Succinct representation of triangulations with a boundary

WADS 2005 - Waterloo

Luca Castelli Aleardi

(joint work with Olivier Devillers and 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

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

- or *compact*: we content of a cost

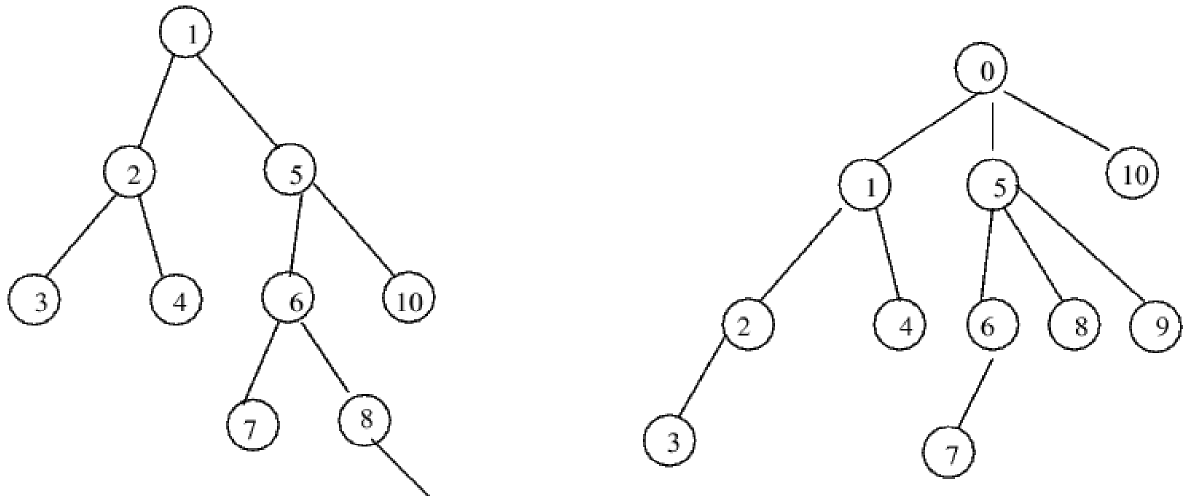
$$size(R) = O(\|C_m\|)$$

- 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:

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

Compact representations: an example

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

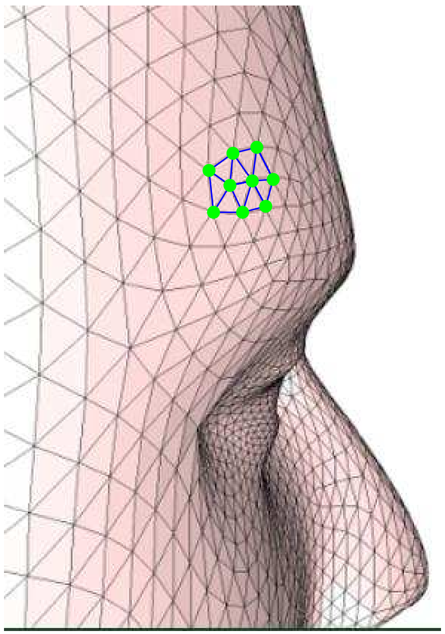
succinct representation (Jacobson 89, Munro et Raman 97)

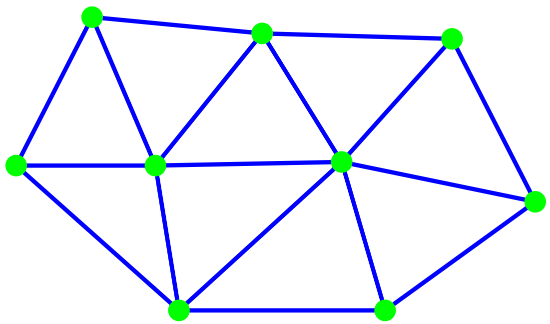
- size: $2n + o(n)$ bits
- adjacency queries in constant time

Motivation

Combinatorial information describing incidence relations

Which information?



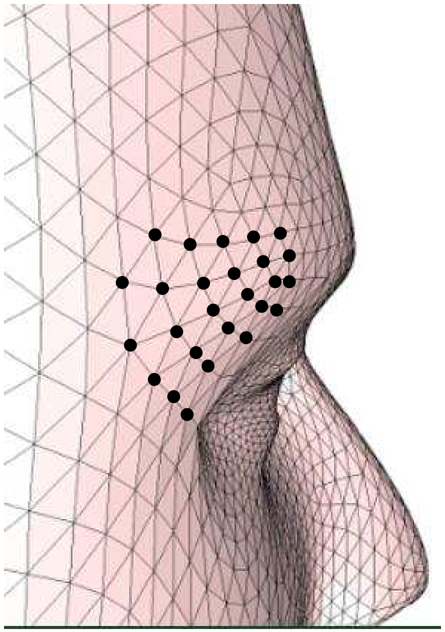


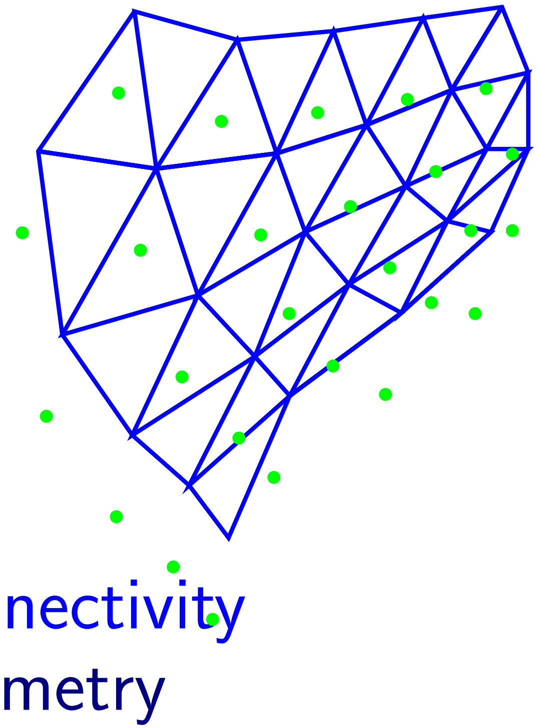
Connectivity

Motivation

Geometry information (vertex coordinates)

Which information?





Motivation

Usual mesh representation

VRML format file

v_1: 0.5389 0.7634 1.3456

v_2: x_2 y_2 z_2

v_3: x_3 y_3 z_3

v_4: x_4 y_4 z_4

. . .   3×32 bits / vertex

. . .

t_1: 1-2-3

t_2: 1-3-4

t_3: *-*-*

t_4: *-*-*

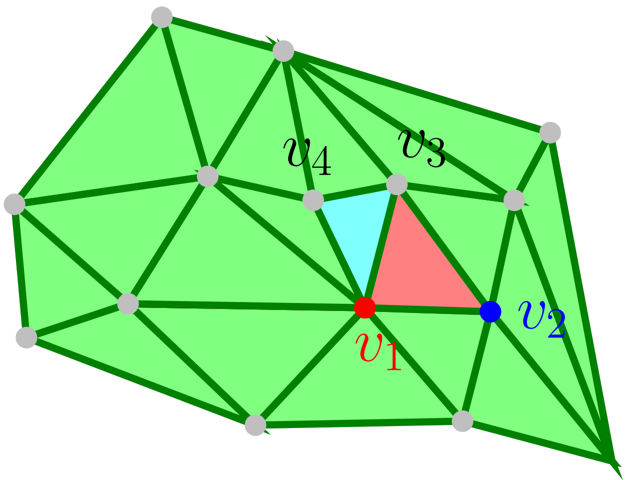
t_5: *-*-*

t_6: *-*-*

. . .

6×32 bits /vertex

288 bits /vertex



Motivation

Mesh compression algorithms

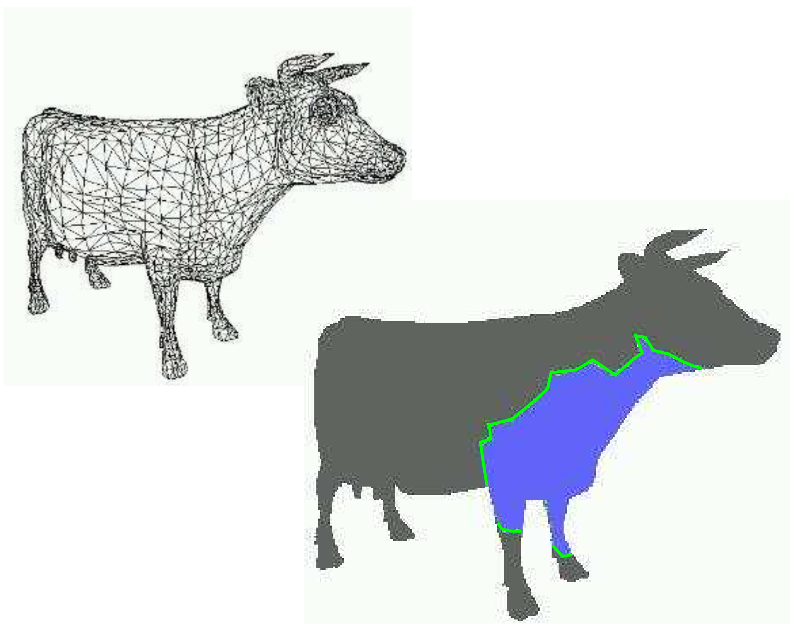
Edgebreaker [Rossignac]

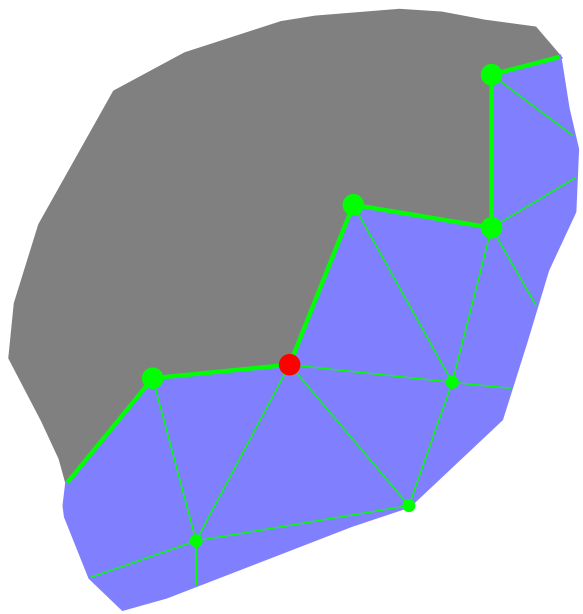
Touma Gotsman

Poulhalon Schaeffer

General underlying idea

Encoding strategies based on a local (global) conquest





Motivation

Mesh compression algorithms

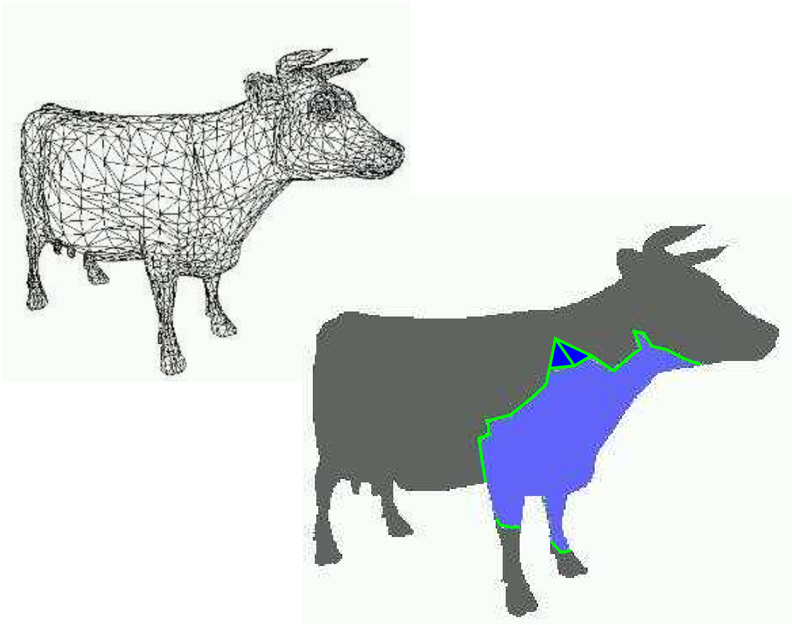
Edgebreaker [Rossignac]

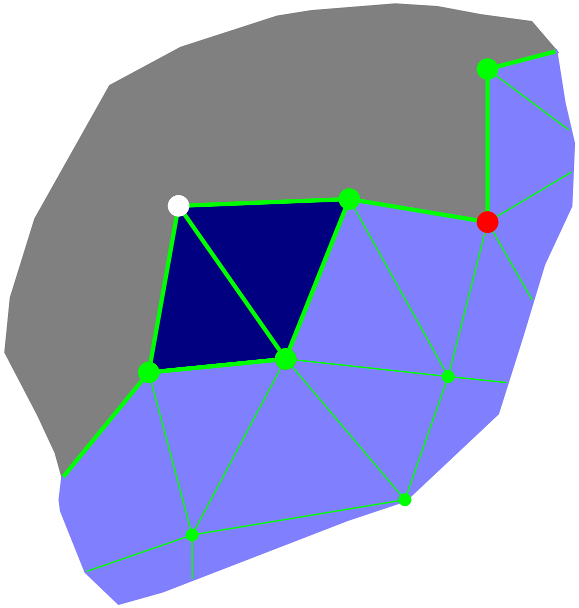
Touma Gotsman

Poulhalon Schaeffer

General underlying idea

Encoding strategies based on a local (global) conquest





Motivation

Mesh compression algorithms

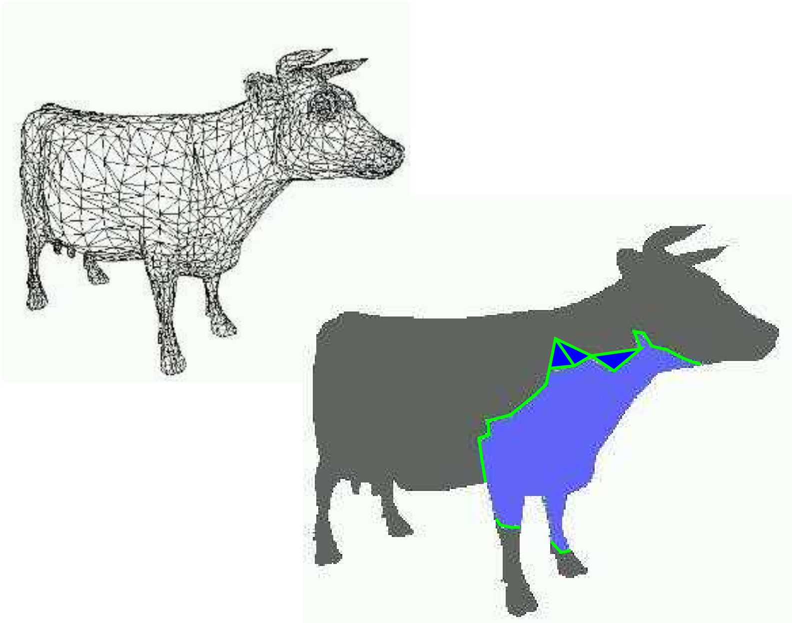
Edgebreaker [Rossignac]

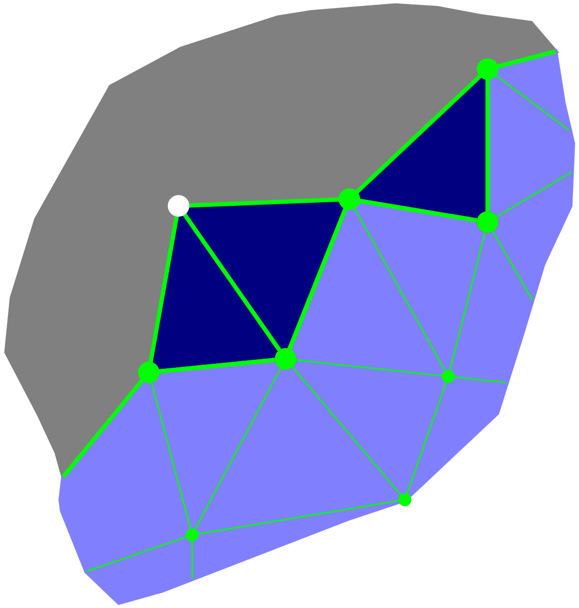
Touma Gotsman

Poulhalon Schaeffer

General underlying idea

Encoding strategies based on a local (global) conquest





Motivation

Mesh compression algorithms

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

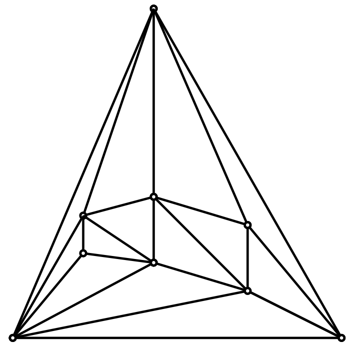
2.175 bits/triangle

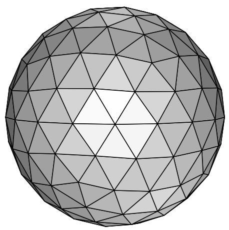
Previous and related works

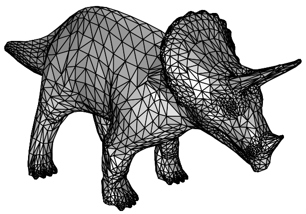
- static trees on n nodes (Jacobson FOCS89): space $2n + o(n)$, navigation in $O(\lg n)$ time;
- planar graphs on n vertices and e edges (Munro Raman FOCS97): space $8n + 2e$, $O(1)$ time navigation;
- 3-connected planar graphs on n vertices(Chuang et al. ICALP98): space $2e + n$, $O(1)$ time navigation;
- 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)







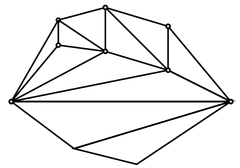
enumeration of rooted planar triangulations on n vertices:

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

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

Our contribution

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

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

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

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

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
our encoding	$O(1)$	$2.175m$	$2.175m$

Basic ideas

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

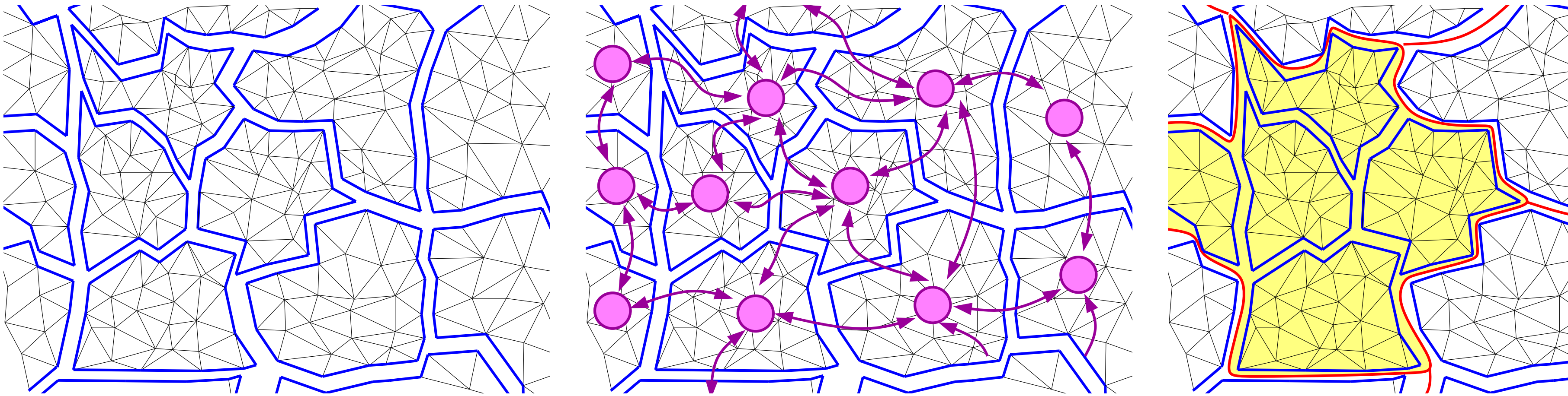
Literary digression

"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.

