Dynamic update of succinct triangulations

CCCG - august 2005

Luca Castelli Aleardi

(joint work with Olivier Devillers and Gilles Schaeffer)

Projet Geometrica
INRIA Sophia-Antipolis

LIX
Ecole Polytechnique

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

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

- or *compact*: we content of a cost

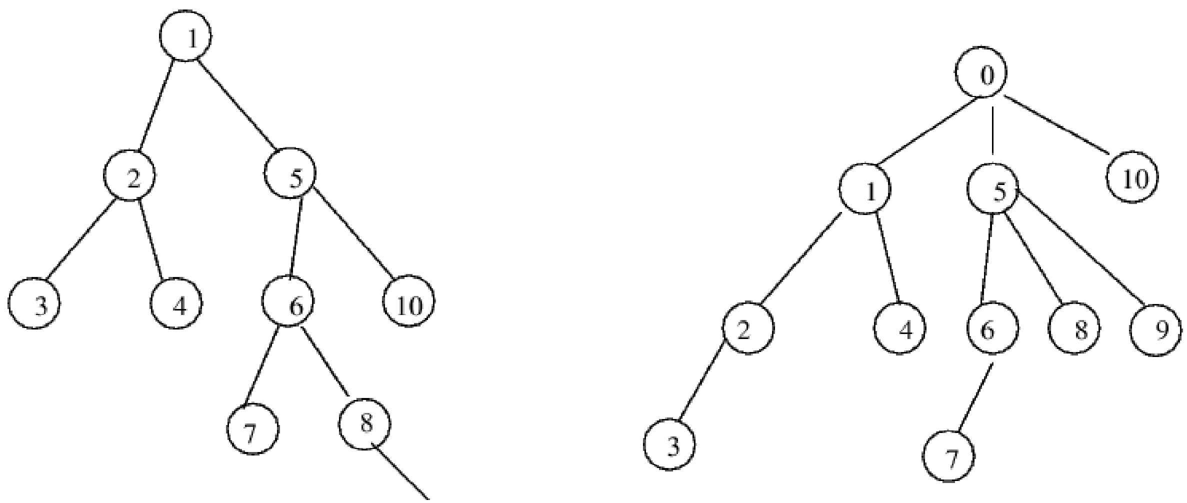
$$size(R) = O(\|C_m\|)$$

- for dynamic data structures: updates are supported in

$O(\lg^c m)$ amortized time

Compact representations

An example: rooted trees with n vertices



enumeration of binary trees with n vertices:

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

Compact representations

An example: rooted trees with n vertices

compact encoding for compression

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

Mesh compression versus compact representation

~~Mesh compression~~

VRML, 288 or 114 bits/vertex

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

[Poulalhon Schaeffer] 3.24bits/vertex (optimal)

Compact representation

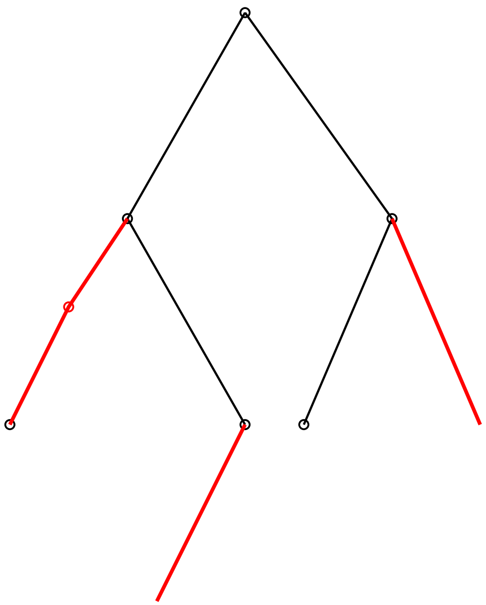
Pointer based representation: 208 bits/triangle

2.175 bits/triangle

Succinct dynamic data structures

Succinct dynamic binary trees on n nodes

Munro Raman Storm (SODA'01)

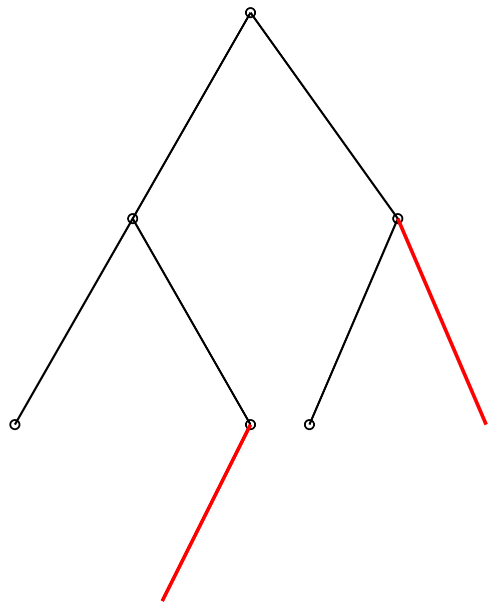


inserting/deleting a leaf

inserting a node along an edge

$O(\lg^2 n)$ amortized time

Raman Rao (ICALP'03)



inserting/deleting a leaf

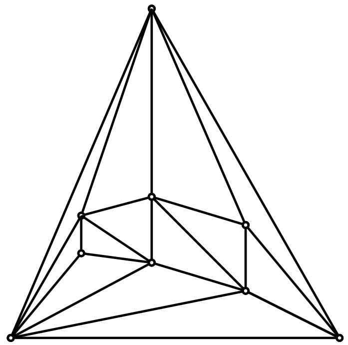
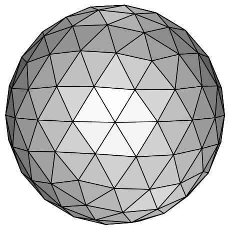
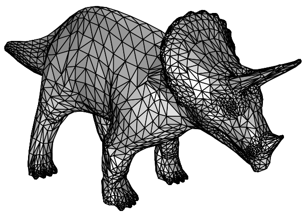
$O((\lg\lg n)^{1+\varepsilon})$ amortized ti

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)

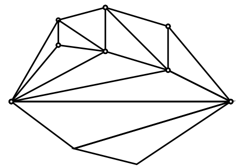
enumeration of rooted planar triangulations on n vertices:

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

Tutte's entropy (1962):

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

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

$$F(m) = \lg(\sum_{k\geq 3}^{m} f'(m,k)) \approx 2.175m$$

3.24 bits/vertex $= 1.62$ bits/face < 2.17 bits/face

Static succinct triangulations

(to be presented at WADS 2005)

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

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

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

$$2.175m + 36(g-1)\lg m + O(m\frac{\lg\lg m}{\lg m} + g\lg\lg m) \;\; bits$$

Comparison: space efficiency

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

Encoding	queries	planar	higher genus
Jacobson (FOCS 89)	$O(\lg n)$		no
Munro Raman (FOCS 97)	$O(1)$	$8n + 2e$ or $7m$	no
Chuang et al. (ICALP 98)	$O(1)$	$2e + n$ or $3.5m$	no
Chiang et al. (SODA 01)	$O(1)$	$2e + n$ or $3.5m$	no
Castelli Aleardi et al. WADS 2005	$O(1)$	$2.175m$	$2.175m$

Basic ideas

- Multi-level hierarchical structure
- Exhaustive enumeration
- Optimal encoding
- Information sharing

Additional tools for the dynamic case

- Memory organization based on *space efficient dynamic arrays*
- New strategy for local redecomposition

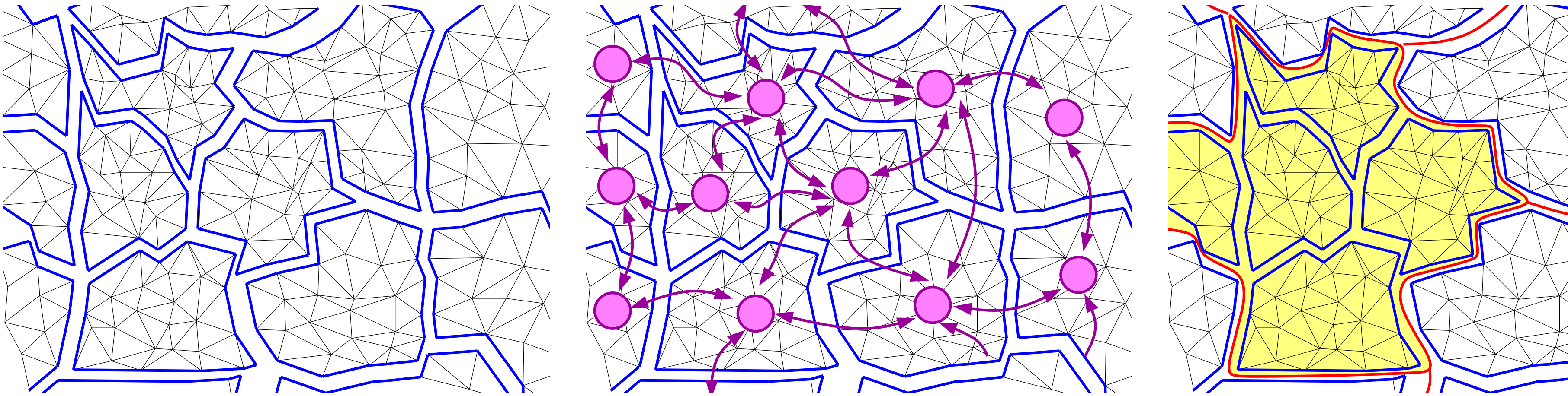
Literary digression

"The lesson", a Eugène Ionesco's play (1951)

During a private lesson, a very young student, preparing herself for the total doctorate, talks about arithmetics with her teacher.
(teacher) If you cannot deeply understand these principles, these arithmetic archetypes, you will never perform correctly a "polytechnicien" job. For example, what is $3.755.918.261$ multiplied by $5.162.303.508$?
(student, very quickly) The result is 193891900145...
(teacher, very astonished): How have you computed it, if you do not know the principles of arithmetic reasoning?
(student) Simple: I have learned by heart all possible results of all possible multiplications.

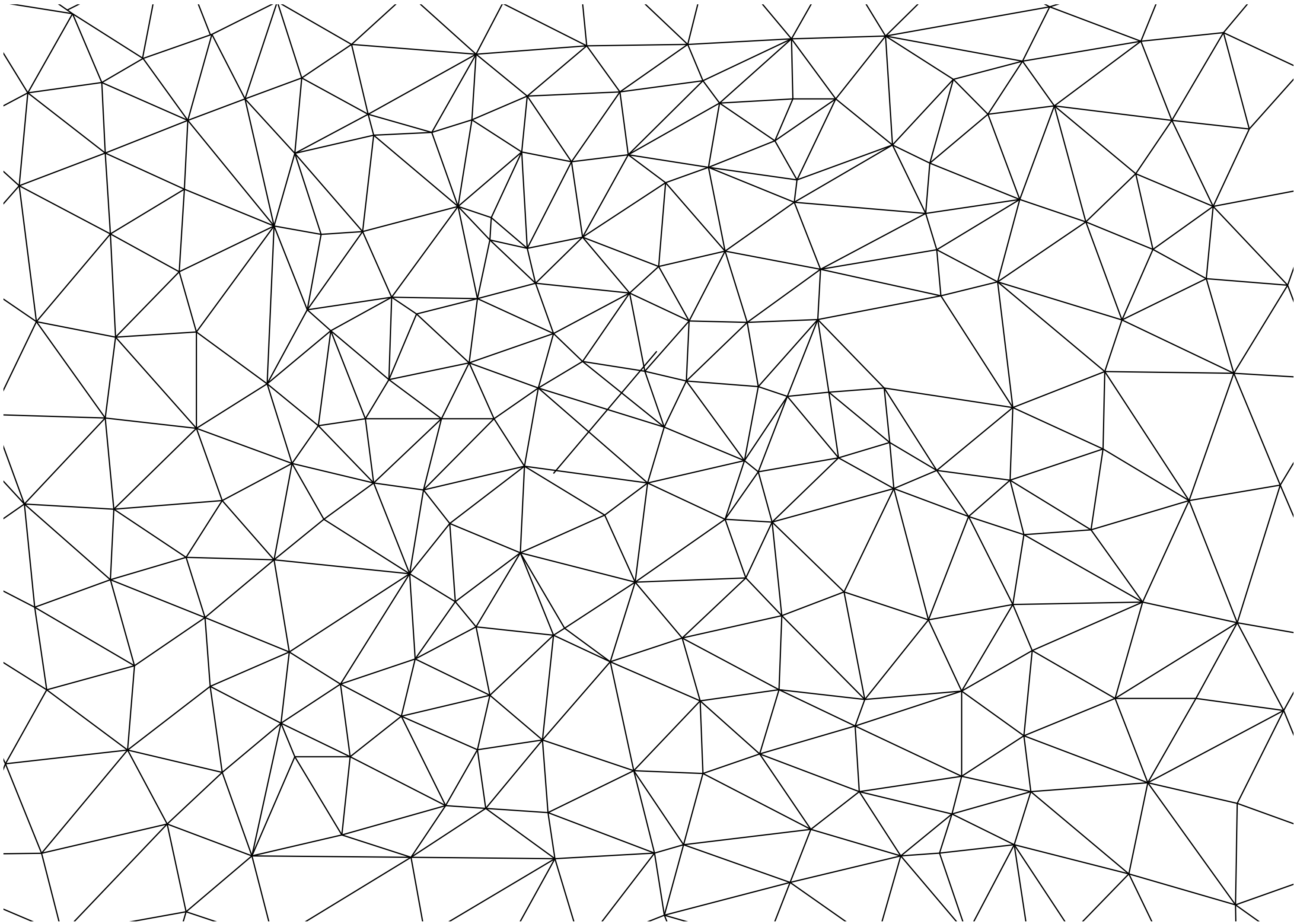
Decomposing $\mathcal{T}$ into sub-triangulation

- we compute tiny triangulations having between $\frac{1}{12} \lg m$ and $\frac{1}{4} \lg m$ triangles;
- we regroup tiny triangulations to form small triangulations containing $\Theta(\lg m)$ tiny triangulations.



