RÉPÉSENTATION SUCCINTE DE TRIANGULATIONS
13 OCTOBRE 2005 - IHP

Luca Castelli Aleardi

(en collaboration avec Olivier Devillers et Gilles Schaeffer)

Projet Geometrica
INRIA Sophia-Antipolis

LIX
Ecole Polytechnique
Succinct and compact representations

Given a class $C_m$ of objects of size $m$, the goal is to design a space efficient data structure such that:

- queries on objects are answered in constant time;
- the encoding is succinct: the cost of an object $R \in C_m$ matches asymptotically the entropy of the class

$$\text{size}(R) = \log_2 \|C_m\|(1 + o(1))$$

- or compact: the content of a cost

$$\text{size}(R) = O(\|C_m\|)$$

- for dynamic data structures: updates are supported in

$$O(\lg^c m)$$ amortized time
Compact representations: an example

Rooted trees with $n$ vertices

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

enumeration of binary trees with $n$ vertices:
Compact representations: an example

compact encoding for compression

• size: $\log_2 \| B_n \| = 2n + O(\lg n)$ bits

• no efficient navigation

explicit pointers-based representation

• size: $2n \lg n$ bits

• constant time navigation

succinct representation (Jacobson 89, Munro et Raman 97)

• size: $2n + o(n)$ bits

• adjacency queries in constant time
Motivation

Combinatorial information describing incidence relations

Which information?

Connectivity
Motivation

Geometry information (vertex coordinates)

Which information?

Connectivity
Geometry
Motivation

Mesh compression algorithms

Edgebreaker [Rossignac]
Touma Gotsman
Poulhalon Schaeffer

General underlying idea
Encoding strategies based on a local (global) conquest
Motivation

Mesh compression algorithms

VRML, 288 or 114 bits/vertex

[Touma Gotsman] 2 bits/vertex (near-optimal)
[Poulalhon Schaeffer] 3.24 bits/vertex (optimal)

Compact representation

Pointer based representation: 208 bits/triangle

2.175 bits/triangle
Previous and related works

- static trees on $n$ nodes (Jacobson FOCS89): space $2n + o(n)$, navigation in $O(\lg n)$ time;

- planar graphs on $n$ vertices and $e$ edges (Munro Raman FOCS97): space $8n + 2e$, $O(1)$ time navigation;

- 3-connected planar graphs on $n$ vertices (Chuang et al. ICALP98): space $2e + n$, $O(1)$ time navigation;

- separable graphs (Blandford et al. SODA03): space $O(n)$, navigation in $O(1)$ time.

- dynamic binary trees (Munro et al. SODA01, Raman Rao ICALP03): space $2n + o(n)$, navigation in $O(1)$ updates in poly-logarithmic amortized time;
Tutte’s entropy (triangulations)

(information theory asymptotic lower bound)

enumeration of rooted planar triangulations on \( n \) vertices:

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

Tutte’s entropy (1962):

\[
e = \frac{1}{n} \log_2 \Psi_n \approx \log_2 \left(\frac{256}{27}\right) \approx 3.2451 \text{ bits/vertex}
\]
Planar Triangulations with a boundary

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

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

$$f'(m, k) = \frac{2 \cdot (2k - 3)! (2m - 1)!}{(k - 1)! (k - 3)! (\frac{m-k}{2} + 1)!}$$

counting planar triangulations 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
Our contribution

(presented at WADS 2005)

**Theorem.** For planar triangulations with a boundary having \( m \) faces, there exists an optimal succinct representation supporting efficient navigation in \( O(1) \) time, requiring

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

For triangulations of genus \( g \) surfaces \( (g = o(\frac{m}{\lg m})) \) the same representation requires

\[
2.175m + 36(g - 1) \lg m + O(m \frac{\lg \lg m}{\lg m} + g \lg \lg m) \text{ bits}
\]
Comparison: space efficiency

Compact representations of triangulations with $n$ vertices, $e$ edges, $m$ faces (lower order term are omitted)