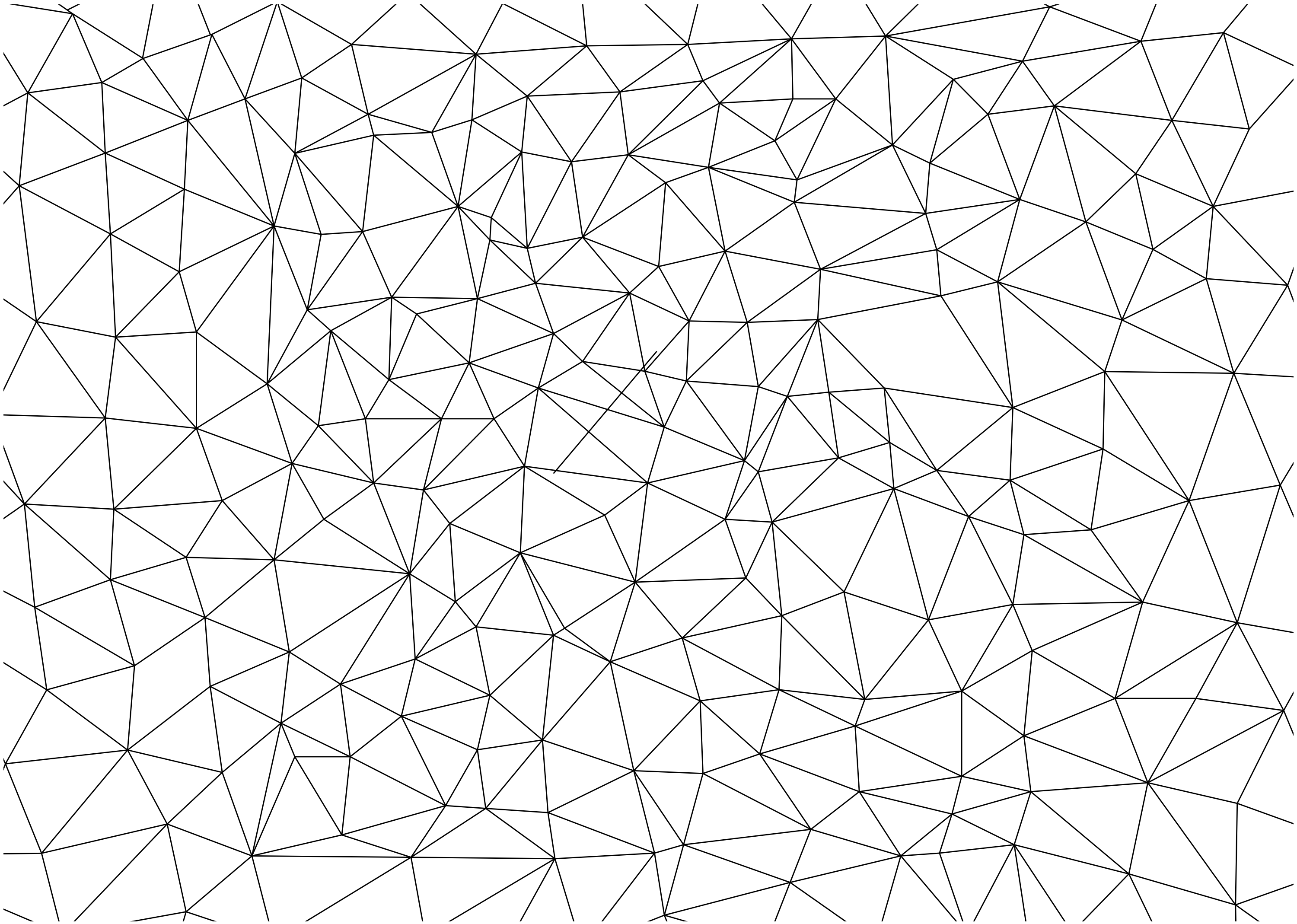
Decomposing $\mathcal{T}$ into sub-triangulation

- 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.