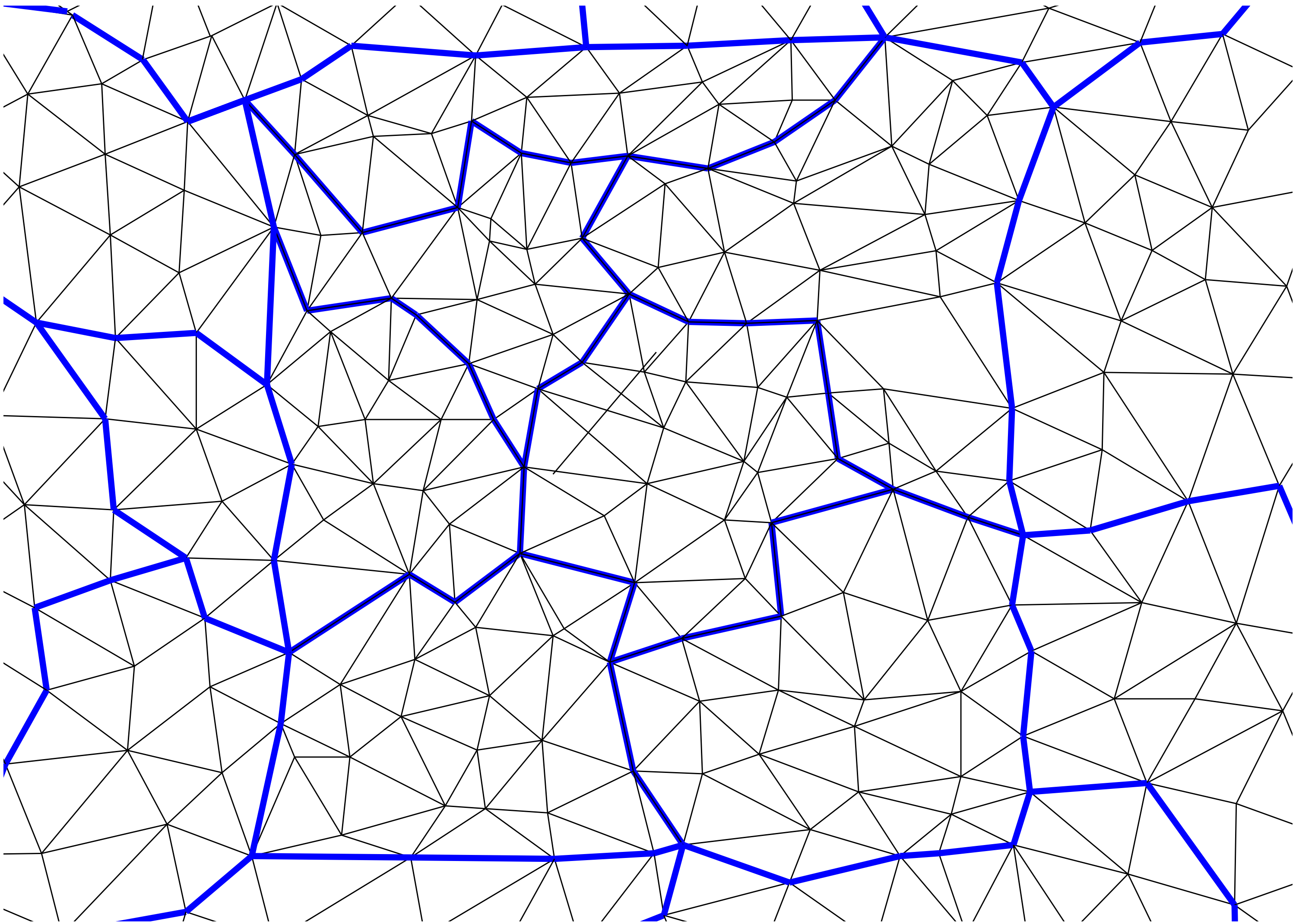
Decomposition phase

We start with a triangulation having m triangles



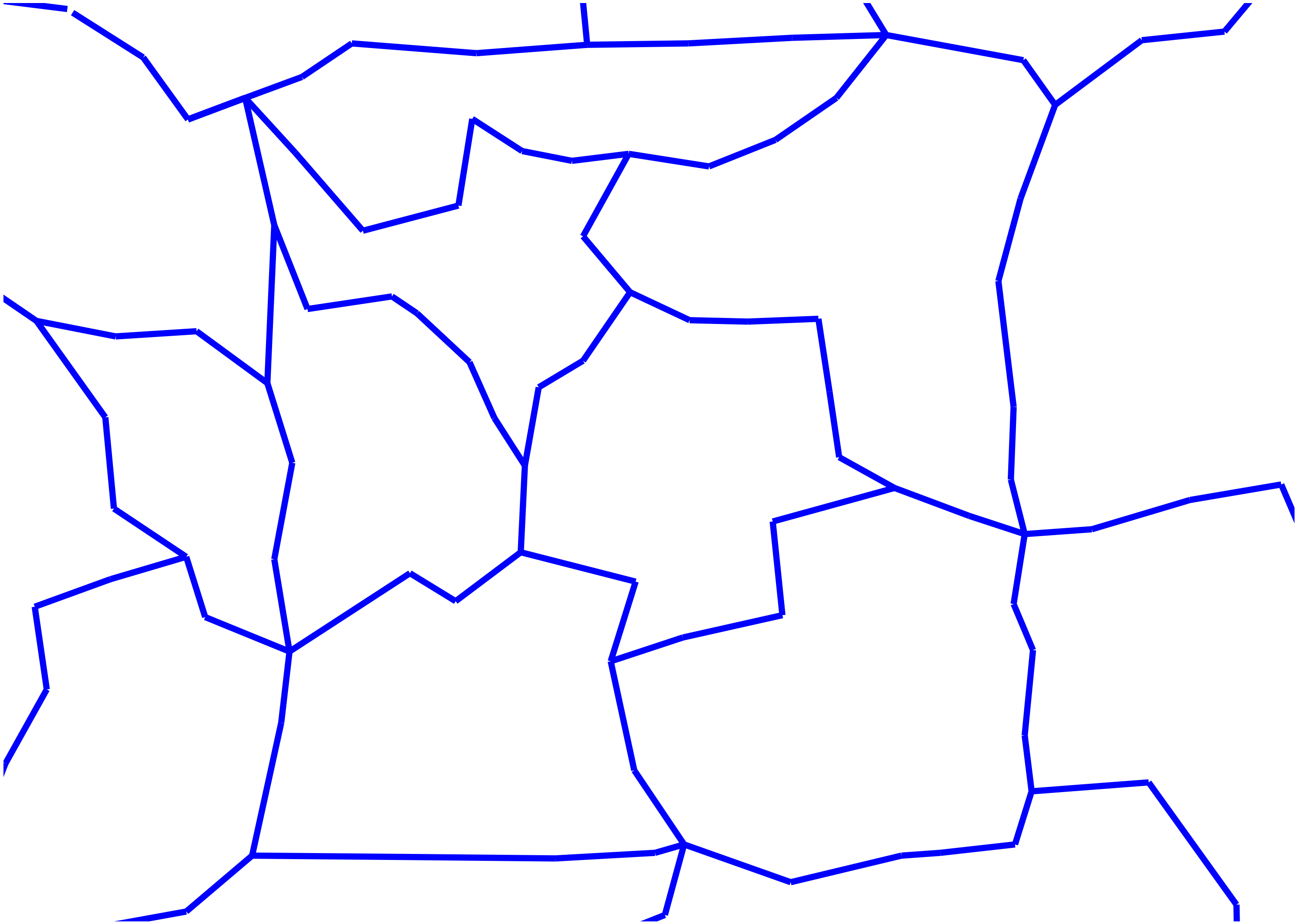
Decomposition phase

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



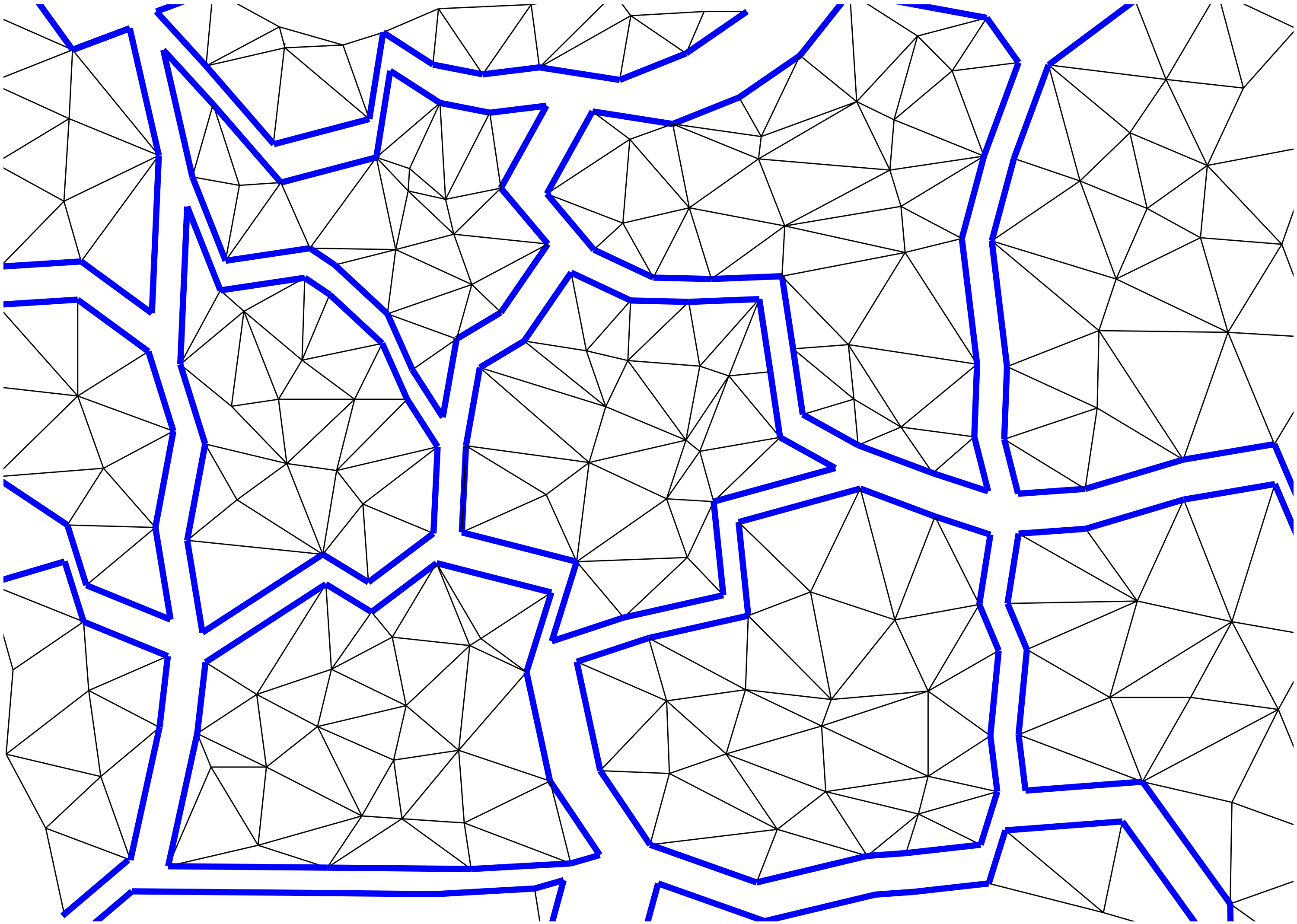
Decomposition phase

There are $\Theta(\frac{m}{\lg m})$ tiny triangulations



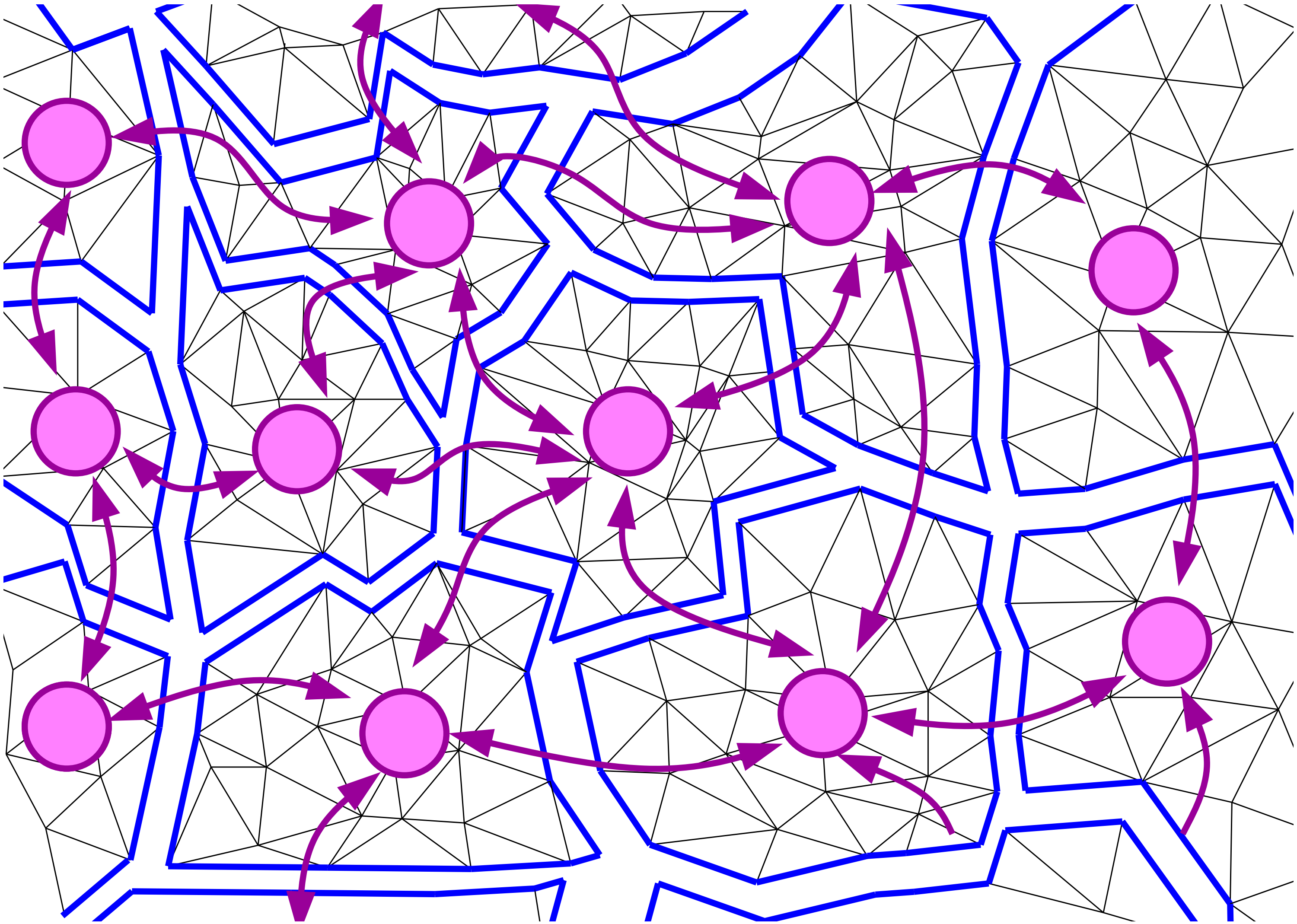
Decomposition phase

Only boundary edges are shared by tiny triangulations



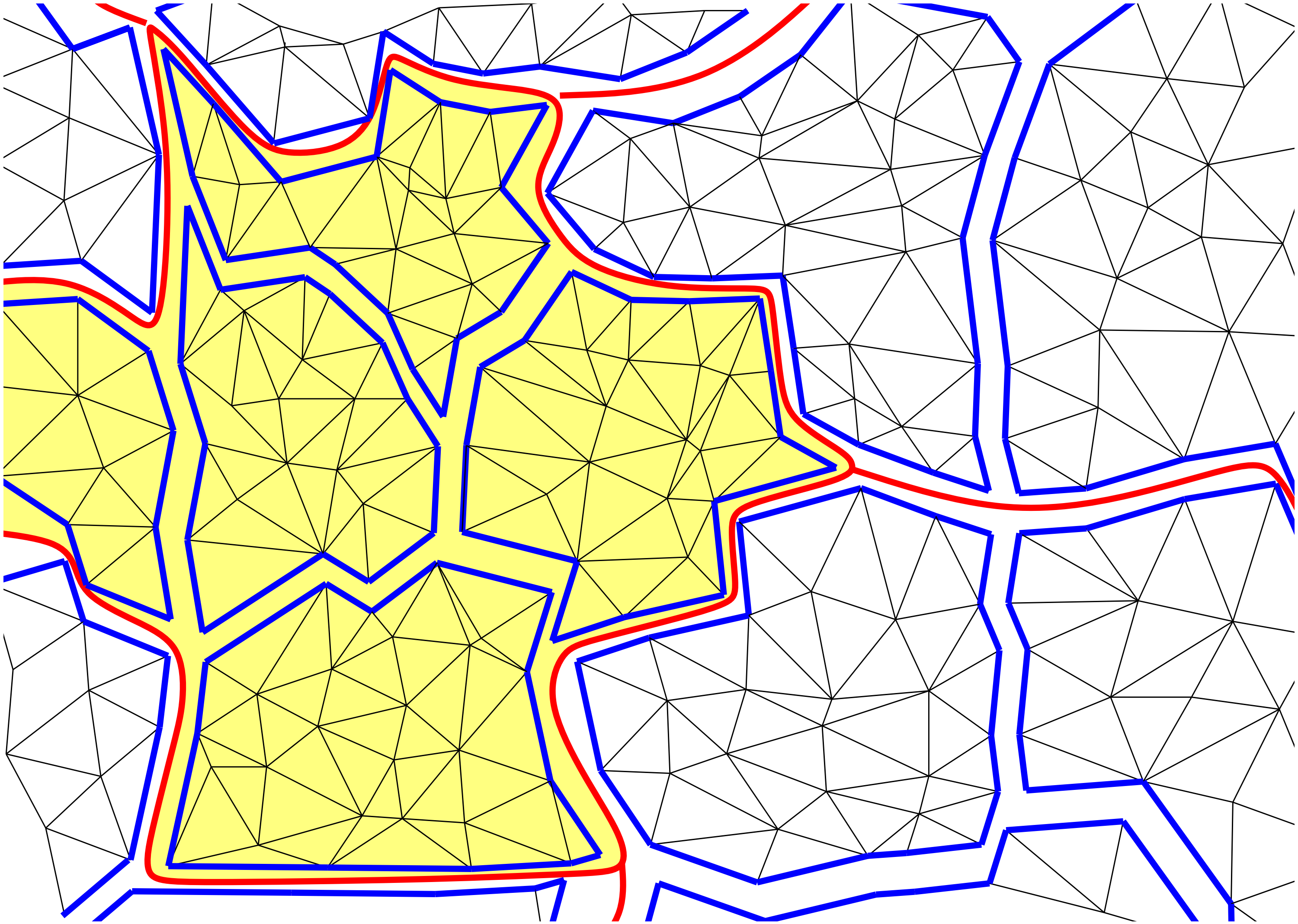
