Topology Reconstruction for B-Rep Modeling from 3D Mesh in Reverse Engineering Applications

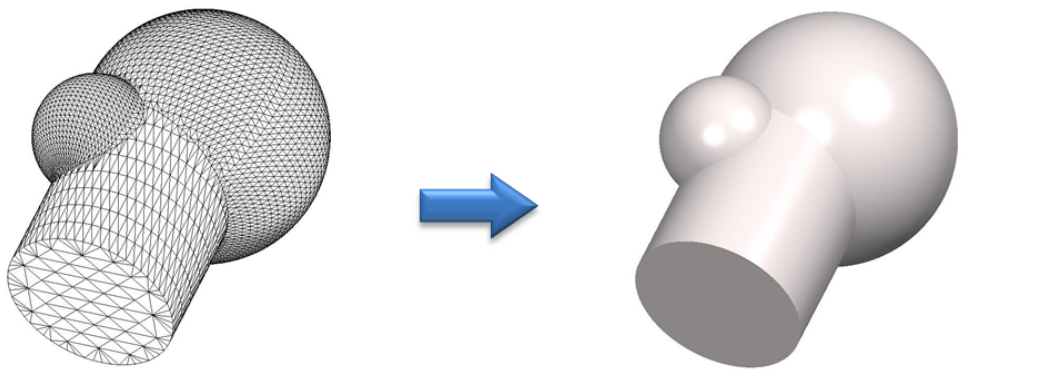
Roseline Bénière[1,2]
G. Subsol[1], G. Gesquière[3], F. Le Breton[2] and W. Puech[1]

LIRMM, University of Montpellier 2/CNRS, France (1)
C4W, Montpellier, France (2)
Aix Marseille University, LSIS, France(3)

January 25, 2012

Objective

- Reverse engineering:
  discretized mesh $\Rightarrow$ continuous representation ,

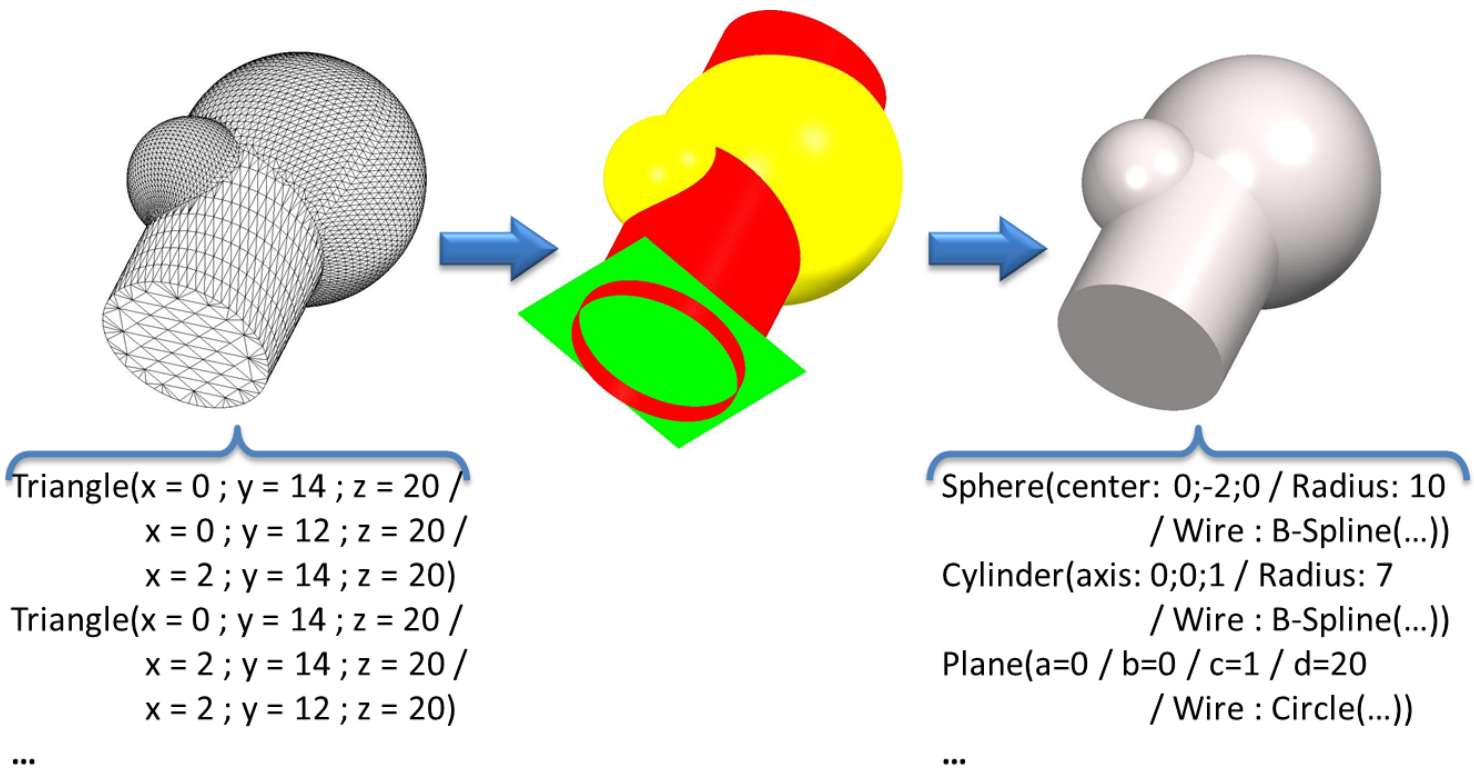


Triangle(x = 0 ; y = 14 ; z = 20 /
x = 0 ; y = 12 ; z = 20 /
x = 2 ; y = 14 ; z = 20)
Triangle(x = 0 ; y = 14 ; z = 20 /
x = 2 ; y = 14 ; z = 20 /
x = 2 ; y = 12 ; z = 20)
...

Sphere(center: 0;-2;0 / Radius: 10
/ Wire : B-Spline(...))
Cylinder(axis: 0;0;1 / Radius: 7
/ Wire : B-Spline(...))
Plane(a=0 / b=0 / c=1 / d=20
/ Wire : Circle(...))
...

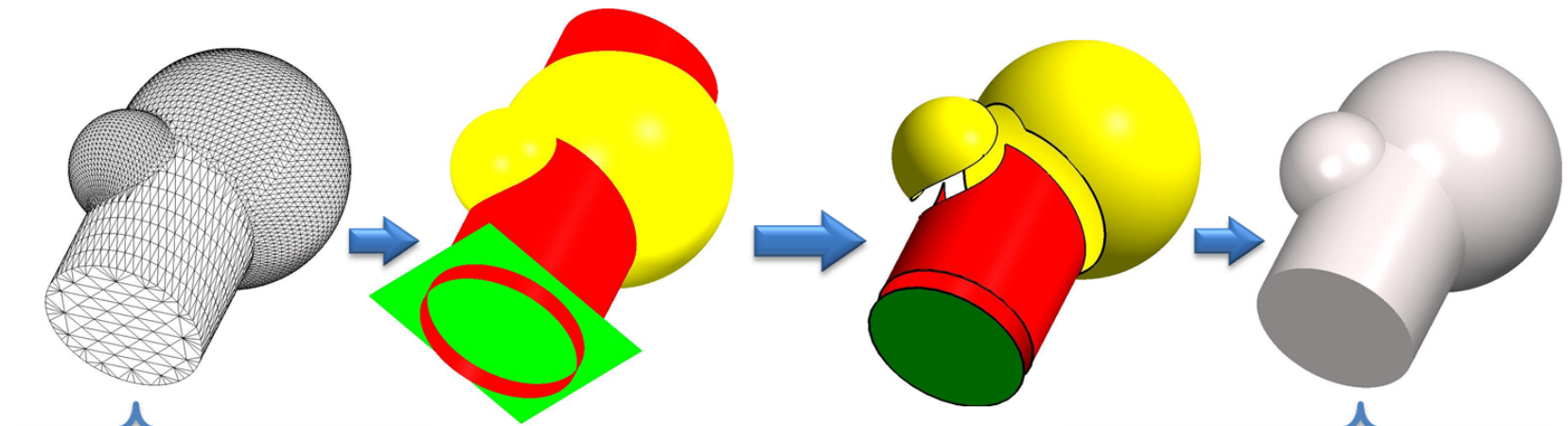
Objective

- Reverse engineering:
  discretized mesh $\Rightarrow$ continuous representation ,
- First problem: detect the primitives $\Rightarrow$ many papers,



Objective

- Reverse engineering:
  discretized mesh $\Rightarrow$ continuous representation ,
- First problem: detect the primitives $\Rightarrow$ many papers,
  But
- Second problem: topology reconstruction $\Rightarrow$ few papers.

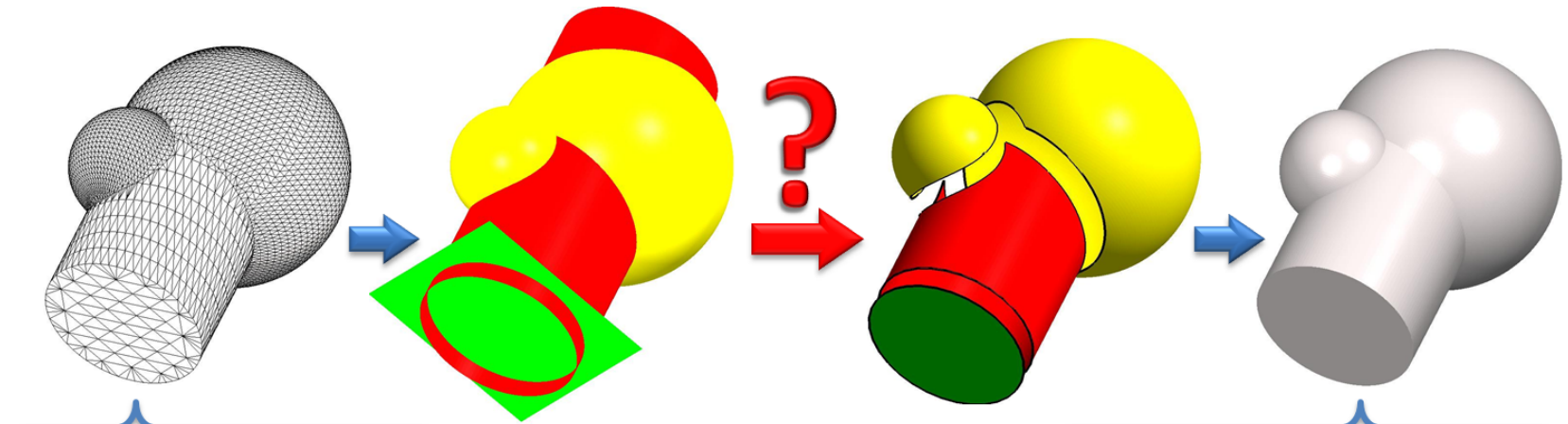


Triangle(x = 0 ; y = 14 ; z = 20 /
x = 0 ; y = 12 ; z = 20 /
x = 2 ; y = 14 ; z = 20)
Triangle(x = 0 ; y = 14 ; z = 20 /
x = 2 ; y = 14 ; z = 20 /
x = 2 ; y = 12 ; z = 20)
...

Sphere(center: 0;-2;0 / Radius: 10
/ Wire : B-Spline(...))
Cylinder(axis: 0;0;1 / Radius: 7
/ Wire : B-Spline(...))
Plane(a=0 / b=0 / c=1 / d=20
/ Wire : Circle(...))
...

Objective

- Reverse engineering:
  discretized mesh $\Rightarrow$ continuous representation ,
- First problem: detect the primitives $\Rightarrow$ many papers,
  But
- Second problem: topology reconstruction $\Rightarrow$ few papers.
  This paper deals with this second problem



Triangle(x = 0 ; y = 14 ; z = 20 /
x = 0 ; y = 12 ; z = 20 /
x = 2 ; y = 14 ; z = 20)
Triangle(x = 0 ; y = 14 ; z = 20 /
x = 2 ; y = 14 ; z = 20 /
x = 2 ; y = 12 ; z = 20)
...

Sphere(center: 0;-2;0 / Radius: 10
/ Wire : B-Spline(...))
Cylinder(axis: 0;0;1 / Radius: 7
/ Wire : B-Spline(...))
Plane(a=0 / b=0 / c=1 / d=20
/ Wire : Circle(...))
...

Previous work

Comprehensive process

Comprehensive reverse engineering process are proposed in few papers.

Algorithms for reverse engineering boundary representation models, *P. Benkö et al., Computer-Aided Design, 33(11):839–851, 2001*

Previous work

Comprehensive process

Comprehensive reverse engineering process are proposed in few papers.

Algorithms for reverse engineering boundary representation models, *P. Benkö et al., Computer-Aided Design, 33(11):839–851, 2001*

Neighborhood

The neighborhood definition can be used to improve the remeshing, computing the real intersections.

Improving surface meshing from discrete data by feature recognition, *C. Chappuis et al., Engineering with Computer (20), 202–209, 2004*

Previous work

Comprehensive process

Comprehensive reverse engineering process are proposed in few papers.

Algorithms for reverse engineering boundary representation models, *P. Benkö et al., Computer-Aided Design, 33(11):839–851, 2001*

Neighborhood

The neighborhood definition can be used to improve the remeshing, computing the real intersections.

Improving surface meshing from discrete data by feature recognition, *C. Chappuis et al., Engineering with Computer (20), 202–209, 2004*

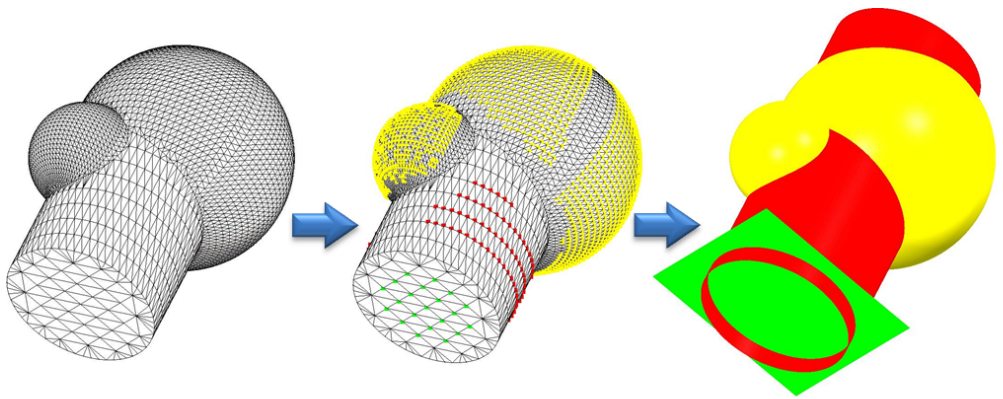
Edges and wires

The wires are built assembling curve intersection parts. This problem is related to Boundary Evaluation in CSG.

