric object

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

between 30 et 96 bits/vertex

Connectivity information

Connectivity object

vertex

1 reference to a triangle

triangle

3 references to vertices

3 references to triangles

Combinatorial object

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

\[
n \times 1 \times \log n
\]

13n \log n

\log n ou 32 bits

416n bits connectivity
Entropy bounds are known...

Triangulations with a boundary
- 2.17 bits/edge

Triangulations of the sphere
- 1.62 bits/triangle

3-connected planar maps (polygonal meshes)
- 2 bits/edge
Enumeration and entropy of planar maps

Enumeration of planar triangulations (Tutte, 1962)

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

Entropy

\[ \frac{1}{n} \log_2 \Psi_n \approx \log_2 \left(\frac{256}{27}\right) \approx 3.2451 \text{ bits/vertex} \]
Enumeration and entropy of planar maps

Triangulations with a boundary

$n + 1$ internal vertices, $m = 2n + k$ faces

\[ f(n, k) = \frac{2 \cdot (2k - 3)! (2k + 4n - 1)!}{(k - 1)! (k - 3)! (n + 1)! (2k + 3n)!} \]

\[ f'(m, k) = \frac{2 \cdot (2k - 3)! (2m - 1)!}{(k - 1)! (k - 3)! \left(\frac{m-k}{2}\right) + 1)!} \]

enumeration of polygons with $m$ faces

\[ F(m) = \lg\left(\sum_{k \geq 3} f'(m, k)\right) \approx 2.175m \]

3.24 bits/vertex = 1.62 bits/face < 2.17 bits/face
Mesh compression
3D triangle meshes

VRML, 288 or 114 bits/vertex

[Edgebreaker] 3.67 bits/vertex (guaranteed upper bound)

[Poulalhon Schaeffer] 3.24 bits/vertex (optimal)

[Touma Gotsman] ≈ 2 bits/vertex (heuristic)

PS 110100011000001001000001100100000000

TG $V_5 V_5 V_6 V_5 V_4 V_5 V_8 V_5 V_5 V_4 S_4 V_3 V_4$

EB $CCCCRCCRCCRECRRRELCRE$
Graph encodings and spanning trees

General visual framework (Isenburg Snoeyink)

Edgebreaker

Canonical orderings and multiple parenthesis words
- Chuang et al. (98)
- Chiang et al. (01)

Touma Gotsman ('98)

No queries

Poulalhon Schaeffer (2003)
Geometric data structures
Geometric data structures
Explicit representations

Triangle meshes

Connectivity

```
struct triangle{
    triangle *t1, *t2, *t3;
    vertex *v1, *v2, *v3;
}
struct vertex{
    triangle * root;
    int label;
}
```

Geometry

```
struct point{
    float x;
    float y;
}
point vertex_geometry[max_label];
```

Polygonal meshes

```
struct edge{
    edge * oppo;
    edge * next;
    edge * prev;
    vertex * origin;
}
struct vertex{
    edge * root;
    int label;
}
```

```
struct point{
    float x;
    float y;
}
point vertex_geometry[max_label];
```

not succinct encodings
$\Theta(n \lg n)$ bits
Succinct encodings of graphs: existing works
Succinct encodings of graphs: existing works

planar graphs: book embeddings and canonical orderings

<table>
<thead>
<tr>
<th>Codage</th>
<th>3-c.</th>
<th>triang.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacobson</td>
<td>64n</td>
<td>64n</td>
</tr>
<tr>
<td>Munro Raman</td>
<td>8n + 2e</td>
<td>7m</td>
</tr>
<tr>
<td>Chiang et al.</td>
<td>2e + 2n</td>
<td>3.5m</td>
</tr>
<tr>
<td>Chiang et al.</td>
<td>2e + 2n</td>
<td>4m</td>
</tr>
<tr>
<td>Blandford et al.</td>
<td>(O(n))</td>
<td>(O(m))</td>
</tr>
</tbody>
</table>

\(G - T\)
Our solution for succinctly representing graphs
First idea: decomposition

Decomposition of quadrangulations...by the french artist Léon Gischia (1903-1991)
Second idea: precomputing and storing

Literary digression (The lesson, Eugène Ionesco, 1951)

During a private lesson, a very young student, preparing herself for the total doctorate, talks about arithmetics with her teacher

(the young student cannot understand how to subtract integers)

**teacher** Listen to me, If you cannot deeply understand these principles, these arithmetic archetypes, you will never perform correctly a "polytechnicien" job... you will never obtain a teaching position at "Ecole Polytechnique". For example, what is 3.755.918.261 multiplied by 5.162.303.508?

**student (very quickly)** the result is 193891900145...

**teacher (very astonished)** yes ... the product is really... But, how have you computed it, if you do not know the principles of arithmetic reasoning?