Decomposition phase

Graph G linking adjacent tiny triangulations



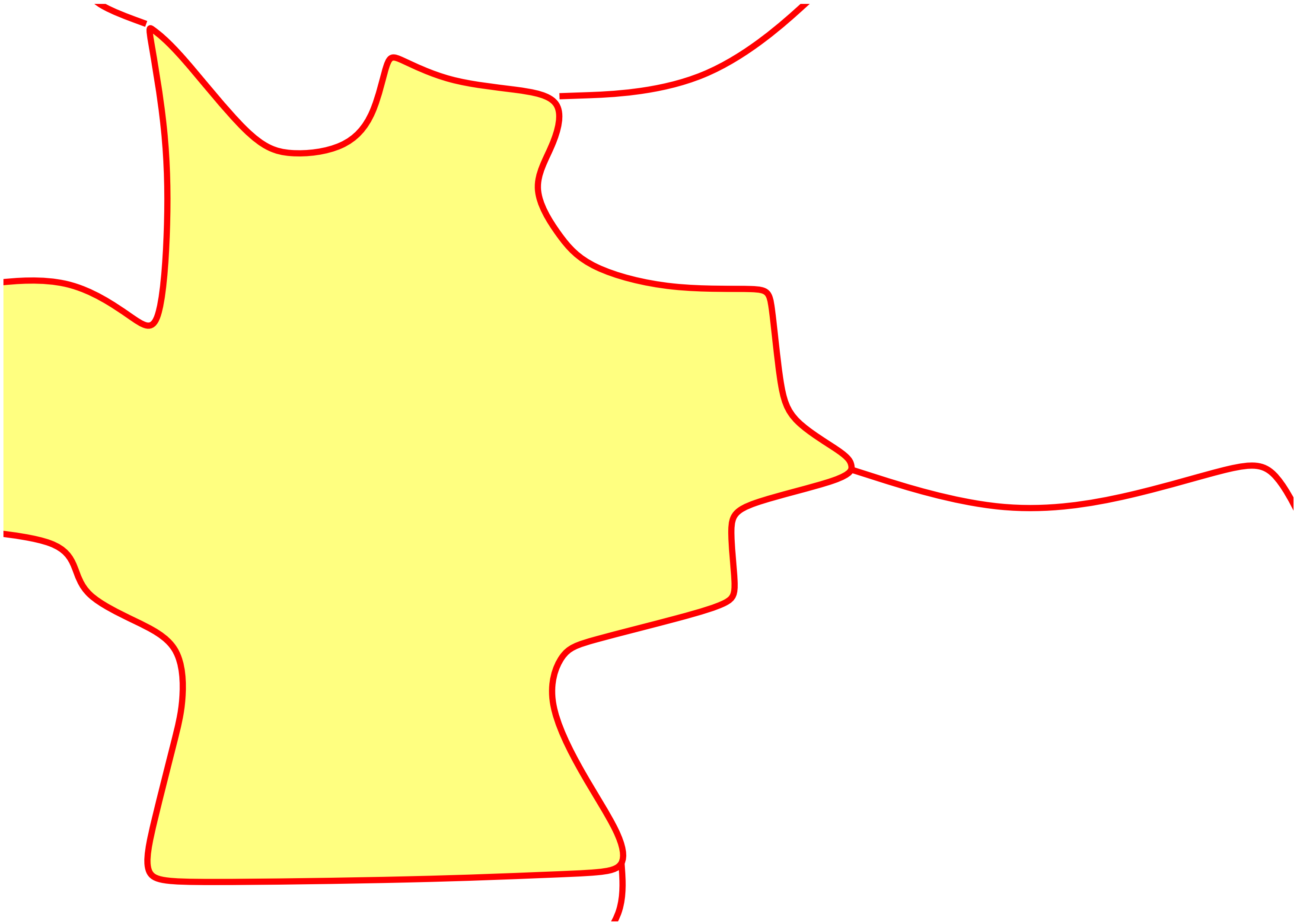
Decomposition phase

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



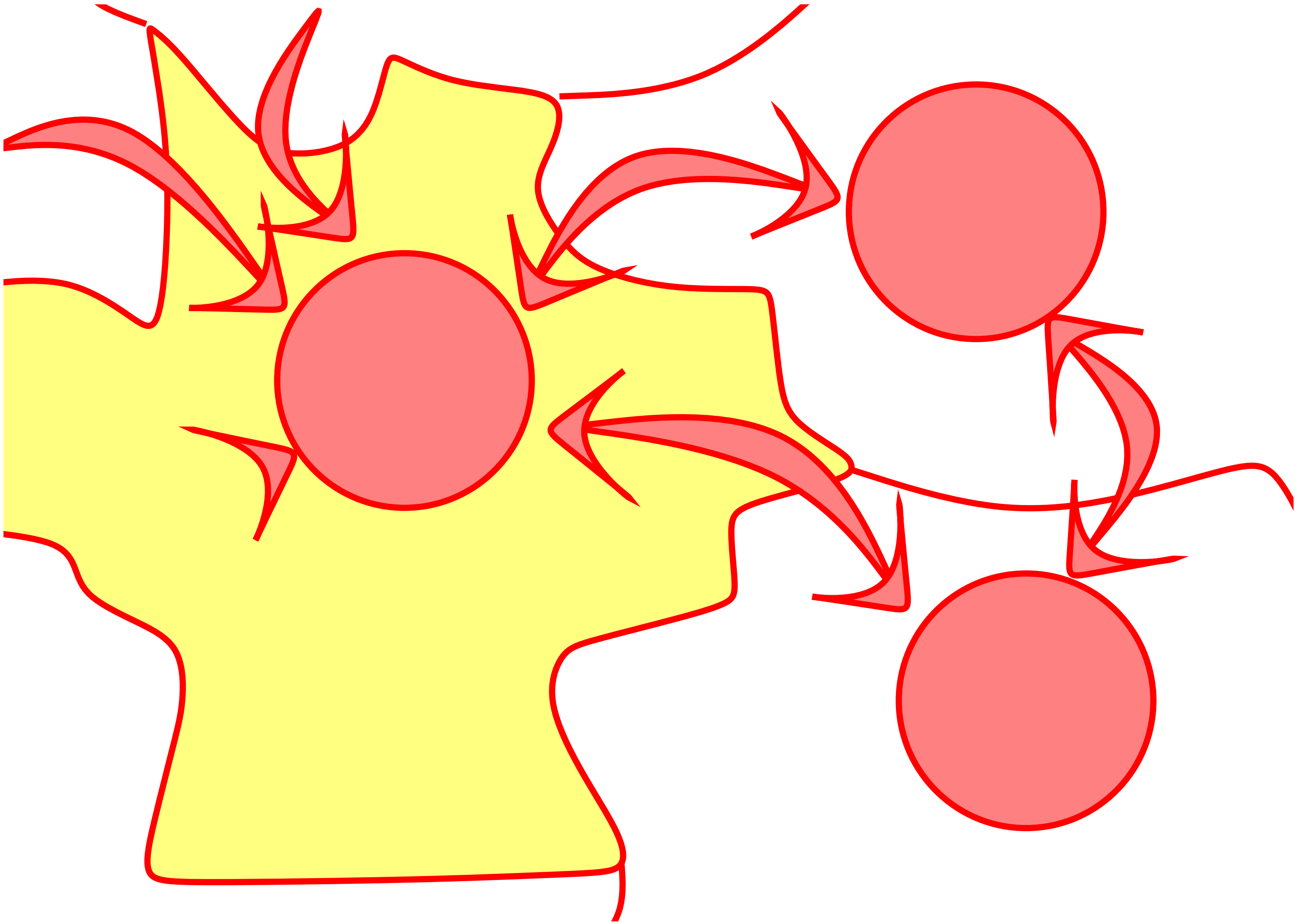
Decomposition phase

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



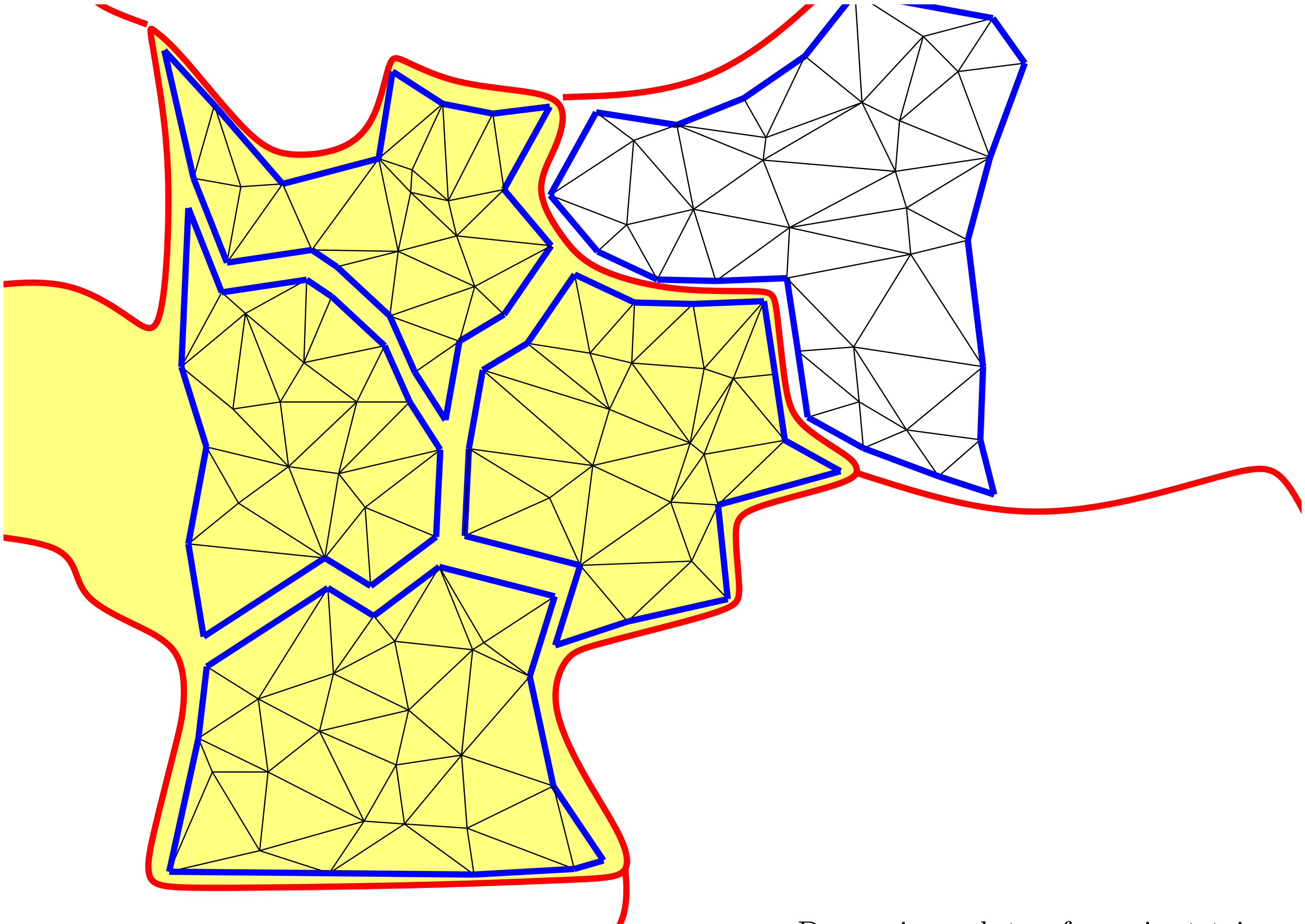
Decomposition phase

Graph F linking adjacent small triangulations



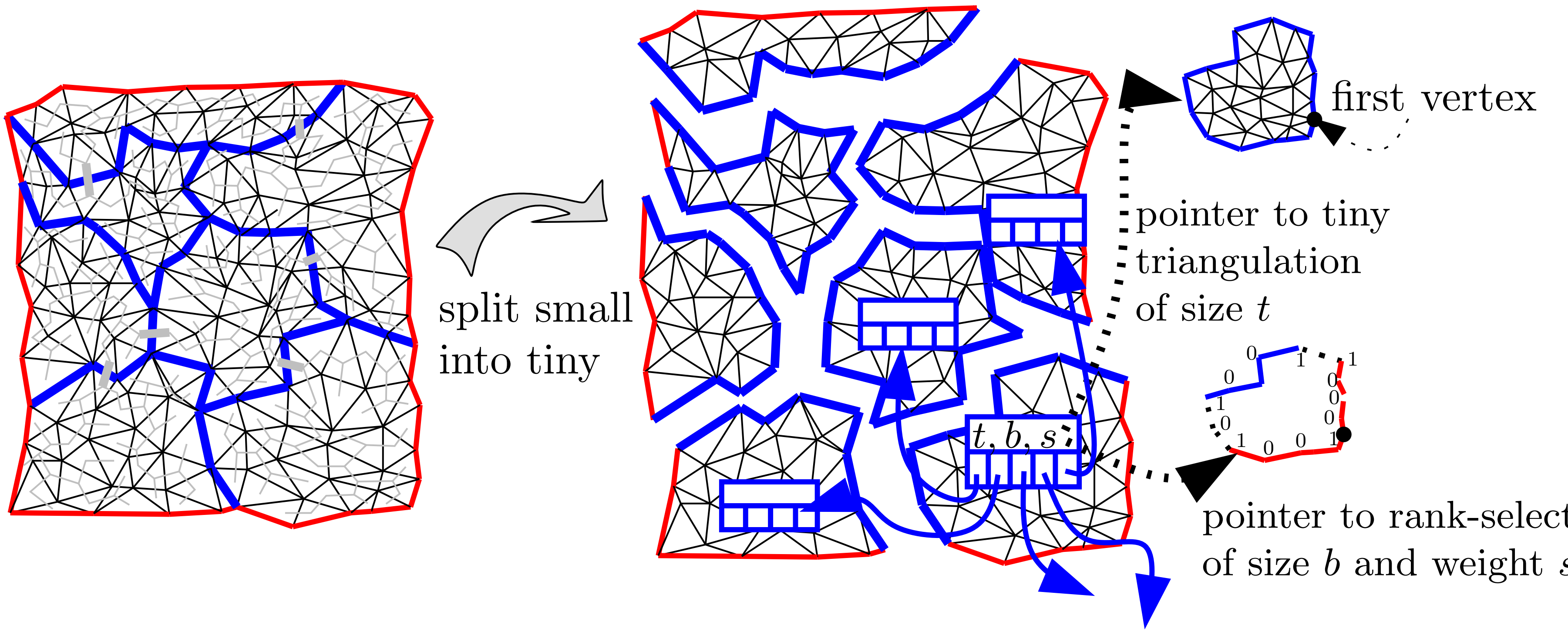
Decomposition phase