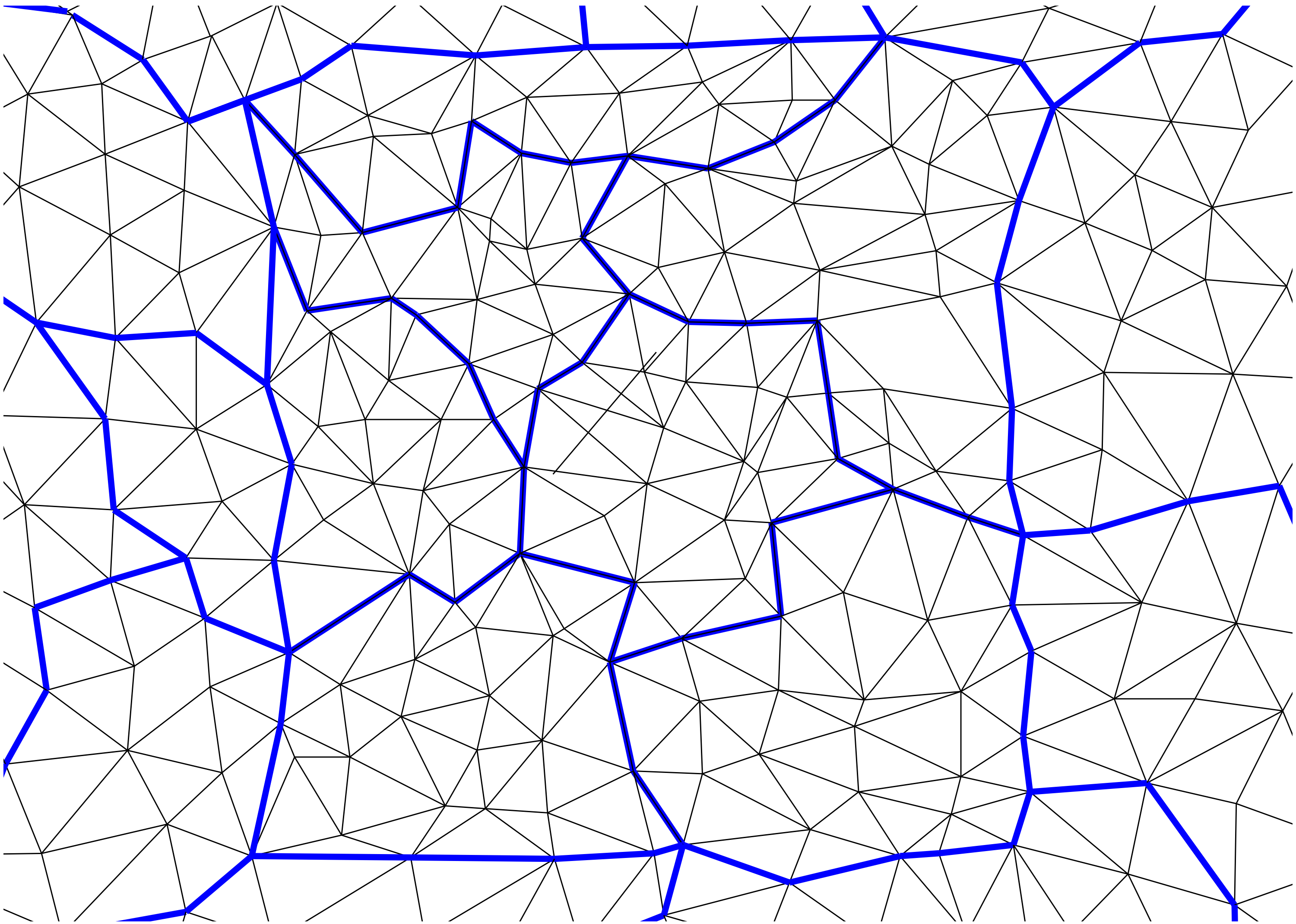
Decomposition phase

We start with a triangulation having m triangles



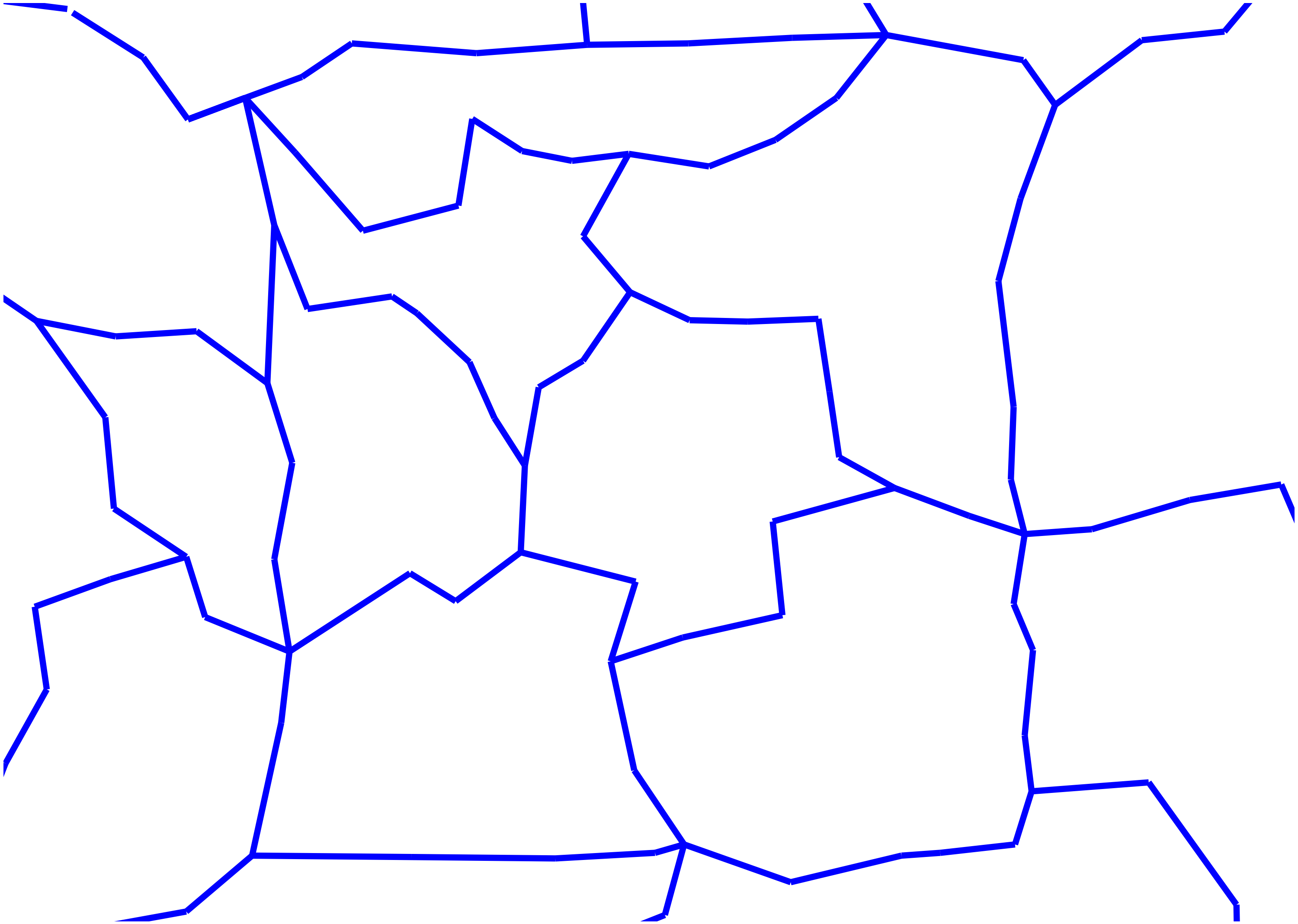
Decomposition phase

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



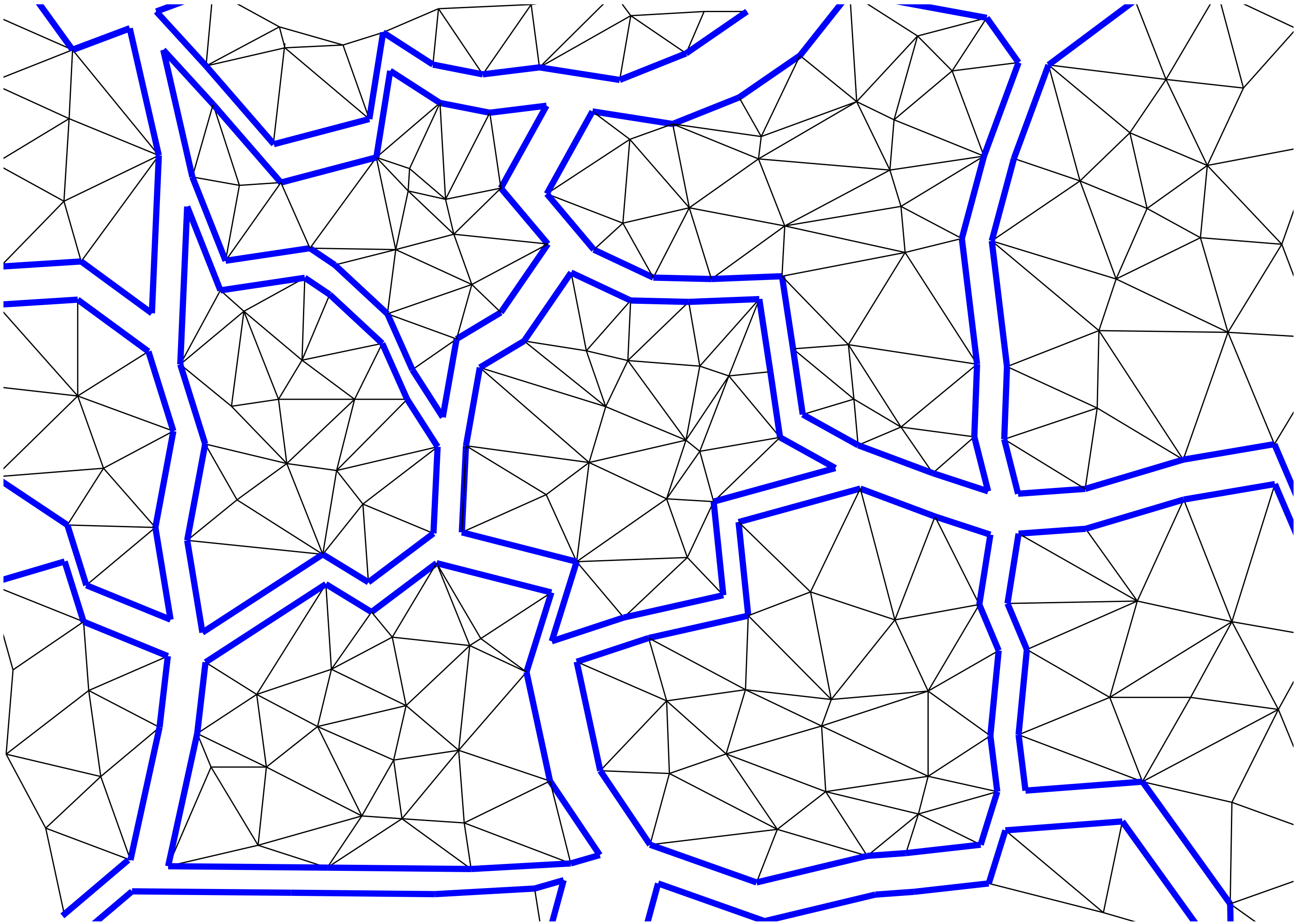
Decomposition phase

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



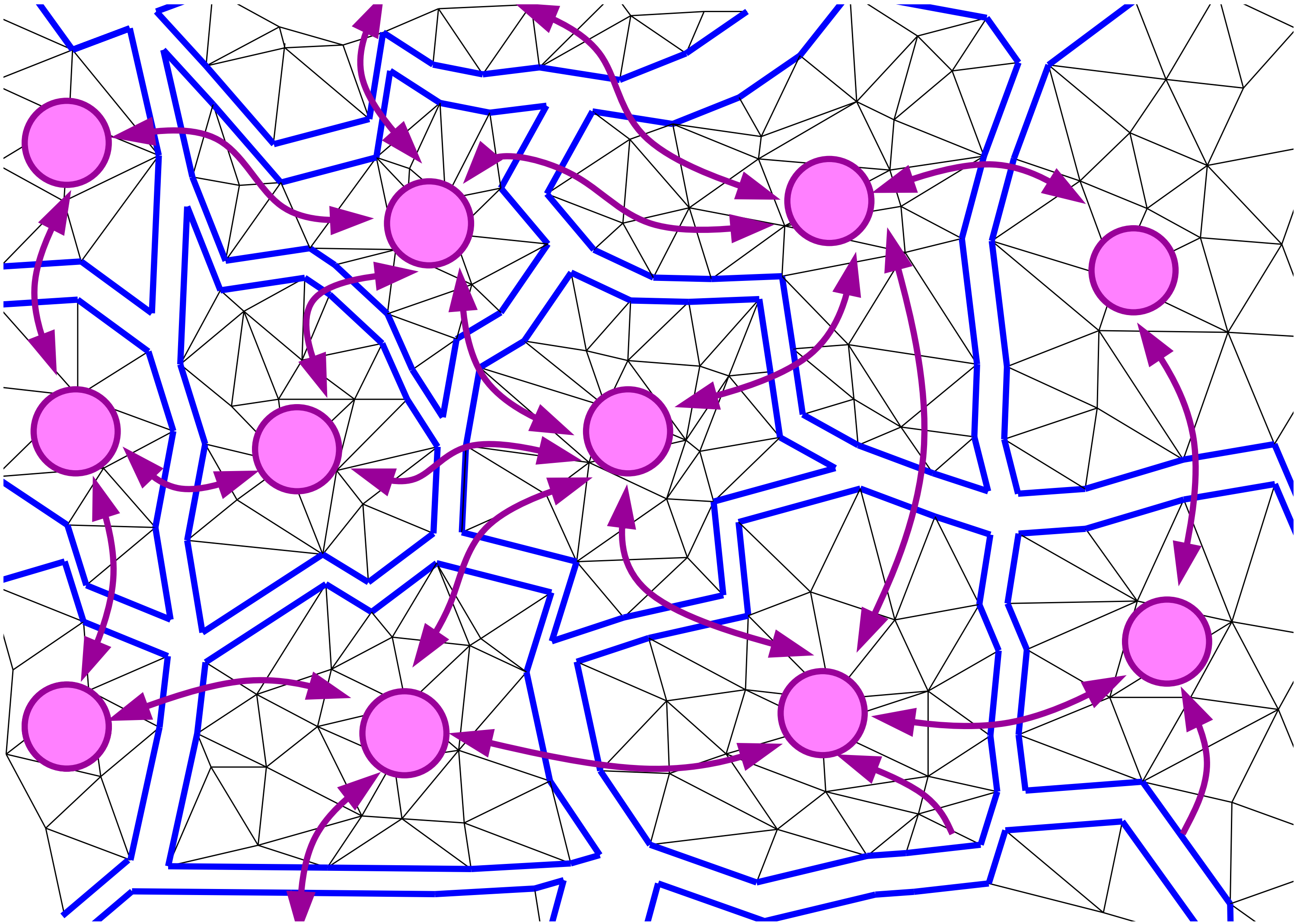
Decomposition phase

Only boundary edges are shared by tiny triangulations



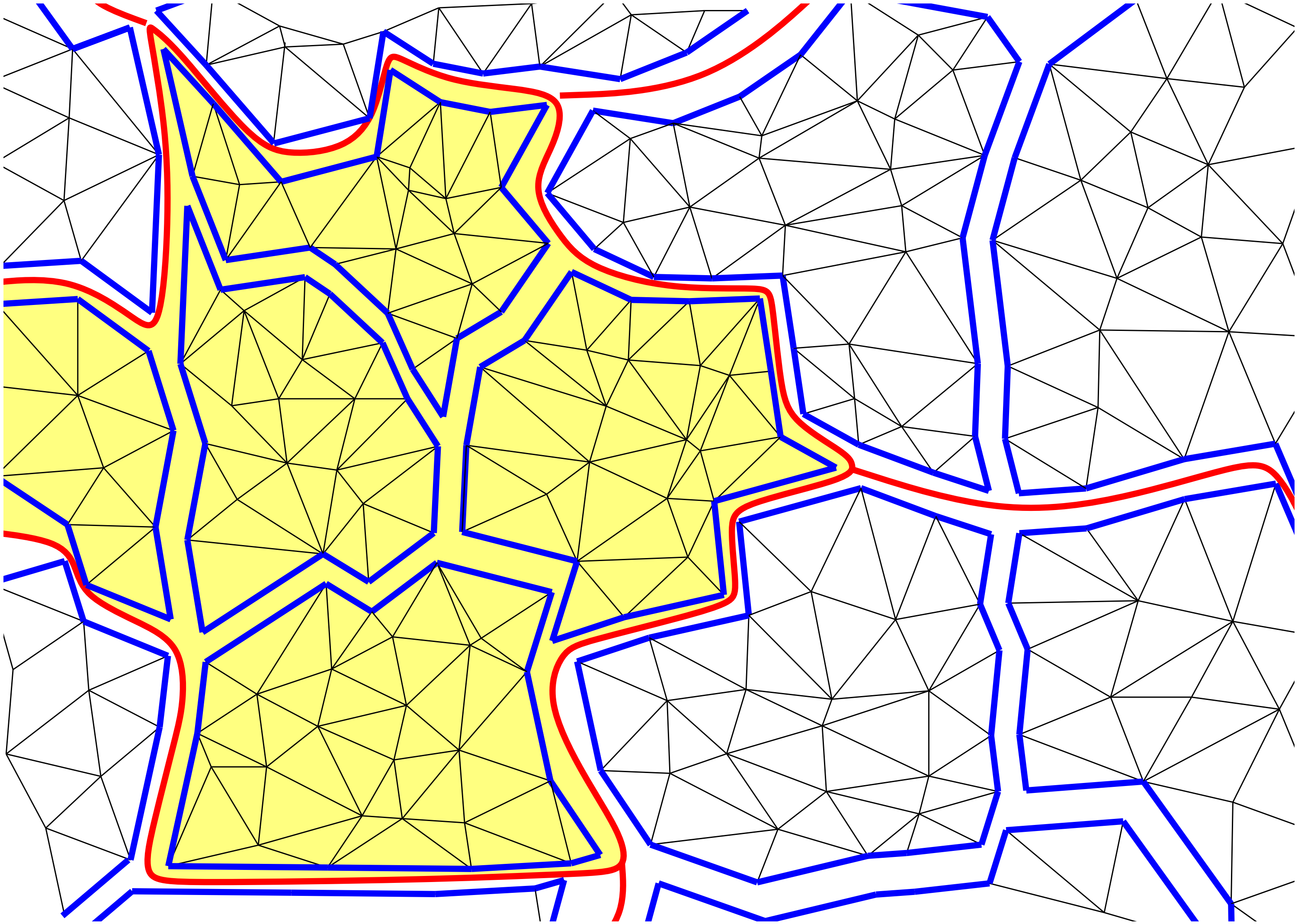
Decomposition phase

Graph G linking adjacent tiny triangulations



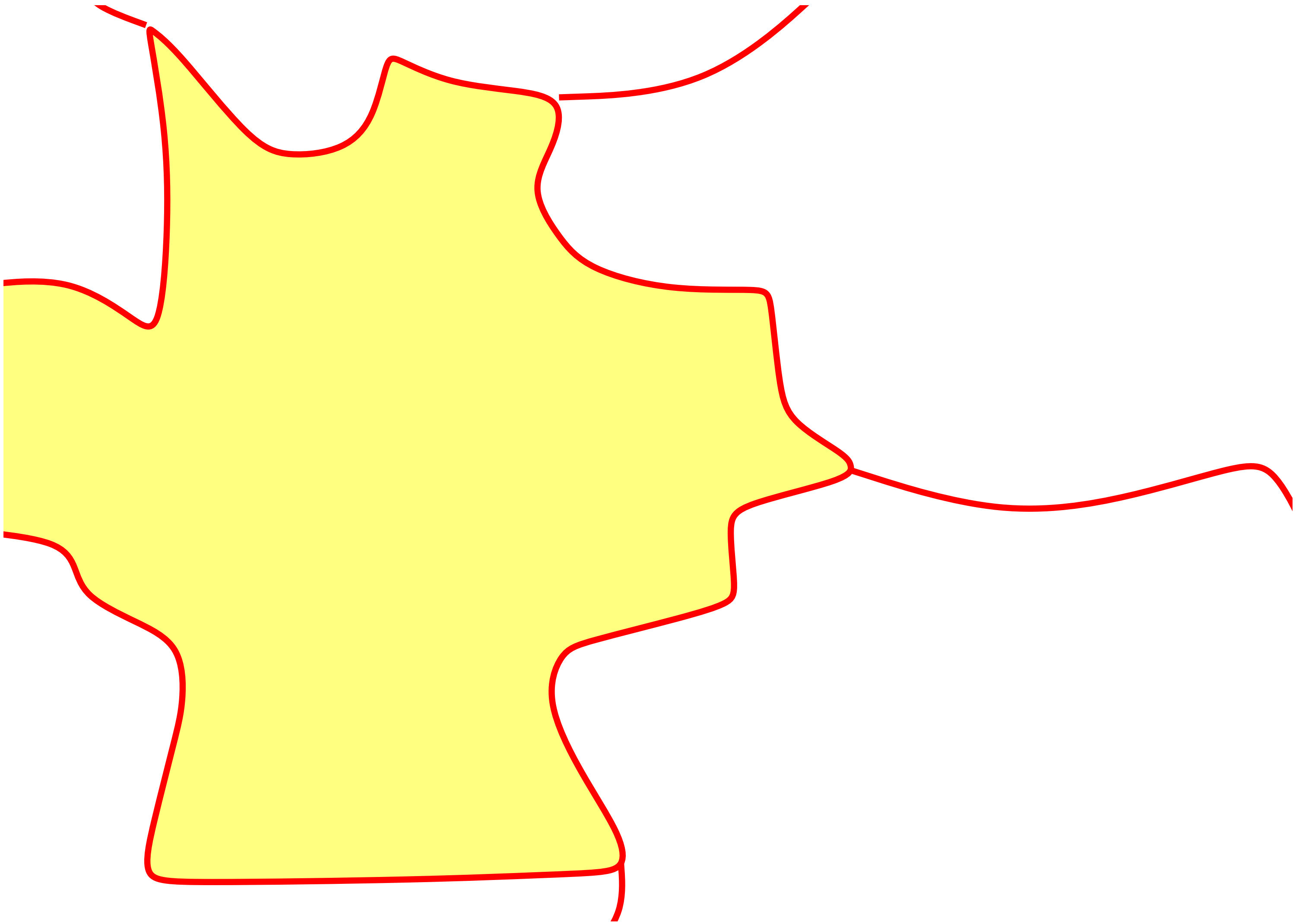
Decomposition phase

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



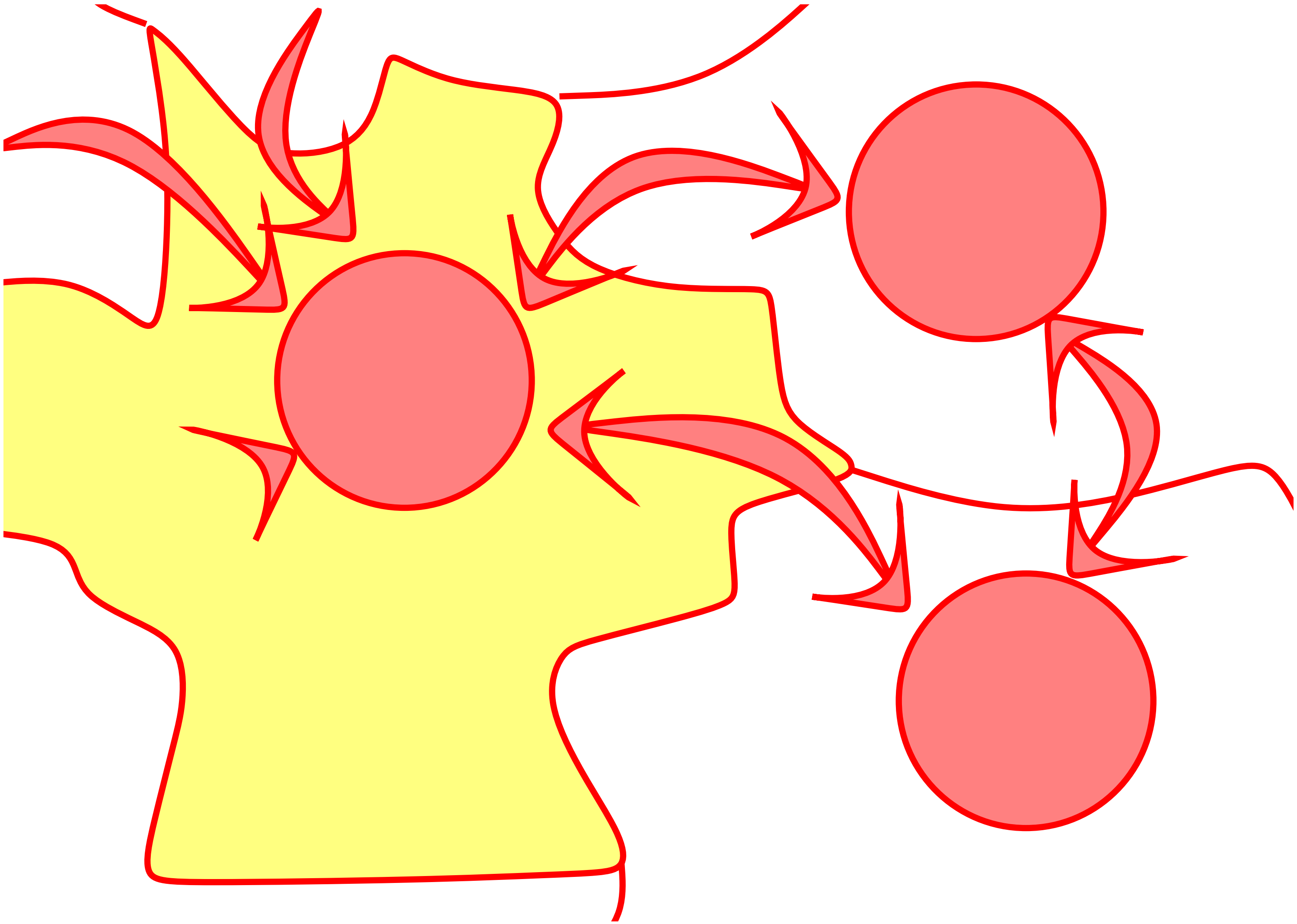
Decomposition phase

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