**student**: it is simple: I have learned by heart all possible results of all possible different multiplications.
Our scheme

Overview of the hierarchical structure

**Level 1:**
- $\Theta\left(\frac{n}{\log^2 n}\right)$ regions of size $\Theta(\log^2 n)$, represented by pointers to level 2

**Level 2:**
- in each of the $\frac{n}{\log^2 n}$ regions
  - $\Theta(\log n)$ regions of size $C \log n$, represented by pointers to level 3

**Level 3:** exhaustive catalog of all different region of size $i < C \log n$:
- complete explicit representation.
Our scheme

Overview of the hierarchical structure

Level 1:
- $\Theta(\frac{n}{\log^2 n})$ regions of size $\Theta(\log^2 n)$, represented by pointers to level 2
- Global pointers of size $\log n$

Level 2:
in each of the $\frac{n}{\log^2 n}$ regions
- $\Theta(\log n)$ regions of size $C\log n$, represented by pointers to level 3
- Local pointers of size $\log\log n$

Level 3: exhaustive catalog of all different region of size $i < C\log n$:
- Complete explicit representation.
Our scheme

Overview of the hierarchical structure

Level 1:
- $\Theta\left(\frac{n}{\log^2 n}\right)$ regions of size $\Theta(\log^2 n)$, represented by pointers to level 2
- global pointers of size $\log n$

Level 2:
- in each of the $\frac{n}{\log^2 n}$ regions
- $\Theta(\log n)$ regions of size $C\log n$, represented by pointers to level 3
- local pointers of size $\log \log n$

Level 3: exhaustive catalog of all different region of size $i < C\log n$:
- complete explicit representation.

Dominant memory cost
Let us consider a triangulation of size $m$ (with $m$ faces)
Compact representations of graphs

We decompose into sub triangulations of polylog size

micro triangulations of size between $\frac{1}{12} \lg m$ et $\frac{1}{4} \lg m$
Graph of micro regions

graph $G$:
- a node for each micro region,
- an arc for each pair of adjacent micro regions

$G$ is a local planar map

$O\left(\frac{m}{\log m}\right)$ nodes
Sub-graph of micro regions

$G_i$ is a sub-map of $G$ containing $O(\lg m)$ micro triangulations

$O(\log^2 m)$ triangles

graph $G_i$

$O(\log m)$ nodes
Graph of mini regions

descrip:
- a node for each mini region
- an arc for each pair of adjacent mini regions

$F$ est une carte planaire

$O\left(\frac{m}{\log^2 m}\right)$ nodes
Asymptotic cost of the representation
Our solution

Cost of graphs $G$ and $F$

$O\left(\frac{m}{\log m}\right)$ micro regions
Because of Euler’s relation:

\[ E(G) = O\left(\frac{m}{\log m}\right) \]

Our solution

\( G \) is a sparse graph

\( G \) is local planar, having all faces of degree at least 3
Our solution

We use local pointers of size $\log \log m$ for explicitly representing graphs $G_i$.

Graph $G_i$ with $O(\log m)$ nodes:

\[ \sum_i \|E(G_i)\| = O\left(\frac{m}{\log m}\right) \]

$O\left(\frac{m}{\log m} \log \log m\right)$ bits

Graph $G$ for explicitly representing graphs $G_i$.
Our solution

We can use pointers of size $\log m$

$F$ is local planar and then sparse

$O\left(\frac{m}{\log^2 m}\right)$ arcs

$O\left(\frac{m}{\log^2 m} \log m\right)$ bits
Compact representations of graphs

Cost of the catalog

Exhaustive catalog of all different micro regions

\[ ||A|| \approx 2^{\frac{1}{4}2.175r} \leq 2^{\frac{1}{4}2.175 \lg m} \]

The catalog of all micro regions having at most \( r \) triangles has negligible size

2.175 entropy of the original class
Cost of references to the catalog

The cost of a reference depends on the size of the catalog.

\[ \log_2 \| A \| = 2.175r \text{ bits} \]

2.175 entropy of the original class
Compact representations of graphs

Dominant term

The dominant term is given by the sum of references to the catalog

\[
\sum_j 2.175r_j = 2.175m \text{ bits}
\]

\[ r \text{ triangles} \]

2.175 entropy of the original class
Compact representations of graphs

**Theorem** (Castelli Aleardi, Devillers et Schaeffer, WADS05)
For planar triangulations with $m$ triangles there exists a compact representation, supporting local adjacency queries in $O(1)$ time, which requires

$$2.175m + O(m \frac{\lg \lg m}{\lg m}) = 2.175m + o(m) \text{ bits}$$

For higher genus $g$ triangulations we have

$$2.175m + 36(g - 1) \lg m + O(m \frac{\lg \lg m}{\lg m} + g \lg \lg m) \text{ bits}$$
Compact representations of graphs

No assumptions on the size of the boundaries

\[ \| A \| \approx 2^{1.75r} \]

\[ \log_2 \| A \| = 2.175r \text{ bits} \]

\[ \sum_j 2.175r_j = 2.175m \text{ bits} \]