Partitioning graph G: graphs G_i link tiny triangulations lying in a same small triangulation



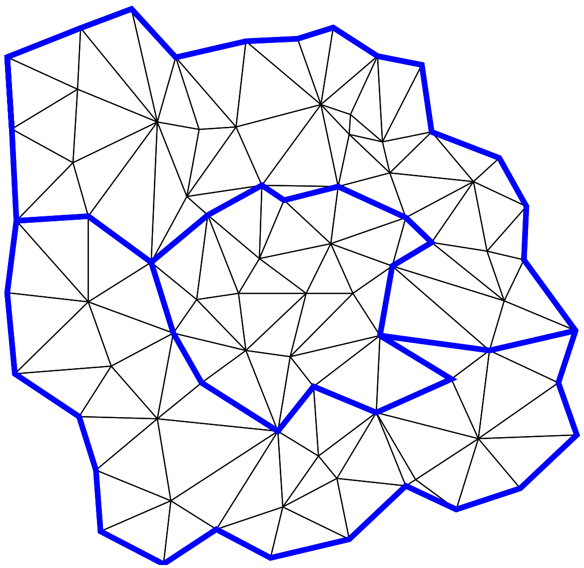
rview: representation of a small triangu

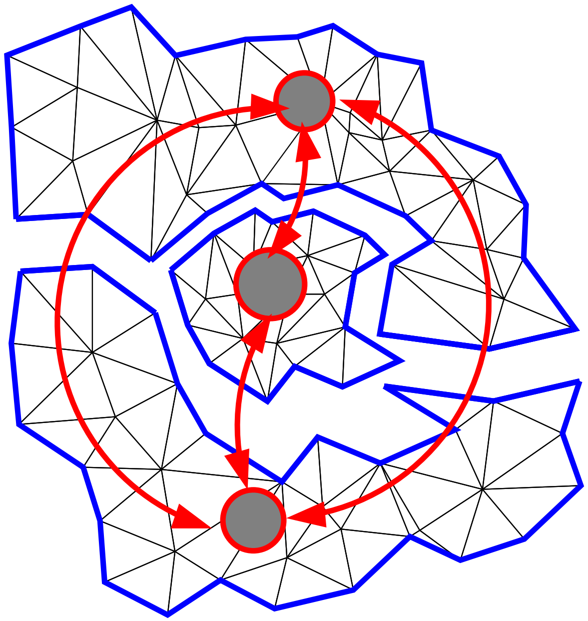
- adjacency relations are described by map G_i;
- internal connectivity is implicitly represented (variable size pointers)
- boundary neighboring relations are represented by boundary coloring (variable length bit-vector)



raph G_i linking adjacent tiny triangulat

- 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;

