A Fast Algorithm for Computing Irreducible Triangulations of Closed Surfaces in $\mathbb{E}^d$ and Its Applications to the TriQuad Problem

Marcelo Siqueira

UFRN, Brazil

mfsiqueira@mat.ufrn.br

A Fast Algorithm for Computing Irreducible Triangulations of Closed Surfaces in $\mathbb{E}^d$ and Its Applications to the TriQuad Problem

A Fast Algorithm for Computing Irreducible Triangulations of Closed Surfaces in $\mathbb{E}^d$ and Its Applications to the TriQuad Problem

Joint work with

Thiago Lemos
UFPR, Brazil
thalemos@inf.ufpr.br

Suneeta Ramaswami
Rutgers University, USA
rsuneeta@camden.rutgers.edu

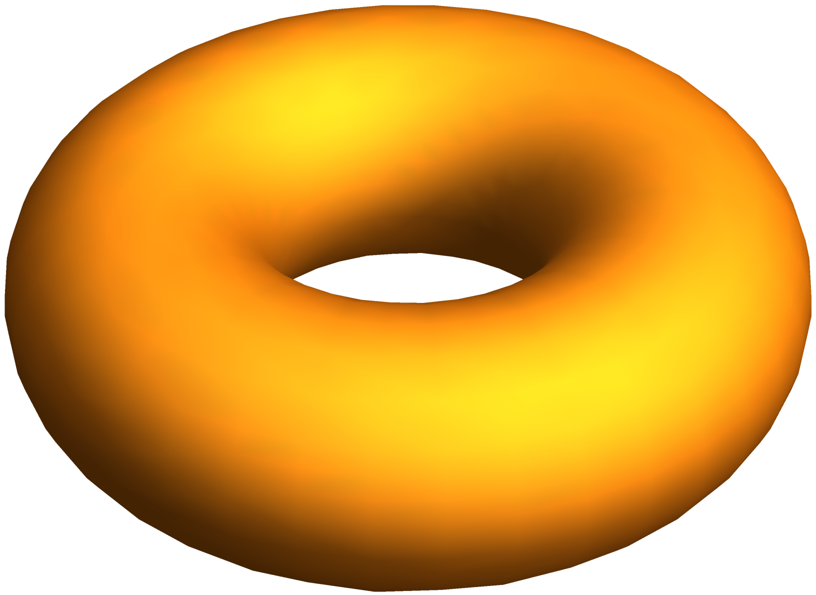
UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

1766 RUTGERS THE STATE UNIVERSITY OF NEW JERSEY

Problem Statement

Problem Statement

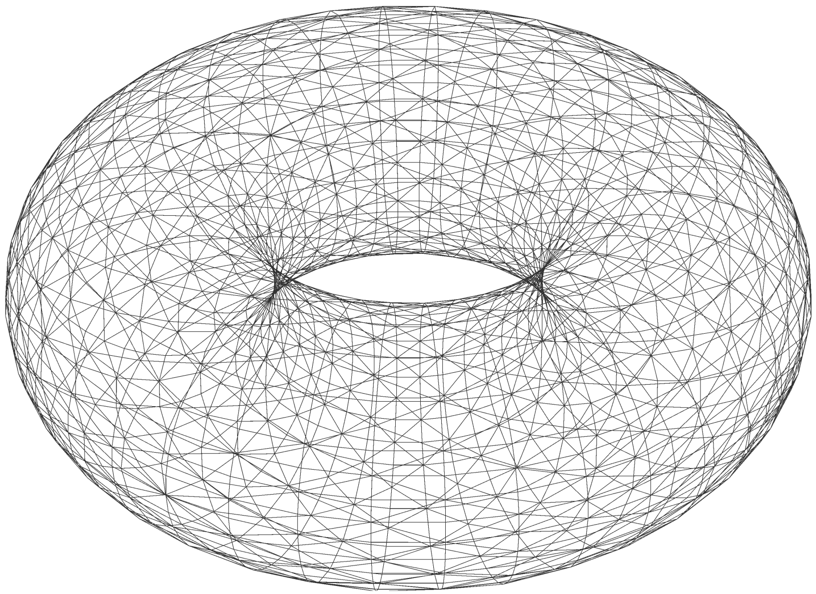
Let $\mathcal{S}$ be a connected, boundaryless, and compact surface in $\mathbb{E}^d$.



Problem Statement

Problem Statement

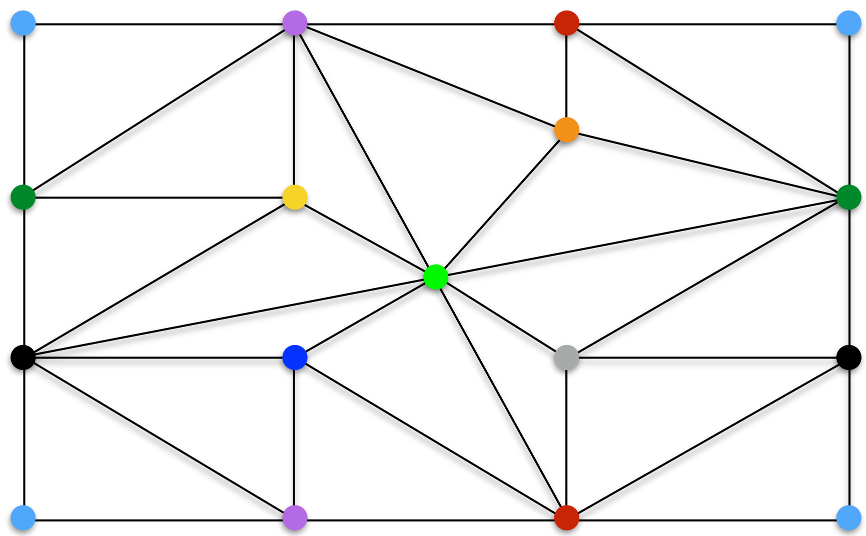
Given a triangulation $\mathcal{T}$ of $\mathcal{S}$,



Problem Statement

Problem Statement

find an **irreducible triangulation** $\mathcal{T}'$ of $\mathcal{S}$ from $\mathcal{T}$:

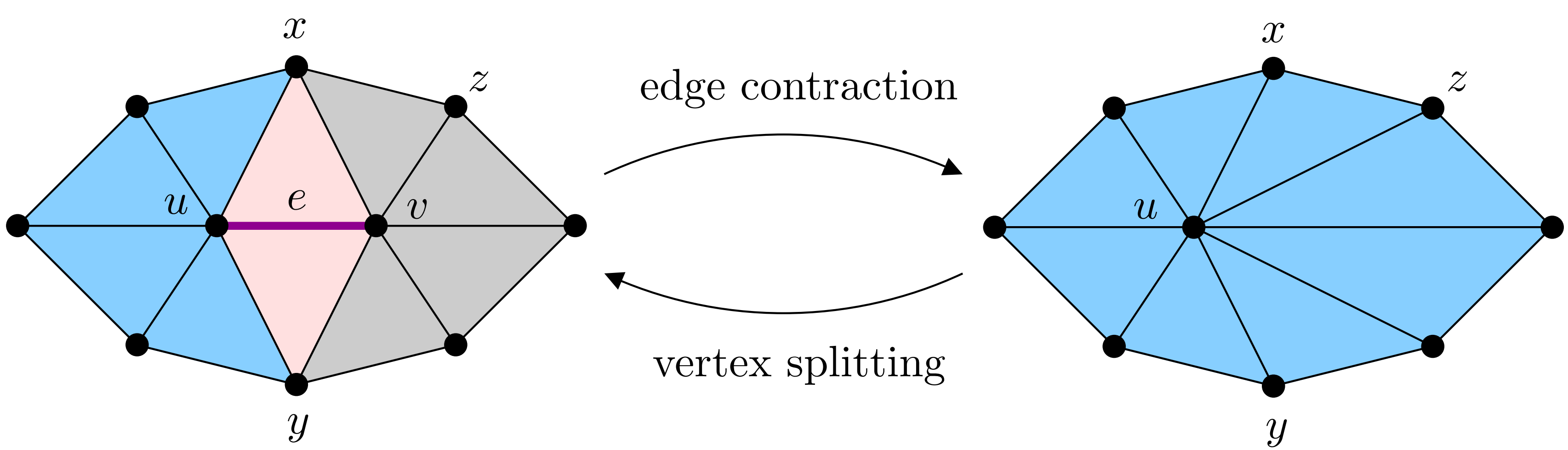


$$V - E + F = 10 - 30 + 20 = 0 = 2 \cdot 0 = 2 \cdot (1 - 1) = 2 \cdot (1 - g)$$

Problem Statement

Problem Statement

$\mathcal{T}'$ is irreducible if and only if $\mathcal{T}'$ has no contractible edges