Incremental boundary evaluation using inference of edge classification, *J.R. Miller, IEEE Computer Graphics & Applications* ***0272****(17), 71–78, 1993*

Neighborhood determination

In [Bénière *et al.*,2011] $\Rightarrow$ extraction of a set of primitives based on point areas defined by curvature informations.

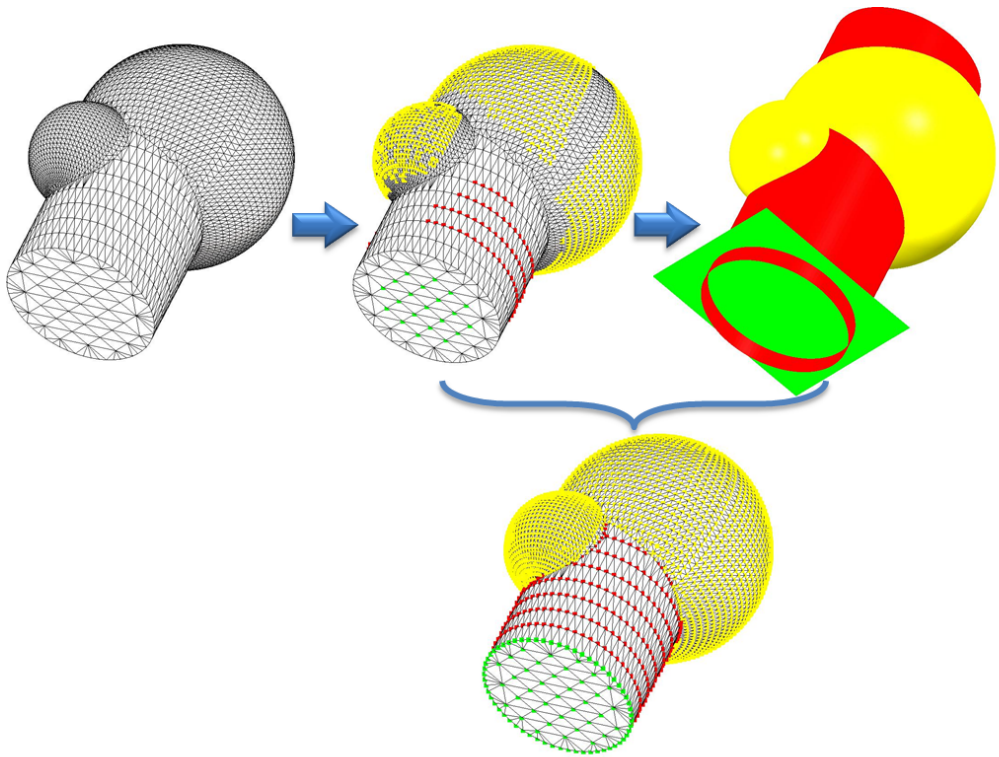


***Recovering primitives in 3D CAD meshes**, R. Bénière et al., SPIE 2011*

Neighborhood determination

In [Bénière *et al.*,2011] $\Rightarrow$ extraction of a set of primitives based on point areas defined by curvature informations. Then now:

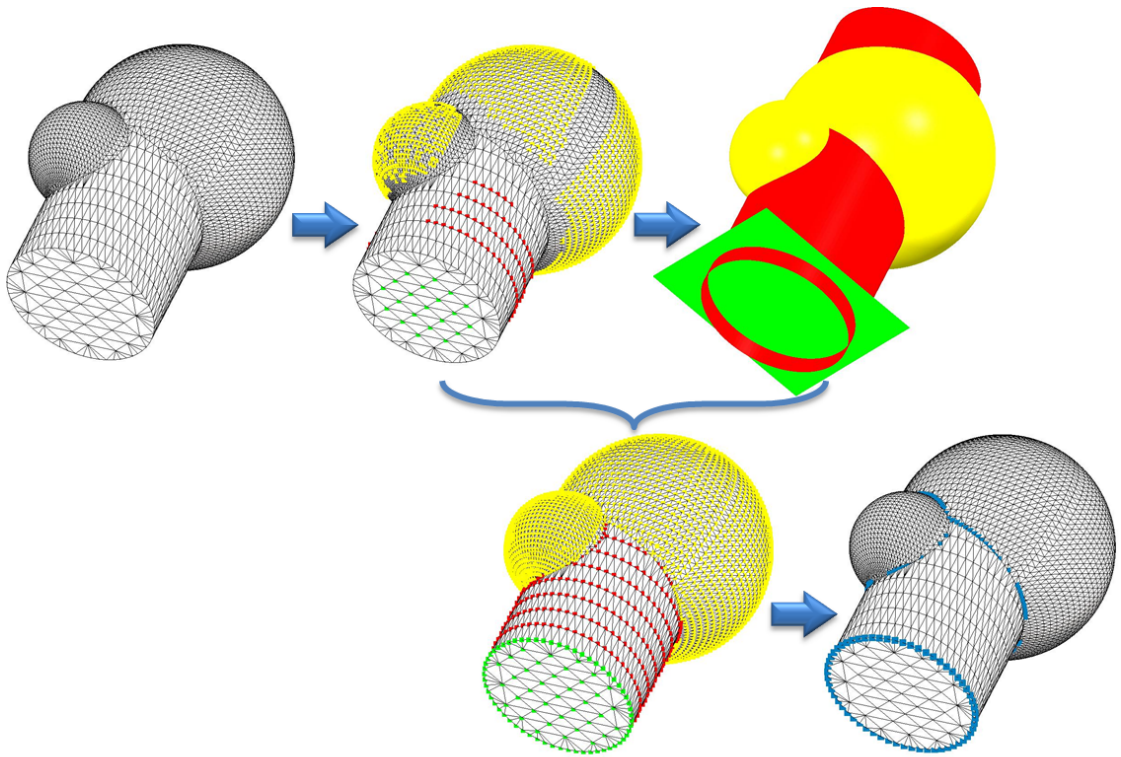
- Primitives + Point Areas $\Rightarrow$ Extended Areas



Neighborhood determination

In [Bénière *et al.*,2011] ⇒ extraction of a set of primitives based on point areas defined by curvature informations. Then now:

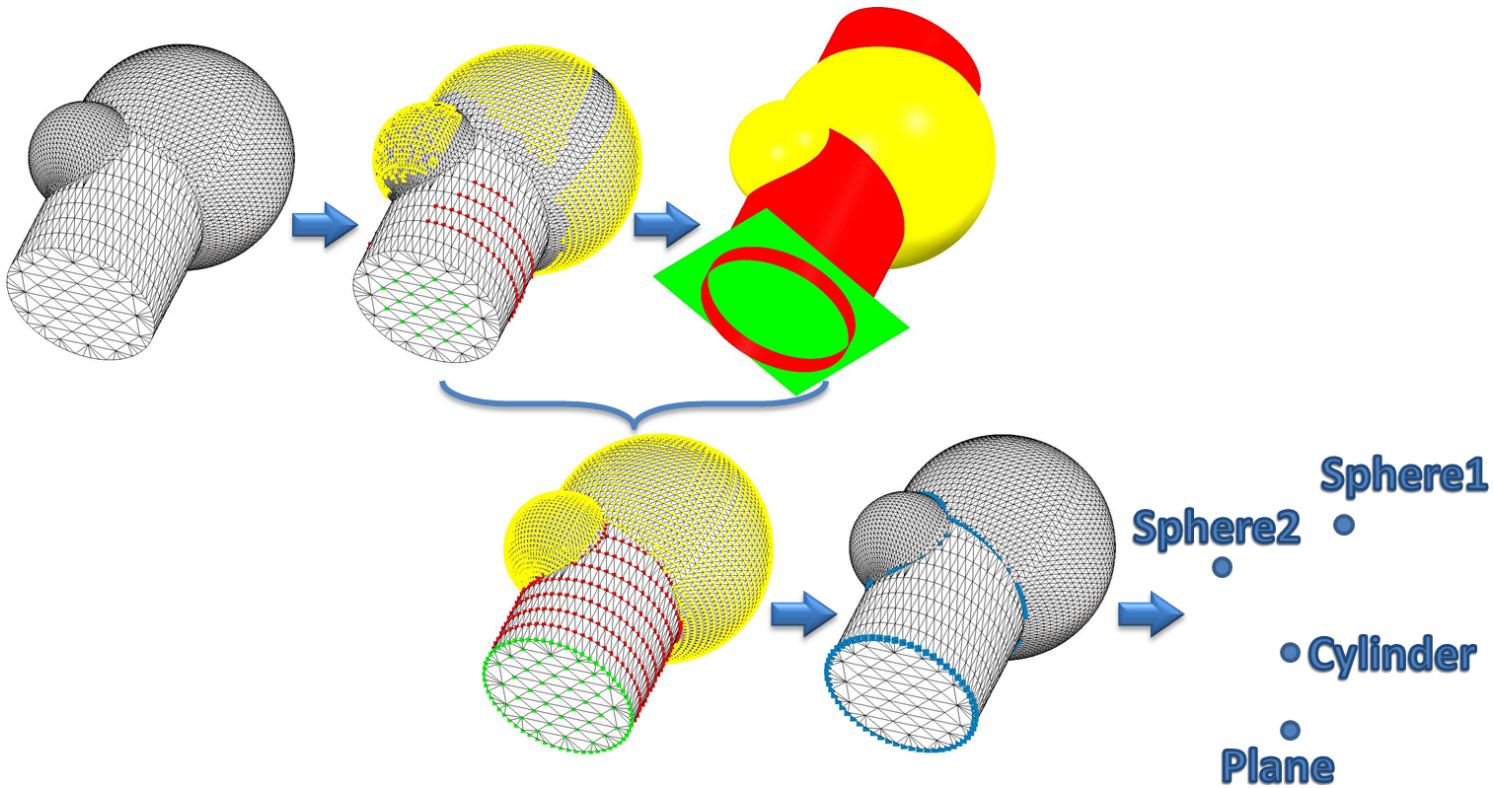
- Primitives + Point Areas ⇒ Extended Areas
- Extended Areas ⇒ Common Points



Neighborhood determination

In [Bénière *et al.*,2011] ⇒ extraction of a set of primitives based on point areas defined by curvature informations. Then now:

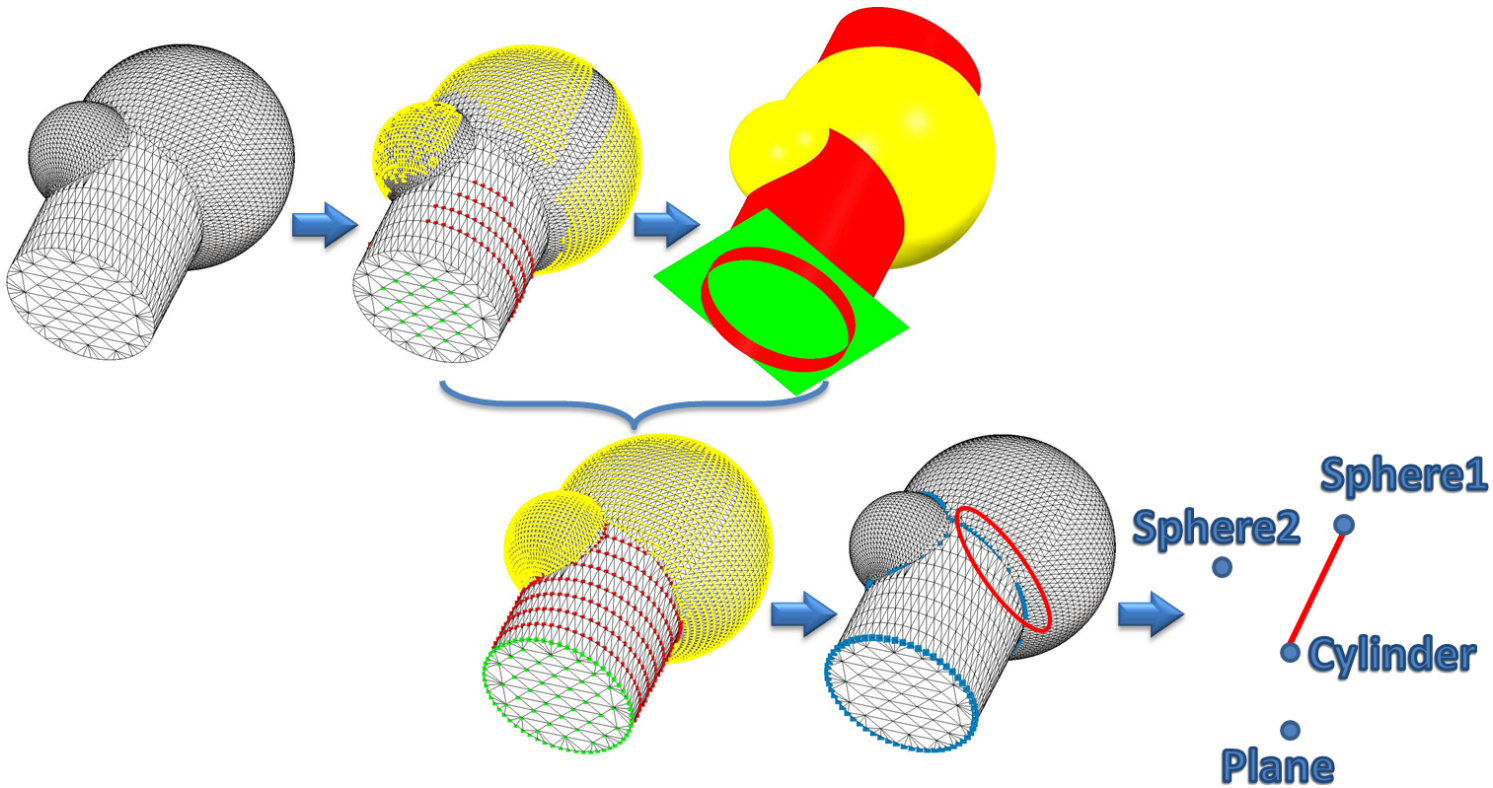
- Primitives + Point Areas ⇒ Extended Areas
- Extended Areas ⇒ Common Points
- Adjacency Graph ⇒ node = primitive