1.62m is the entropy of the original class
Is it possible to achieve optimality?
A more clever decomposition strategy

\[ \| A \| = ? \]

\[ \log_2 \| A \| = ? \text{ bits} \]

a smaller catalog

1.62m entropy of the original class
More clever decomposition strategy... using "canonical" spanning trees

\[ \| A \| \ll 2^{2.175r} \]

A smaller catalog

We decompose into micro trees

We count in term of vertices

1.62m is the entropy of the original class
Succinct planar maps
(Castelli Aleardi, Devillers et Schaeffer, SoCG06)

Theorem
For some classes of planar maps (triangulations and 3-connected planar maps) there exists an optimal succinct representation, supporting navigation in $O(1)$ time. The memory cost matches asymptotically the entropy:

- $1.62m$ bits, for triangulations of the sphere with $m$ faces;
- $2e$ bits, for 3-connected graphs with $e$ edges.
Planar maps and trees

(planar triangulations, Poulalhon et Schaeffer 2003)

$n + 2$ nodes

$n$ nodes
exactly 2 stems per node

(3-connected planar maps, Fusy Poulalhon et Schaeffer 2005)

3-connected planar graph

quadrangulation

binary tree
A new local closure operation

1101000100000000000
1001010000
1
1
1
1

0
0
0
0
0
0
0
0
0

1101000100000000000
1001010000
1001010000

1
1
1
1

1
1
1
1

1
1
1
1

1
1
1
1

1
1
1
1

1
1
1
1
A better decomposition strategy
memory cost: dominant term

\[ \| A_r \| = \| \sum_w A_{r,w} \| \]

\[ \| A_{r,w} \| = 2^{3.24r} \cdot \binom{2r}{w} \]

3.24r + w \log r \text{ bits}

cost of a reference

3.24n or 1.62m

entropy of the original class
memory cost: dominant term

\[ \sum_i r_i \approx n \]
\[ \sum_i w_i = O\left(\frac{n}{\lg n}\right) \]

\[ \sum_j 3.24r_i + w_i \lg r_i = 3.24n + o(n) \text{ bits} \]

overall cost of pointers

3.24n ou 1.62m
entropy of the original class
## Compact representations of graphs

### Comparison of existing results

<table>
<thead>
<tr>
<th>Encoding</th>
<th>queries</th>
<th>3-connected</th>
<th>triangl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacobson (Focs89)</td>
<td>$O(\lg n)$</td>
<td>64$n$</td>
<td>64$n$</td>
</tr>
<tr>
<td>Munro Raman (Focs97)</td>
<td>$O(1)$</td>
<td>$8n + 2e$</td>
<td>7$m$</td>
</tr>
<tr>
<td>Chuang et al. (Icalp98)</td>
<td>$O(1)$</td>
<td>$2e + 2n$</td>
<td>3.5$m$</td>
</tr>
<tr>
<td>Chiang et al. (Soda01)</td>
<td>$O(1)$</td>
<td>$2e + 2n$</td>
<td>4$m$</td>
</tr>
<tr>
<td>Blandford et al. (Soda03)</td>
<td>$O(1)$</td>
<td>$O(n)$</td>
<td>$O(m)$</td>
</tr>
<tr>
<td>Castelli et al. (Wads05 Cccg05)</td>
<td>$O(1)$</td>
<td>no</td>
<td>2.175$m$</td>
</tr>
<tr>
<td>Castelli et al. (SoCG06)</td>
<td>$O(1)$</td>
<td>$2e$</td>
<td>1.62$m$</td>
</tr>
</tbody>
</table>
A dynamic representation
Succinct dynamic triangulations

**Theorem** (Castelli Aleardi, Devillers et Schaeffer, CCCG05)

For triangulations with a boundary having $m$ faces, it is possible to maintain a succinct representation under insertion/deletion of vertices and edge flips, while supporting local navigation in $O(1)$ time. The size of the representation is still asymptotically optimal:

\[ 2.175m + o(m) \text{ bits} \]

The cost for an update is:

- $O(1)$ amortized for the vertex insertion;
- $O(\lg^2 m)$ amortized time for vertex deletion and edge flip.
better use of memory (dynamic arrays)

\[ \| t_{ij} \| = r + 4 \]
Succinct dynamic triangulations

- more efficient decomposition strategies
And in practice?

\[2.175m + 36(g - 1) \lg m + O\left(m \frac{\lg \lg m}{\lg m} + g \lg \lg m\right) \text{ bits}\]
A practical implementation
(joint work with Abdelkrim Mebarki, CCCG06)

\[ \frac{1}{12} \log m \text{ to } \frac{1}{4} \log m \]

usual structure
6 references to triangles
1 reference/vertex

Catalog
8 references/quadrangle

gain 9/13
And the geometry?
External geometric data

too many multiple references

$\lg n$ bits

multiple nodes

$\lg n$ bits

$\lg \lg n$ bits
futur work
Futur work

- Extension to higher genus surfaces
- Volume meshes

Encoding the geometry