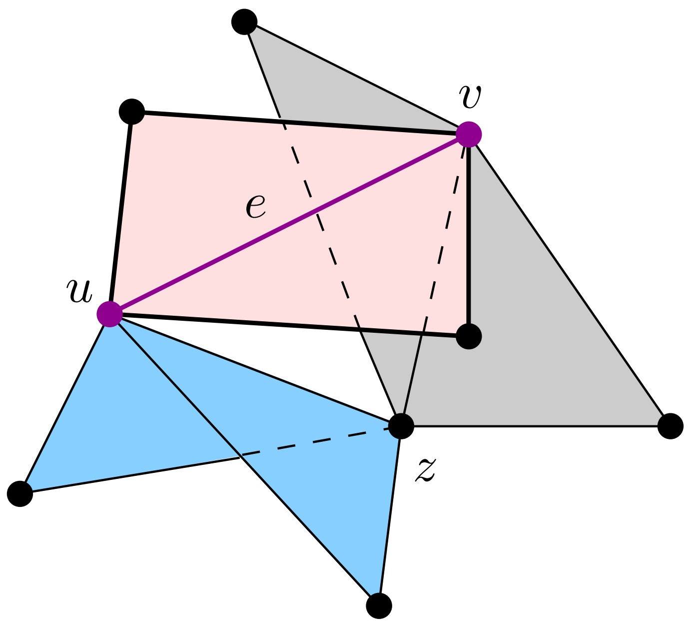


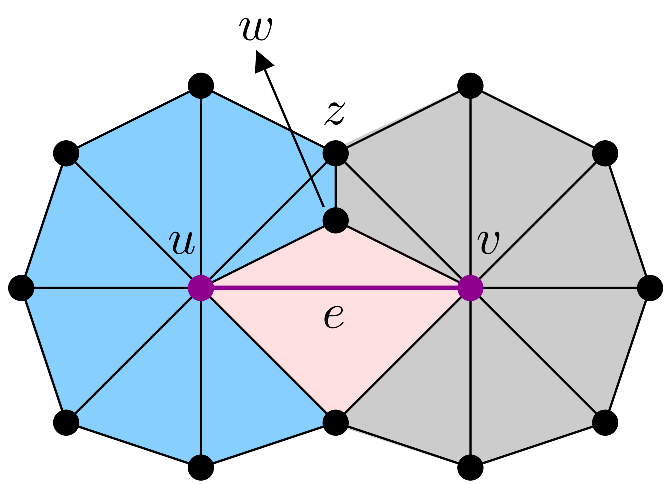
Problem Statement

$\mathcal{T}'$ is irreducible if and only if $\mathcal{T}'$ has no contractible edges

Problem Statement

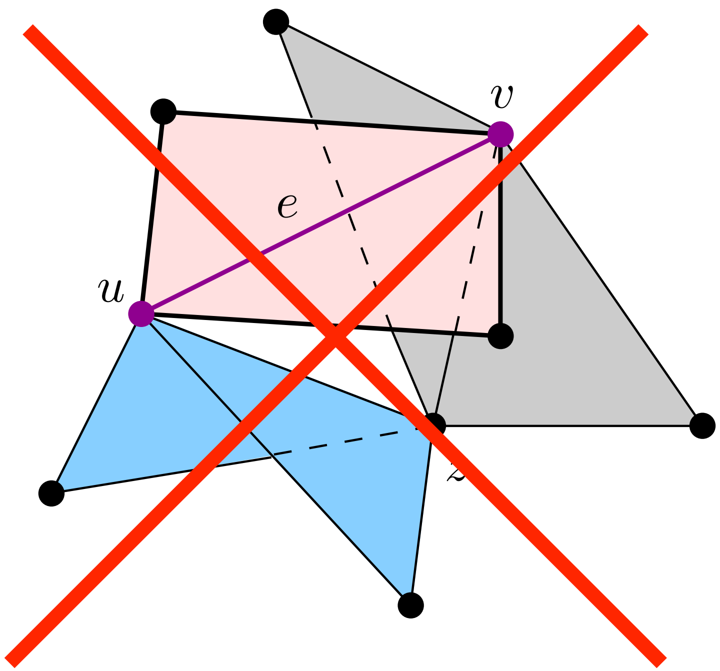
$\mathcal{T}'$ **is irreducible if and only if** $\mathcal{T}'$ **has no contractible edges**

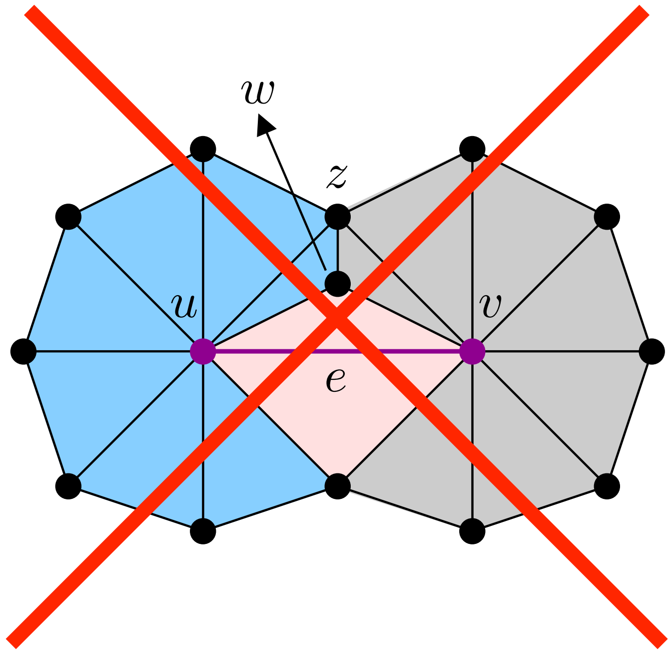




Problem Statement

$\mathcal{T}'$ is irreducible if and only if $\mathcal{T}'$ has no contractible edges

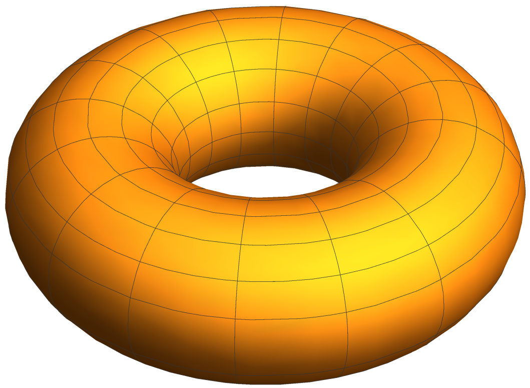




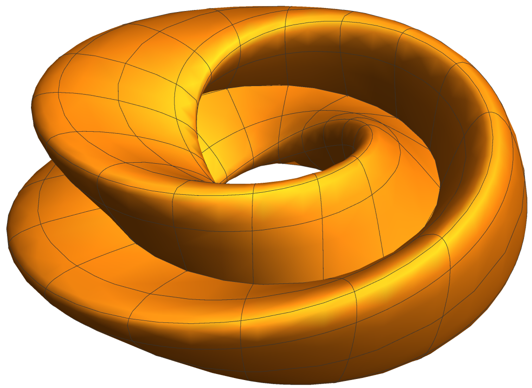
Facts

Facts

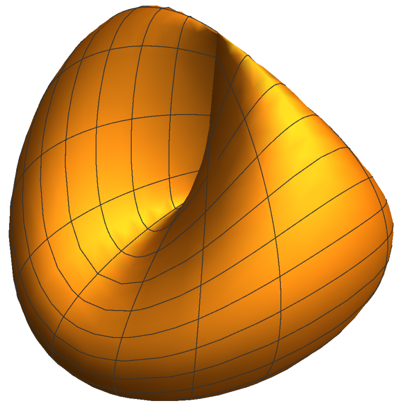
All closed surfaces have finitely many irreducible triangulations (Barnette & Edelson, 1989).



21, with $n_v \in \{7, 8, 9, 10\}$



29, with $n_v \in \{8, 9, 10, 11\}$



2, with $n_v \in \{6, 7\}$

Facts

Facts

If the genus g of $\mathcal{S}$ is positive, then any irreducible triangulation of *smallest* size (known as *minimal*) has $\Theta(\sqrt{g})$ vertices. Also, for any irreducible triangulation of $\mathcal{S}$,

$$n_v \leq 13 \cdot h - 4\,,$$

where h is the *Euler* genus of $\mathcal{S}$ (if $\mathcal{S}$ is orientable, then $h = 2g$. Otherwise, $h = g$.)

(Joret & Wood, 2010)

Facts

Facts

The largest known irreducible triangulation of an orientable surface of genus g has

$$n_v = \left\lfloor \frac{17}{2} \cdot g \right\rfloor$$

(Sulanke, 2006)

Why Should One Care?

Why Should One Care?

All irreducible triangulations of $\mathcal{S}$ form a "basis" for all triangulations of $\mathcal{S}$.

Why Should One Care?

All irreducible triangulations of $\mathcal{S}$ form a ″basis″ for all triangulations of $\mathcal{S}$.

Have been used for

- proving the existence of geometric realizations;
- studying properties of diagonal flips on surfaces triangulations;
- characterizing the structure of flexible triangulations;
- finding bounds for the number of cliques in graphs on surfaces.

Why Do We Care?

Why Do We Care?