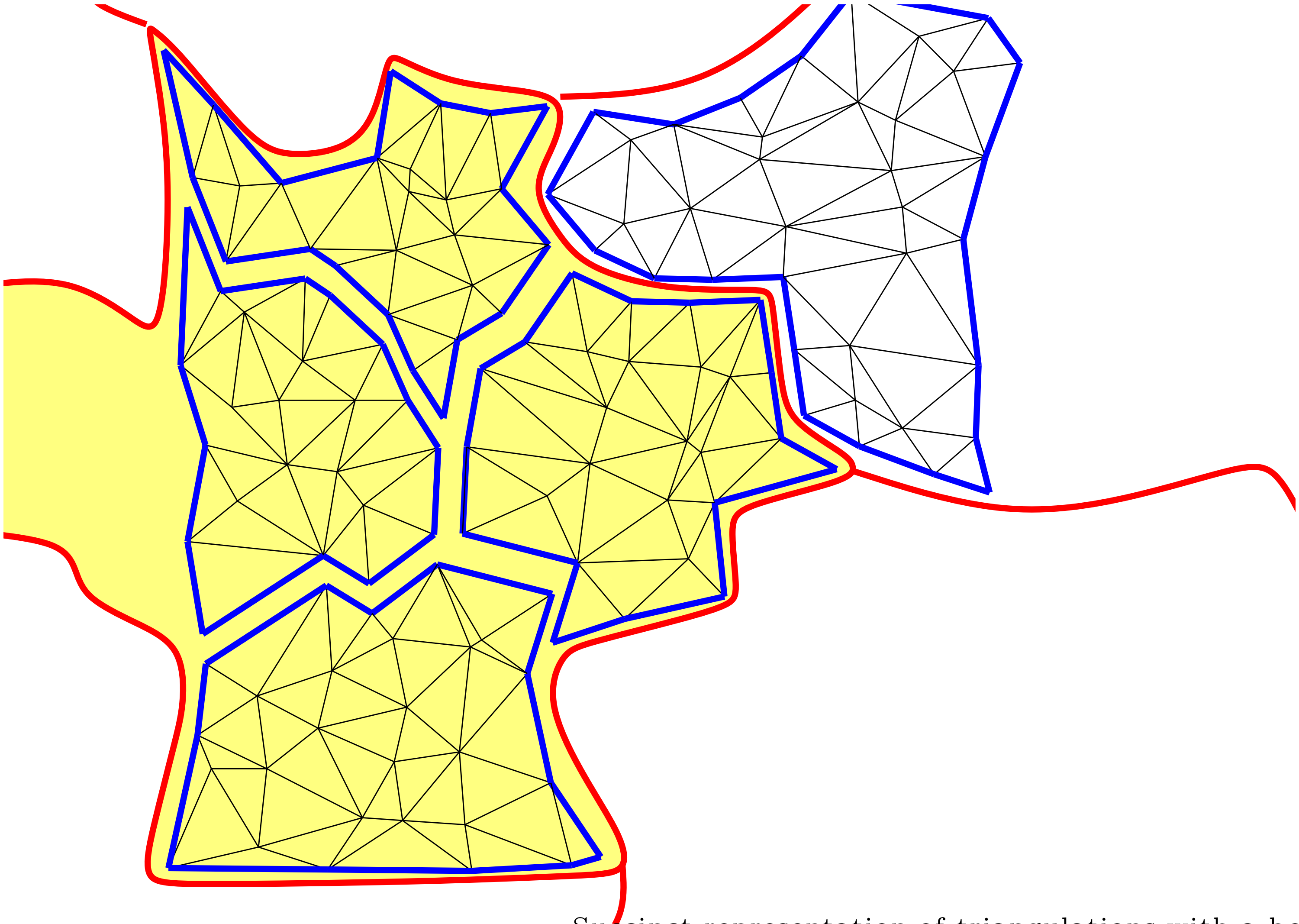
Decomposition phase

Graph F linking adjacent small triangulations



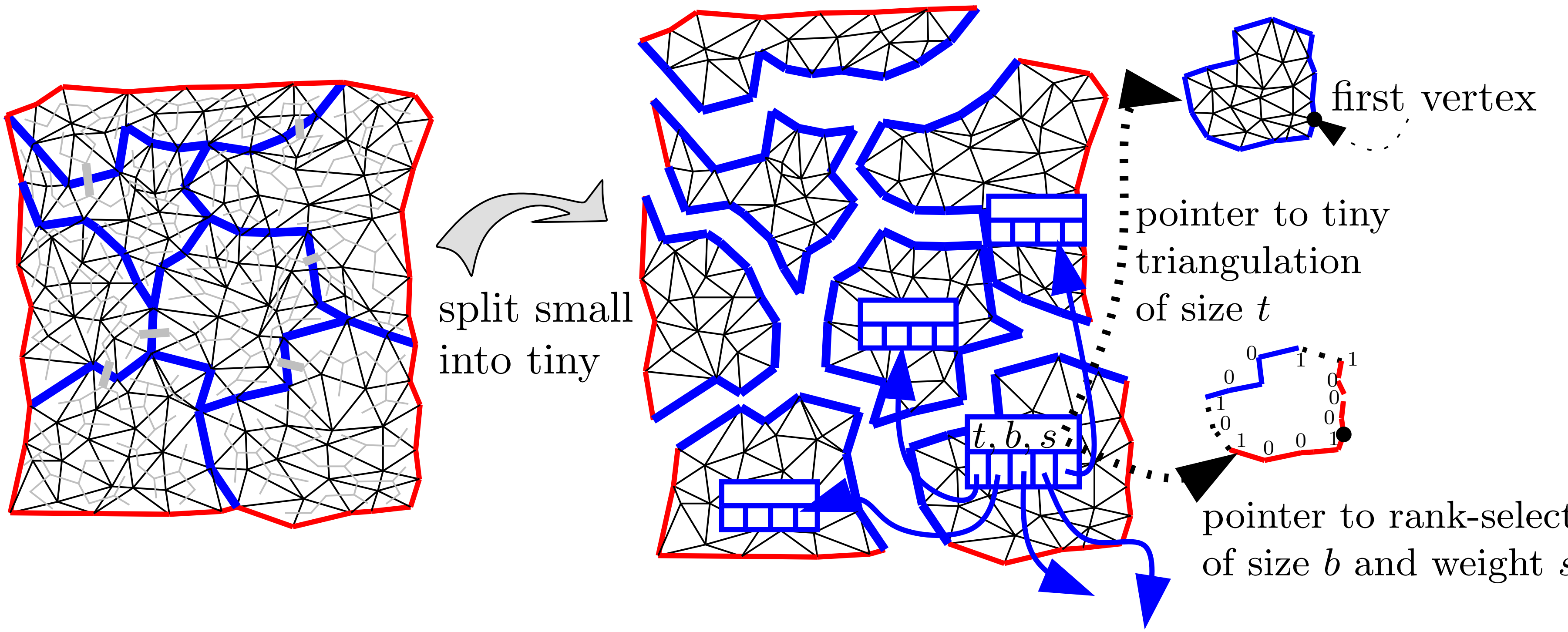
Decomposition phase

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



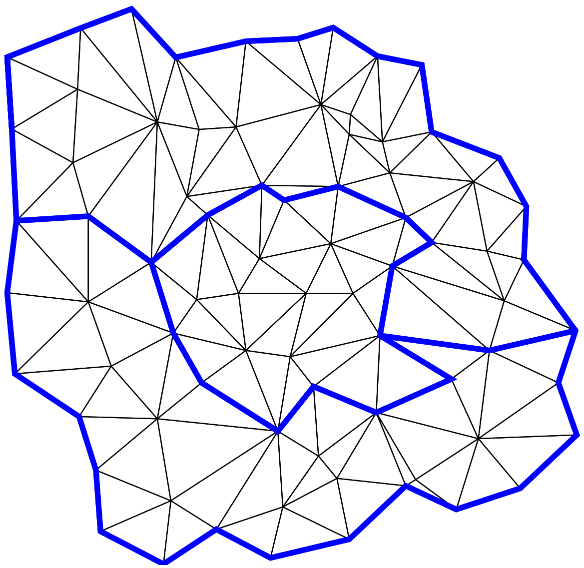
rview: representation of a small triangu

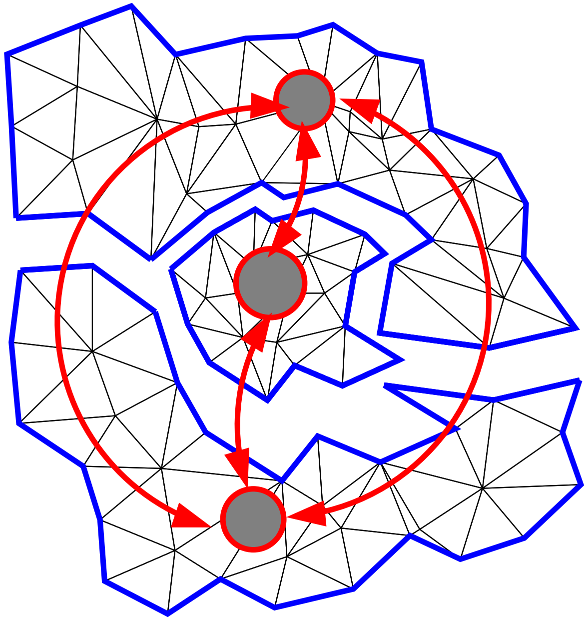
- adjacency relations are described by map G_i;
- internal connectivity is implicitly represented (variable size pointers)
- boundary neighboring relations are represented by boundary coloring (variable length bit-vector)



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;

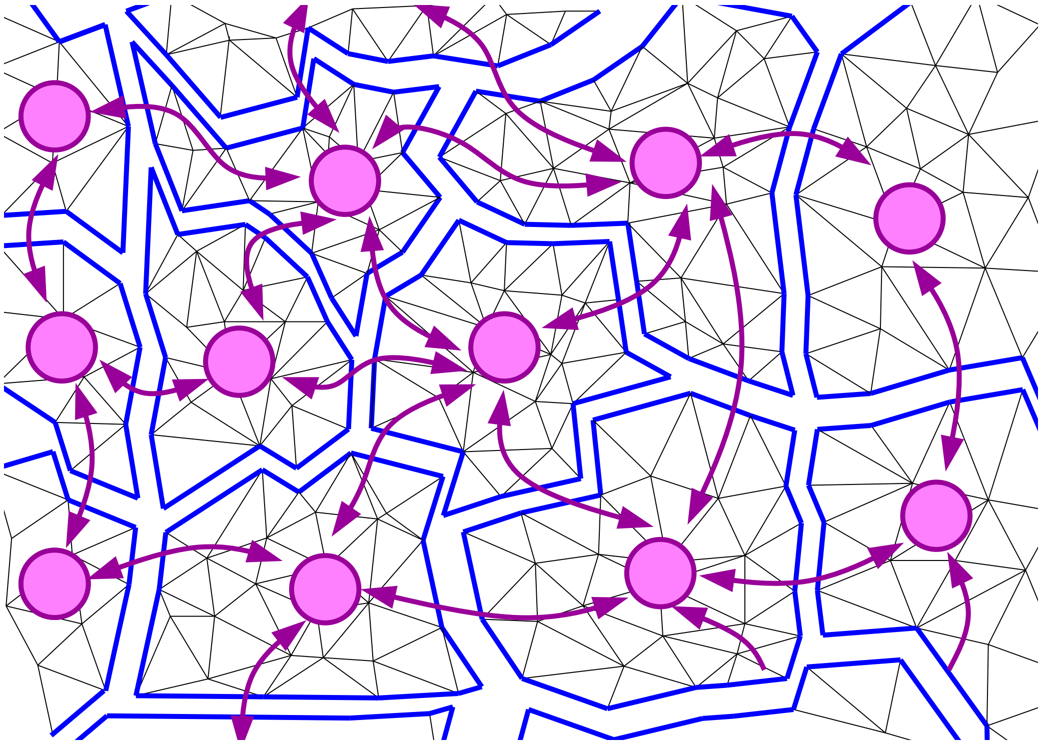




ljacency relations between tiny triangu

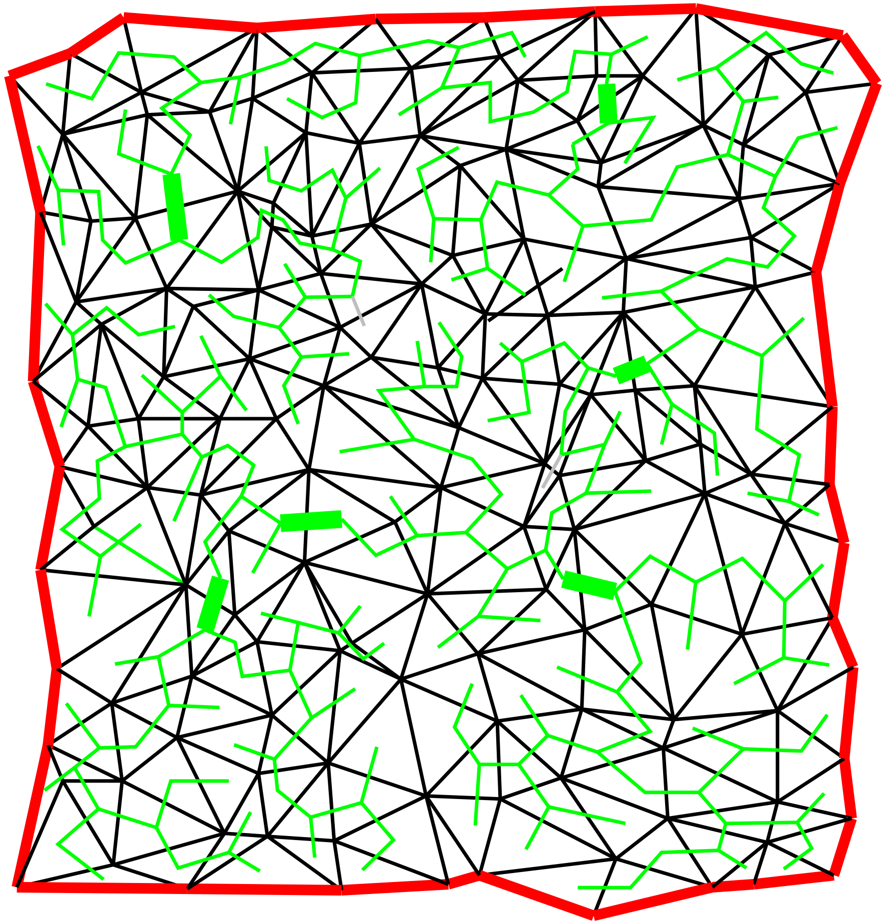
- Because of Euler's formula, the overall number of arcs in maps G_i is:

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



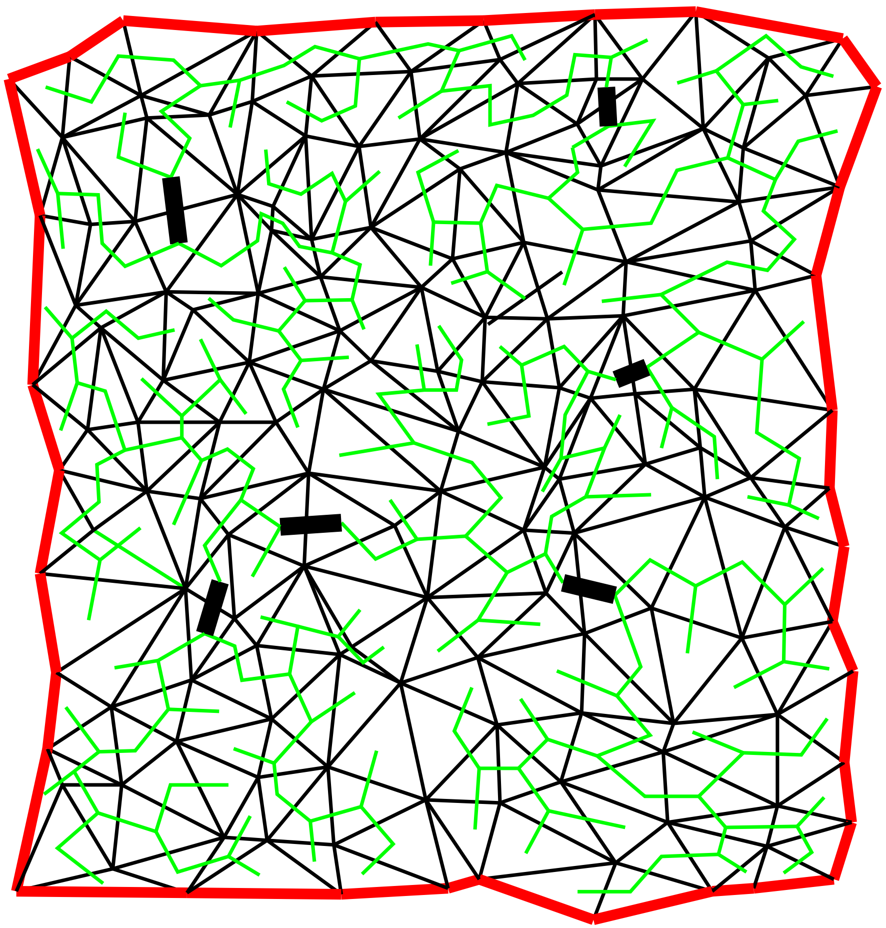
Decomposition

Initial small triangulation with a dual spanning tree



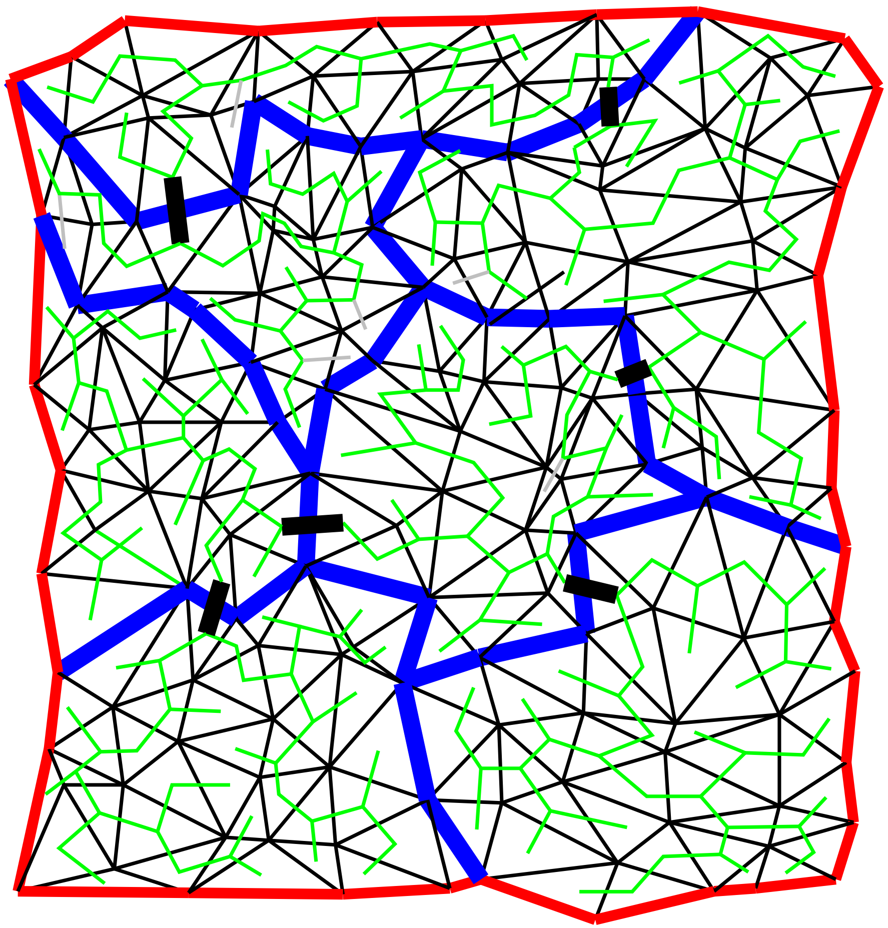
Decomposition

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



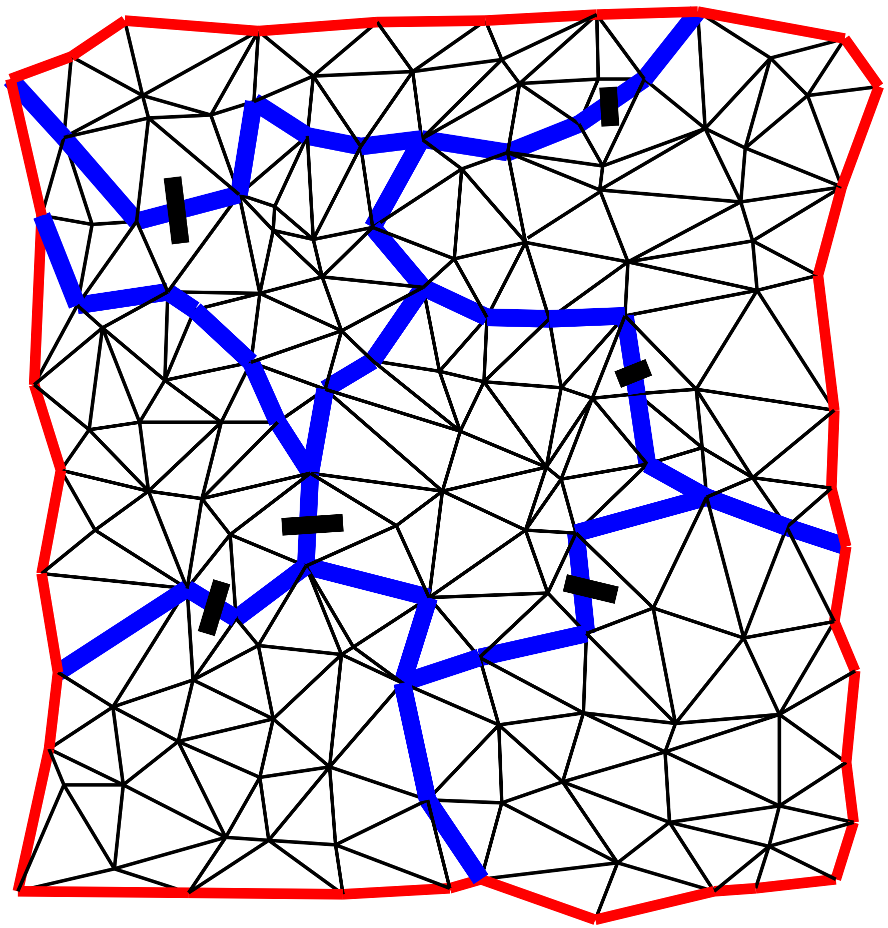
Decomposition

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



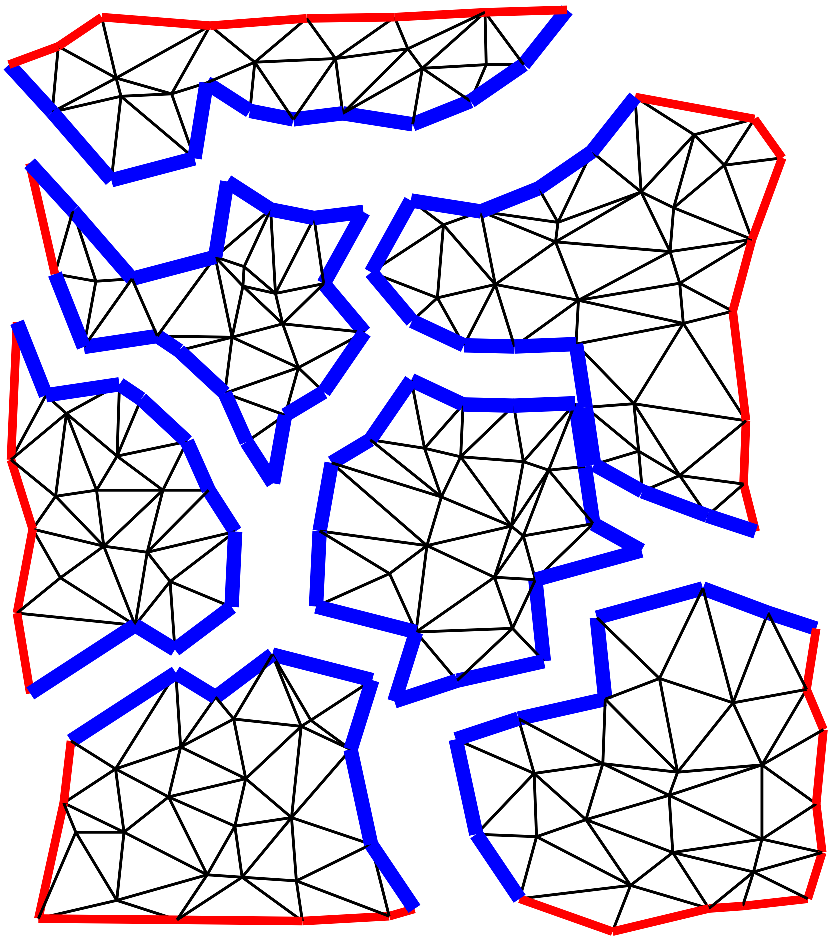
Decomposition

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

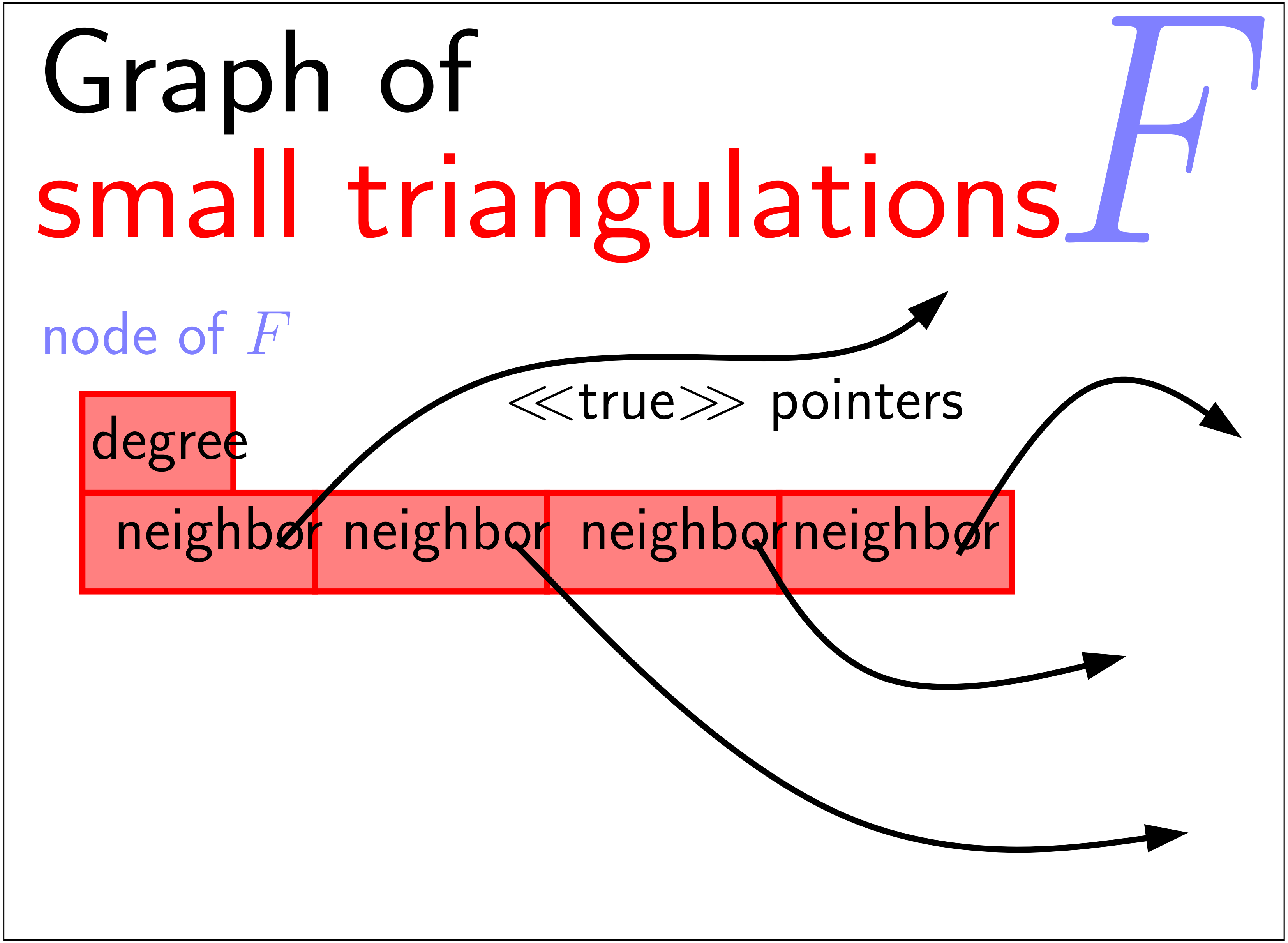


Decomposition

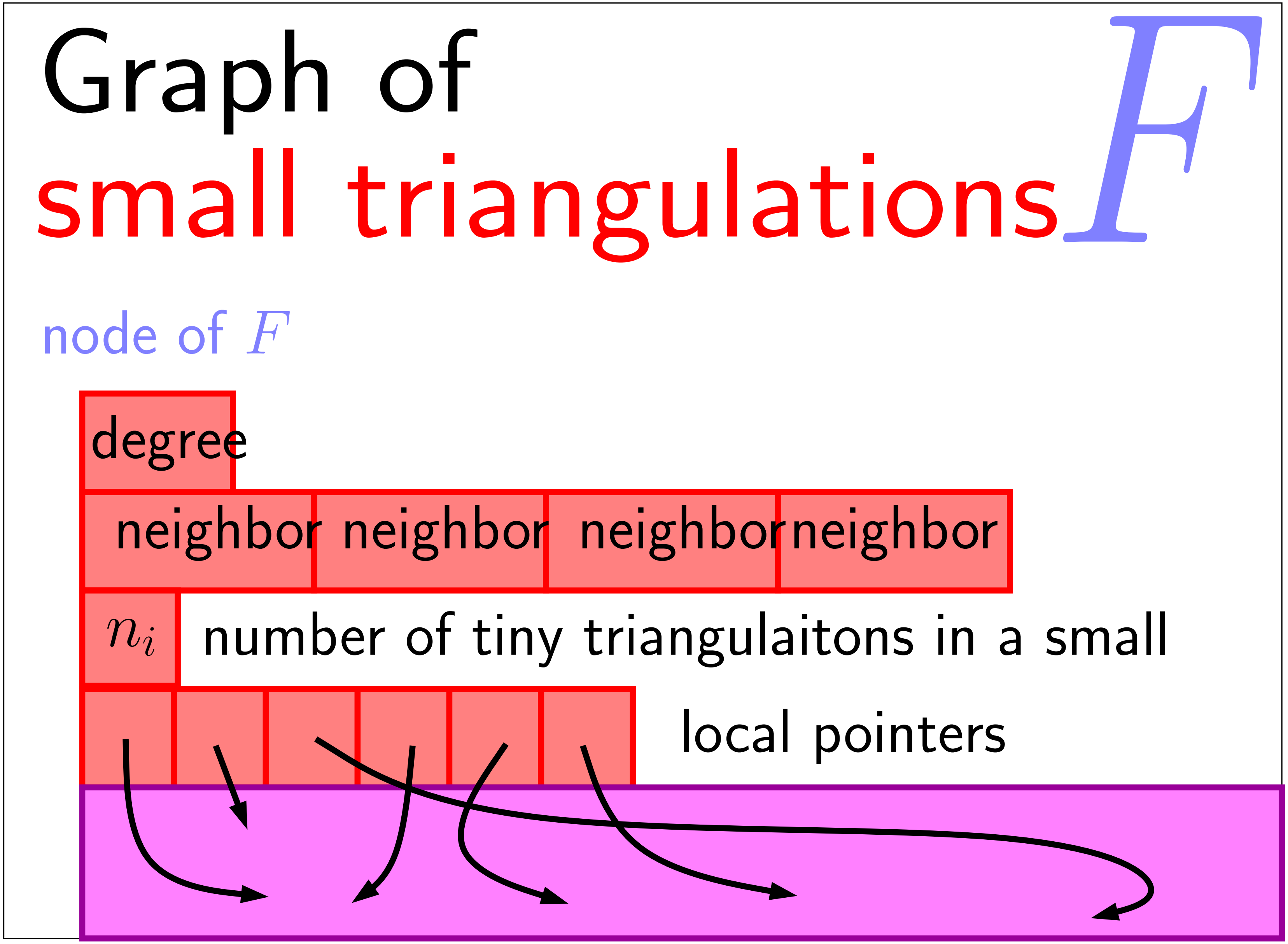
A small triangulations contains $\Theta(\lg m)$ tiny triangulations



Memory organization



Memory organization

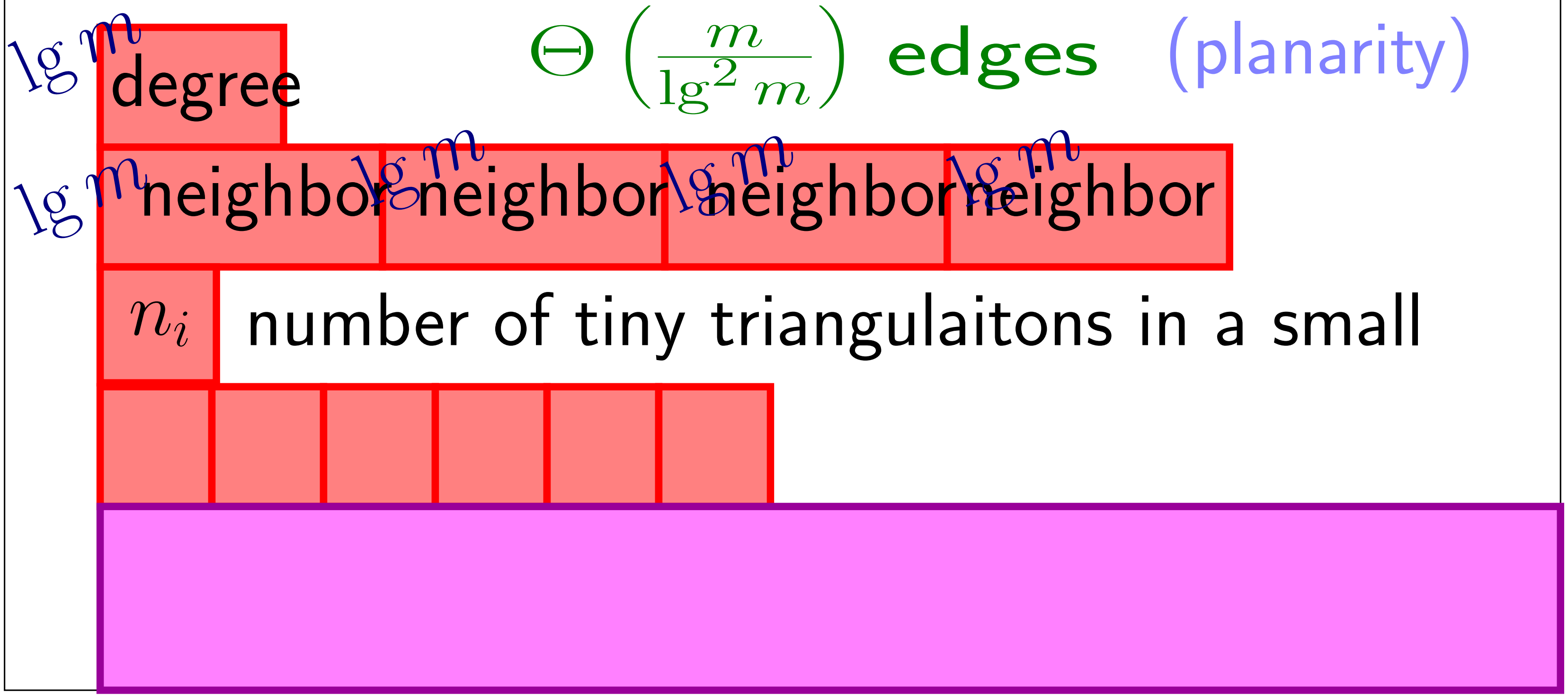


Memory organization

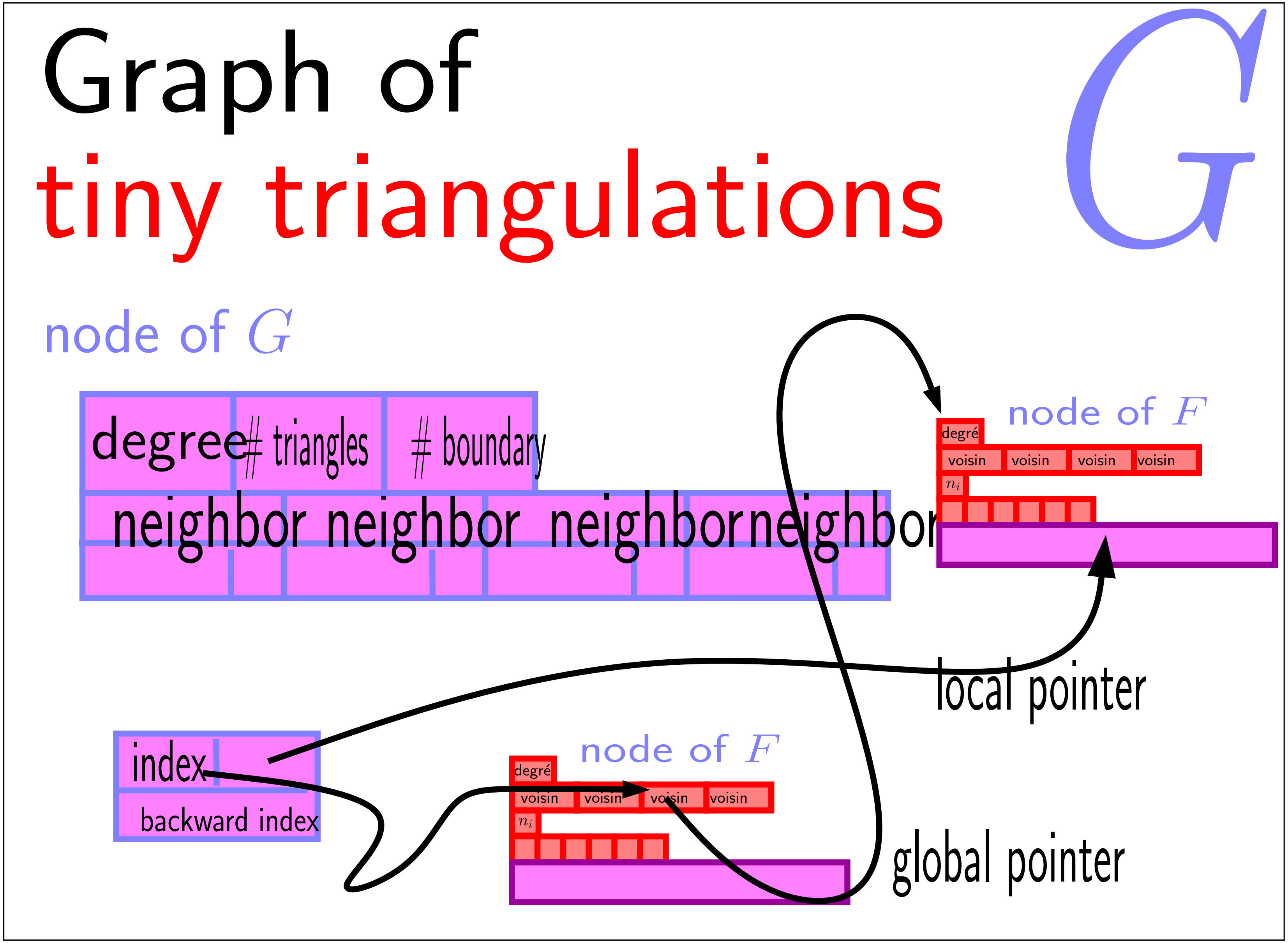
Graph of
small triangulations F

node of F $\quad \lg m \Theta\left(\frac{m}{\lg^2 m}\right) = \Theta\left(\frac{m}{\lg m}\right)$ **bits**