Neighborhood determination

In [Bénière *et al.*,2011] ⇒ extraction of a set of primitives based on point areas defined by curvature informations. Then now:

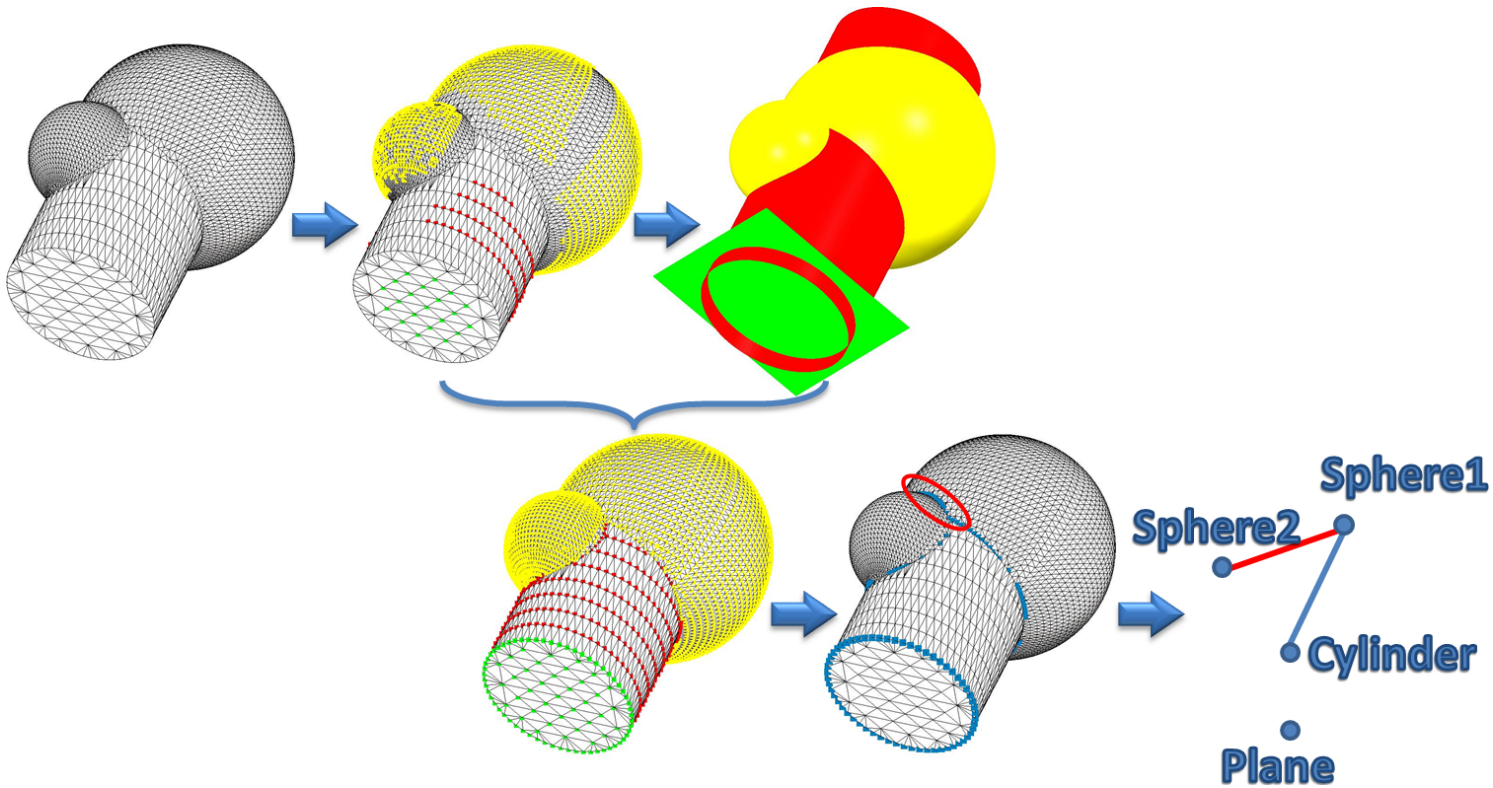
- Primitives + Point Areas ⇒ Extended Areas
- Extended Areas ⇒ Common Points
- Adjacency Graph ⇒ node = primitive
  ⇒edge = existence of common points



Neighborhood determination

In [Bénière *et al.*,2011] ⇒ extraction of a set of primitives based on point areas defined by curvature informations. Then now:

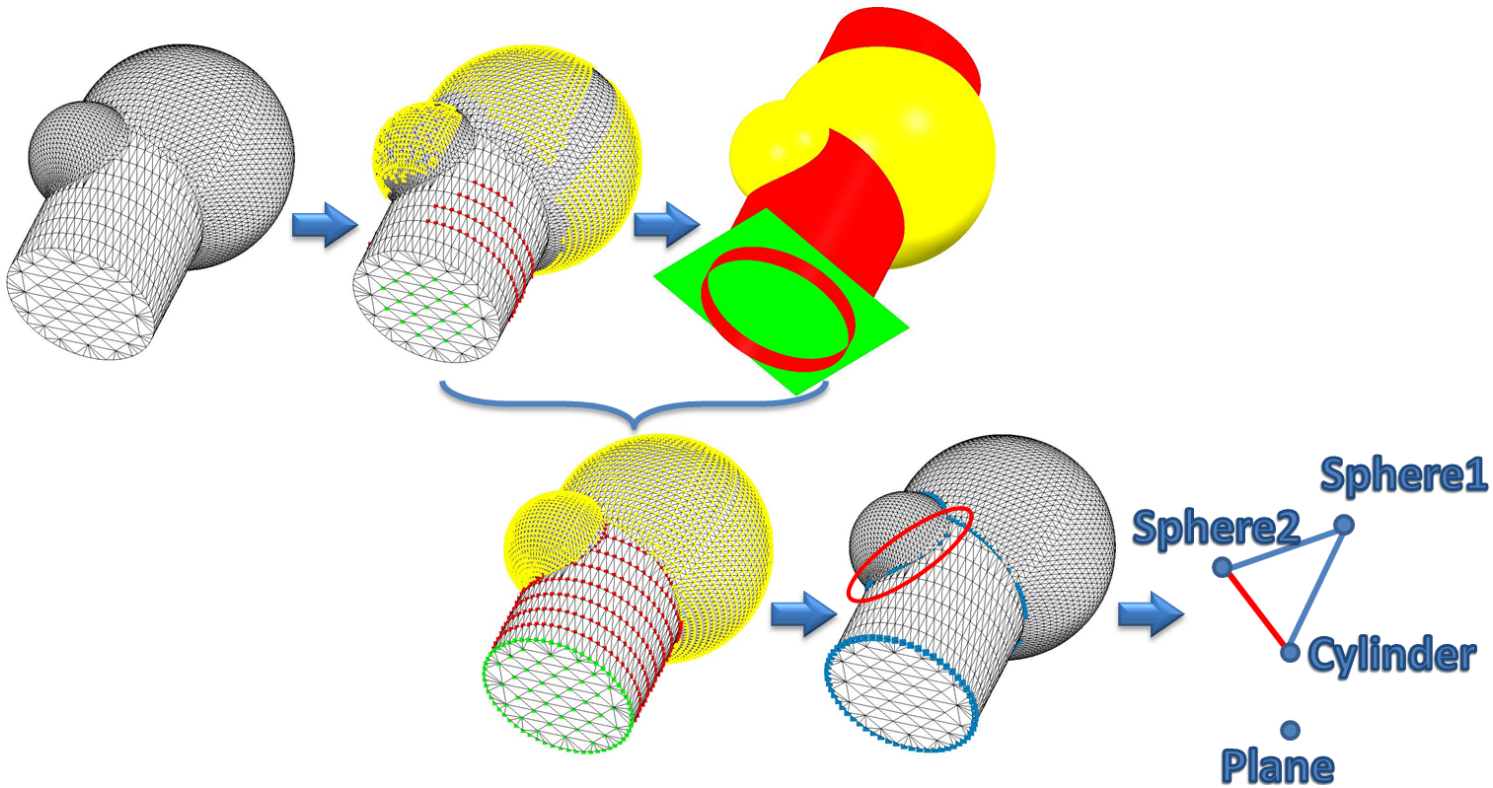
- Primitives + Point Areas ⇒ Extended Areas
- Extended Areas ⇒ Common Points
- Adjacency Graph ⇒ node = primitive
  ⇒edge = existence of common points



Neighborhood determination

In [Bénière *et al.*,2011] ⇒ extraction of a set of primitives based on point areas defined by curvature informations. Then now:

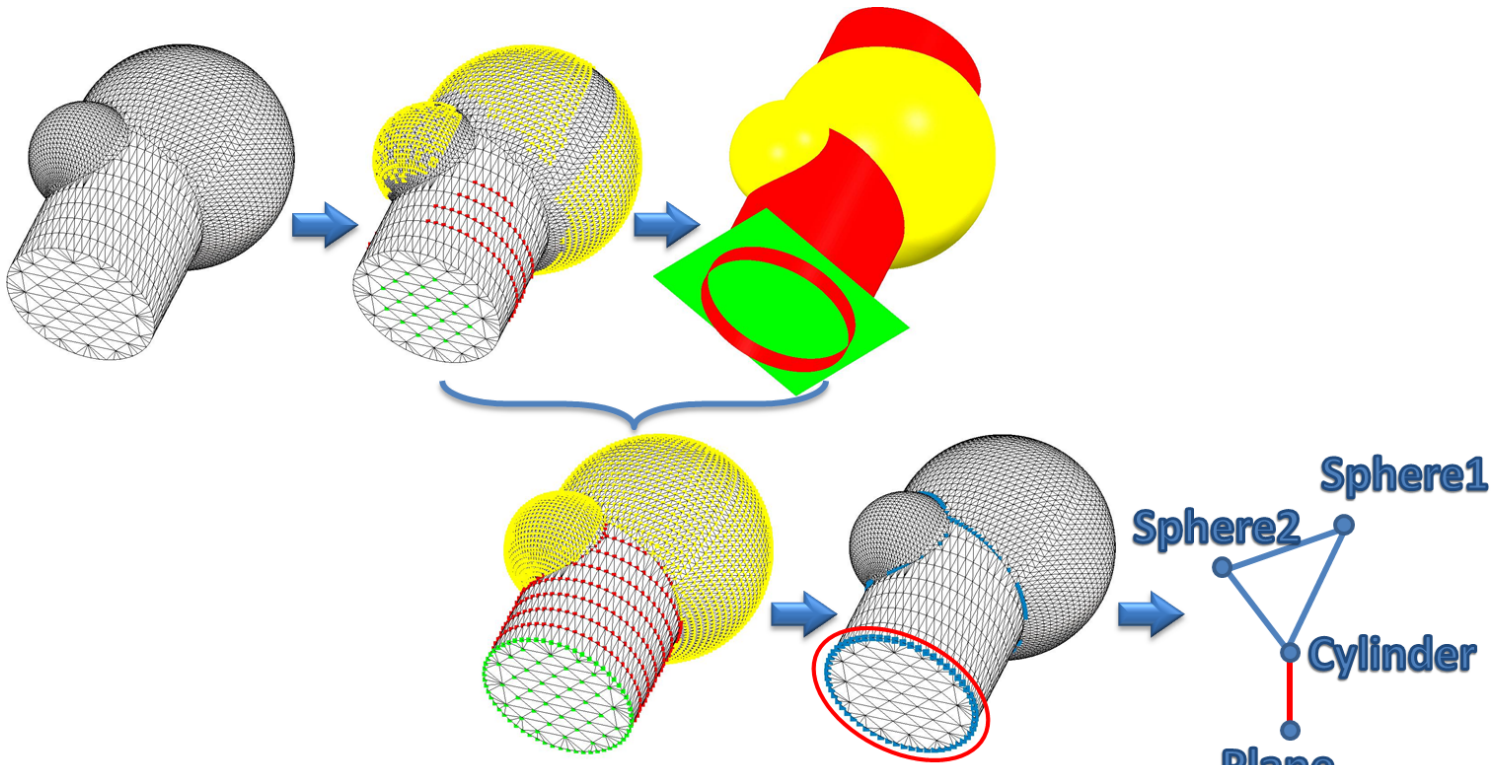
- Primitives + Point Areas ⇒ Extended Areas
- Extended Areas ⇒ Common Points
- Adjacency Graph ⇒ node = primitive
  ⇒edge = existence of common points



Neighborhood determination

In [Bénière *et al.*,2011] ⇒ extraction of a set of primitives based on point areas defined by curvature informations. Then now:

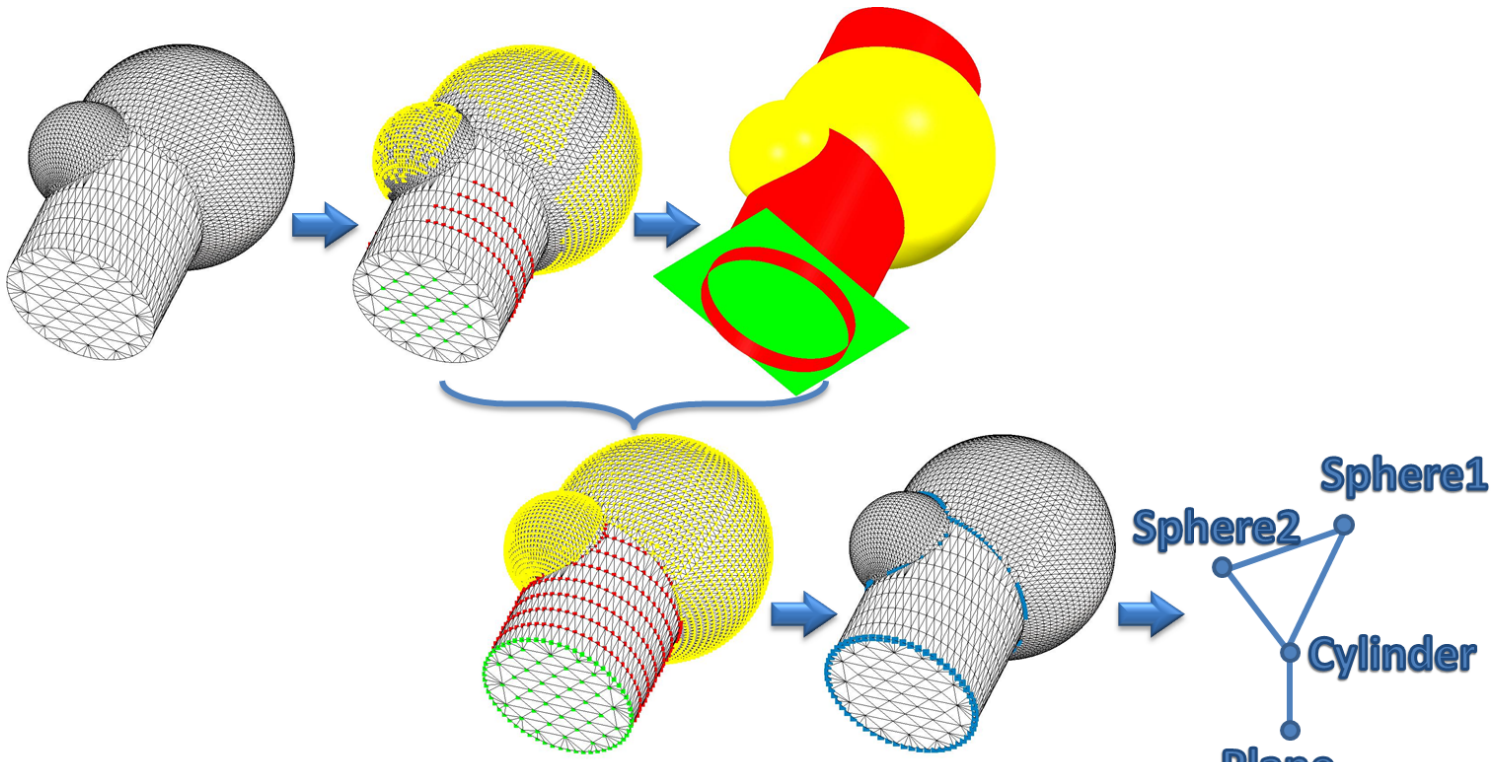
- Primitives + Point Areas ⇒ Extended Areas
- Extended Areas ⇒ Common Points
- Adjacency Graph ⇒ node = primitive
  ⇒edge = existence of common points



Neighborhood determination

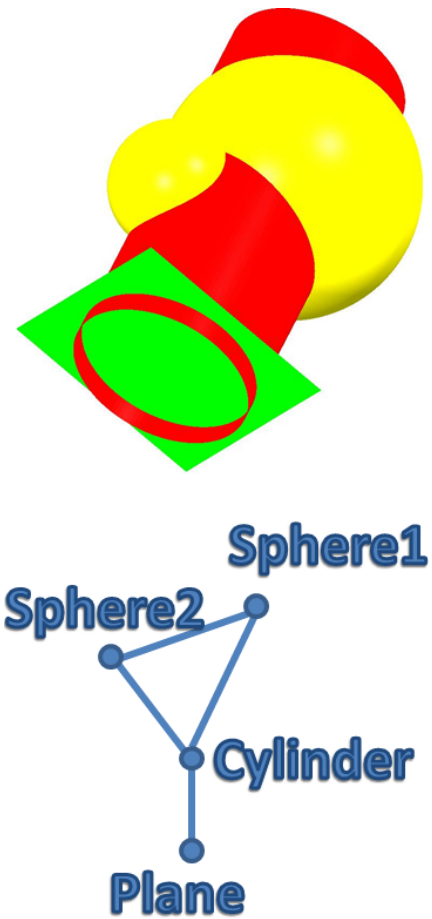
In [Bénière *et al.*,2011] ⇒ extraction of a set of primitives based on point areas defined by curvature informations. Then now:

- Primitives + Point Areas ⇒ Extended Areas
- Extended Areas ⇒ Common Points
- Adjacency Graph ⇒ node = primitive
  ⇒edge = existence of common points



Valid intersection computation

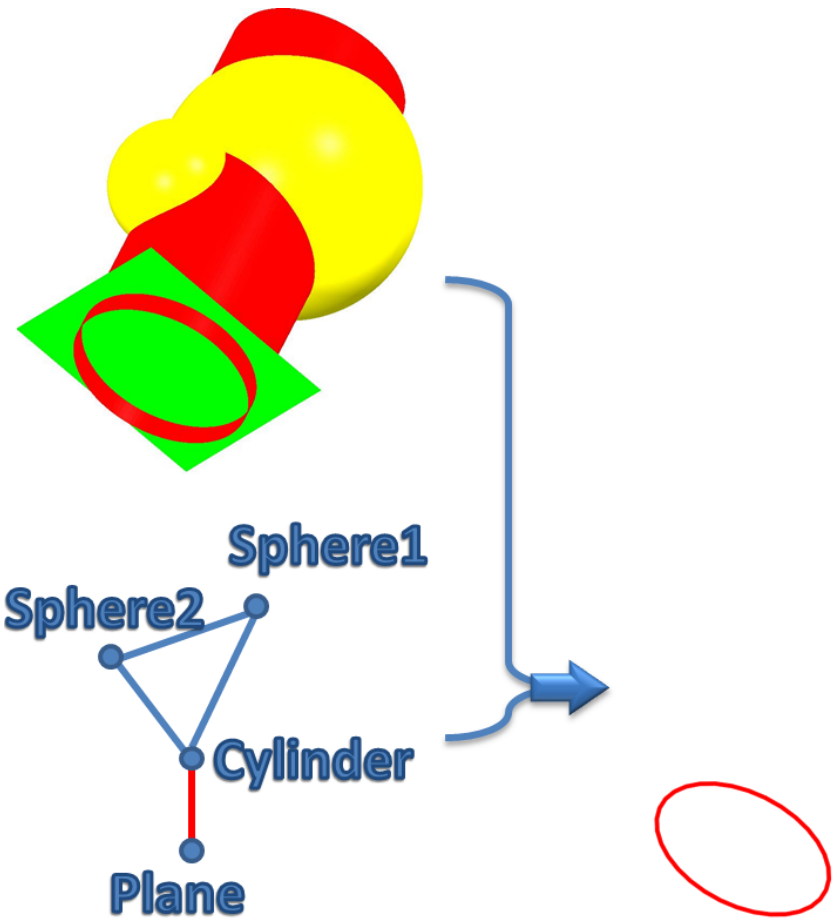
Using the Primitives and the Adjacency Graph:



Valid intersection computation

Using the Primitives and the Adjacency Graph:

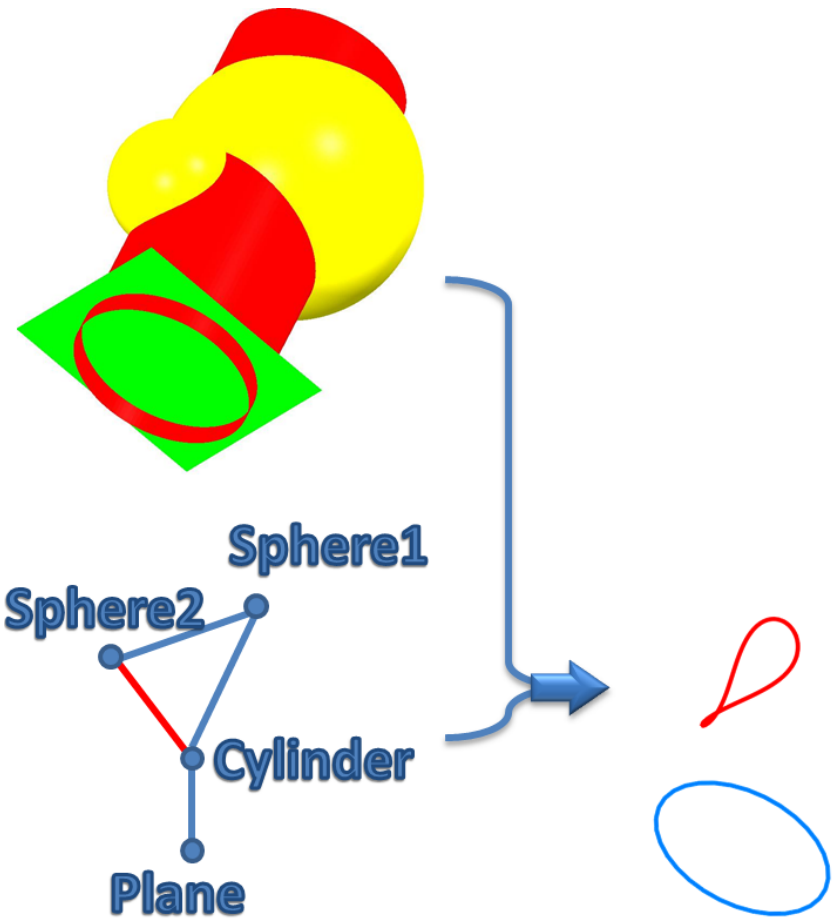
- For each Adjacency Graph edge $\Rightarrow$ intersection curves,



Valid intersection computation

Using the Primitives and the Adjacency Graph:

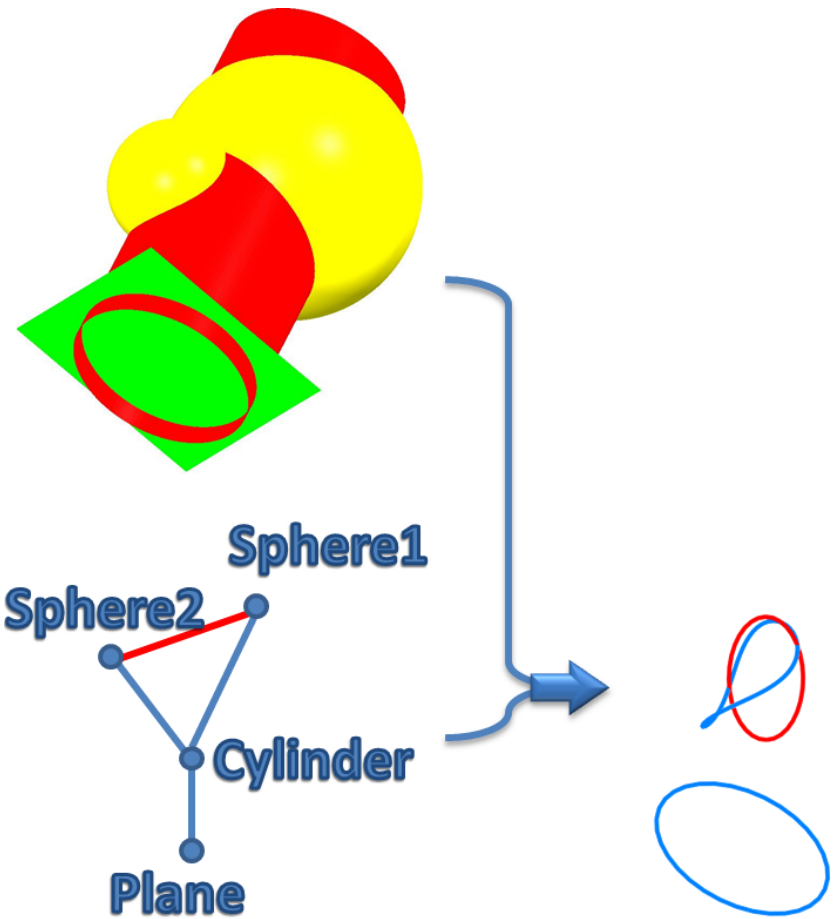
- For each Adjacency Graph edge $\Rightarrow$ intersection curves,



Valid intersection computation

Using the Primitives and the Adjacency Graph:

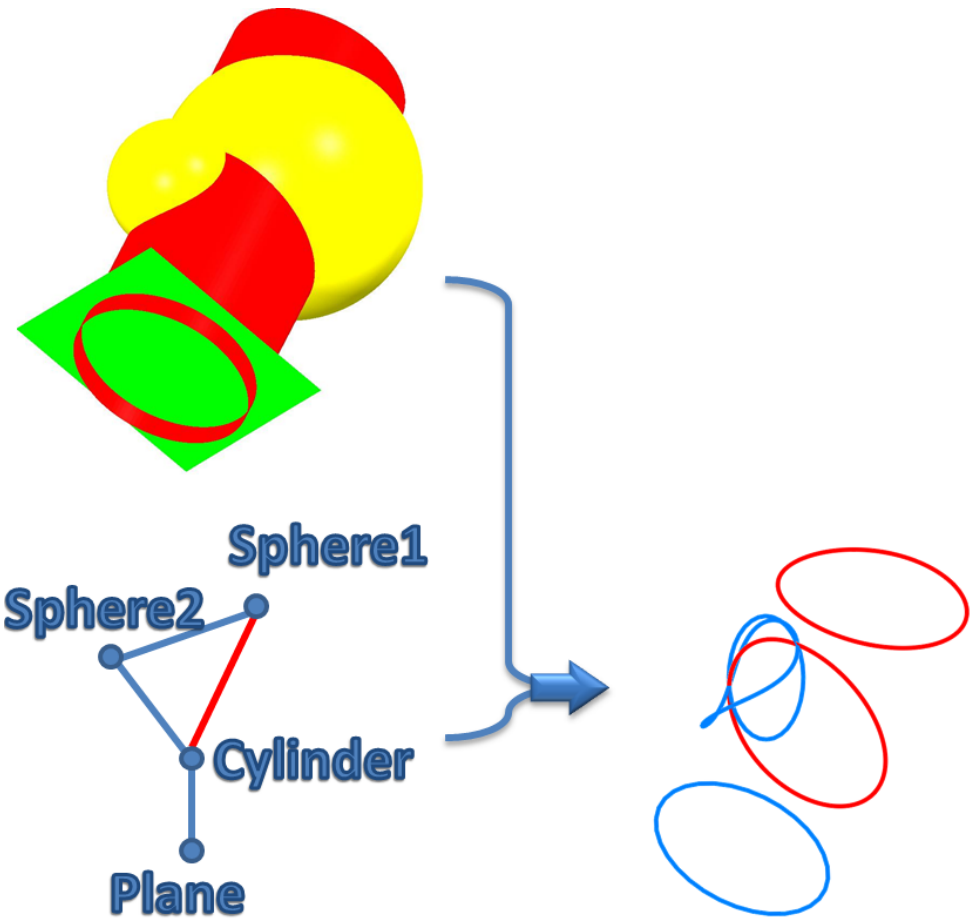
- For each Adjacency Graph edge $\Rightarrow$ intersection curves,