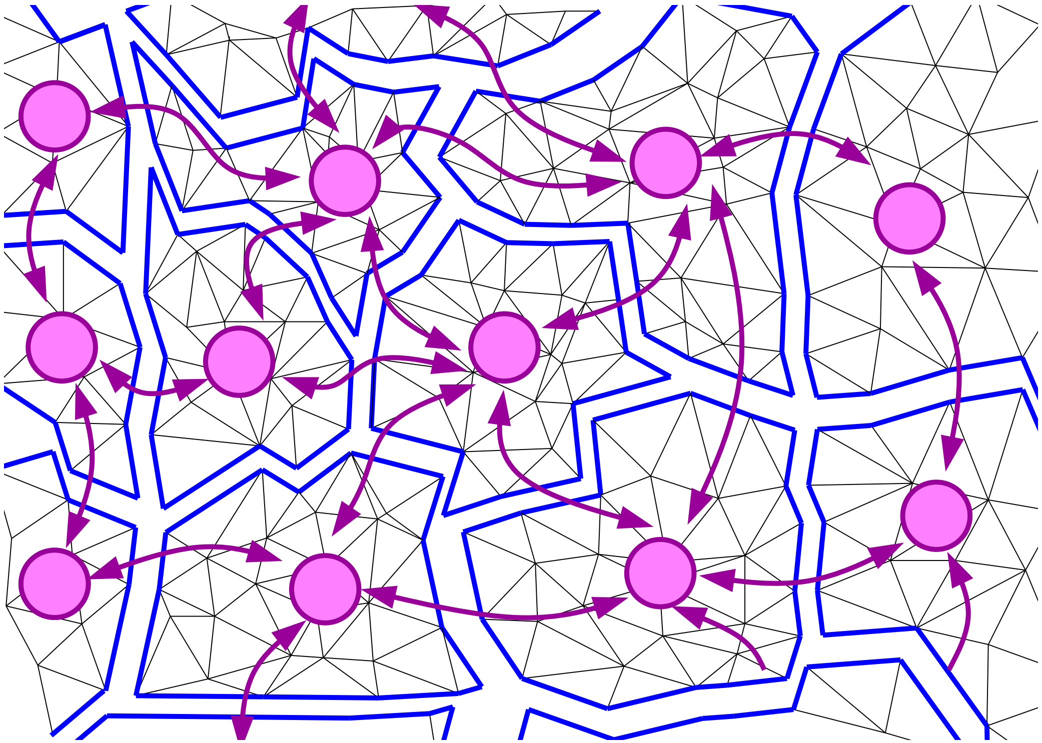




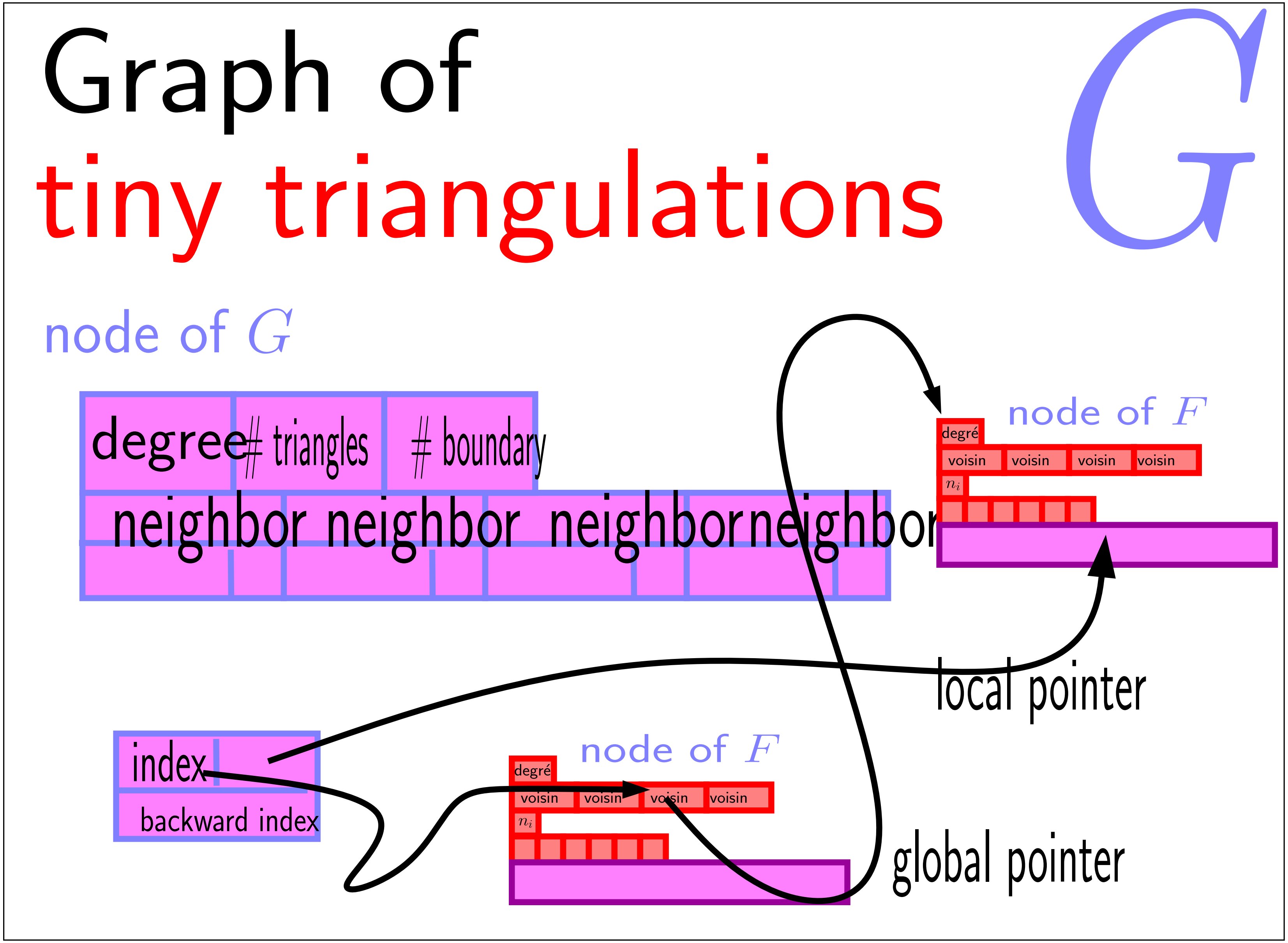
djacency relations between tiny triangu

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

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



Memory organization overview



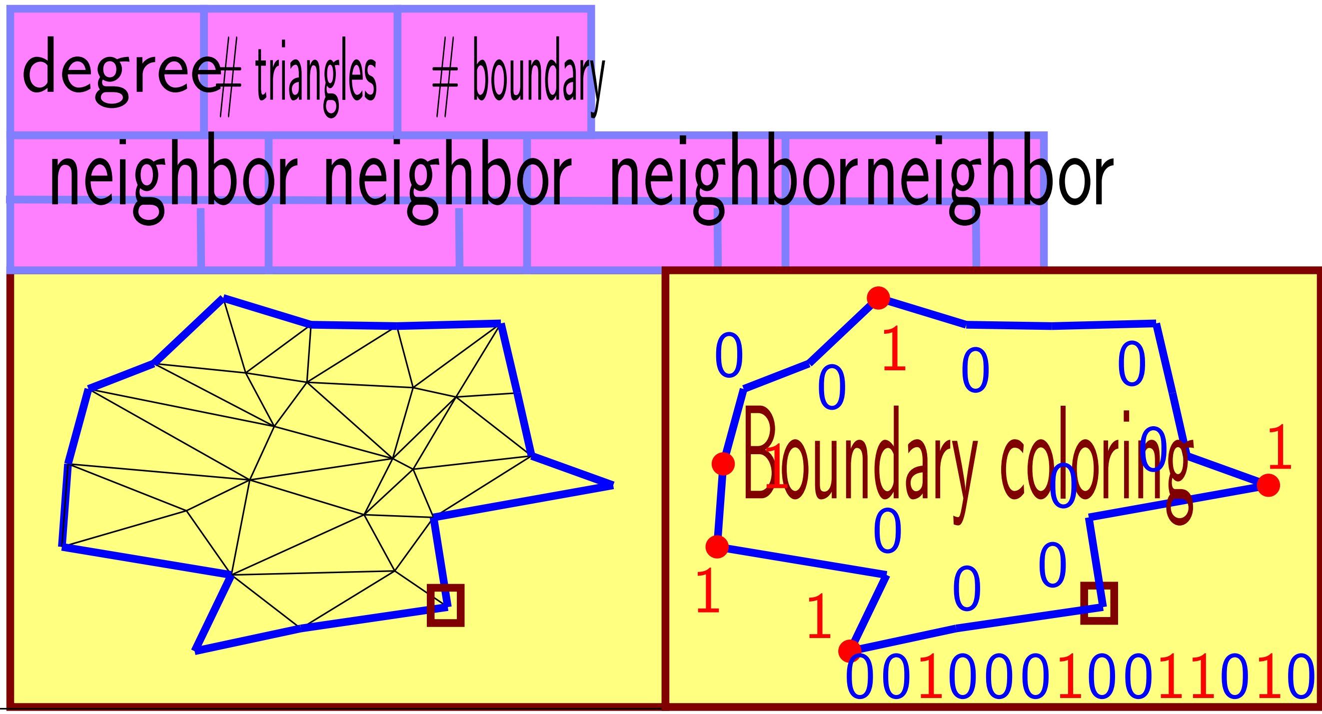
Memory organization overview

Graph of

tiny triangulations

G

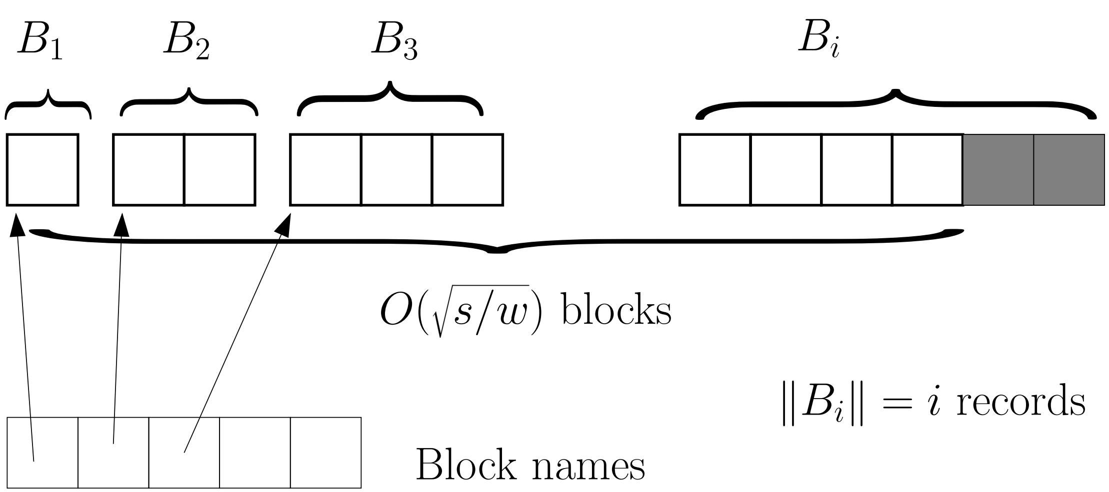
node of G



Space efficient dynamic data structur

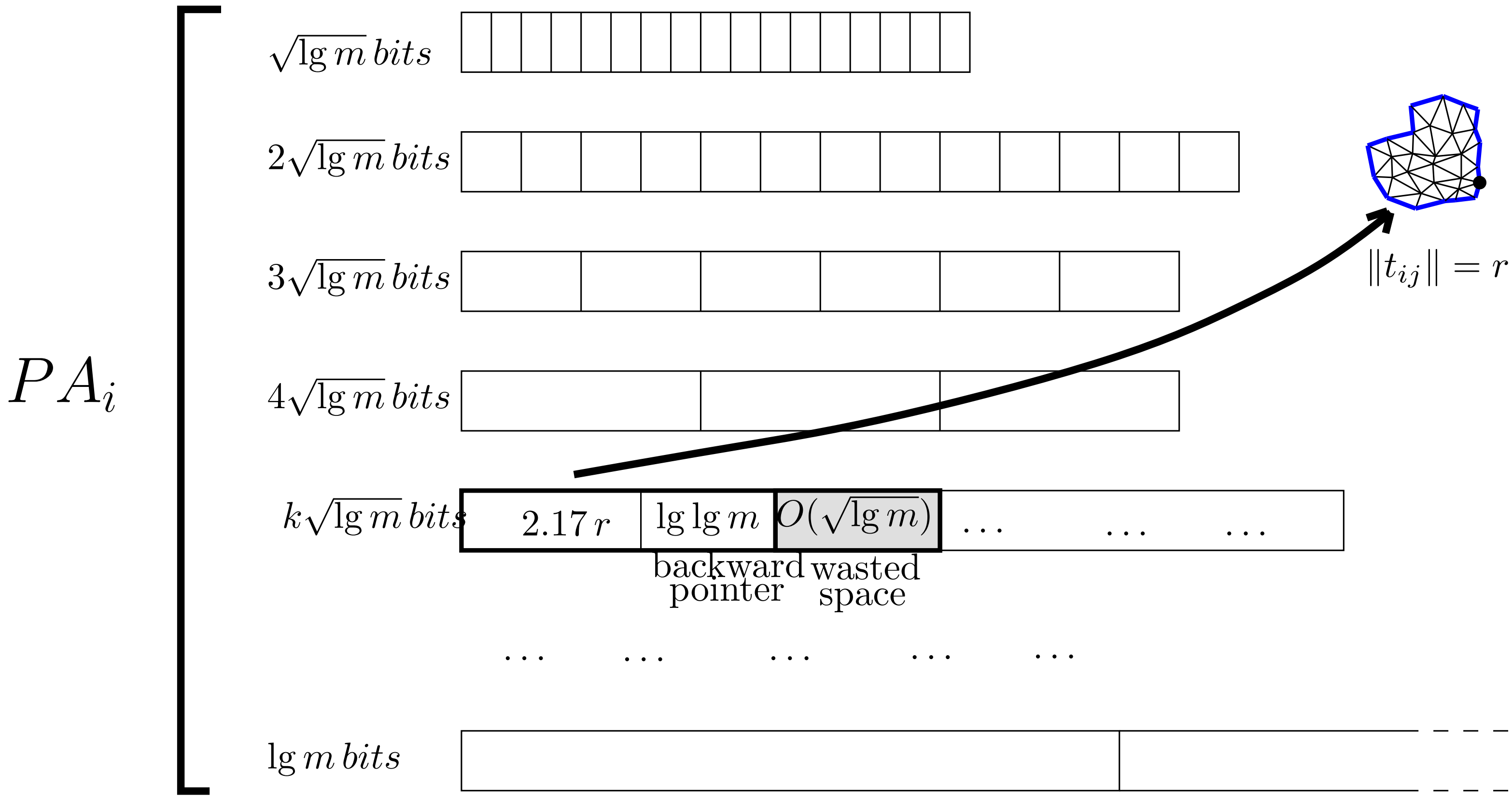
Extendible arrays (Raman Rao ICALP'03)

Proposition. *It is possible to maintain n records of r bits each under insertion of new records, while supporting access in $O(1)$ worst-case time. The updates (*grow *and* shrink*) are performed in $O(1)$ amortized time and the wasted space is $O(w + \sqrt{nrw})$ (w being the size of a word machine).*