<table>
<thead>
<tr>
<th>Encoding</th>
<th>queries</th>
<th>planar</th>
<th>higher genus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacobson (FOCS 89)</td>
<td>$O(\lg n)$</td>
<td></td>
<td>no</td>
</tr>
<tr>
<td>Munro Raman (FOCS 97)</td>
<td>$O(1)$</td>
<td>$8n + 2e$ or $7m$</td>
<td>no</td>
</tr>
<tr>
<td>Chuang et al. (ICALP 98)</td>
<td>$O(1)$</td>
<td>$2e + n$ or $3.5m$</td>
<td>no</td>
</tr>
<tr>
<td>Chiang et al. (SODA 01)</td>
<td>$O(1)$</td>
<td>$2e + n$ or $3.5m$</td>
<td>no</td>
</tr>
<tr>
<td><strong>our encoding</strong></td>
<td>$O(1)$</td>
<td>$2.175m$</td>
<td>$2.175m$</td>
</tr>
</tbody>
</table>
Basic ideas

- Multi-level hierarchical structure
- Exhaustive enumeration
- Optimal encoding
- Information sharing
Au cours d’une leçon privée en préparation du doctorat total, une jeune fille discute d’arithmétique avec son professeur.

(le professeur) Écoutez-moi, Mademoiselle, si vous n’arrivez pas à comprendre profondément ces principes, ces archétypes arithmétiques, vous n’arriverez jamais à faire correctement un travail de polytechnicien... combien font, par exemple, 3.755.918.261 multiplié par 5.162.303.508?

(l’élève, très vite) Ça fait 193891900145...

(le professeur, de plus en plus étonné calcule mentalement) Oui... Vous avez raison... le produit est bien... (il bredouille inintelligiblement)... Mais comment le savez-vous, si vous ne connaissez pas les principes du raisonnement arithmétique?

(l’élève) C’est simple. Ne pouvant me fier à mon raisonnement, j’ai appris par cœur tous les résultats possibles de toutes les multiplications possibles.
Decomposing $\mathcal{T}$ into sub-triangulations

- we compute **small triangulations** having between $\frac{1}{3} \lg^2 m$ and $\lg^2 m$ triangles;
- we decompose small triangulations into **tiny triangulations** containing between $\frac{1}{12} \lg m$ and $\frac{1}{4} \lg m$ triangles.
Decomposition phase

We start with a triangulation having $m$ triangles
Decomposition phase

Computing tiny triangulations having $\Theta(\lg m)$ triangles
Decomposition phase

There are $\Theta\left(\frac{m}{\lg m}\right)$ tiny triangulations
Decomposition phase

Only boundary edges are shared by tiny triangulations
Decomposition phase

Graph $G$ linking adjacent tiny triangulations
Decomposition phase

A small triangulation contains $\Theta(lg^2 m)$ triangles
Decomposition phase

There are $\Theta\left(\frac{m}{\lg^2 m}\right)$ small triangulations
Decomposition phase

Graph $F$ linking adjacent small triangulations
Decomposition phase

Partitioning graph $G$: graphs $G_i$ link tiny triangulations lying in a same small triangulation
Overview: representation of a small triangulation

- adjacency relations are described by map $G_i$;
- internal connectivity is implicitly represented (variable size pointers);
- boundary neighboring relations are represented by boundary coloring (variable length bit-vector).
Graph $G_i$ linking adjacent tiny triangulations

- $G_i$ has a node for each tiny triangulation and an arc for each pair of adjacent tiny triangulations;
- $G_i$ is a planar map, having faces of degree at least 3, multiple edges and loops are allowed;
adjacency relations between tiny triangulations

- Because of Euler’s formula, the overall number of arcs in maps $G_i$ is:

$$\sum_i \|E(G_i)\| = O\left(\frac{m}{\lg m}\right)$$
Decomposition

Initial small triangulation with a dual spanning tree
Decomposition

The tree is decomposed into tiny trees of size $\Theta(\lg m)$.
Decomposition

We get tiny triangulations of size $\Theta(\lg m)$
Decomposition

A small triangulations contains $\Theta(\lg m)$ tiny triangulations
Decomposition

A small triangulations contains $\Theta(\lg m)$ tiny triangulations
Memory organization

Graph of tiny triangulations

node of $G$

node of $F$

local pointer

global pointer

$G$

$F$

node of $G$

$F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$

$F$

node of $F$
Memory organization

Graph of tiny triangulations

node of $G$

degree  # triangles  # boundary
neighbor neighbor neighbor neighbor

Description of the triangulation
Memory organization

Graph of tiny triangulations

Node of $G$

degree # triangles # boundary

neighbor neighbor neighbor neighbor

Description of the triangulation

Pointers to the catalog of the triangulations with $t$ triangles
Graph of tiny triangulations