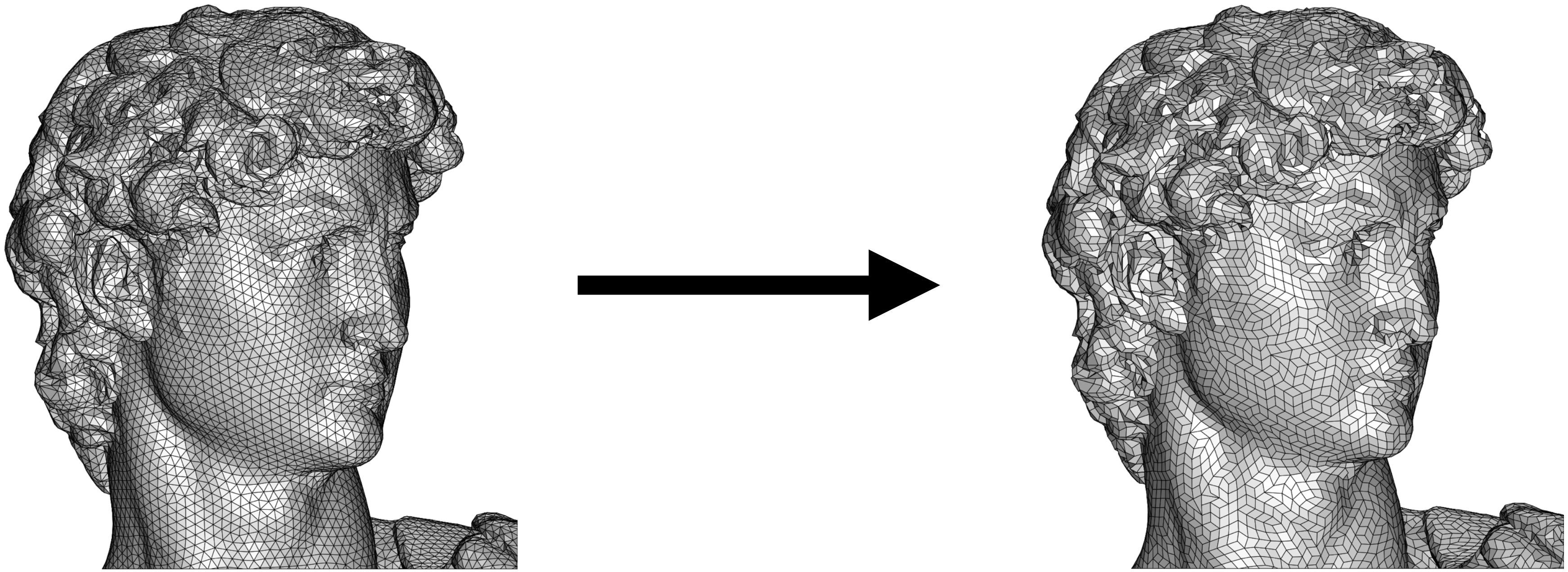
The TriQuad Problem

Given a triangulation, $\mathcal{T}$, of $\mathcal{S}$, obtain a quadrangulation, $\mathcal{Q}$, of $\mathcal{S}$ whose vertex set is a (not necessarily proper) superset of the vertex set of $\mathcal{T}$.

Why Do We Care?

The TriQuad Problem

Given a triangulation, $\mathcal{T}$, of $\mathcal{S}$, obtain a quadrangulation, $\mathcal{Q}$, of $\mathcal{S}$ whose vertex set is a (not necessarily proper) superset of the vertex set of $\mathcal{T}$.



Main Result

Main Result

Given a triangulation $\mathcal{T}$ of connected, boundaryless, compact surface $\mathcal{S}$ of genus g, there exists an algorithm for computing an irreducible triangulation $\mathcal{T}'$ of $\mathcal{S}$ in $\mathcal{O}(g^2 + gn_f)$ time if g is positive; otherwise, $\mathcal{T}'$ can be computed in $\mathcal{O}(n_f)$ time, where n_f is the number of triangles of $\mathcal{T}$.

Main Result

Given a triangulation $\mathcal{T}$ of connected, boundaryless, compact surface $\mathcal{S}$ of genus g, there exists an algorithm for computing an irreducible triangulation $\mathcal{T}'$ of $\mathcal{S}$ in $\mathcal{O}(g^2 + gn_f)$ time if g is positive; otherwise, $\mathcal{T}'$ can be computed in $\mathcal{O}(n_f)$ time, where n_f is the number of triangles of $\mathcal{T}$.

When $g > 0$, our upper bound improves upon the previously best upper bound by a $\lg n_f / g$ factor, which was given by Haijo Schipper in 1991.

Main Result

Given a triangulation $\mathcal{T}$ of connected, boundaryless, compact surface $\mathcal{S}$ of genus g, there exists an algorithm for computing an irreducible triangulation $\mathcal{T}'$ of $\mathcal{S}$ in $\mathcal{O}(g^2 + gn_f)$ time if g is positive; otherwise, $\mathcal{T}'$ can be computed in $\mathcal{O}(n_f)$ time, where n_f is the number of triangles of $\mathcal{T}$.

When $g > 0$, our upper bound improves upon the previously best upper bound by a $\lg n_f / g$ factor, which was given by Haijo Schipper in 1991.

$\mathcal{O}(n_f \lg n_f + g \ln n_f + g^4)$

Common Strategy

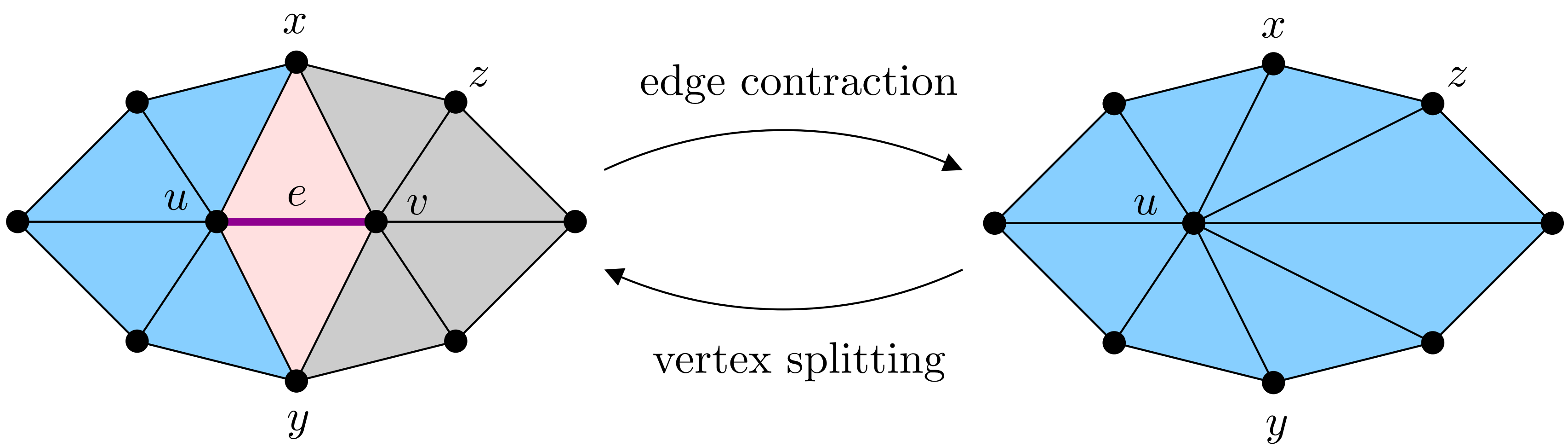
Common Strategy

Let K be the current triangulation (i.e., $K = \mathcal{T}$ initially).

Common Strategy

Let K be the current triangulation (i.e., $K = \mathcal{T}$ initially).

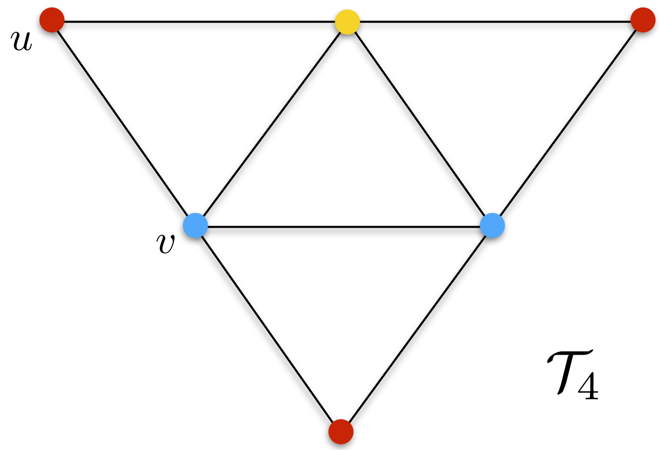
While there exists a *contractible* edge $e = [u, v]$ in the current triangulation K, contract e, identifying v with u and producing triangulation $K - uv$.