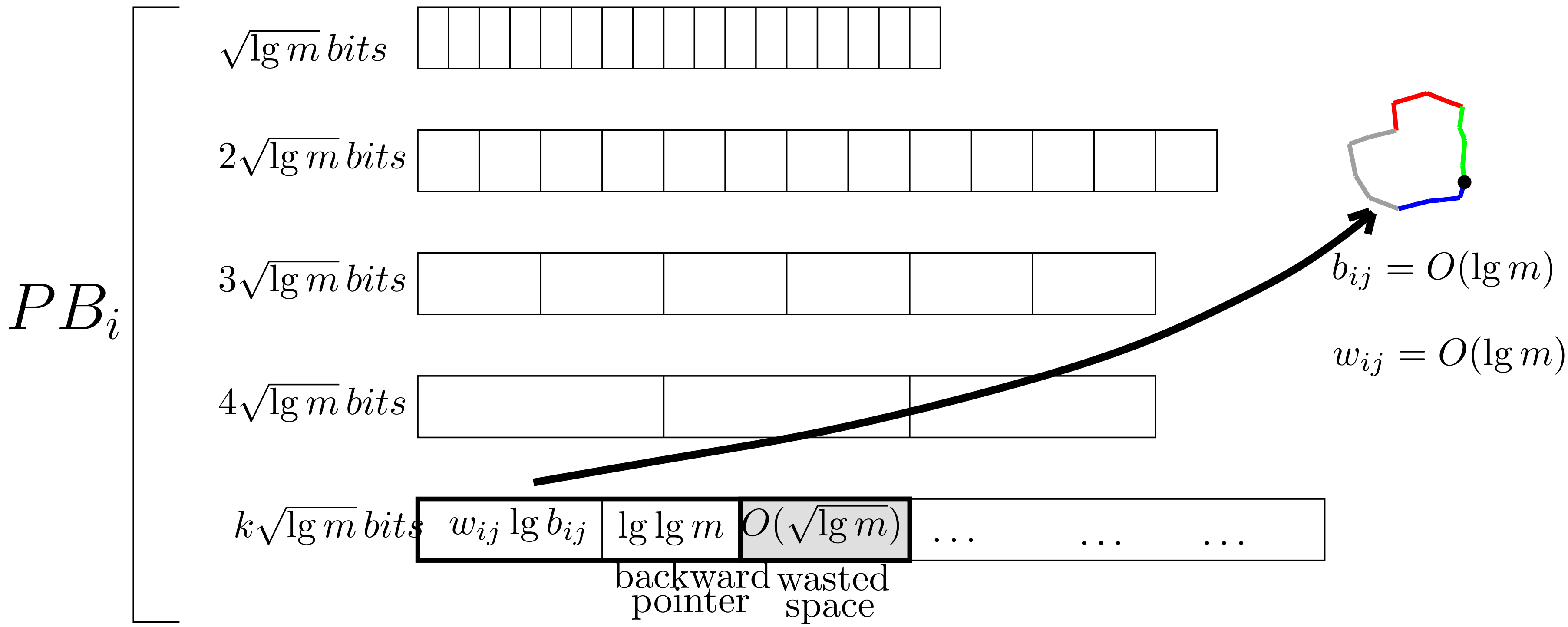
Memory organization

Collection of extendible arrays storing implicitly the tiny triangulations: $2.17r$ bits pointers



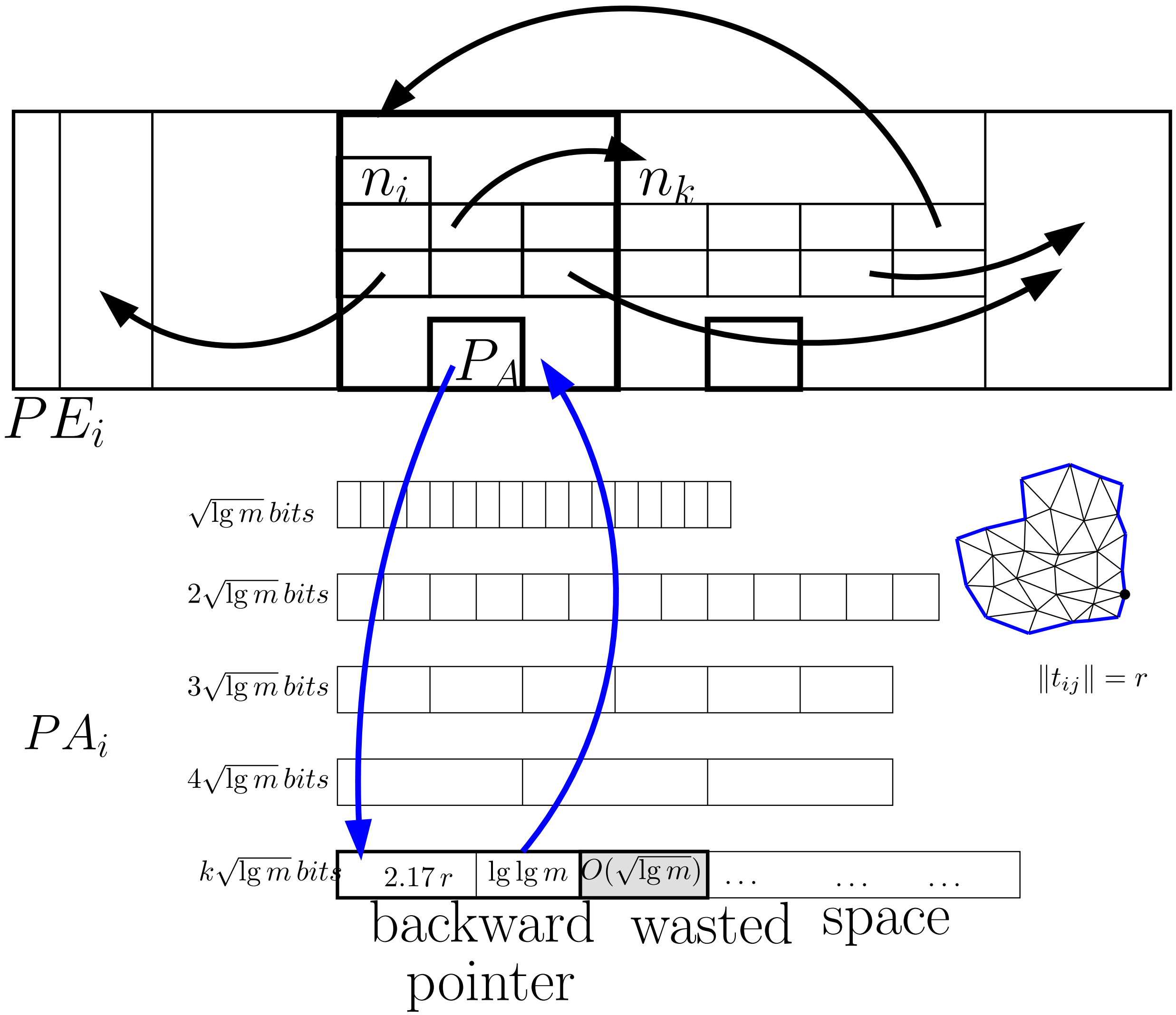
Memory organization

Collection of extendible arrays storing implicitly the boundary colorings: $w_{ij} \lg b_{ij}$ bits pointers



Memory organization

Representation of a node n_{ij} in map G_i



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)

$$\sum_t \|RS(t)\| \leq \sum_t \lg \binom{\lg m}{w(t)} \leq O(m\frac{\lg\lg m}{\lg m})$$

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)

$$O(m^{\frac{1}{4}2.17}\lg m \lg\lg m) = o(m)$$

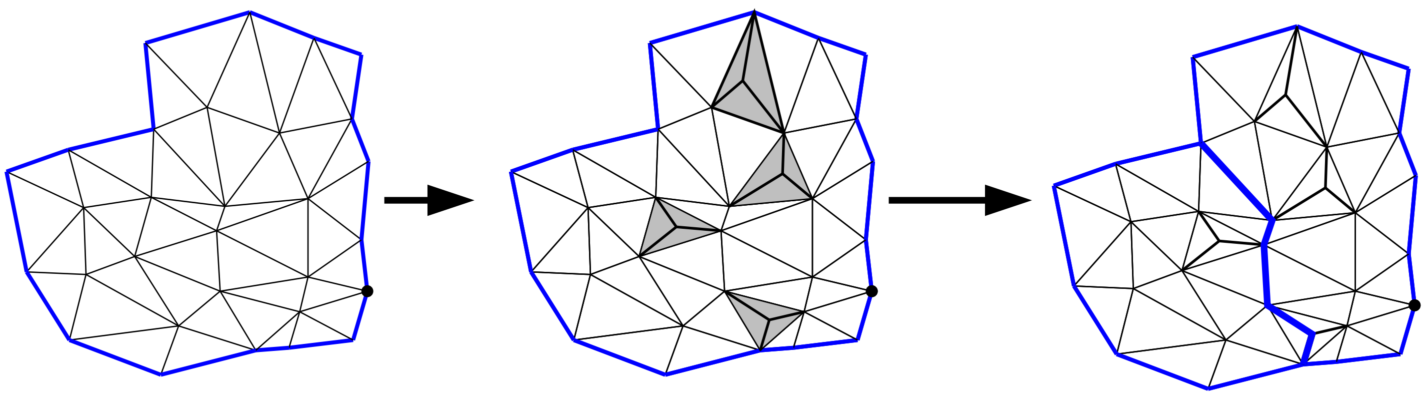
- representation of graph F: $O(\frac{m}{\lg^2 m}\lg m) = o(m)$
- graphs G_i

$$2.17m + O(m\frac{\lg\lg m}{\lg m})$$

Local updates

Problems arising from degree 3 vertex insertion

- increasing of the size of tiny (small) triangulations, after vertex insertion

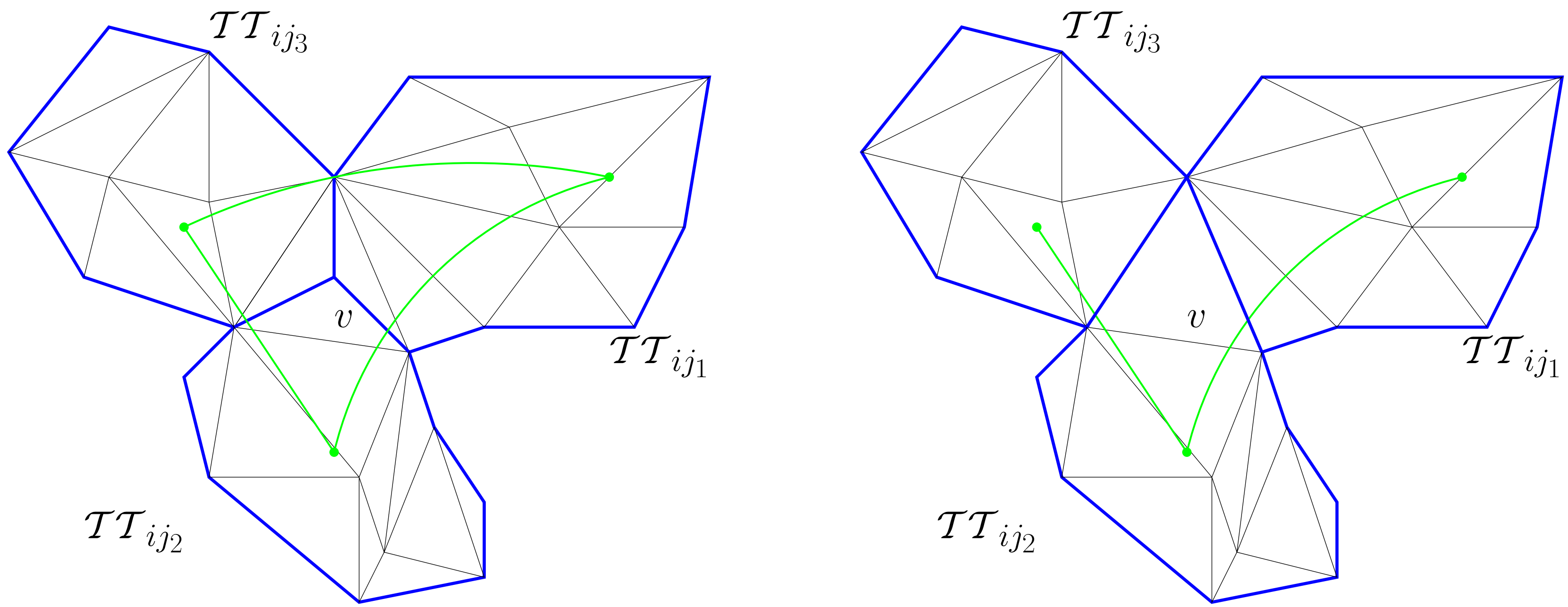


$\|t_r\| \leq \frac{1}{4}\lg m$ $\qquad$ $\|t_r\| > \frac{1}{4}\lg m$ $\qquad$ $\|t_1\| \leq \frac{1}{4}\lg m$, $\|t_2\| \leq \frac{1}{4}\lg m$