node of $G$

degree  # triangles  # boundary
neighbor neighbor neighbor neighbor

Boundary coloring
Memory organization

Graph of tiny triangulations

node of $G$

degree  # triangles  # boundary

neighbor  neighbor  neighbor  neighbor

Boundary coloring

0 0 1 0 0 0 0 1

1 1 1 0 0 0 0 1
Memory organization

Graph of tiny triangulations

node of $G$

Rep\`esentation succinte de triangulations – p.39/56
Memory organization

Graph of tiny triangulations

node of $G$

node of $F$

local pointer

global pointer

$\Theta(lg lg m)$

$2 lg lg m$

$lg lg m$

index

backward index

neighbor neighbor neighbor neighbor

$\text{degree}$

voisin voisin voisin voisin

$\text{index}$
Catalog of tiny triangulations

$t$ triangles

$2^{2.17t}$ triangulations using each $t \lg t$ bits

$\frac{1}{4} \lg m \sum_{t=\frac{1}{12} \lg m} t \lg t \leq m^{0.55}$
Overall cost of graphs $G_i$

- list of neighbors, nodes degrees, size of nodes, ... (local pointers of size $O(\lg \lg m) - O(\frac{m}{\lg m})$ nodes and arcs)

$$O(m \frac{\lg \lg m}{\lg m})$$

- pointers to table $A_r$ (combinatorial information)

$$2.17m + O(\lg m)$$

- pointers to "Rank/Select" tables (boundary coloring)

$$\sum_t \|RS(t)\| \leq \sum_t \lg \left(\frac{\lg m}{w(t)}\right) \leq O(m \frac{\lg \lg m}{\lg m})$$
Total space used

- Catalog of all different tiny triangulations
  \[ O(m^{\frac{1}{4}2.17} \lg^2 m \lg \lg m) = o(m) \]
- catalog of bit-vectors (with Rank/Select)
  \[ O(m^{\frac{1}{4}2.17} \lg m \lg \lg m) = o(m) \]
- representation of graph \( F \):
  \[ O\left(\frac{m}{\lg^2 m} \lg m\right) = o(m) \]
- graphs \( G_i \)
  \[ 2.17m + O\left(m\frac{\lg \lg m}{\lg m}\right) \]
Navigation

node of map $F$

a node in map $G$

00100010011010

triangle index
Navigation

node of map $F$

a node in map $G$

triangle index

00100010011010
Navigation

node of map $F$

a node in map $G$

triangle index

00100010011010
retrieving side information

node of map $F$

a node in map $G$

triangle index

00100010011010

Navigation
Navigation

retrieving side information

node of map $F$

a node in map $G$

another node in $G$

triangle index

Répresentation succinte de triangulations – p.48/56
Navigation

retrieving side information

node of map $F$

a node in map $G$

another node in $G$

triangle index

Répresentation succinte de triangulations – p.49/56
Navigation

retrieving side information

node of map $F$

a node in map $G$

another node in $G$

triangle index
Navigation

Navigation in $O(1)$ time

node of map $F$

a node in map $G$

another node in $G$

triangle index
Concluding remarks

• Reducing storage requirements

Restraining the catalog to a sub-class (e.g. triangulations with bounded vertex degree) automatically reduces the entropy and the pointers size, and hence the amount of space used.

• Other local navigation operations

We can enrich our representation to allow for efficient queries on vertices (testing adjacency, vertex degree, turning around a vertex)

• Geometry information

With some slight modifications we can associate geometric data to faces and vertices
Theorem (Castelli Aleardi, Devillers and Schaeffer). For triangulations with a boundary having $m$ faces, it is possible to maintain a succinct representation under vertex insertion/deletion and edge flip, while supporting navigation in $O(1)$ time. The storage is

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

The cost for an update is:

- $O(1)$ amortized time for degree 3 vertex insertion;
- $O(\lg^2 m)$ amortized time for vertex deletion and edge flip;
A practical solution

(Abdelkrim Mebarki and Olivier Devillers)

C++ implementation based on CGAL library

Idea: gathering triangles in small groups
Proposition. For planar triangulations having \( m \) faces (3-connected planar graphs with \( e \) edges), it is possible to design a succinct representation which supports efficient navigation, requiring for the storage

\[
1.62m + o(m) \text{ bits (resp. } 2e + o(e) \text{ bits)}
\]

Main idea: a face decomposition based on optimal vertex spanning tree codings (Poulalhon/Schaeffer Icalp03, Fusy/Poulalhon/Schaeffer Soda05)
Future work

Triangulations 3D

Any idea?