Link Condition Test

Link Condition Test

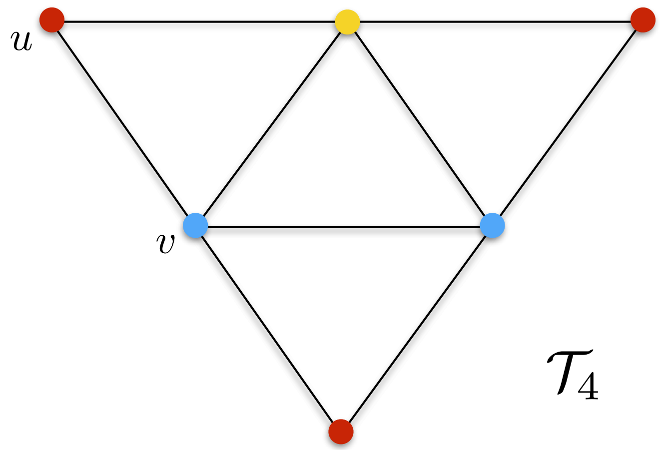
If K is (isomorphic to) $\mathcal{T}_4$ then no edge of K is contractible.



$\mathcal{T}_4$

Link Condition Test

If K is (isomorphic to) $\mathcal{T}_4$ then no edge of K is contractible.

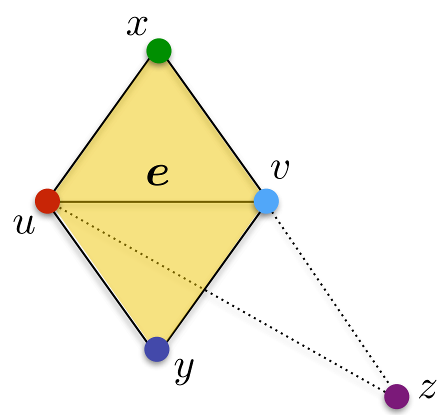


$$K \approx \mathcal{T}_4 \iff d_u = d_v = 3$$

Link Condition Test

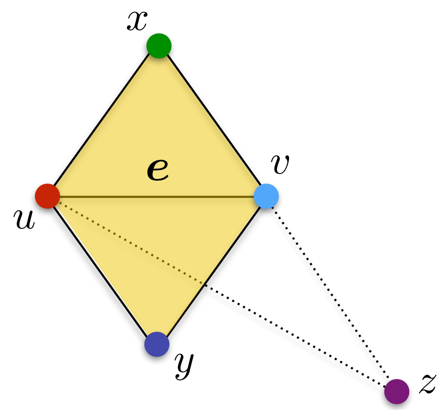
Link Condition Test

If K is not (isomorphic to) $\mathcal{T}_4$, then $e = [u, v]$ is contractible iff e does not belong to a *critical cycle* (i.e., a 3-cycle that does not bound a triangle) in K.



Link Condition Test

If K is not (isomorphic to) $\mathcal{T}_4$, then $e = [u, v]$ is contractible iff e does not belong to a *critical cycle* (i.e., a 3-cycle that does not bound a triangle) in K.



Edge $e = [u, v]$ belongs to a critical cycle iff u and v have a common neighbor other than x and y, where x and y are the vertices of the link of e.

Time Complexity

Time Complexity

Time complexity is dictated by two factors:

Time Complexity

Time complexity is dictated by two factors:

(1) The cost of testing an edge against the link condition, and

Time Complexity

Time complexity is dictated by two factors:

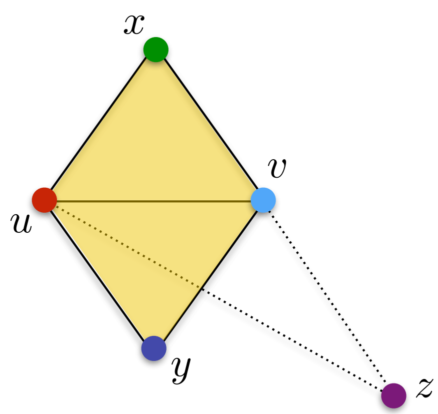
(1) The cost of testing an edge against the link condition, and

(2) The number of link condition tests executed by the algorithm.

Time Complexity

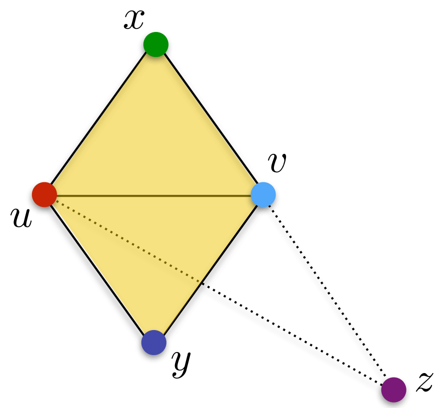
Time Complexity

(1) The cost of testing an edge against the link condition, and



Time Complexity

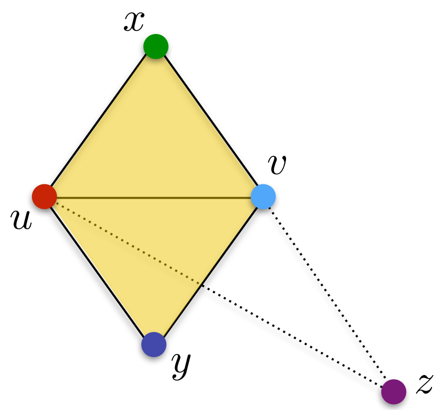
(1) The cost of testing an edge against the link condition, and



$\Omega(d_u \cdot d_v)$

Time Complexity

(1) The cost of testing an edge against the link condition, and



$$\Omega(d_u \cdot d_v)$$

Naïve

$$\Omega(d_u \cdot \lg d_v)$$

Neighborhood dictionary

Time Complexity

Time Complexity

(2) The number of link condition tests executed by the algorithm.

Time Complexity

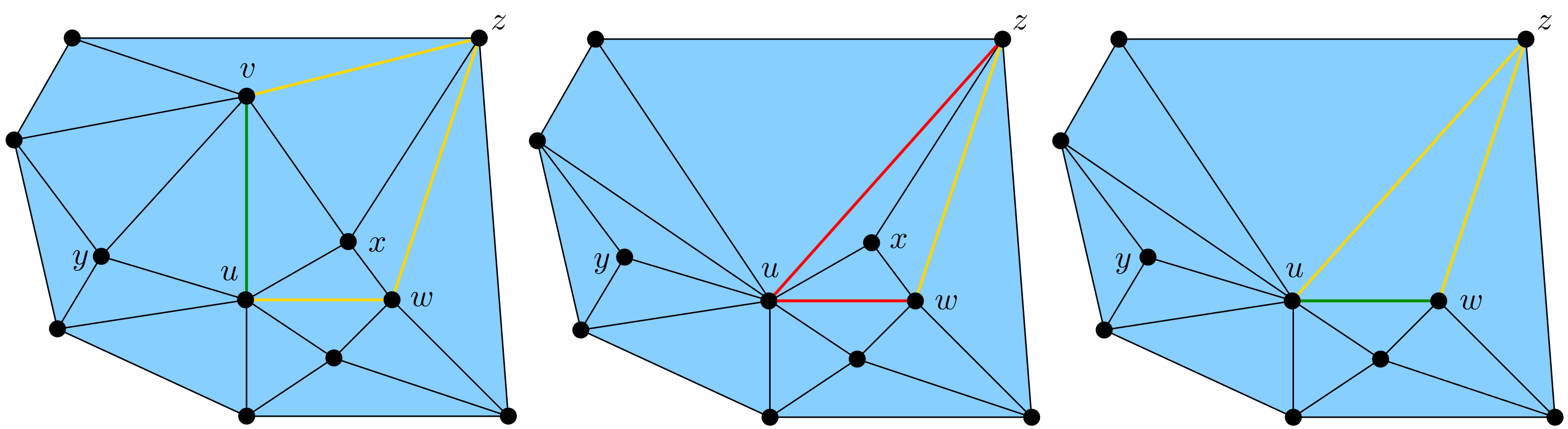
(2) The number of link condition tests executed by the algorithm.

Bounding this number is challenging because the contraction of a contractible edge can make a previously non-contractible edge contractible and vice-versa.

Time Complexity

(2) The number of link condition tests executed by the algorithm.

Bounding this number is challenging because the contraction of a contractible edge can make a previously non-contractible edge contractible and vice-versa.