Local updates

Problems arising from degree 3 vertex deletion

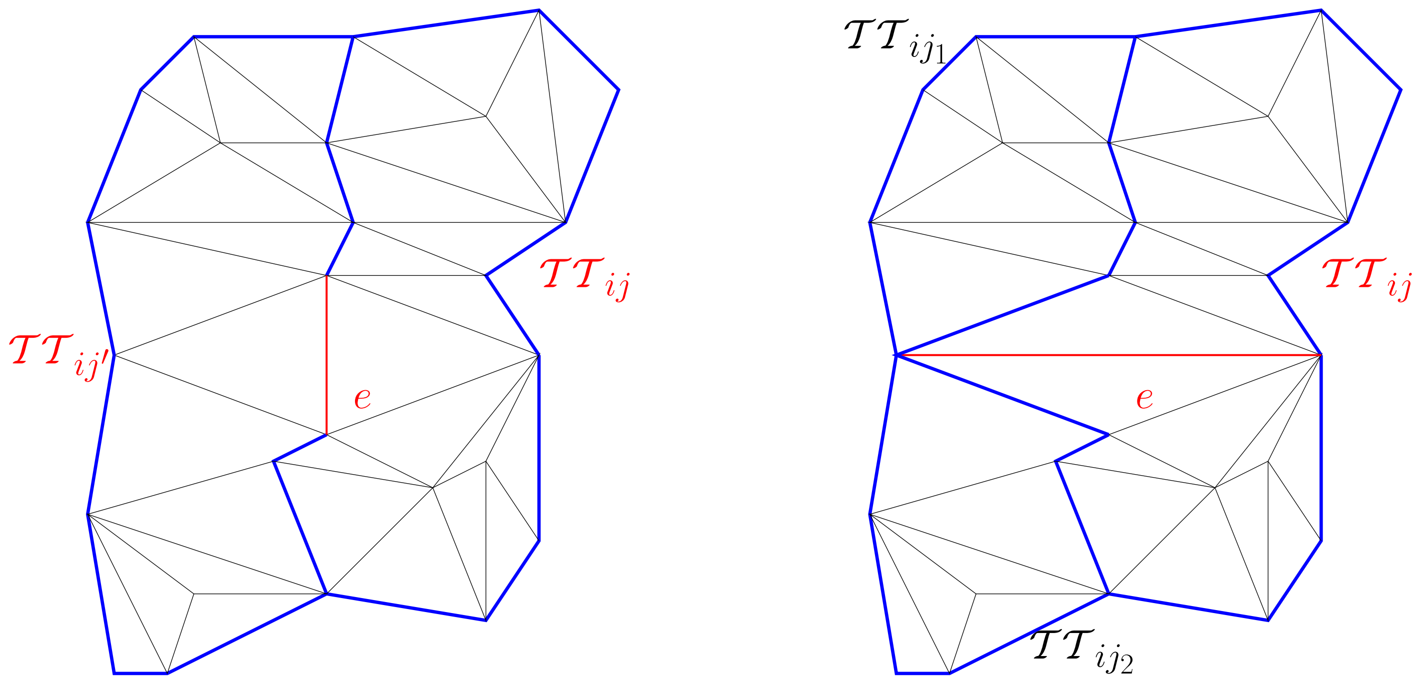
- topology of graphs G_i can change drastically after vertex deletion



Local updates

Problems arising from edge flip

- topology of graphs G_i can change drastically after edge flip



Our contribution

An updatable succinct representation for triangulations

Theorem. *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*

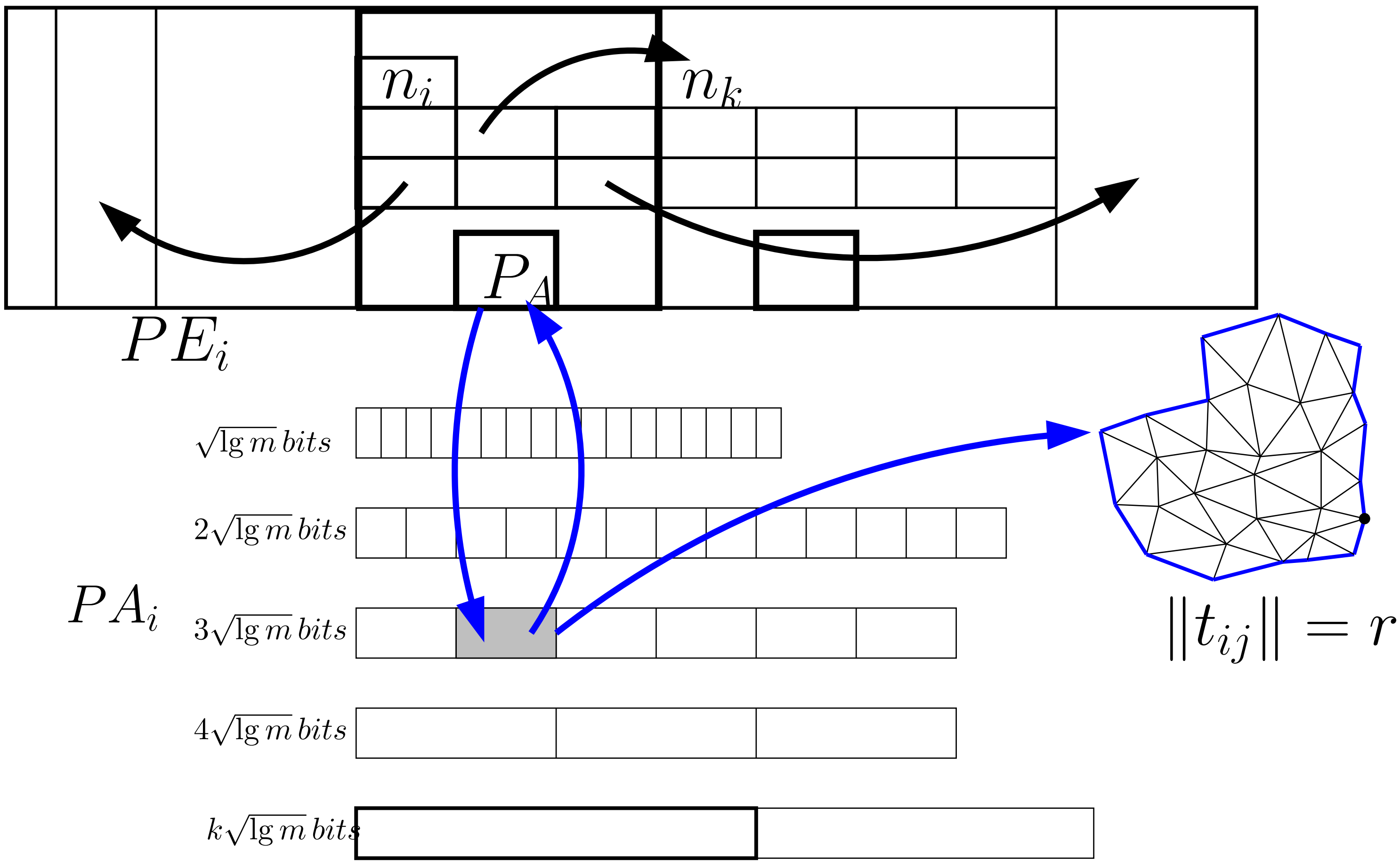
$$2.175m + O\left(m\frac{\lg\lg m}{\lg m}\right) = 2.175m + o(m) \text{ bits}$$