$\Theta\left(\frac{m}{\lg^2 m}\right)$ **edges** (planarity)



Memory organization

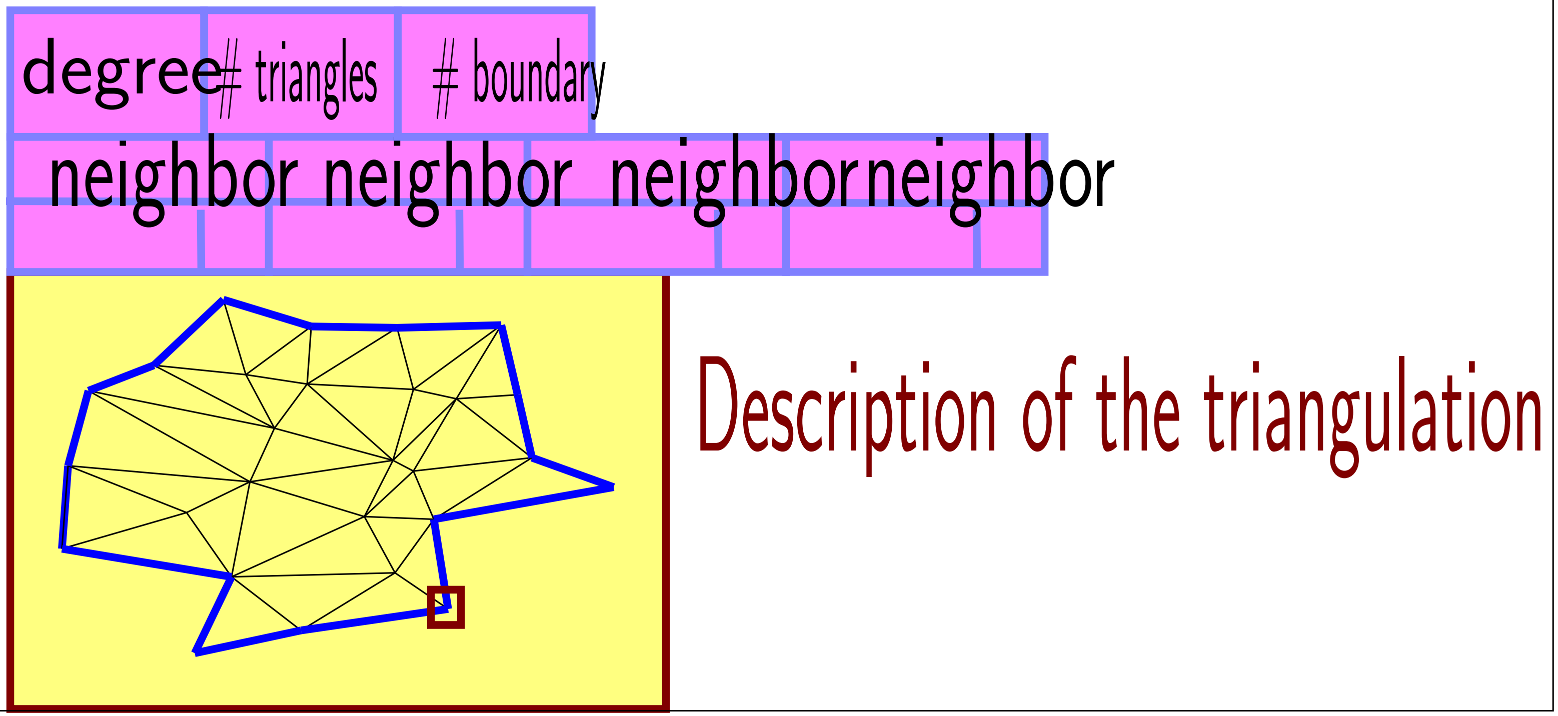


Memory organization

Graph of
tiny triangulations

G

node of G



Memory organization

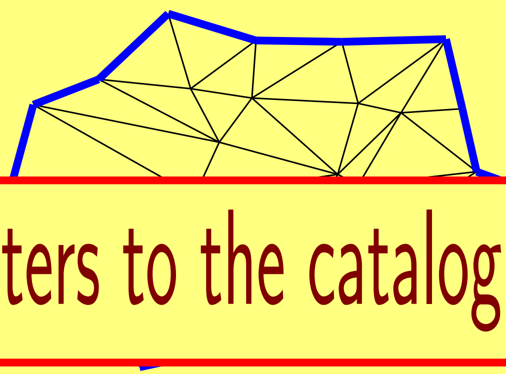
Graph of
tiny triangulations

G

node of G

degree | # triangles | # boundary

neighbor | neighbor | neighbor | neighbor



Description of the triangulation

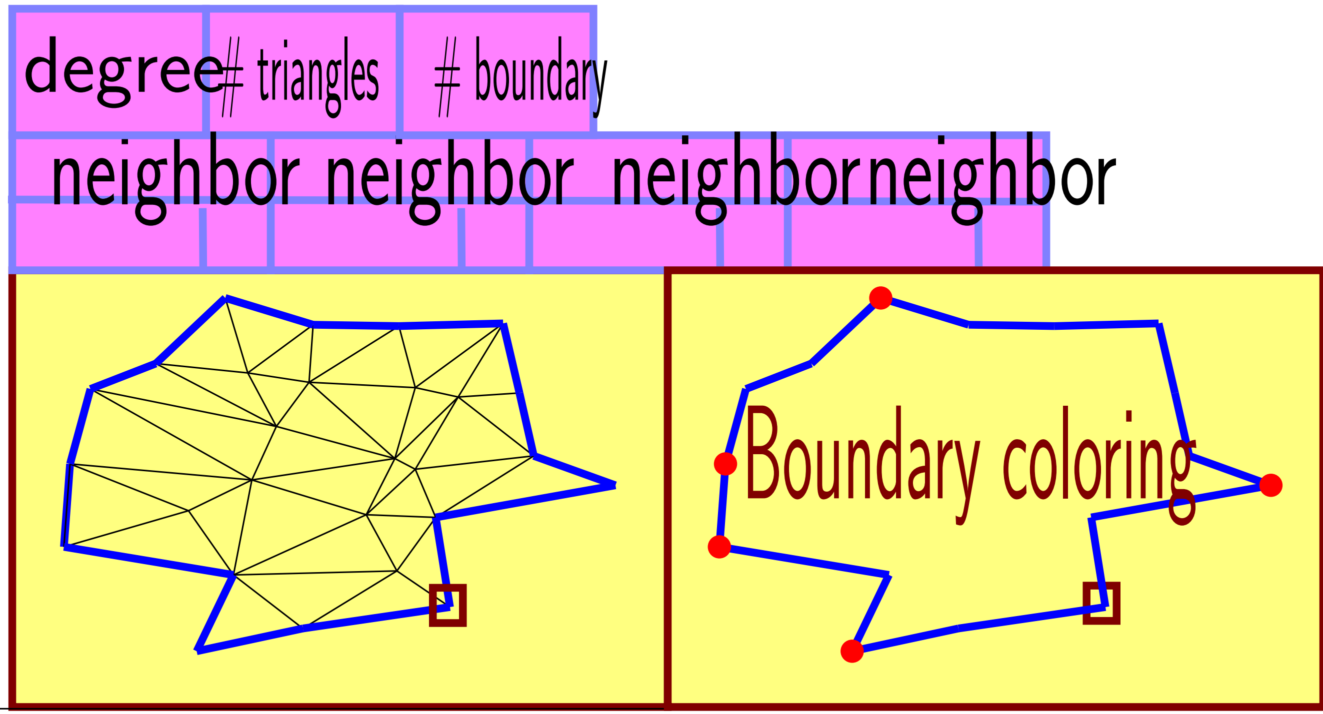
Pointers to the catalog of the triangulations with t triangles

Memory organization

Graph of G

tiny triangulations

node of G



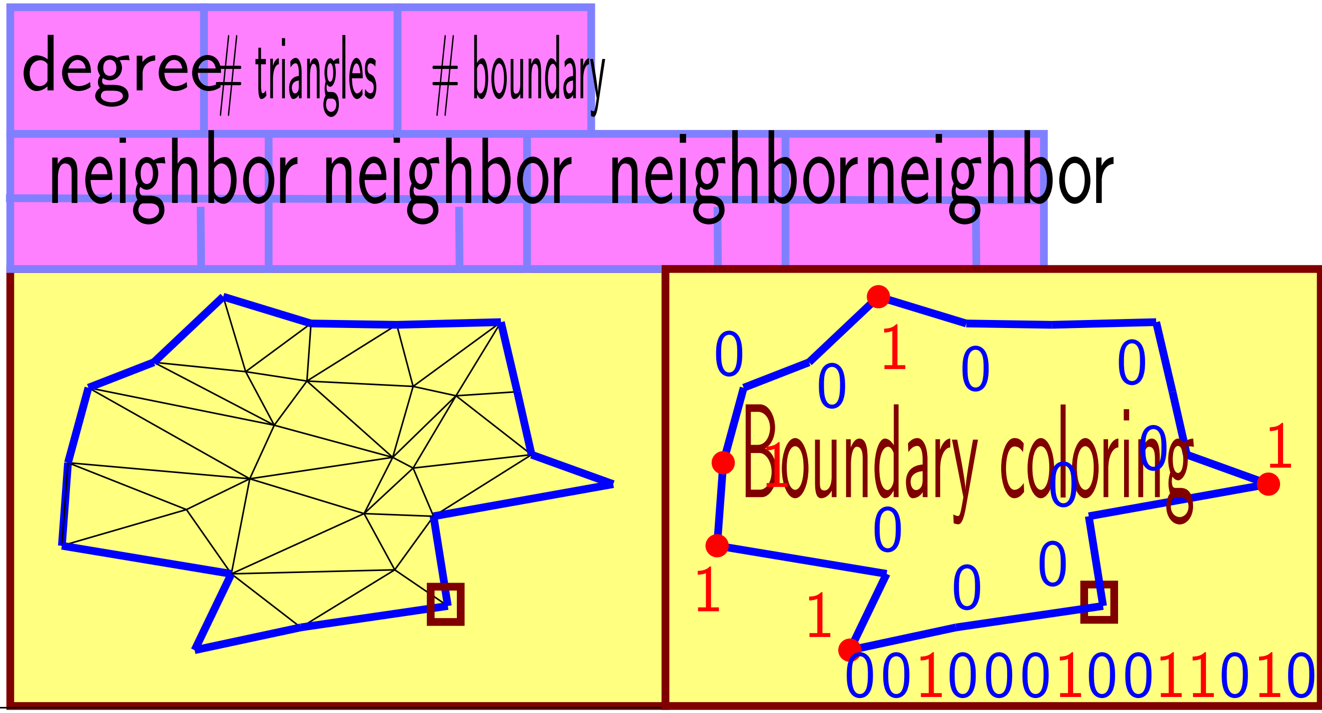
Memory organization

Graph of

tiny triangulations

G

node of G

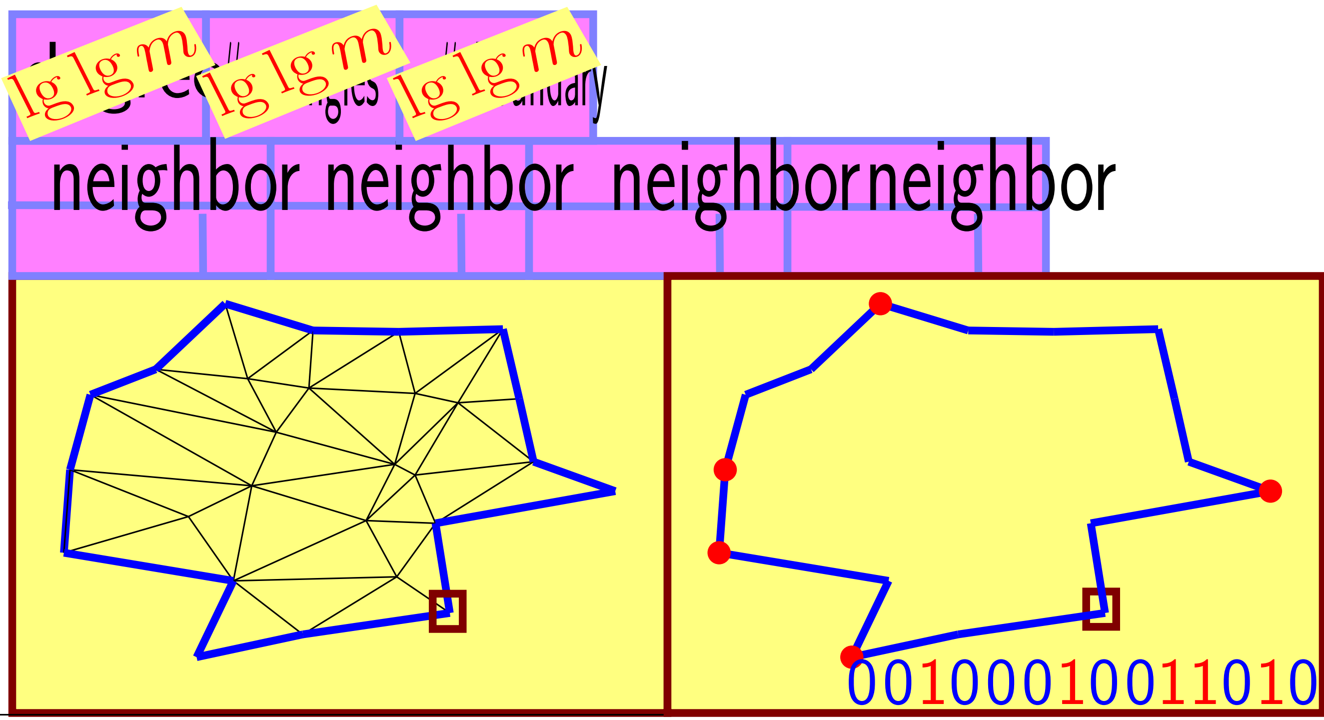


Memory organization

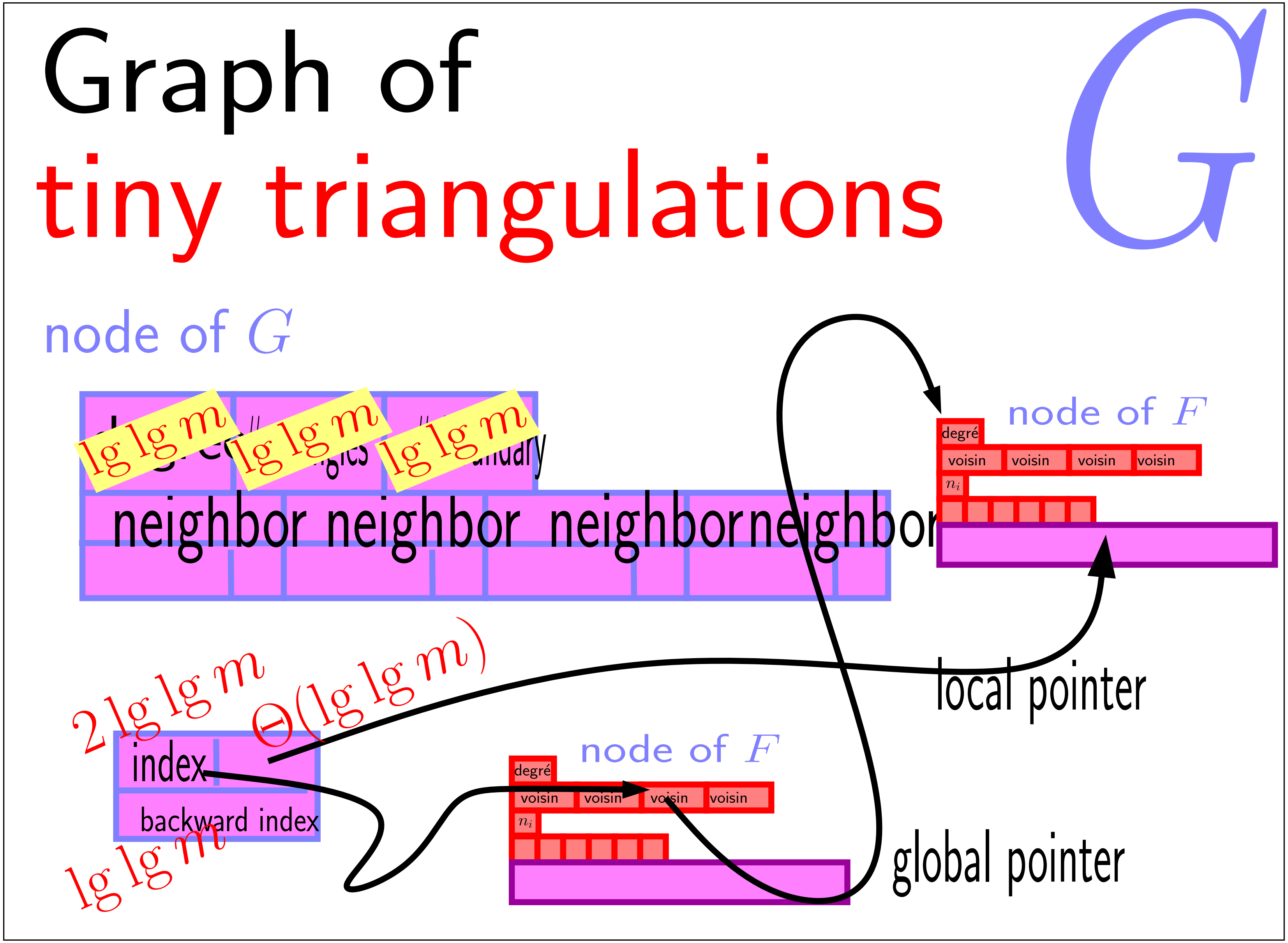
Graph of
tiny triangulations

G

node of G



Memory organization



Memory organization

Catalog of

tiny triangulations

t triangles

$2^{2.17t}$ triangulations

using each $t \lg t$ bits

$$\sum_{t=\frac{1}{12}\lg m}^{\frac{1}{4}\lg m} t \lg t \leq m^{0.55}$$

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)