Valid intersection computation

Using the Primitives and the Adjacency Graph:

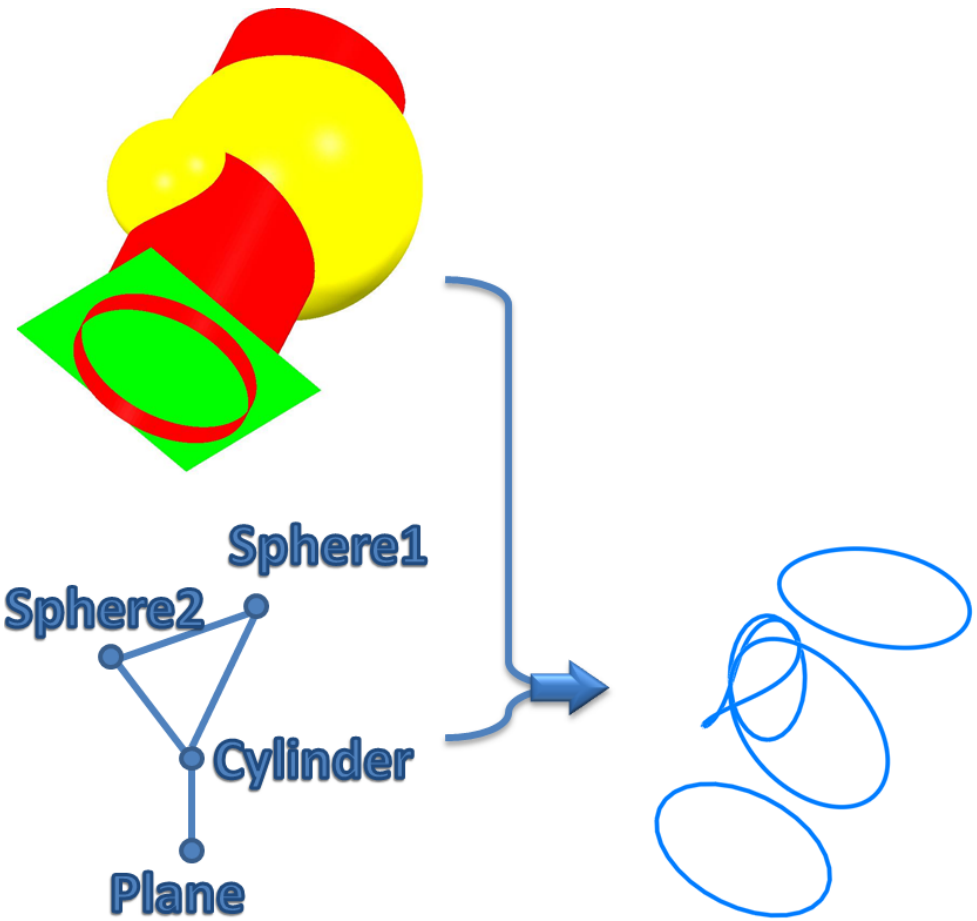
- For each Adjacency Graph edge $\Rightarrow$ intersection curves,



Valid intersection computation

Using the Primitives and the Adjacency Graph:

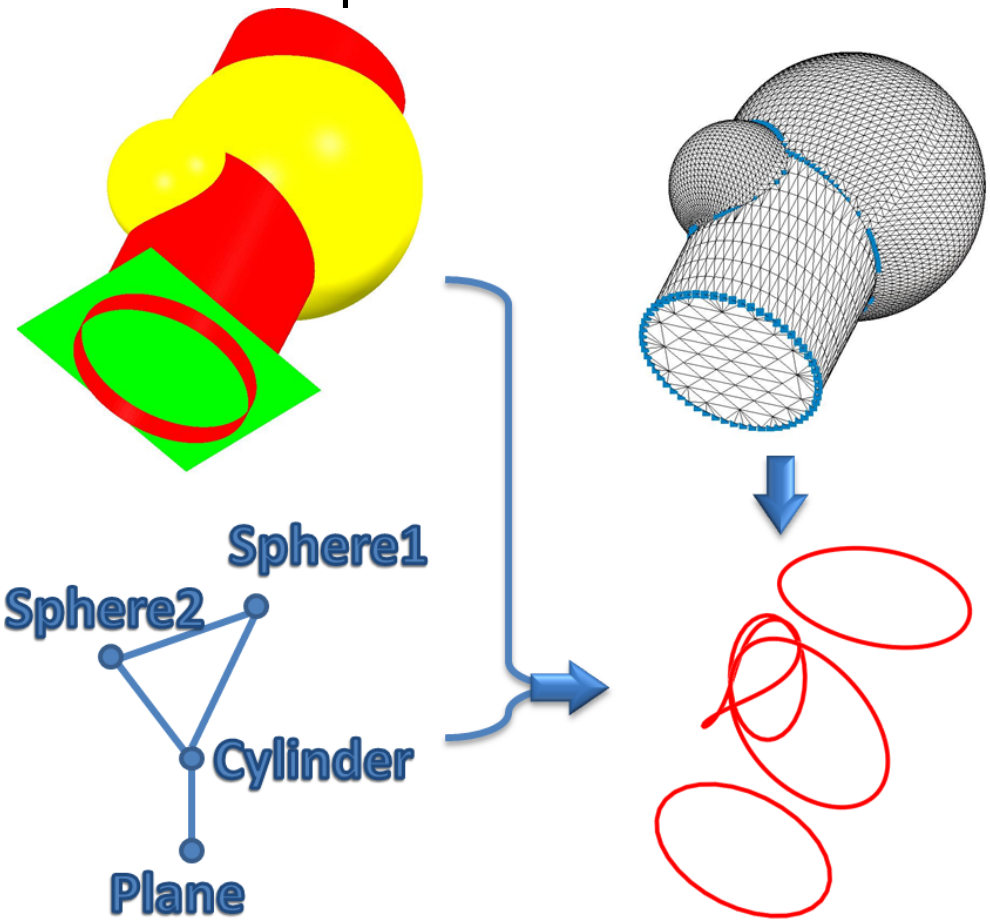
- For each Adjacency Graph edge $\Rightarrow$ intersection curves,



Valid intersection computation

Using the Primitives and the Adjacency Graph:

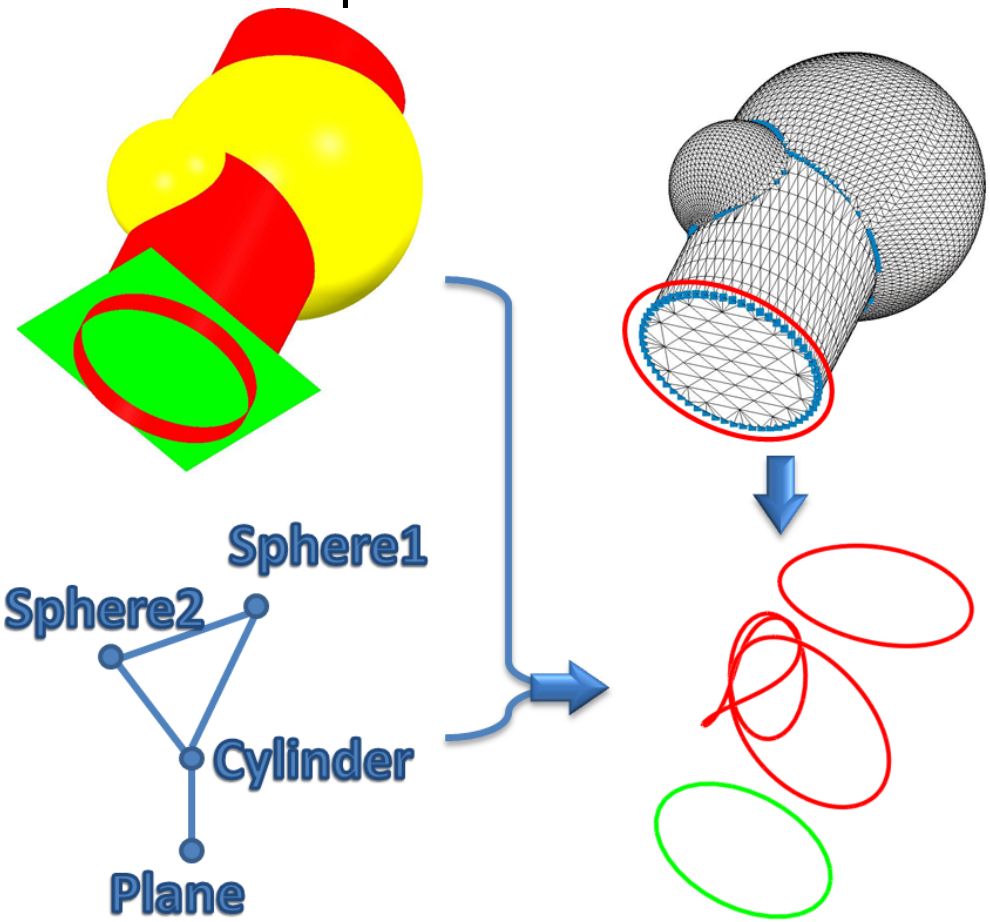
- For each Adjacency Graph edge ⇒ intersection curves,
- Distance to common points ⇒ validation of intersections.



Valid intersection computation

Using the Primitives and the Adjacency Graph:

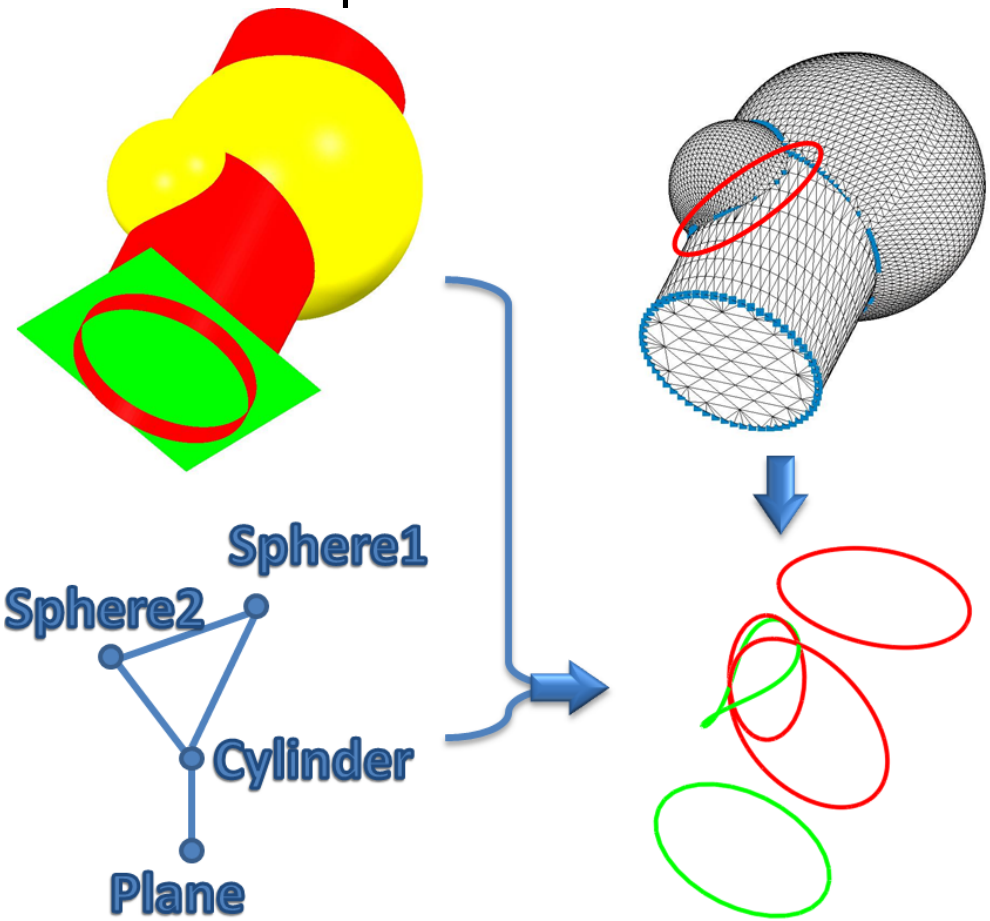
- For each Adjacency Graph edge ⇒ intersection curves,
- Distance to common points ⇒ validation of intersections.