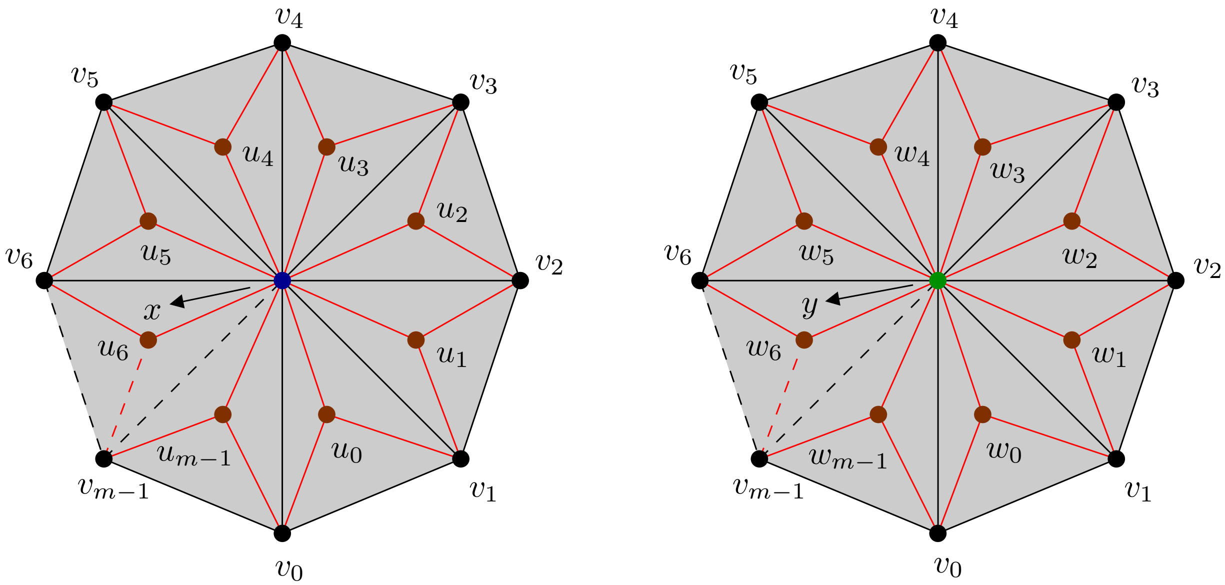
K $\qquad$ $K - uv$ $\qquad$ $(K - uv) - ux$

Time Complexity

(2) The number of link condition tests executed by the algorithm.

Time Complexity

(2) The number of link condition tests executed by the algorithm.



$n_v = 3m + 2$ vertices, with $n_f \in \Theta(n_v)$

Testing all edges $[v_i, x]$ and $[v_i, y]$ *first* may take $\Omega(n_f \lg n_f)$ time

Our Approach

Our Approach

Let K be the current triangulation (i.e., $K = \mathcal{T}$ initially).

Our Approach

Let K be the current triangulation (i.e., $K = \mathcal{T}$ initially).

Choose a vertex u from K.

Our Approach

Let K be the current triangulation (i.e., $K = \mathcal{T}$ initially).

Choose a vertex u from K.

Contract all edges of the form $[u, v]$ until u becomes *trapped*.

Our Approach

Let K be the current triangulation (i.e., $K = \mathcal{T}$ initially).

Choose a vertex u from K.

Contract all edges of the form $[u, v]$ until u becomes *trapped*.

A vertex w in K is **trapped** if all edges incident on w in K are non-contractible edges. Otherwise, vertex w is said to be **loose**. So, K is an irreducible triangulation if and only if every vertex in K is a trapped one.

Our Approach

Our Approach

Lemma 1 (Schipper, 1991)
Once u becomes trapped in K, it remains trapped.

Our Approach

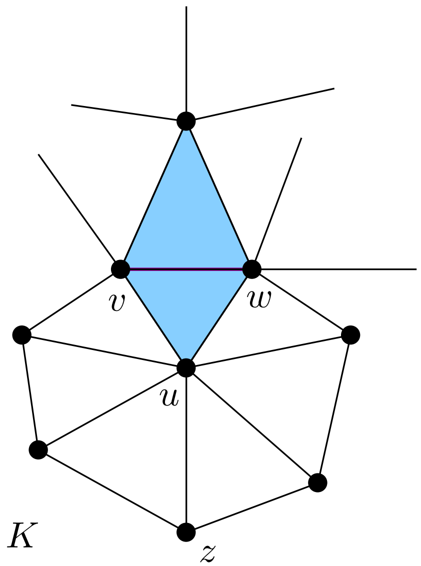
Lemma 1 (Schipper, 1991)
Once u becomes trapped in K, it remains trapped.

So, u will be in the irreducible triangulation, $\mathcal{T}'$.

Our Approach

Lemma 1 (Schipper, 1991)
Once u becomes trapped in K, it remains trapped.

So, u will be in the irreducible triangulation, $\mathcal{T}'$.



Our Approach

Let K be the current triangulation (i.e., $K = \mathcal{T}$ initially).

Choose a vertex u from K.

Contract all edges of the form $[u, v]$ until u becomes *trapped*.

Our Approach

Let K be the current triangulation (i.e., $K = \mathcal{T}$ initially).

Choose a vertex u from K.

Contract all edges of the form $[u, v]$ until u becomes *trapped*.

Pick another vertex u from the current triangulation, K.

Our Approach

Our Approach

Time complexity is dictated by two factors:

(1) The cost of testing an edge against the link condition, and

Our Approach

Time complexity is dictated by two factors:

(1) The cost of testing an edge against the link condition, and

The cost for testing *all* edges *during the processing of* u is

$$\Theta(d_u) + \sum_{v \in \mathcal{A}_u} \Theta(d_v)\,,$$

where $\mathcal{A}_u$ is the set of vertices that are or become adjacent to u.

Our Approach

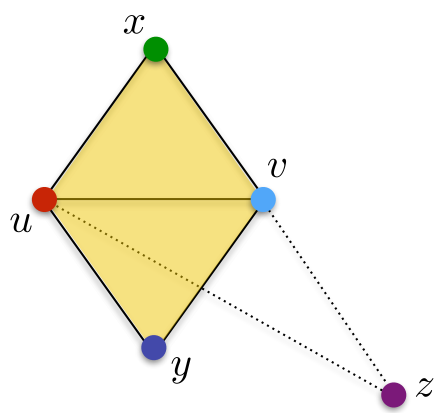
Time complexity is dictated by two factors:

(1) The cost of testing an edge against the link condition, and

Our Approach

Time complexity is dictated by two factors:

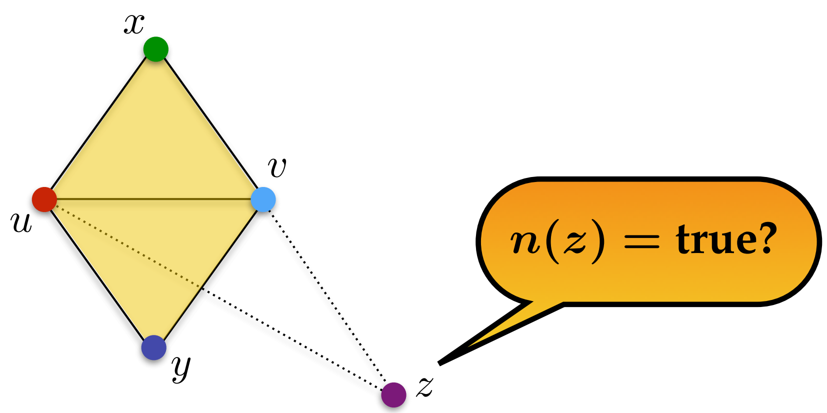
(1) The cost of testing an edge against the link condition, and



Our Approach

Time complexity is dictated by two factors:

(1) The cost of testing an edge against the link condition, and



Our Approach

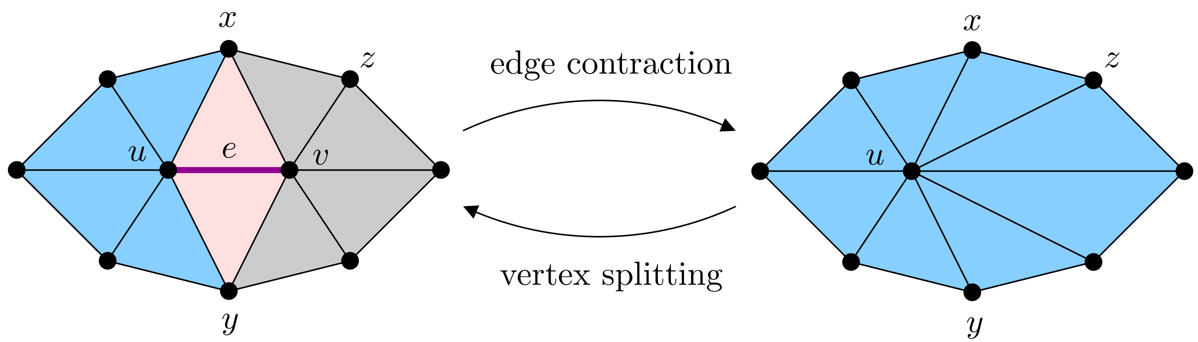
Time complexity is dictated by two factors:

(1) The cost of testing an edge against the link condition, and

Our Approach

Time complexity is dictated by two factors:

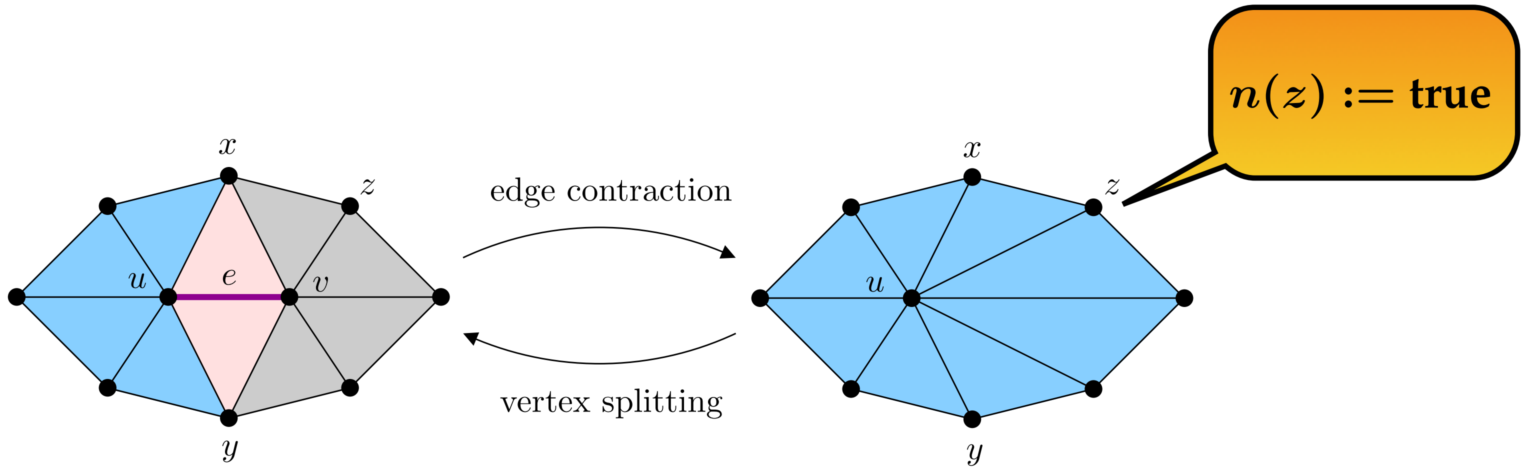
(1) The cost of testing an edge against the link condition, and



Our Approach

Time complexity is dictated by two factors:

(1) The cost of testing an edge against the link condition, and



Our Approach

Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

During the processing of u, edge $[u, v]$ in K is tested against the link condition exactly once if v has not been selected by the algorithm prior to u; else it is not tested.

Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

During the processing of u*, edge* $[u, v]$ *in* K *is tested against the link condition exactly once if* v *has not been selected by the algorithm prior to* u*; else it is not tested.*

Why?

Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

We devised an efficient mechanism to determine when a previously non-contractible, tested edge incident on u becomes contractible (if it does).

Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

We devised an efficient mechanism to determine when a previously non-contractible, tested edge incident on u becomes contractible (if it does).

Keep a counter for the number of critical cycles every edge $[u, v]$ belongs to.

Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

Lemma 2
Let K be a surface triangulation, and let v be any vertex of degree 3 in K. If K is (isomorphic to) $\mathcal{T}_4$, then no edge of K is a contractible edge. Otherwise, each of the 3 edges of K incident on v is a contractible edge in K.

Our Approach

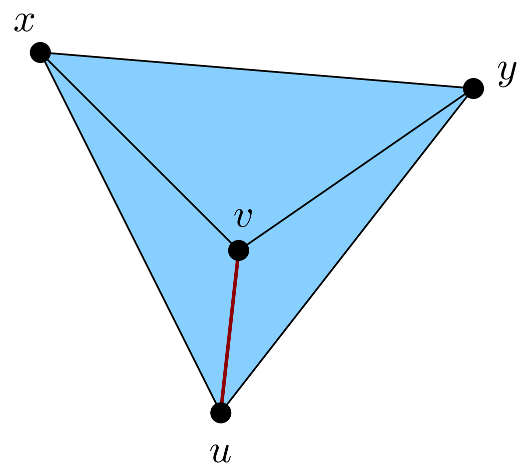
Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.



Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

Lemma 3
Let K be a surface triangulation, and let f be a contractible edge of K. If a non-contractible edge e of K becomes contractible in $K - f$, then f must be incident on a degree-3 vertex v of K and e must belong to $\mathrm{lk}(v, K)$. Moreover, e belongs to a single critical cycle in K, which consists of the edges in $\mathrm{lk}(v, K)$, and this cycle becomes non-critical in $K - f$.

Our Approach

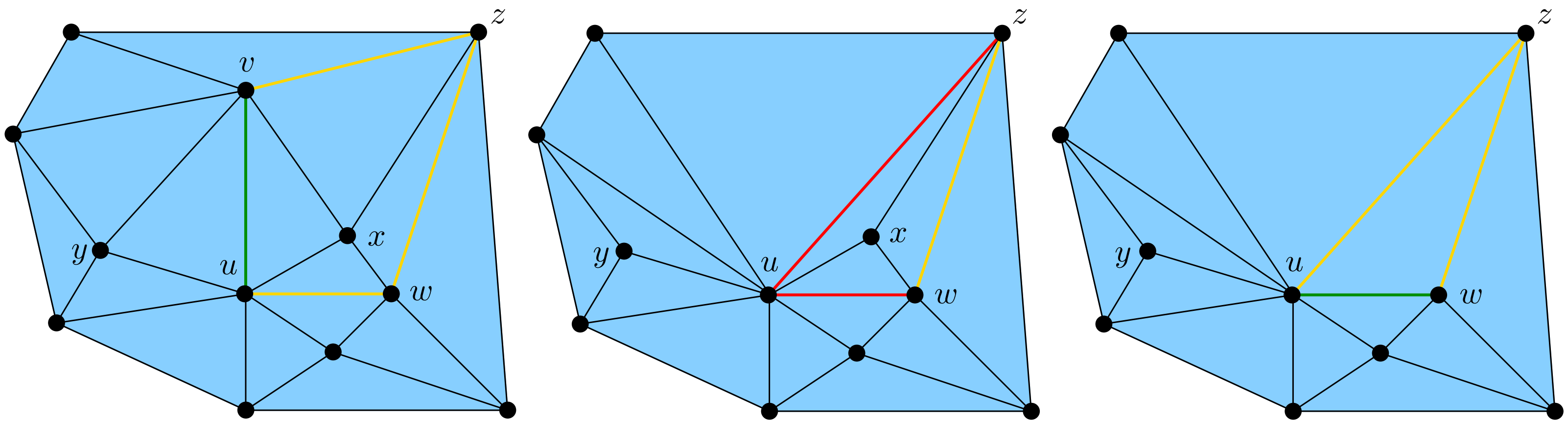
Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.



K $K - uv$ $(K - uv) - ux$

Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

Our Approach

Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

Lemma 4
Let K be a surface triangulation, and let f be a contractible edge of K. If a contractible edge e of K becomes non-contractible in $K - f$, then the vertex v of f identified with the other vertex, u, of f in $K - f$ is such that

$$d_v > 3\,.$$

Our Approach

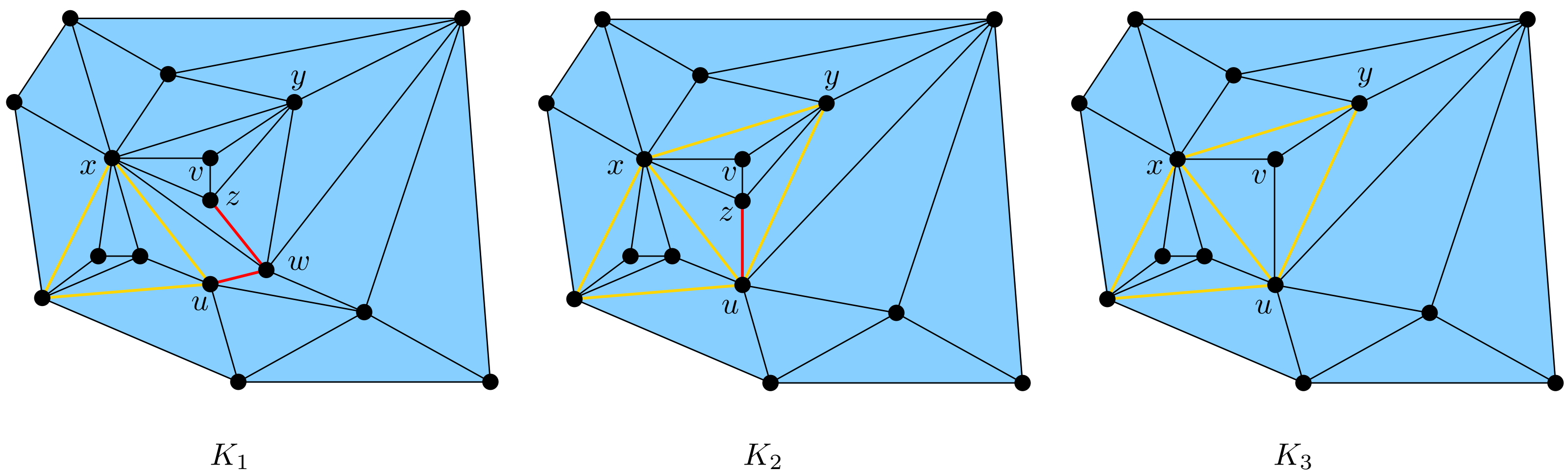
Time complexity is dictated by two factors:

(2) The number of link condition tests executed by the algorithm.