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

Total space used

- Catalog of all different tiny triangulations

$$O(m^{\frac{1}{4}2.17}\lg^2 m \lg\lg m) = o(m)$$

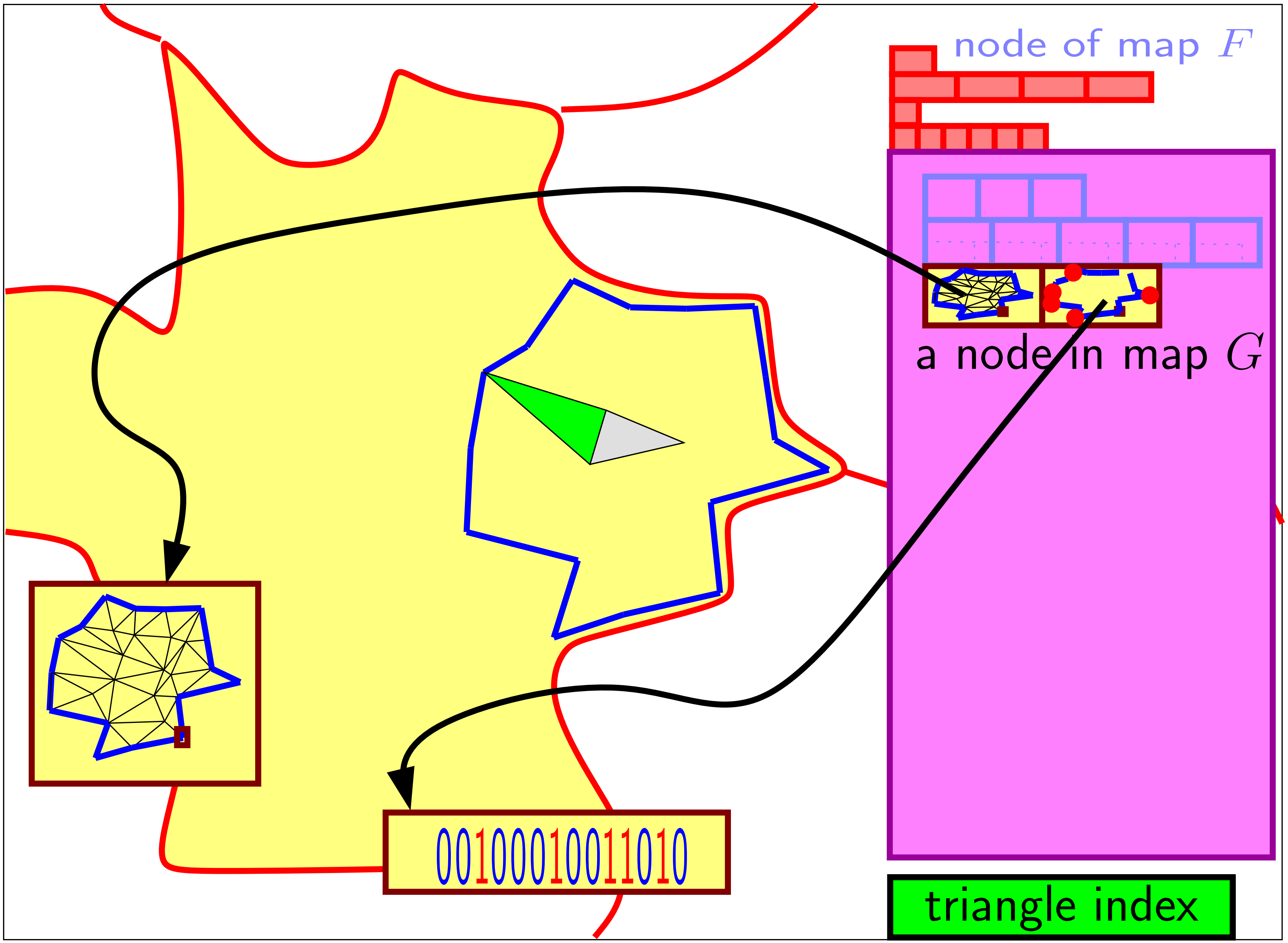
- catalog of bit-vectors (with Rank/Select)

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

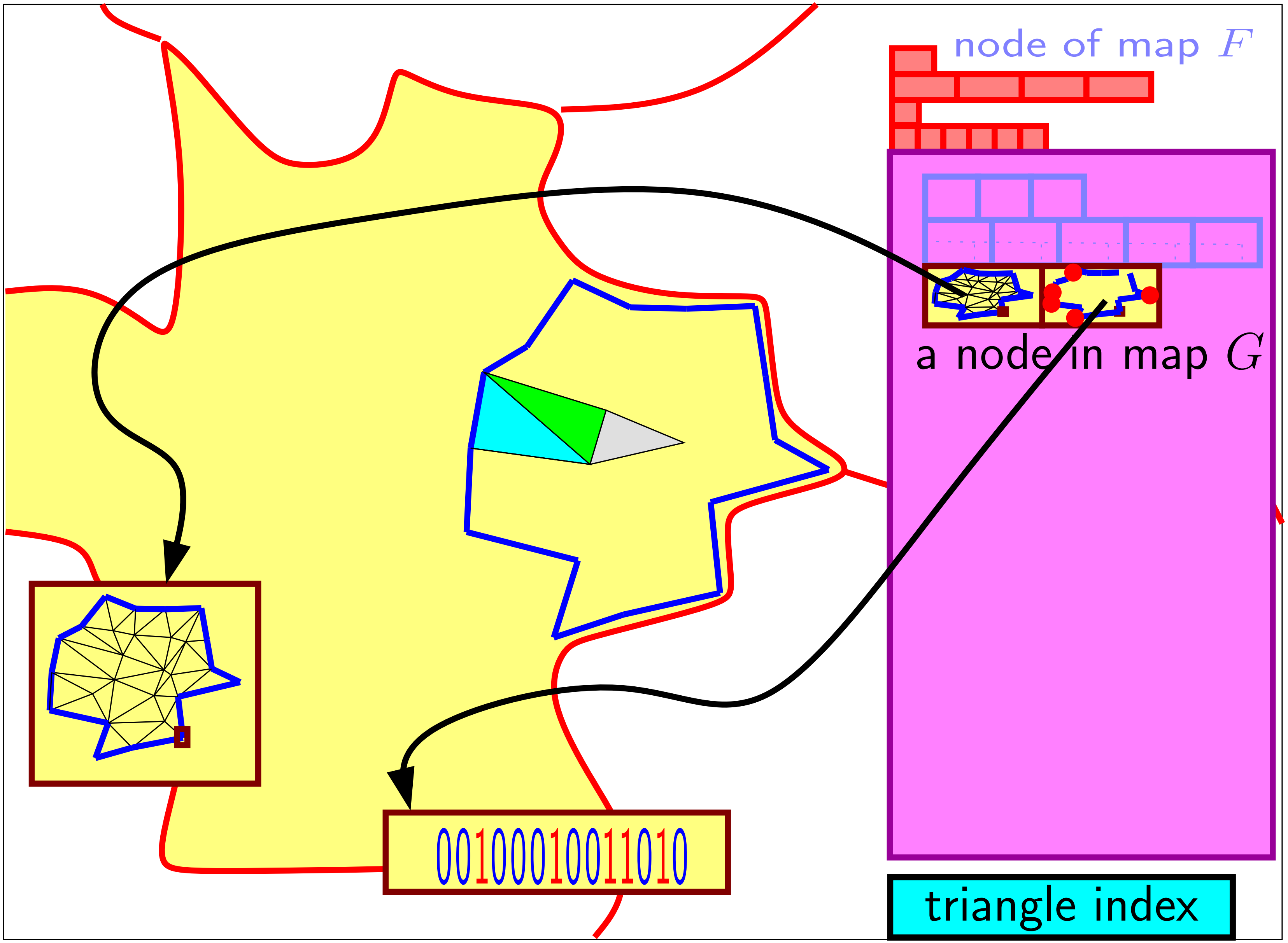
- representation of graph F: $O(\frac{m}{\lg^2 m}\lg m) = o(m)$
- graphs G_i