Valid intersection computation

Using the Primitives and the Adjacency Graph:

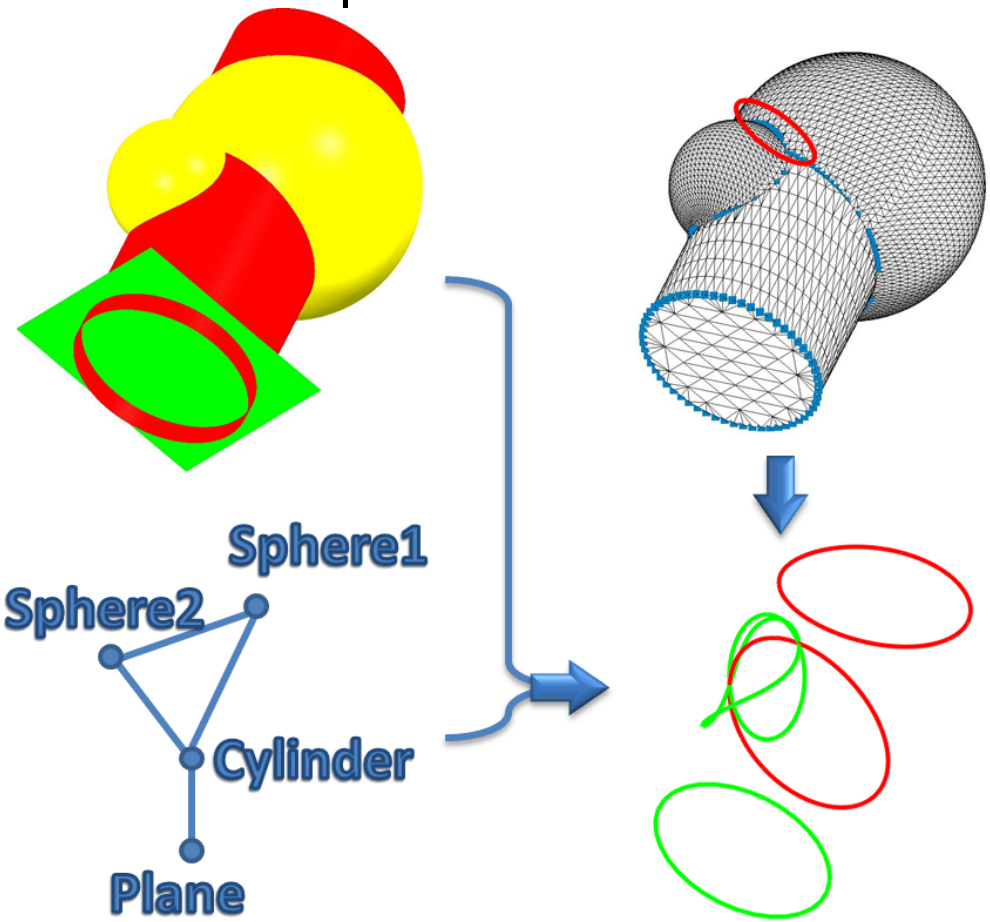
- For each Adjacency Graph edge $\Rightarrow$ intersection curves,
- Distance to common points $\Rightarrow$ validation of intersections.



Valid intersection computation

Using the Primitives and the Adjacency Graph:

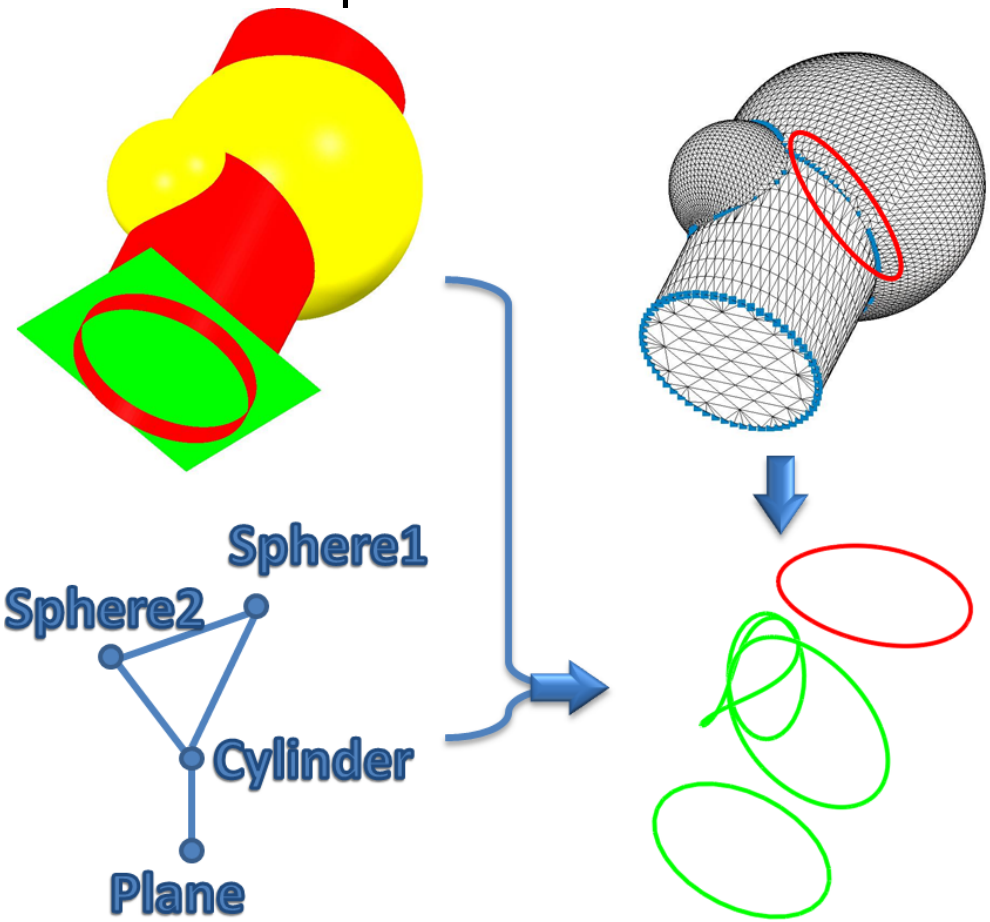
- For each Adjacency Graph edge ⇒ intersection curves,
- Distance to common points ⇒ validation of intersections.



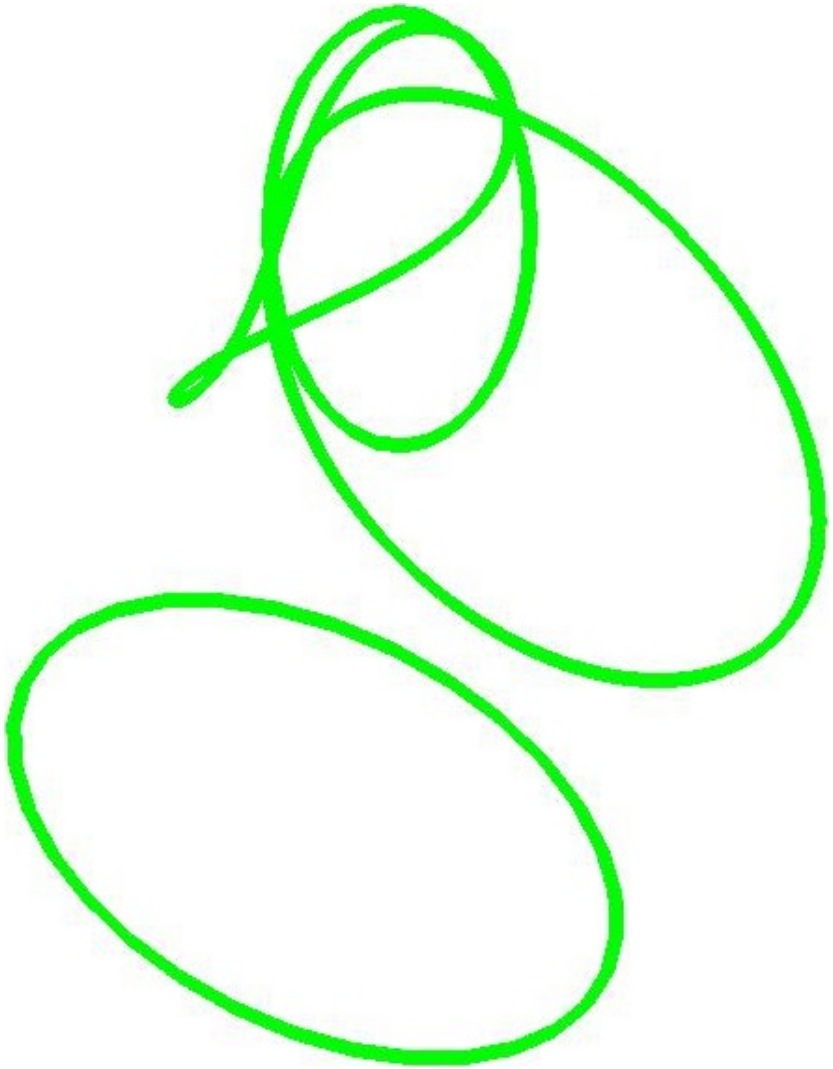
Valid intersection computation

Using the Primitives and the Adjacency Graph:

- For each Adjacency Graph edge ⇒ intersection curves,
- Distance to common points ⇒ validation of intersections.



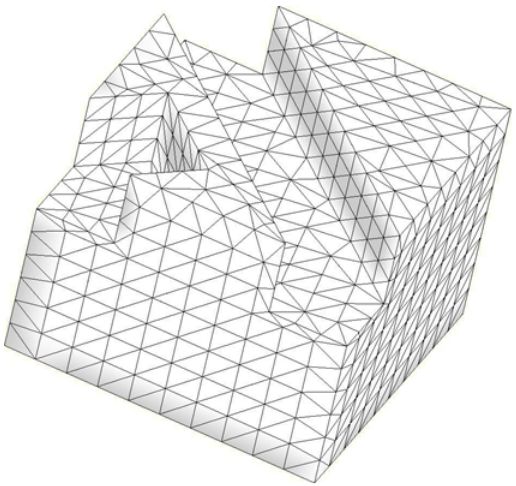
Consistent intersection computation



Decomposition of the intersection curve into edges

With the previous steps:

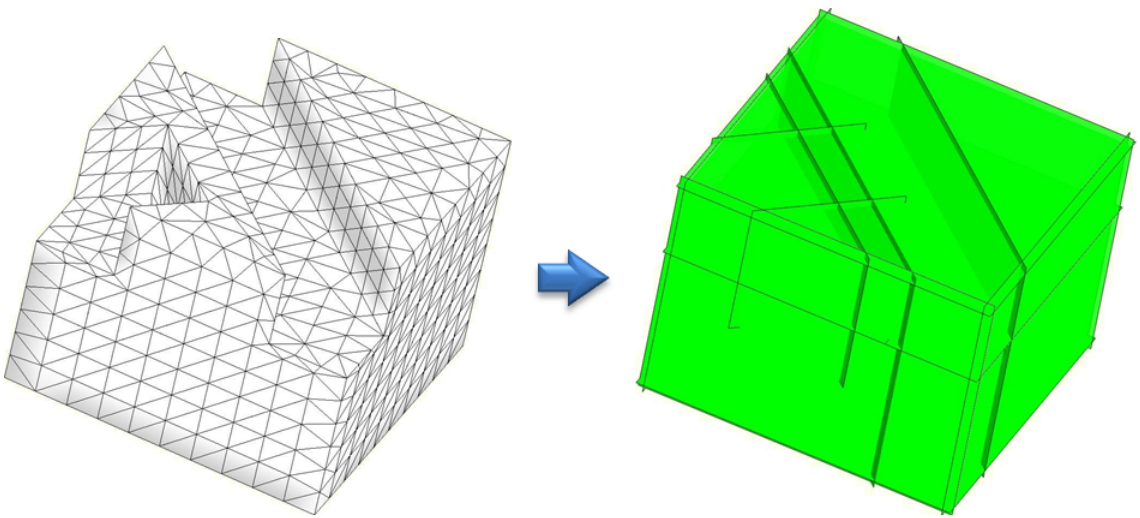
Mesh



Decomposition of the intersection curve into edges

With the previous steps:

Mesh Primitives

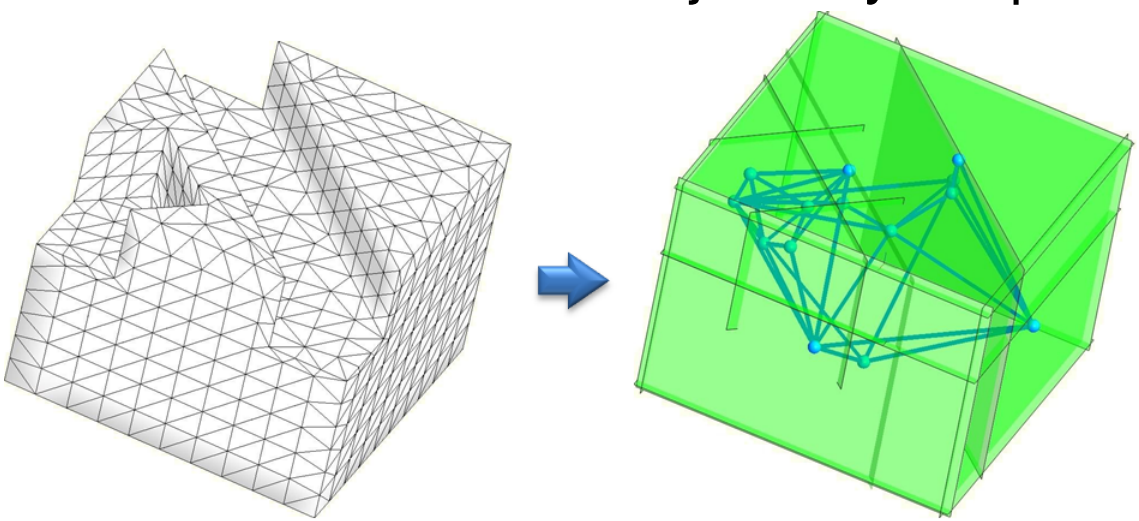


Decomposition of the intersection curve into edges

With the previous steps:

Mesh

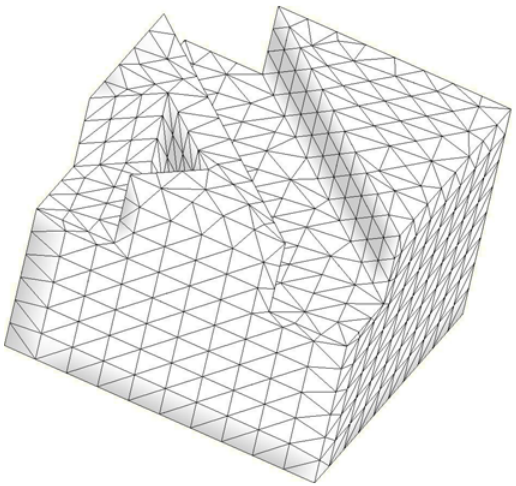
Primitives
+Adjacency Graph



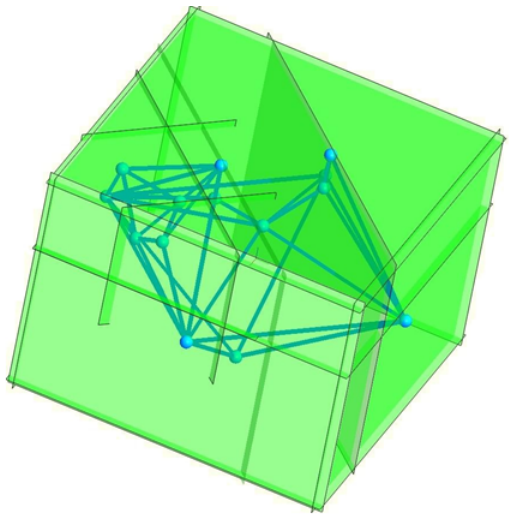
Decomposition of the intersection curve into edges