Our Approach

Time complexity is dictated by two factors:

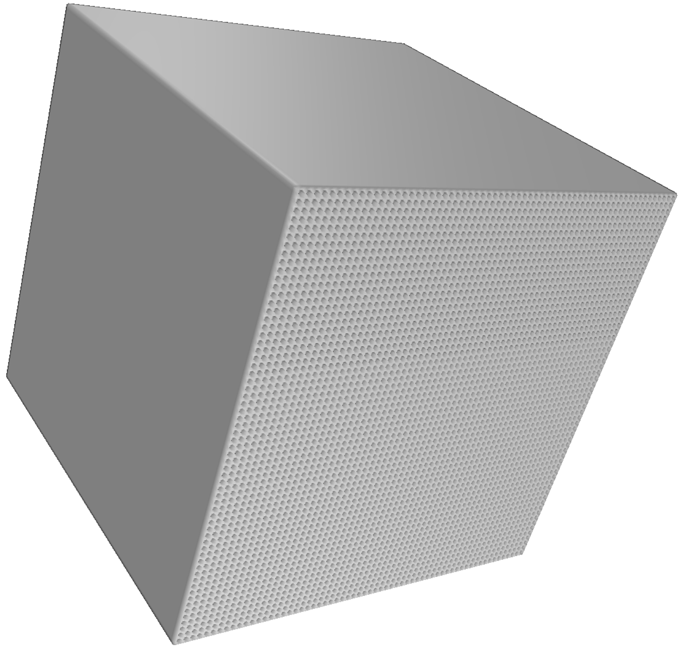
(2) The number of link condition tests executed by the algorithm.