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

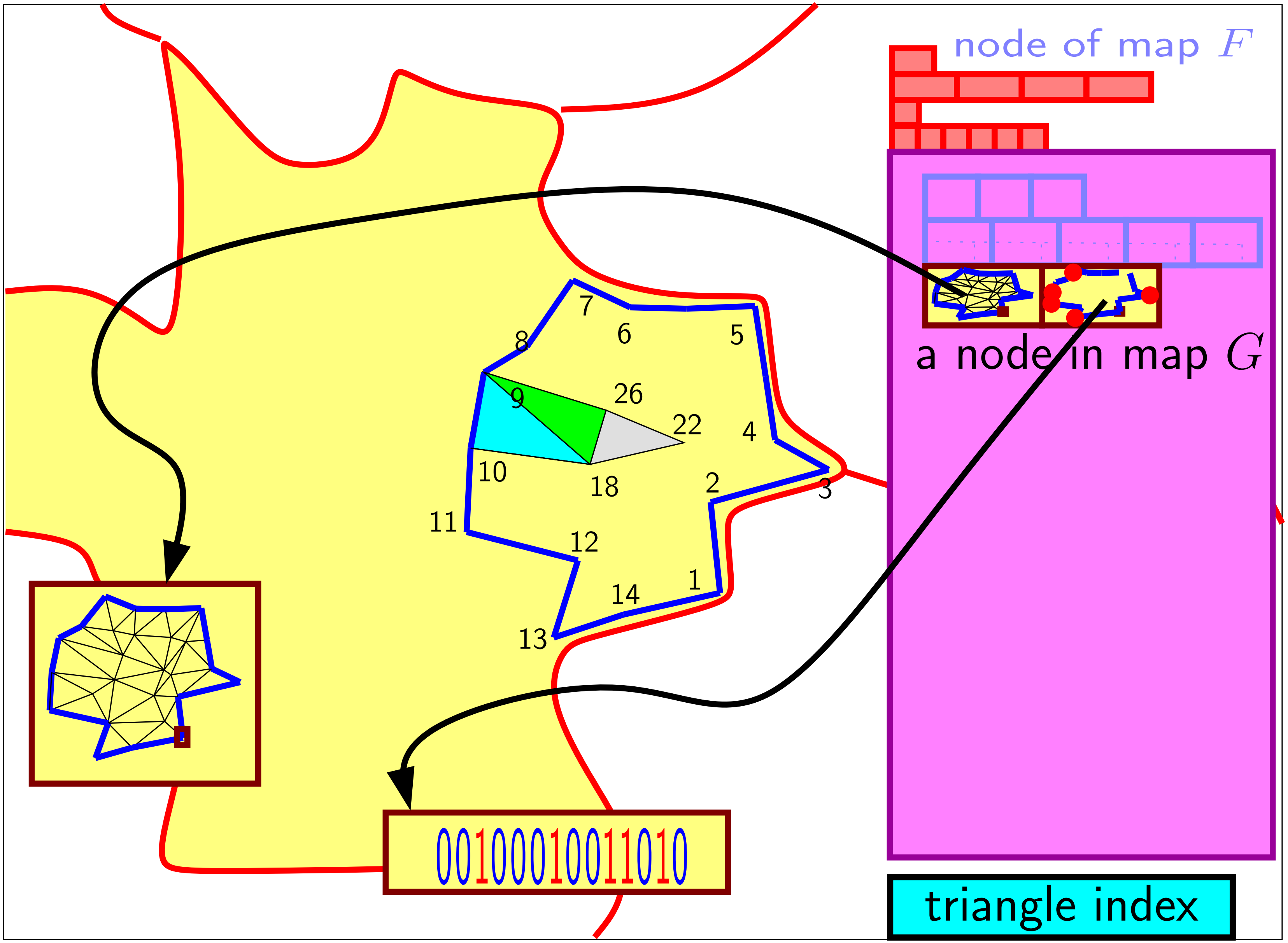
Navigation



Navigation

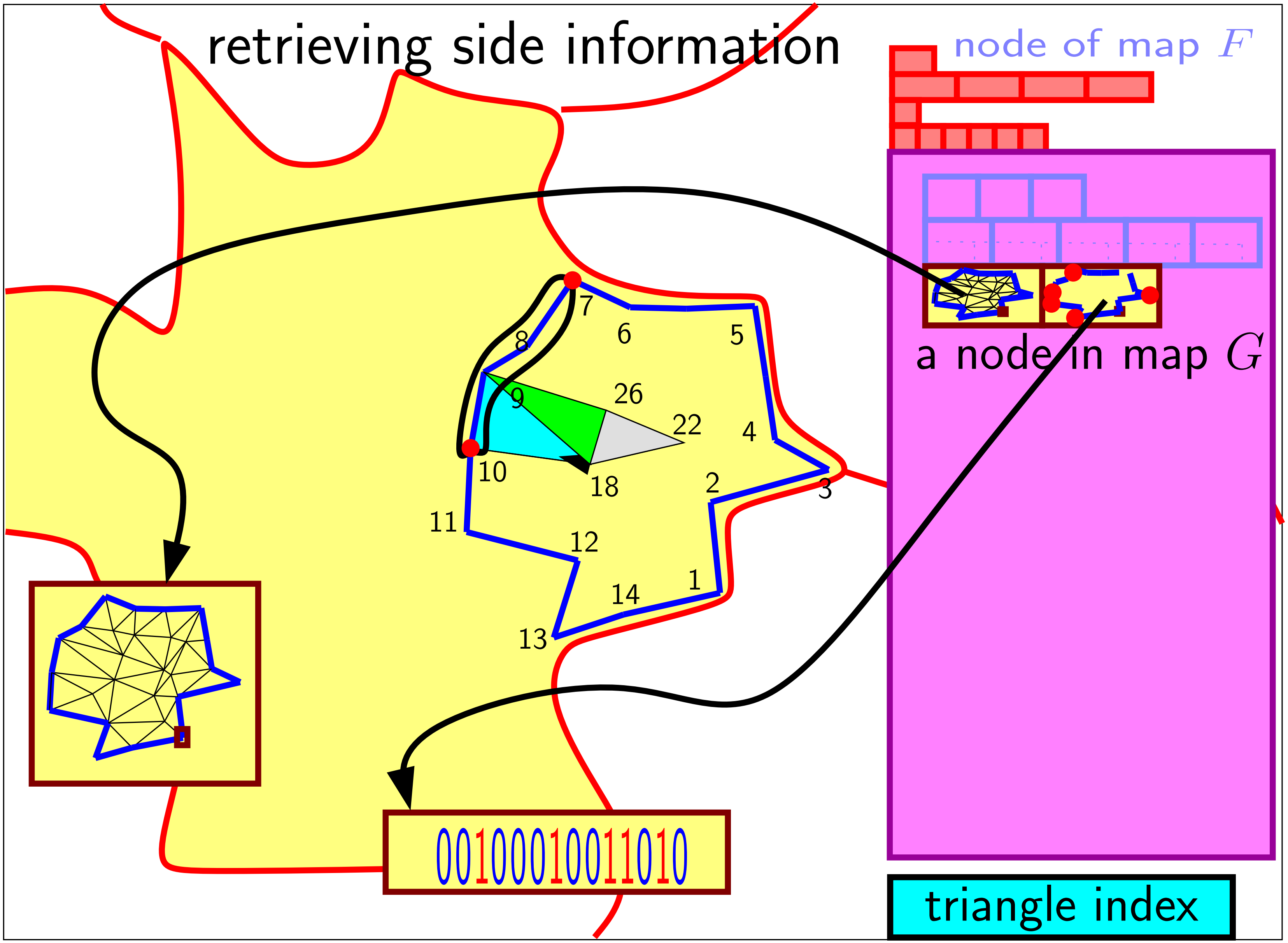

node of map F
a node in map G
0010001001101010
triangle index

Navigation



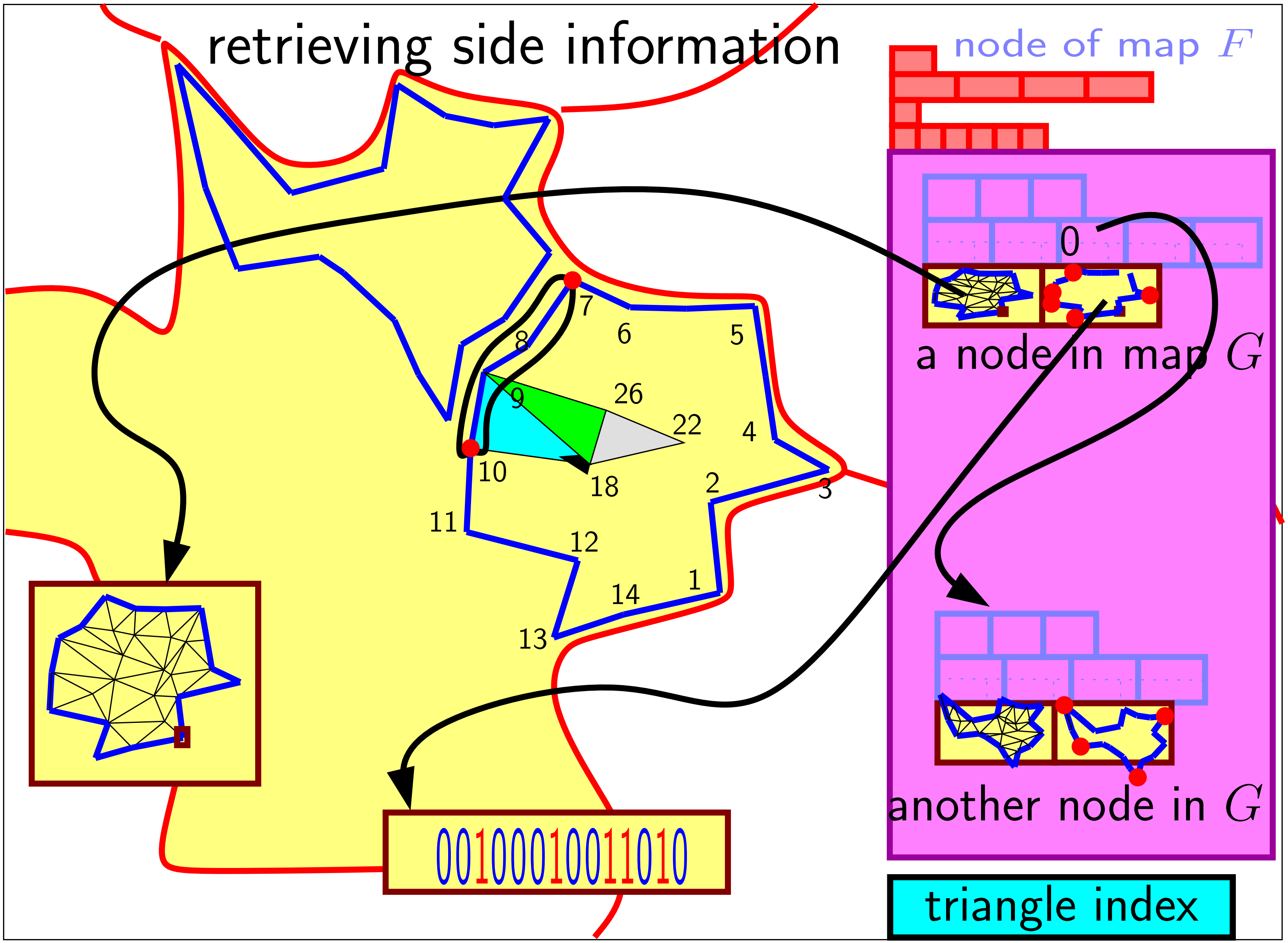
node of map F
a node in map G
7
6
5
8
9
26
22
4
10
18
2
3
11
12
14
1
13
0010001001101 0
triangle index

Navigation



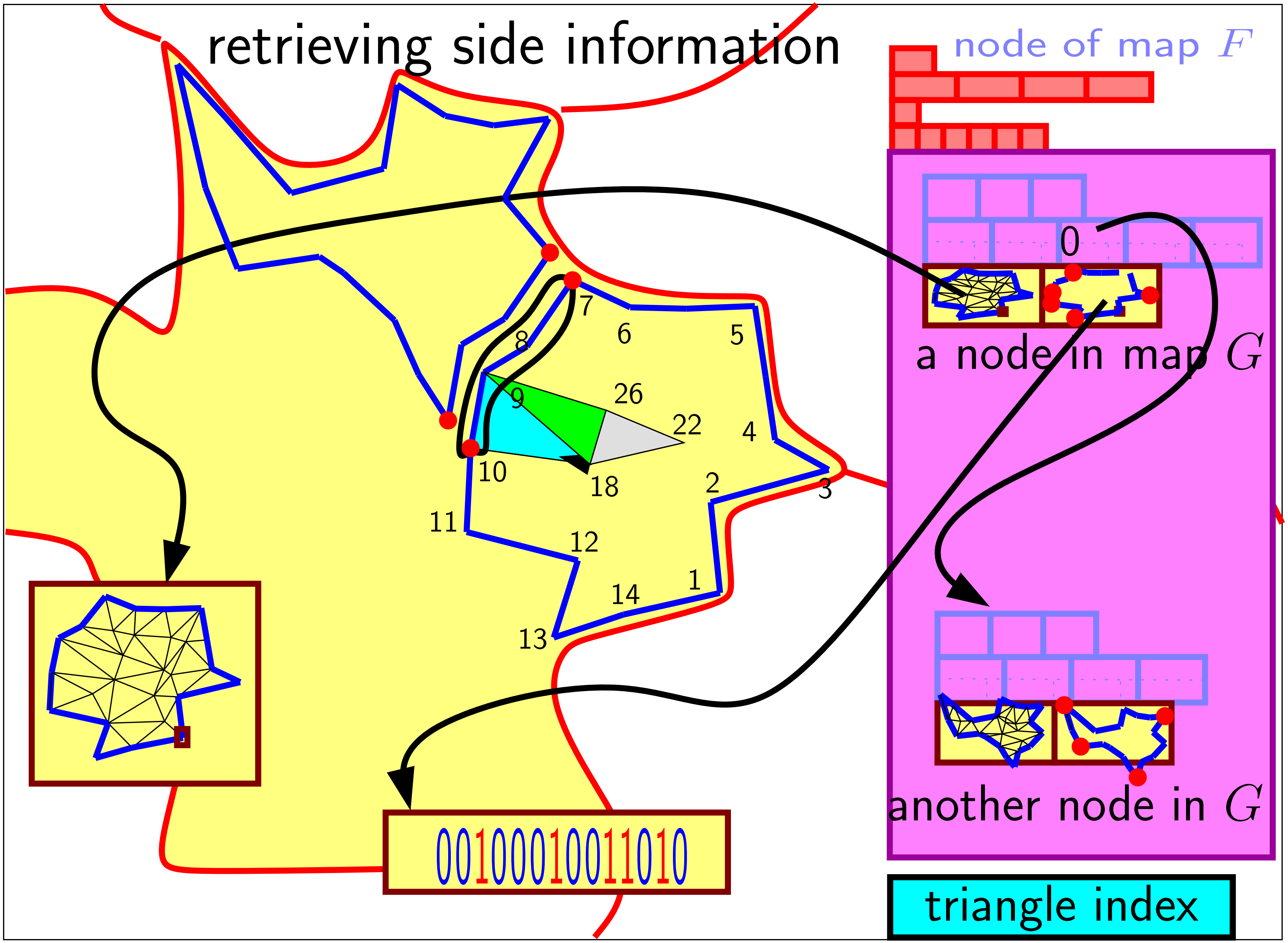
retrieving side information
node of map F
a node in map G
7
6
5
8
26
9
22
4
10
18
2
3
11
12
14
1
13
0010001001101 0
triangle index

Navigation



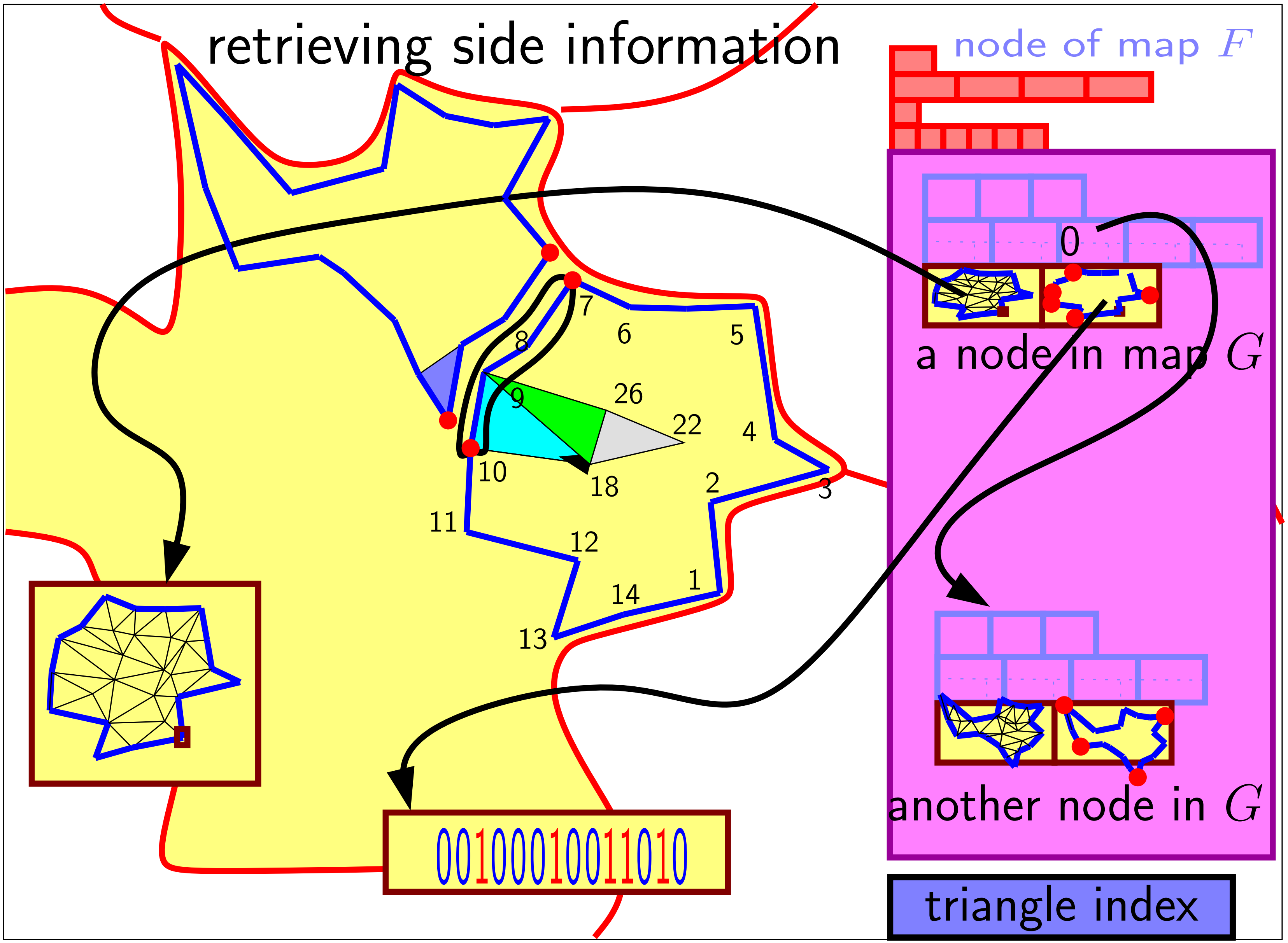
retrieving side information
node of map F
0
a node in map G
7
6
5
8
9
26
22
4
10
18
2
3
11
12
14
1
13
another node in G
0010001001101 0
triangle index

Navigation



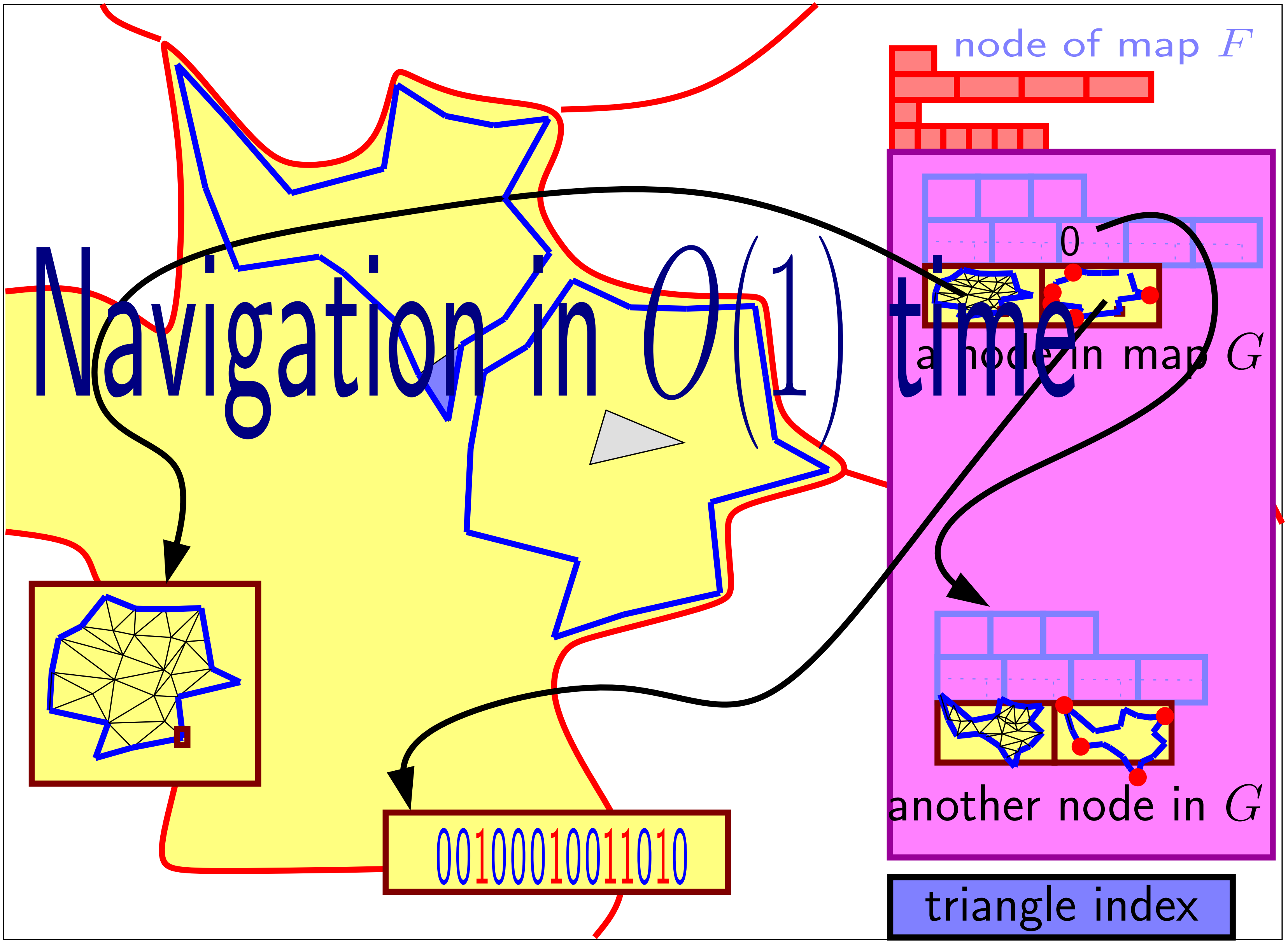
retrieving side information
node of map F
0
a node in map G
7
6
5
8
9
26
22
4
10
18
2
3
11
12
14
1
13
another node in G
0010001001101 0
triangle index

Navigation



retrieving side information
node of map F
0
7
8
6
5
9
26
22
4
10
18
2
3
11
12
1
14
13
a node in map G
another node in G
00100010011010
triangle index

Navigation



node of map F
0
Navigation in $O(1)$ time
a node in map G
another node in G
0010001001101 0
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

Dynamic extension

presented at CCCG 2005

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*

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

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)

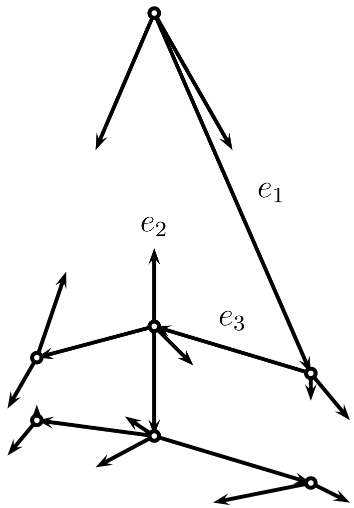
C++ implementation based on CGAL library

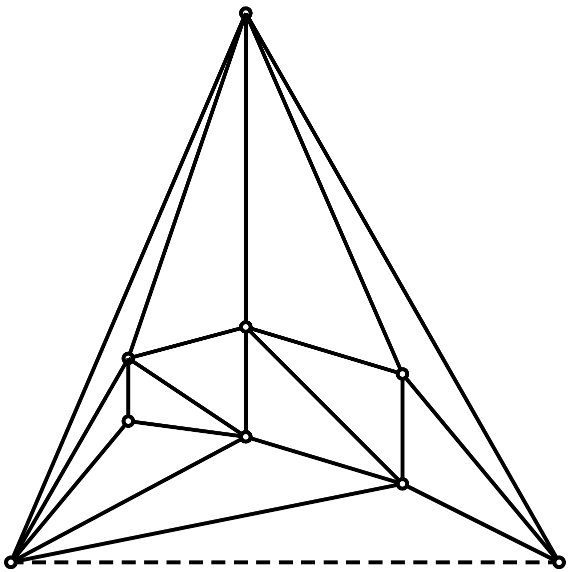
Idea: gathering triangles in small groups

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?