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;*

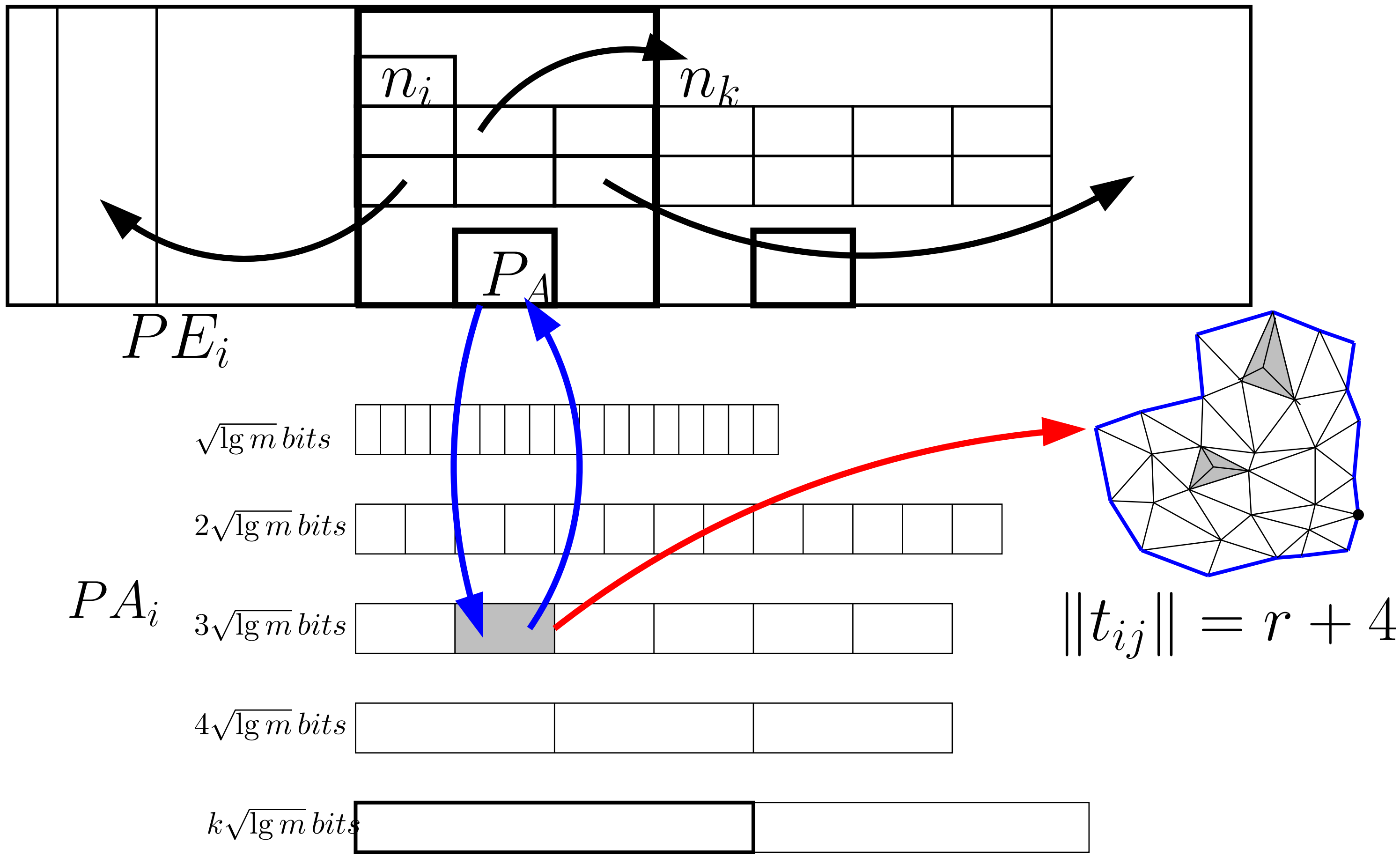
Updating the data structures

Updates of the implicit representation



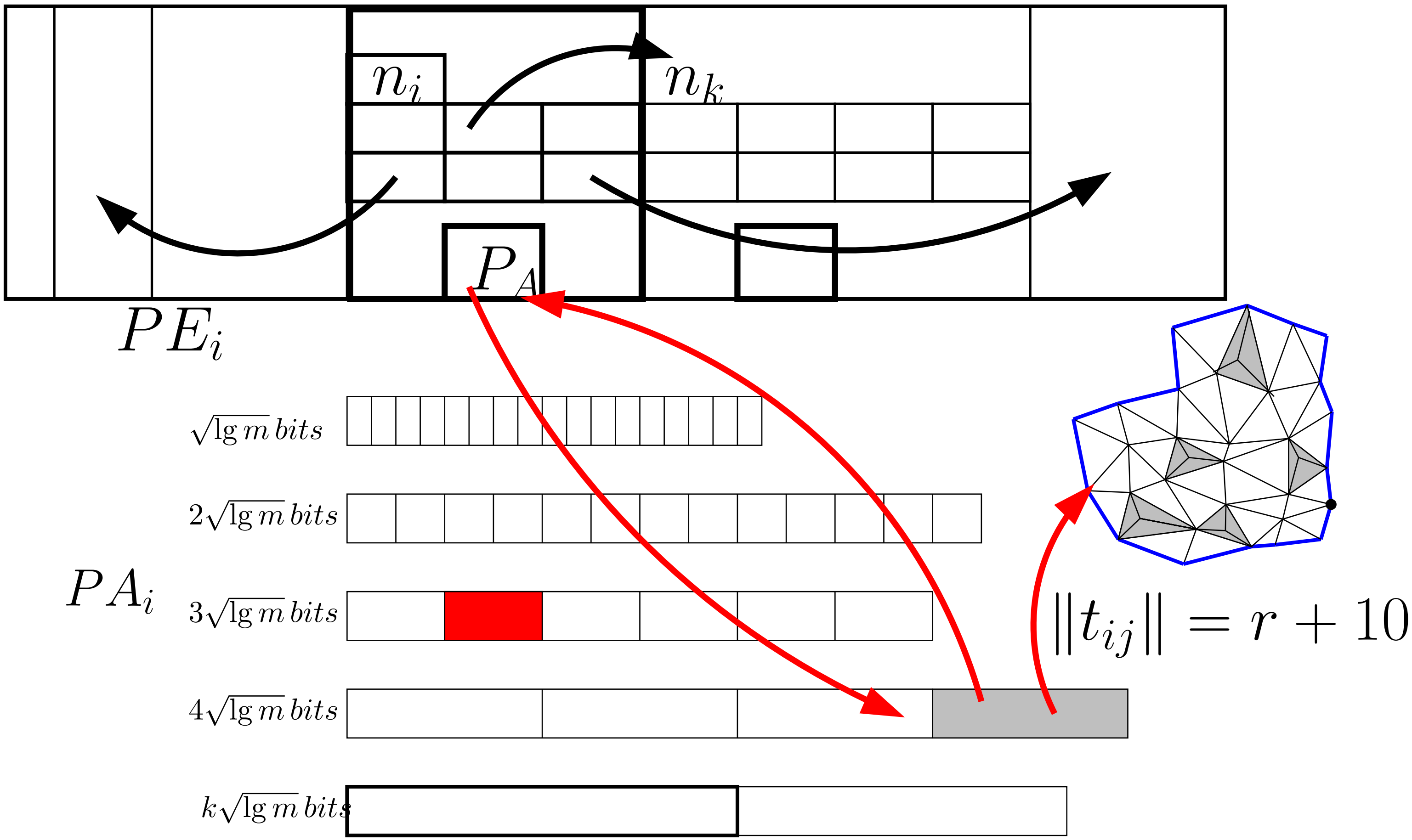
Updating the data structures

Pointers in collection PA_i have to be updated after a vertex insertion



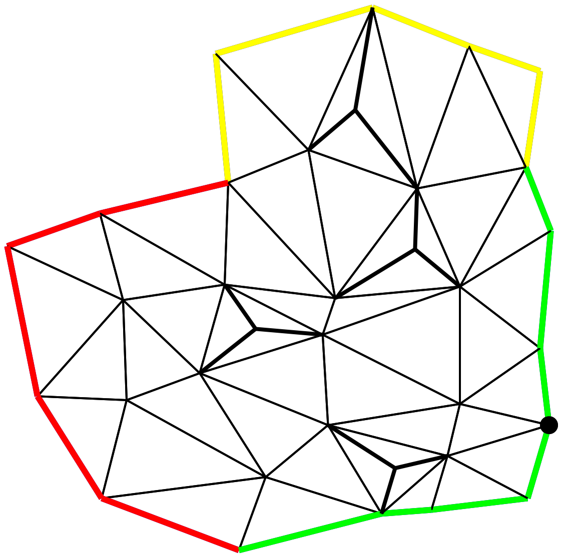
Updating the data structures

The updated tiny triangulation is still valid ($\|t_{ij}\| \leq \frac{1}{4} \lg m$): no decomposition procedure is needed.



Local decomposition

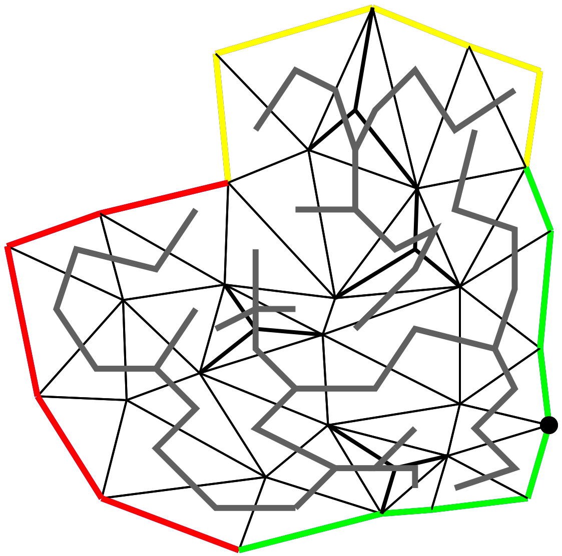
Splitting a tiny triangulation



$$\|t_r\| > \frac{1}{4} \lg m$$

Local decomposition

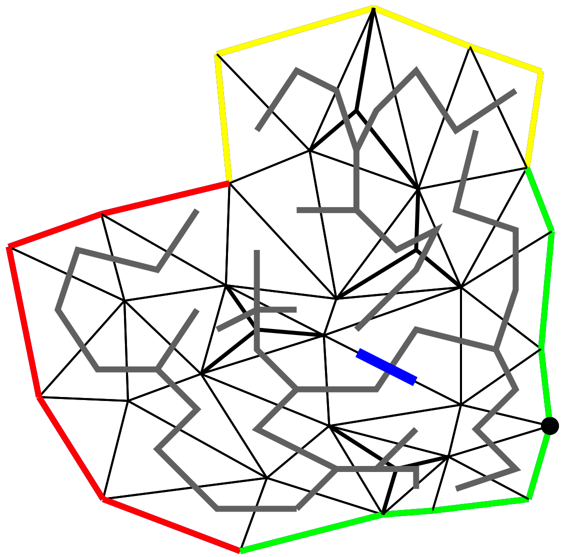
Compute a spanning tree of the dual graph



$$\|t_r\| > \frac{1}{4}\lg m$$

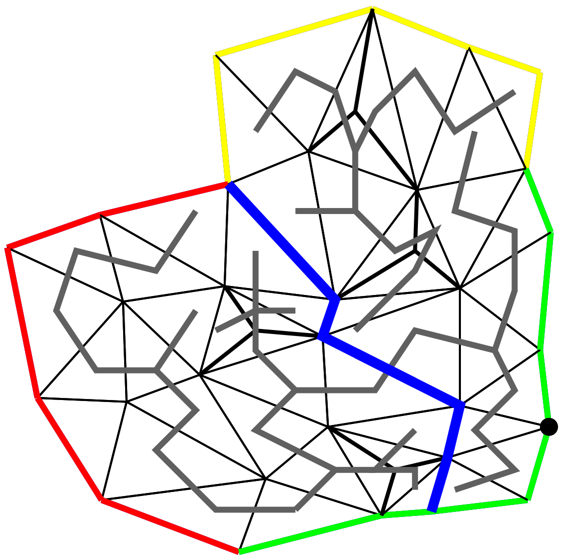
Local decomposition

Apply a decomposition procedure for binary trees (Munro, Raman and Storm SODA 2001)



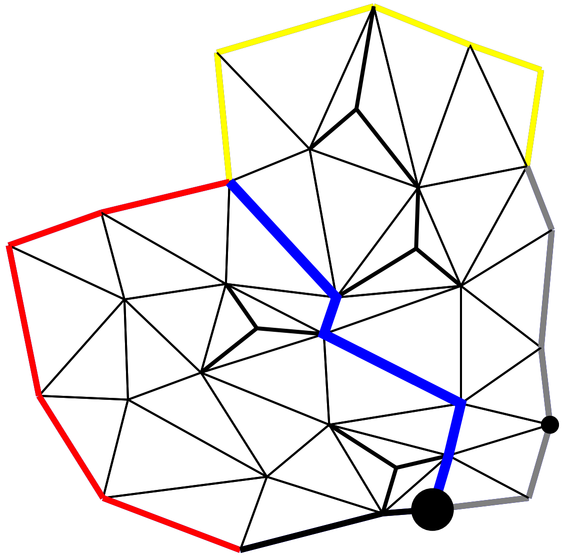
Local decomposition

Obtain two valid tiny triangulations



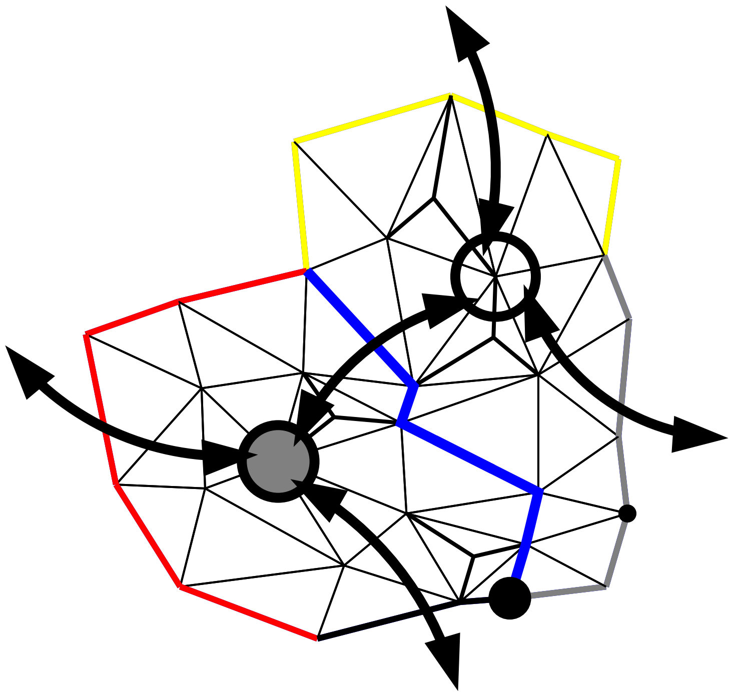
Local decomposition

Update sides and boundary coloring



Local decomposition

Update locally map G_i

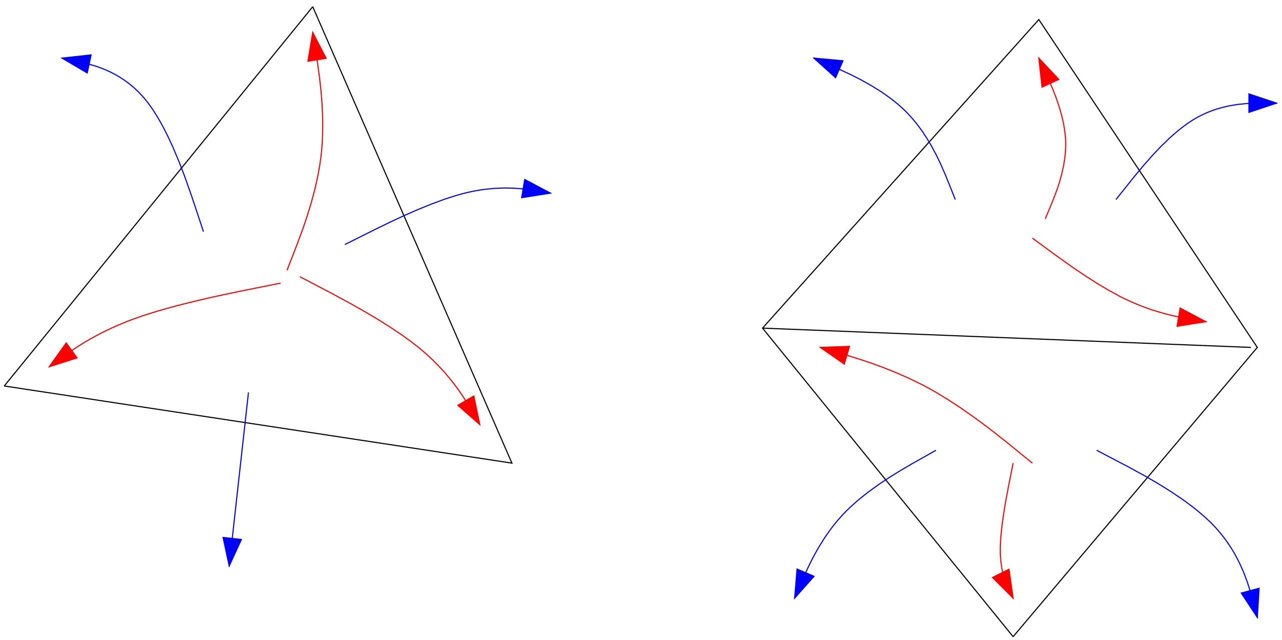


A practical solution

(Abdelkrim Mebarki and Olivier Devillers)

C++ implementation based on CGAL library

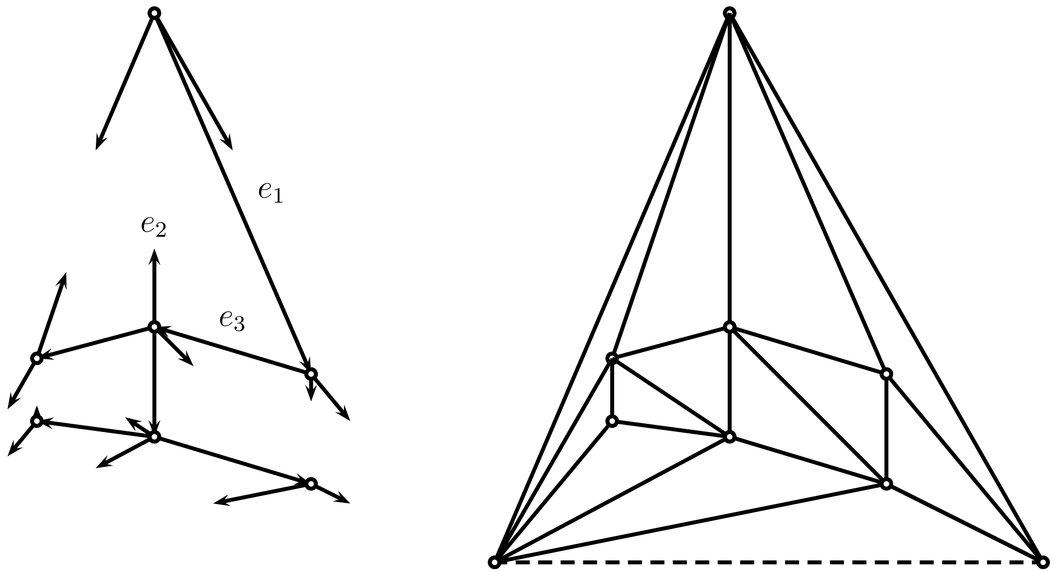
triangle+quad based representation for triangle meshes



Open problem

Optimal succinct encoding for planar triangulations, achieving Tutte's entropy: 1.62 bits/face.

Idea: canonical decomposition strategy based on optimal encoding (Poulalhon et Schaeffer ICALP03)



Future work

Triangulations 3D

Any idea?