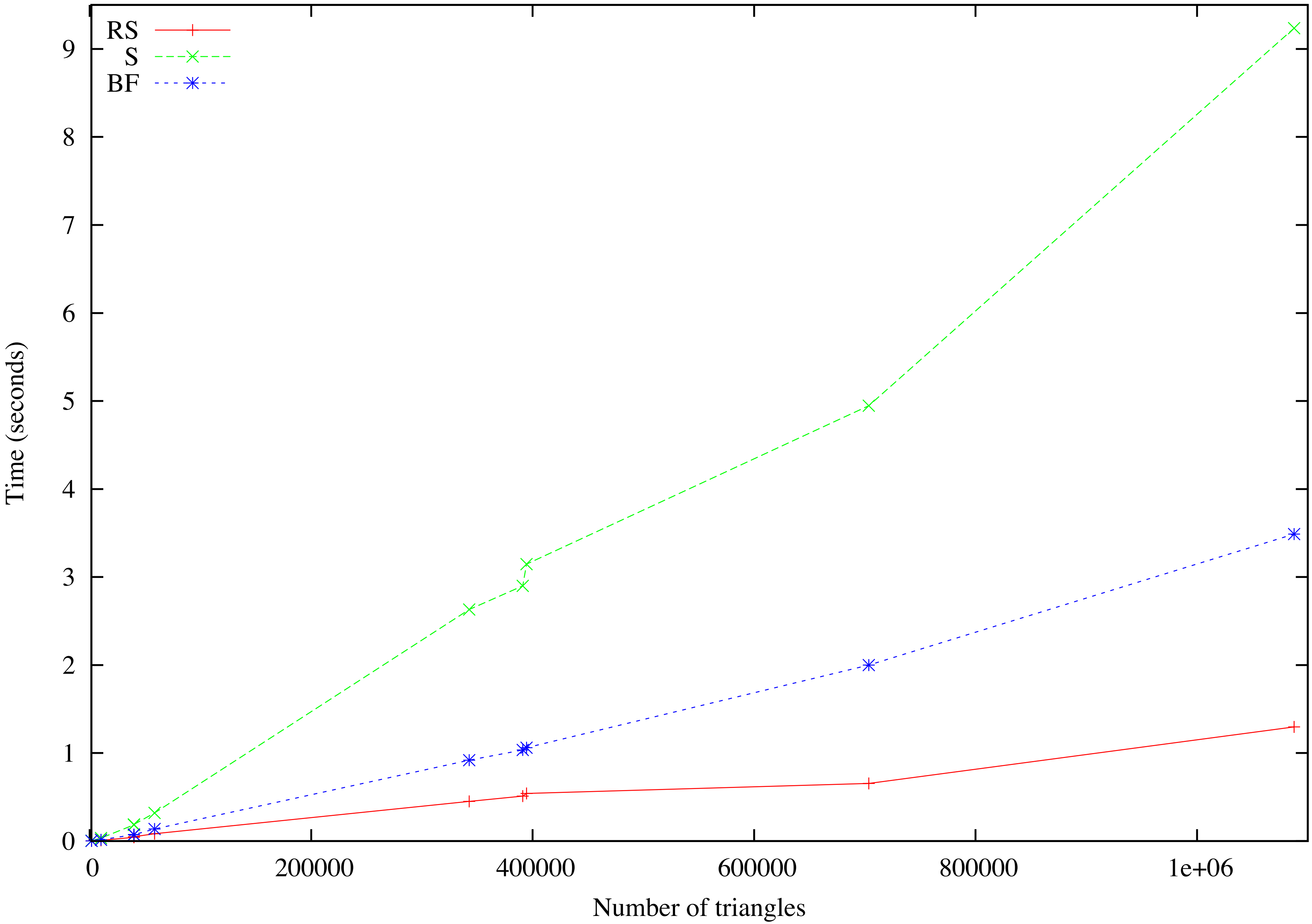
Experimental Results

Experimental Results

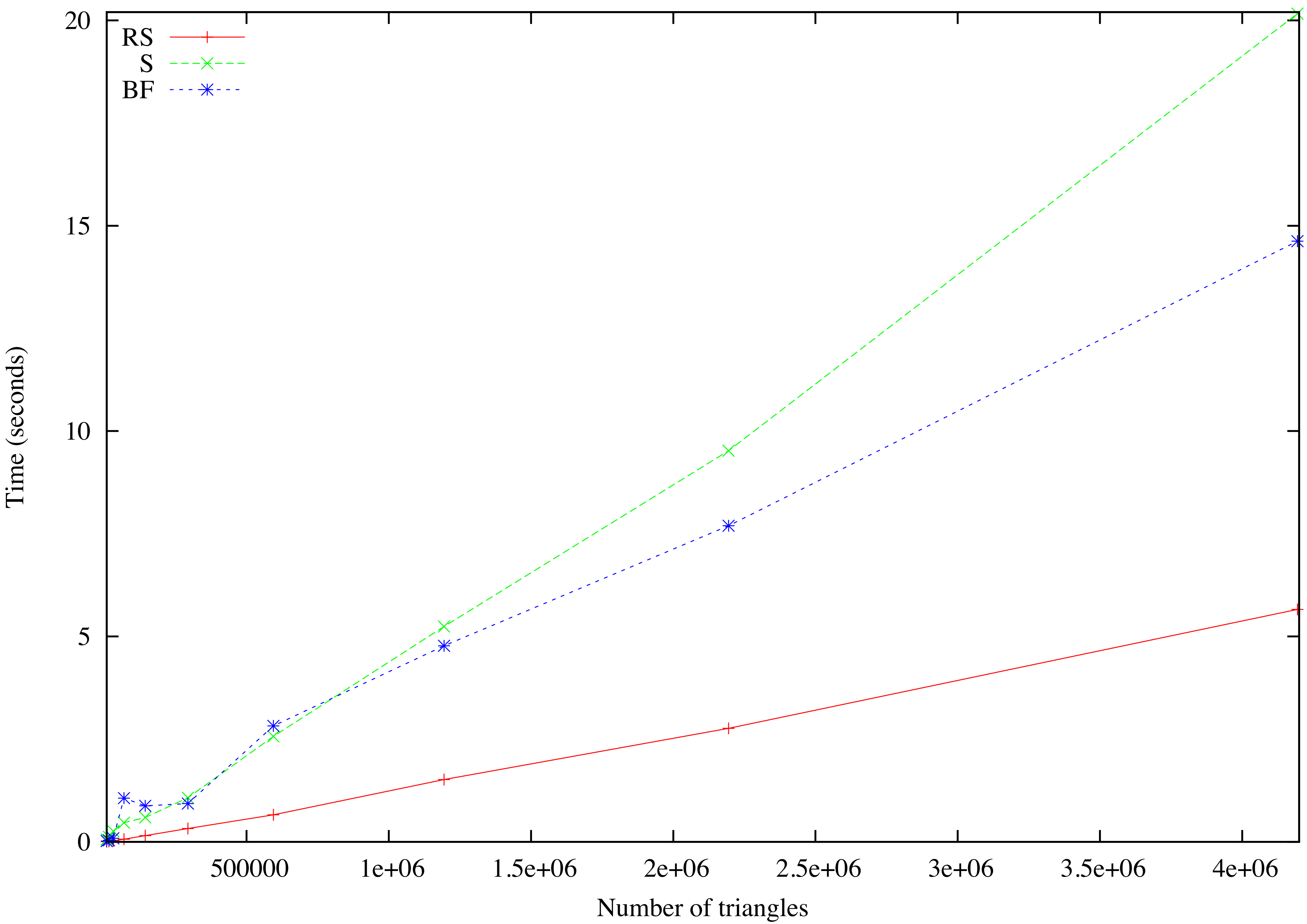
$g \ll n_f$

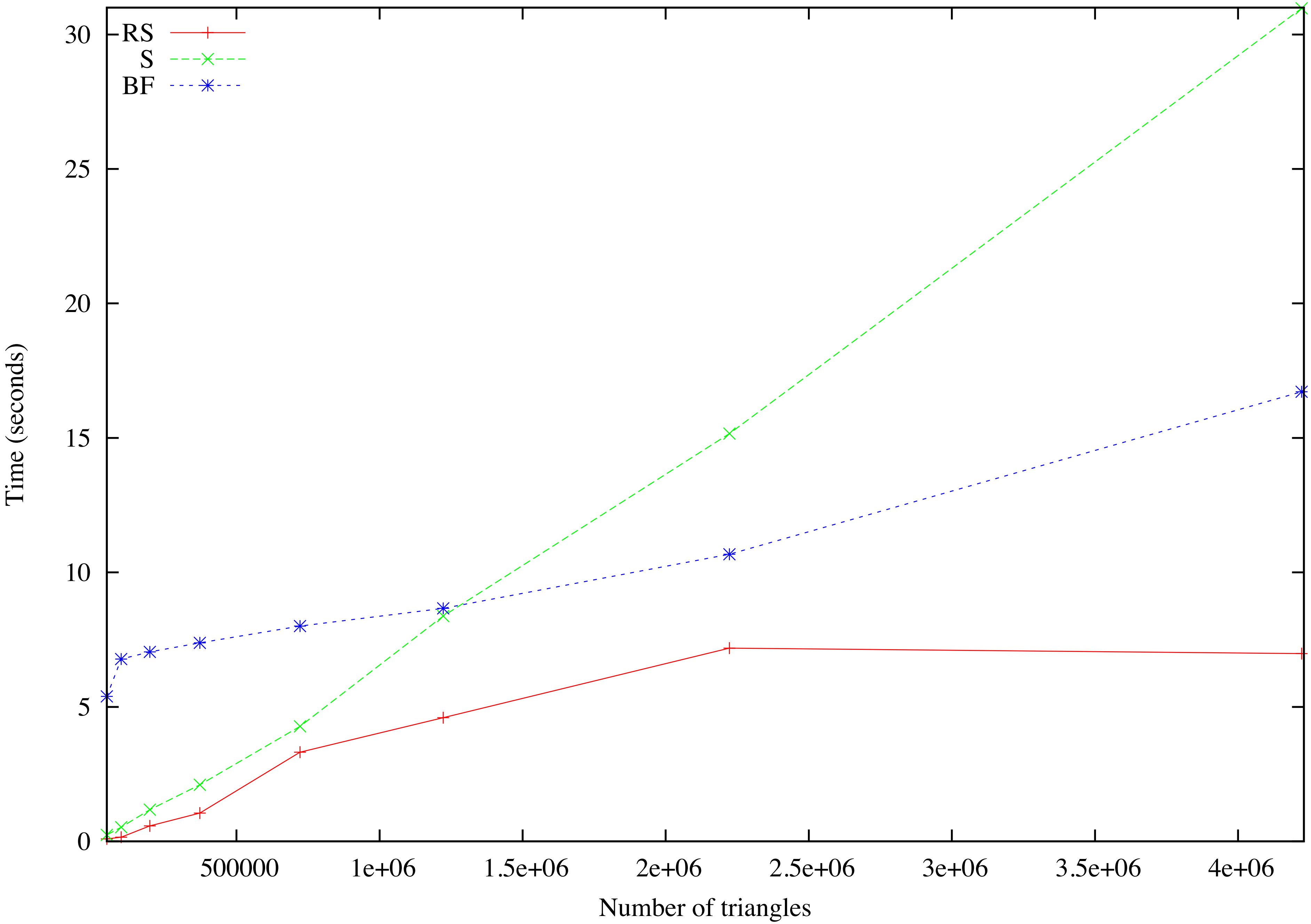


$g \approx 2 \cdot \sqrt{n_f}$

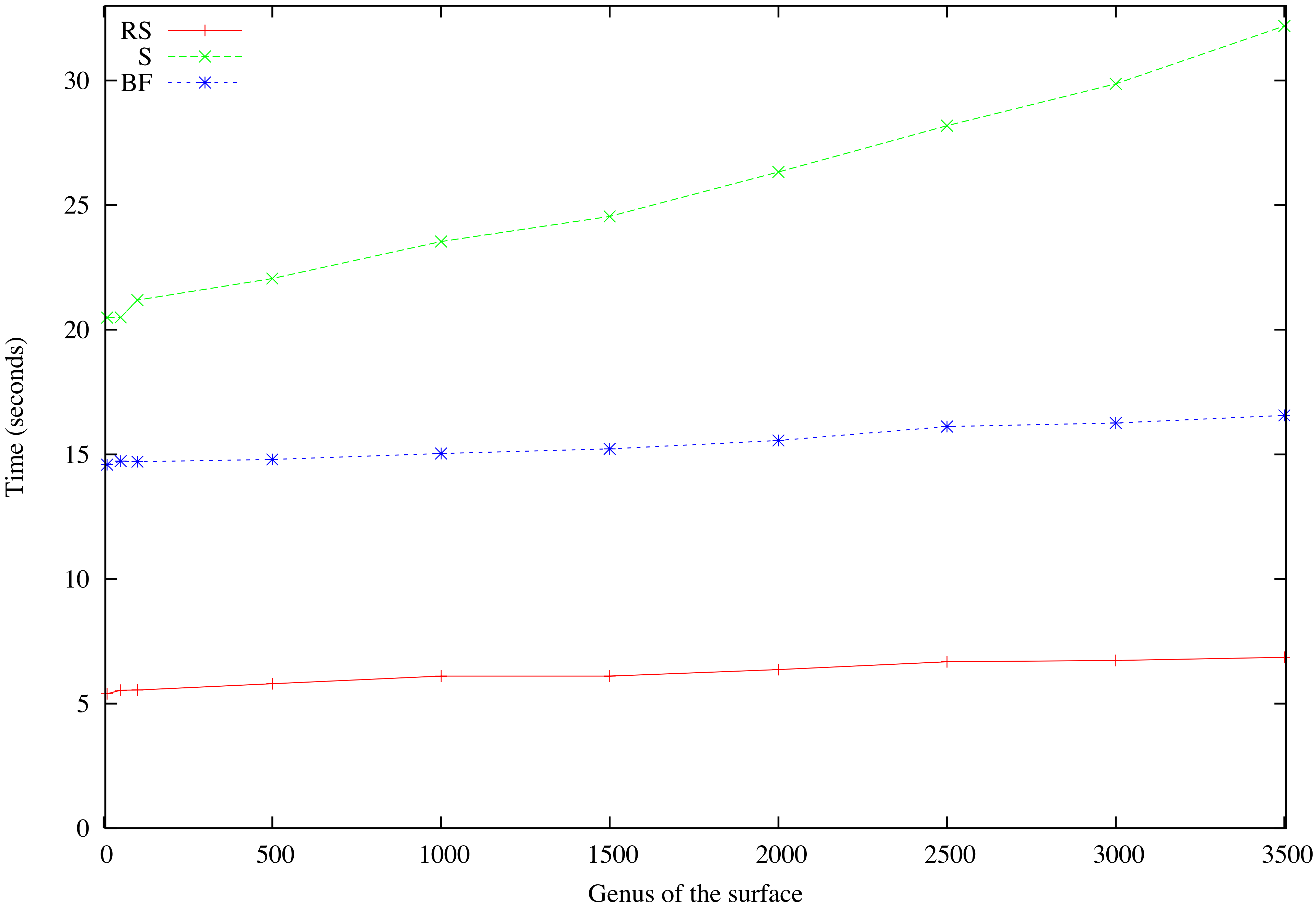


RS
S
BF
Time (seconds)
0
1
2
3
4
5
6
7
8
9
0
200000
400000
600000
800000
1e+06
Number of triangles


RS
S
BF
Time (seconds)
0
5
10
15
20
500000
1e+06
1.5e+06
2e+06
2.5e+06
3e+06
3.5e+06
4e+06
Number of triangles



RS
S
BF
Time (seconds)
Number of triangles
0
5
10
15
20
25
30
500000
1e+06
1.5e+06
2e+06
2.5e+06
3e+06
3.5e+06
4e+06


RS
S
BF
Time (seconds)
0
5
10
15
20
25
30
0
500
1000
1500
2000
2500
3000
3500
Genus of the surface

More Details

More Details

Suneeta Ramaswami and Marcelo Siqueira

A fast algorithm for computing irreducible triangulations of closed surfaces in $\mathbb{E}^d$

CoRR, arXiv:1409.6015, 2014

Code:
`http://www.mat.ufrn.br/~mfsiqueira/Marcelo_Siqueiras_Web_Spot/Software.html`

More Details

Suneeta Ramaswami and Marcelo Siqueira

A fast algorithm for computing irreducible triangulations of closed surfaces in $\mathbb{E}^d$

CoRR, arXiv:1409.6015, 2014

Code:
`http://www.mat.ufrn.br/~mfsiqueira/Marcelo_Siqueiras_Web_Spot/Software.html`

Questions?