With the previous steps:

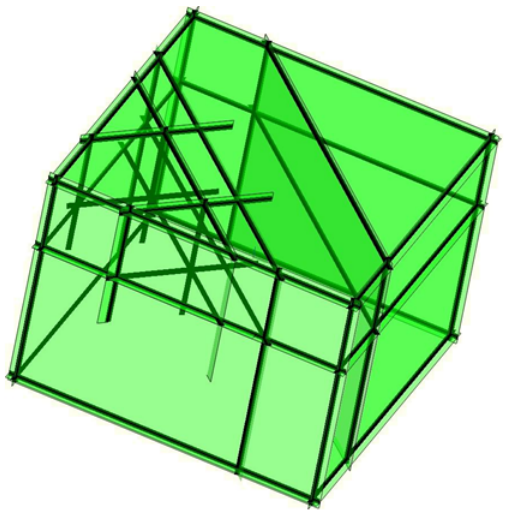
Mesh



Primitives
+Adjacency Graph



Valid
Intersection



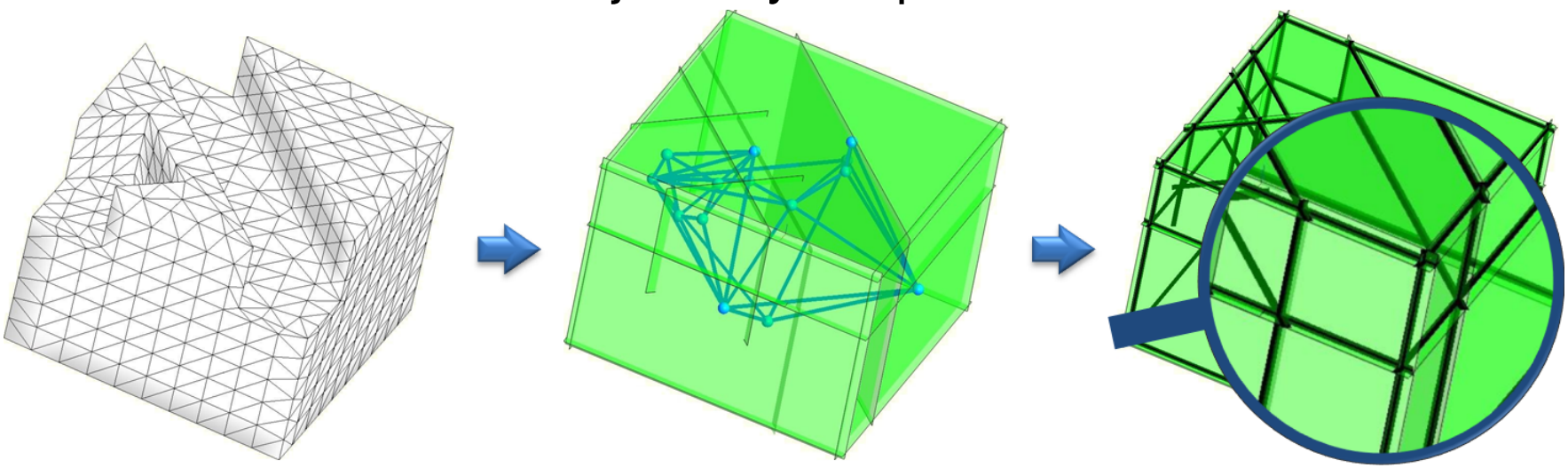
Decomposition of the intersection curve into edges

With the previous steps:

Mesh

Primitives
+Adjacency Graph

Valid
Intersection



But the intersection curves between two primitives must be trimmed.

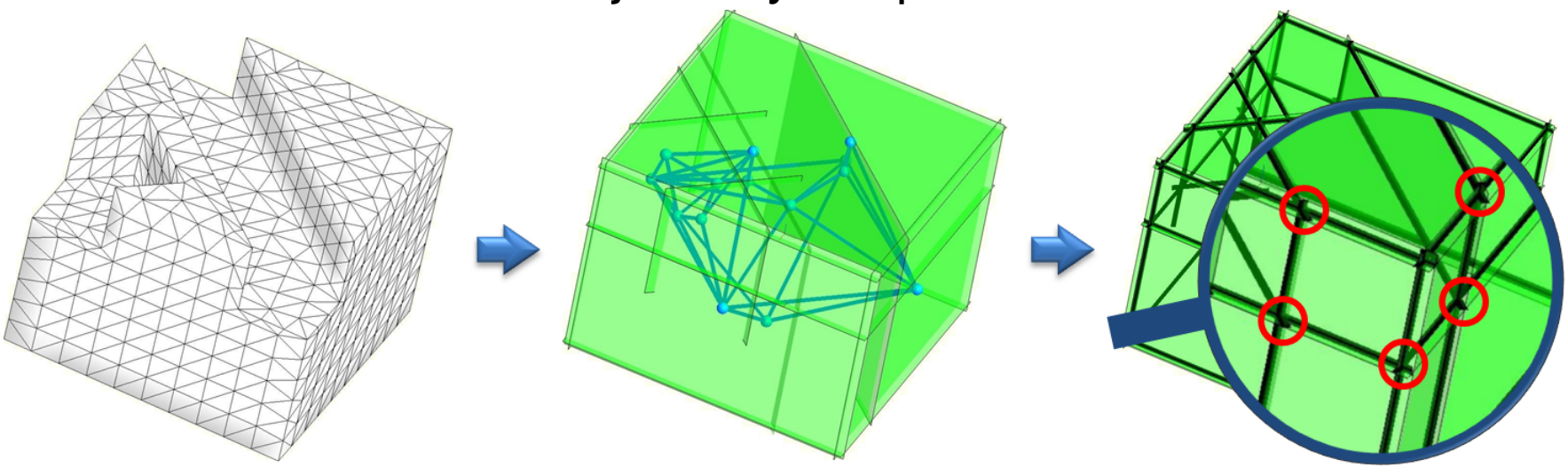
Decomposition of the intersection curve into edges

With the previous steps:

Mesh

Primitives
+Adjacency Graph

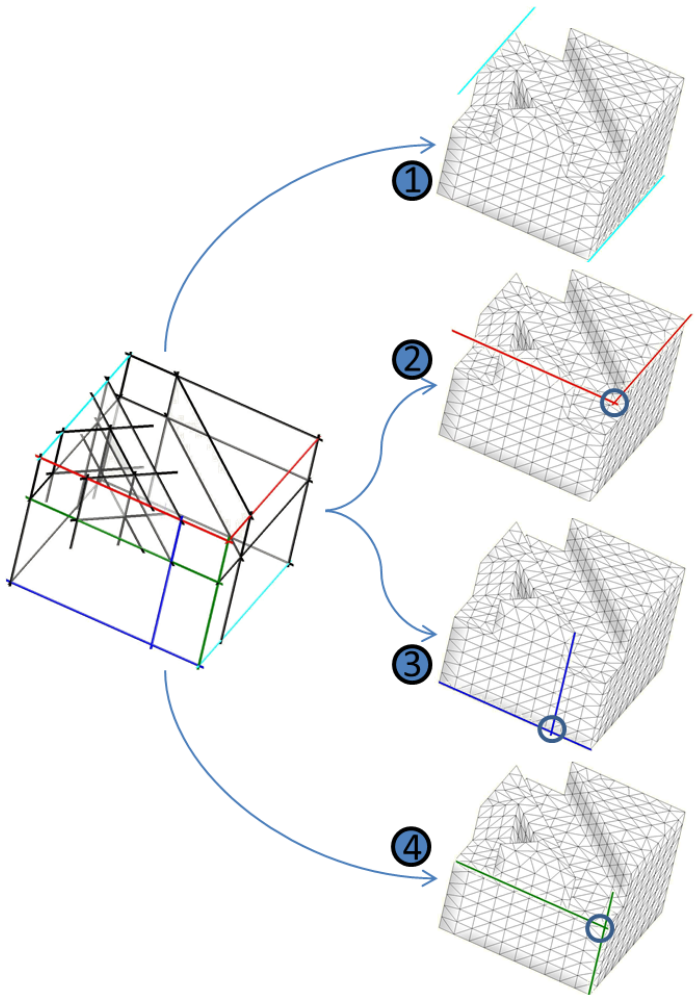
Valid
Intersection



But the intersection curves between two primitives must be trimmed.
Using specific points: **THE JUNCTIONS**.

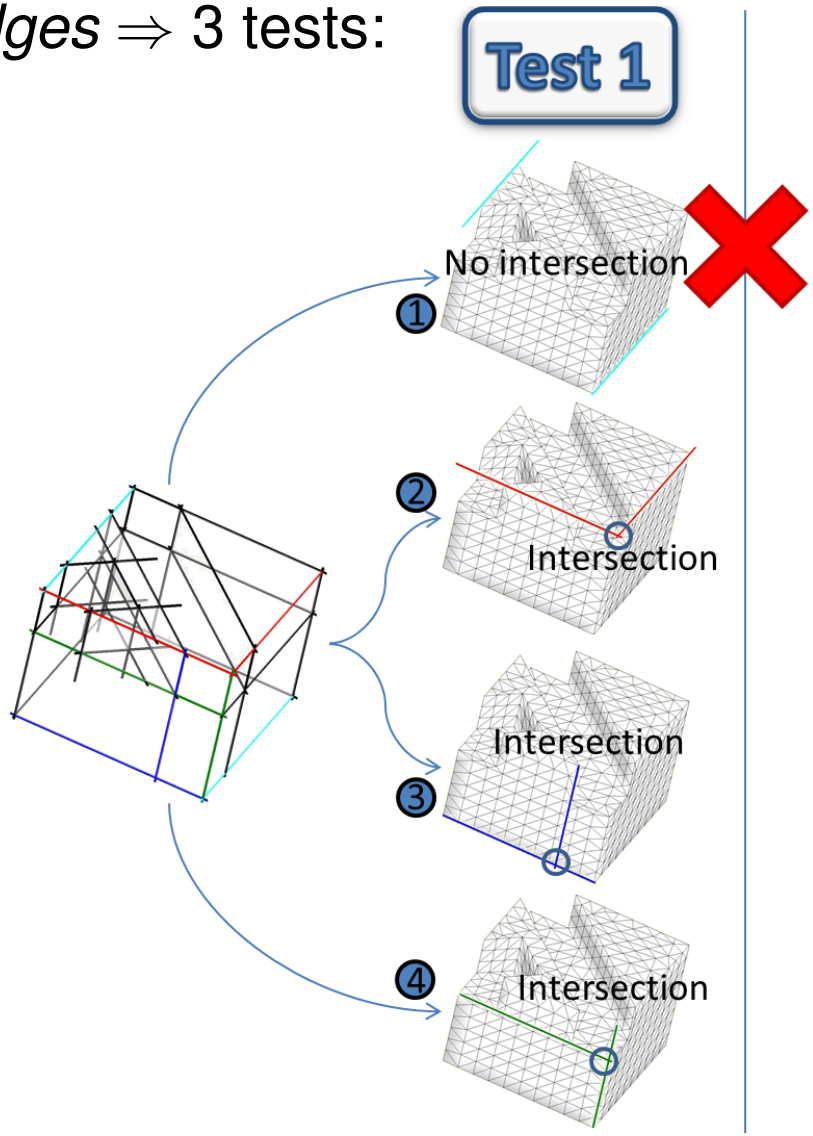
Junction extraction

To extract junctions and decompose intersection curve into *edges* $\Rightarrow$ 3 tests:



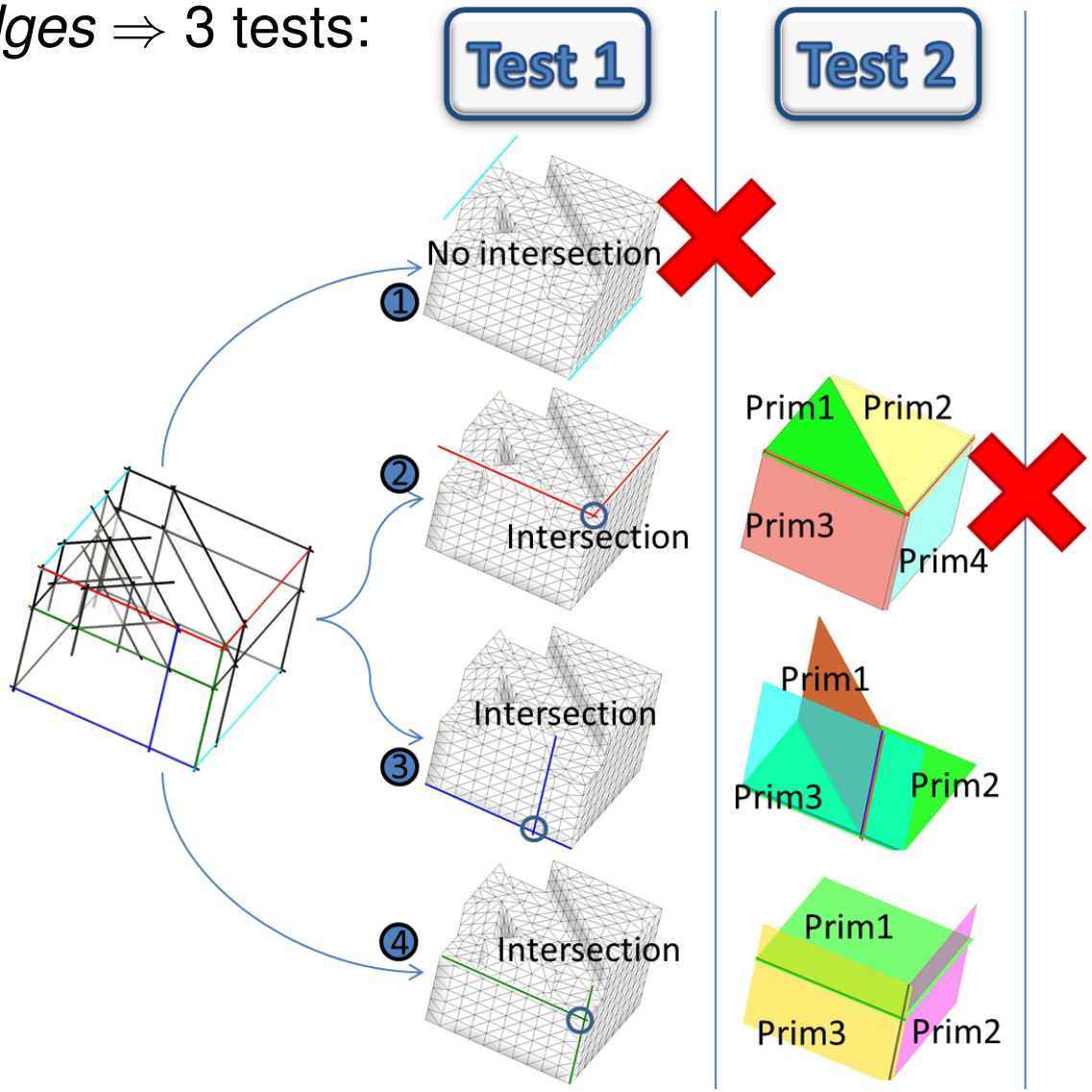
Junction extraction

To extract junctions and decompose intersection curve into *edges* $\Rightarrow$ 3 tests:



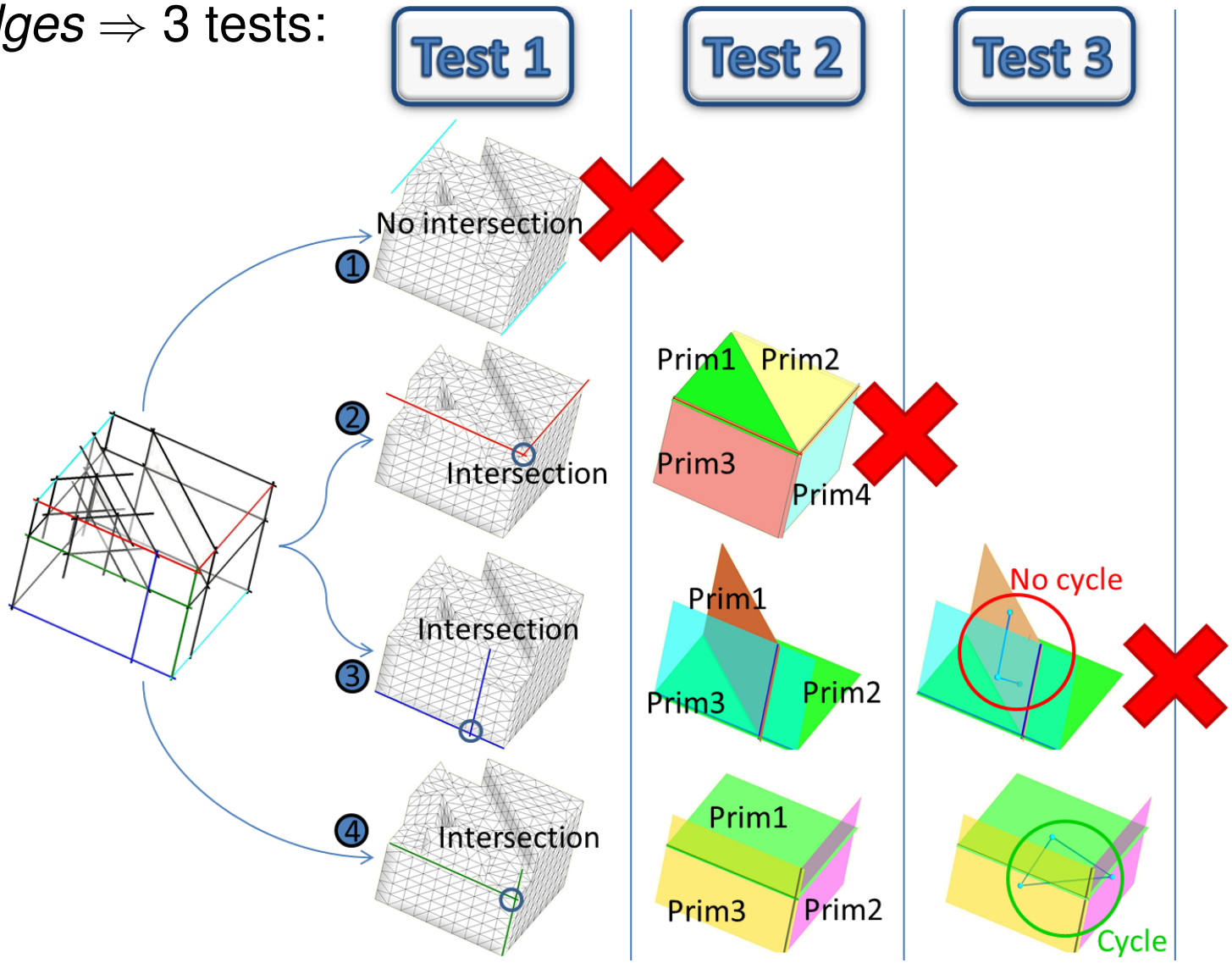
Junction extraction

To extract junctions and decompose intersection curve into *edges* $\Rightarrow$ 3 tests:



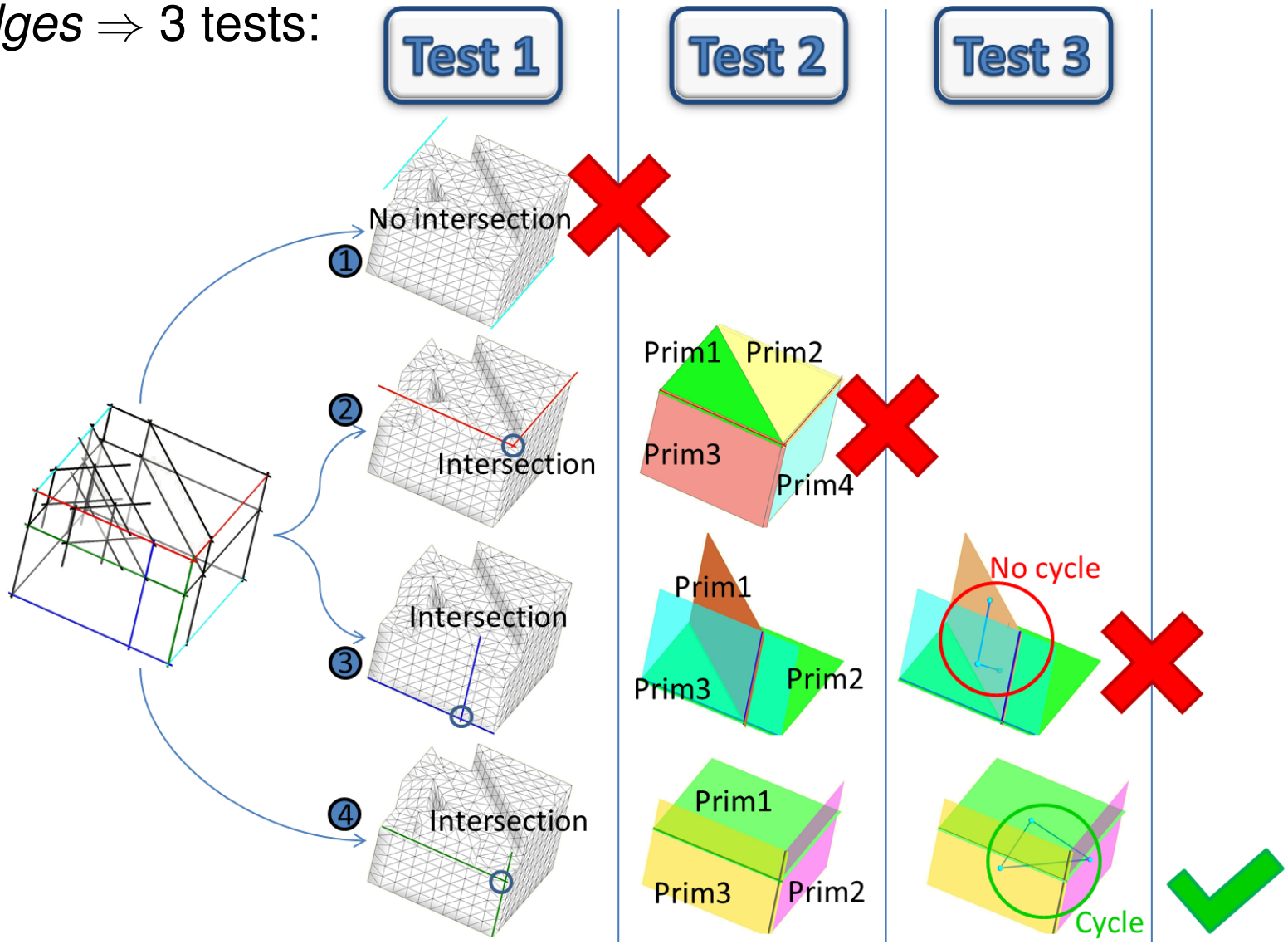
Junction extraction

To extract junctions and decompose intersection curve into *edges* ⇒ 3 tests:

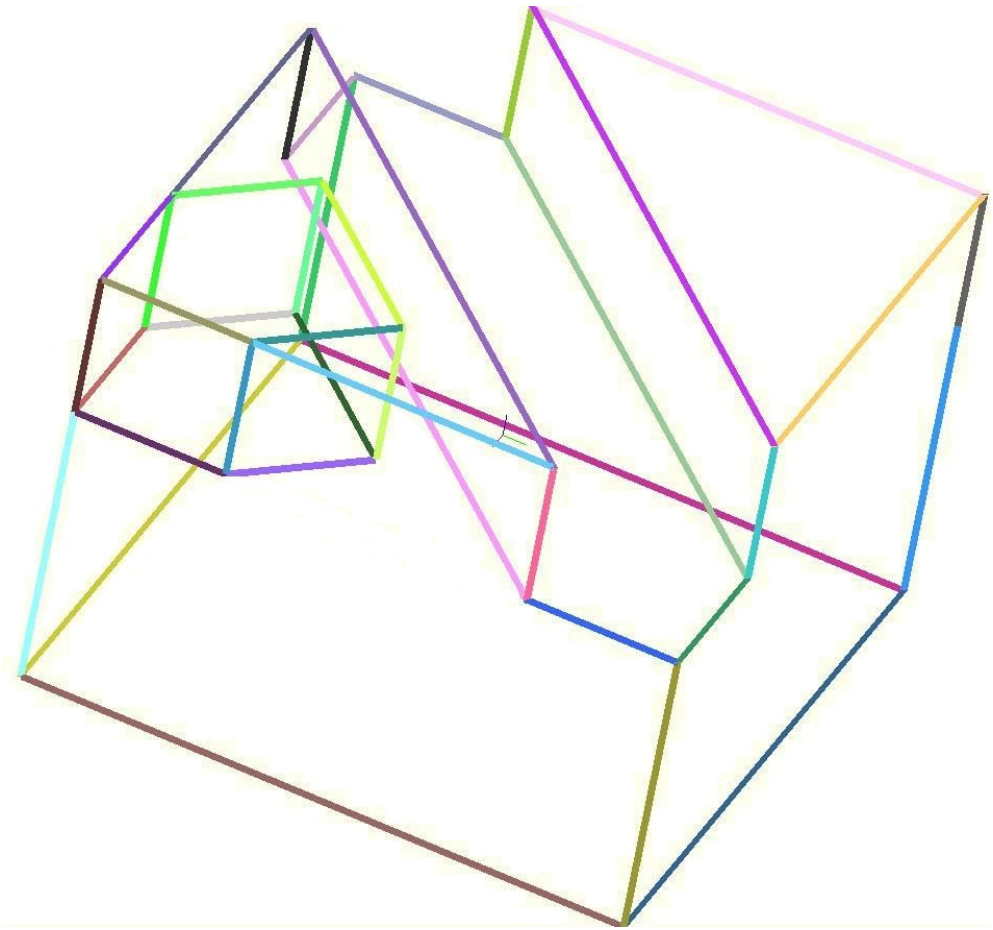


Junction extraction

To extract junctions and decompose intersection curve into *edges* ⇒ 3 tests:

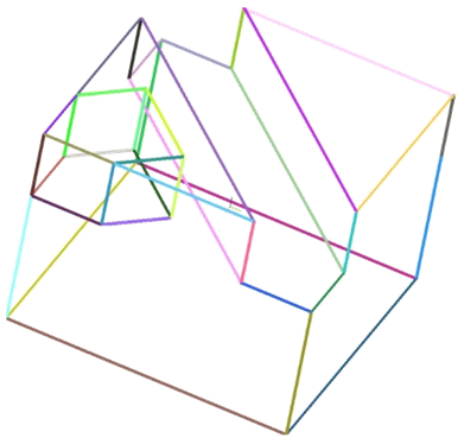


Junction extraction



Wire construction

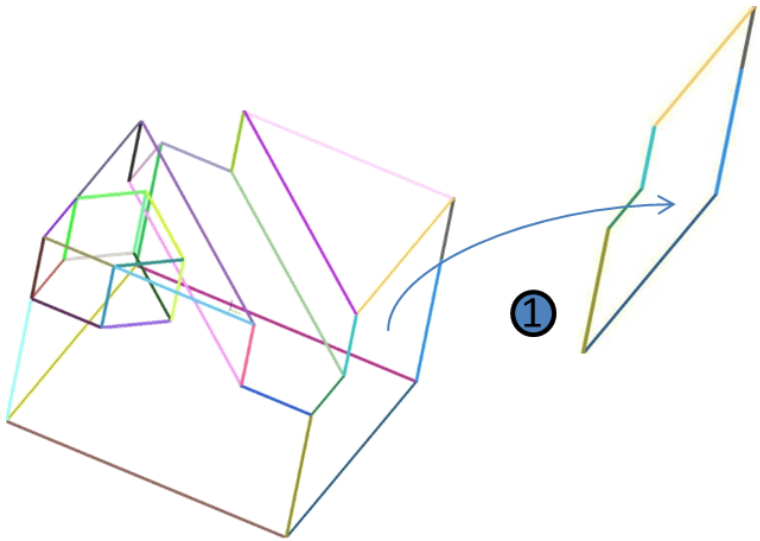
Then the edges are assembled into wires.



Wire construction

Then the edges are assembled into wires. There is two cases:

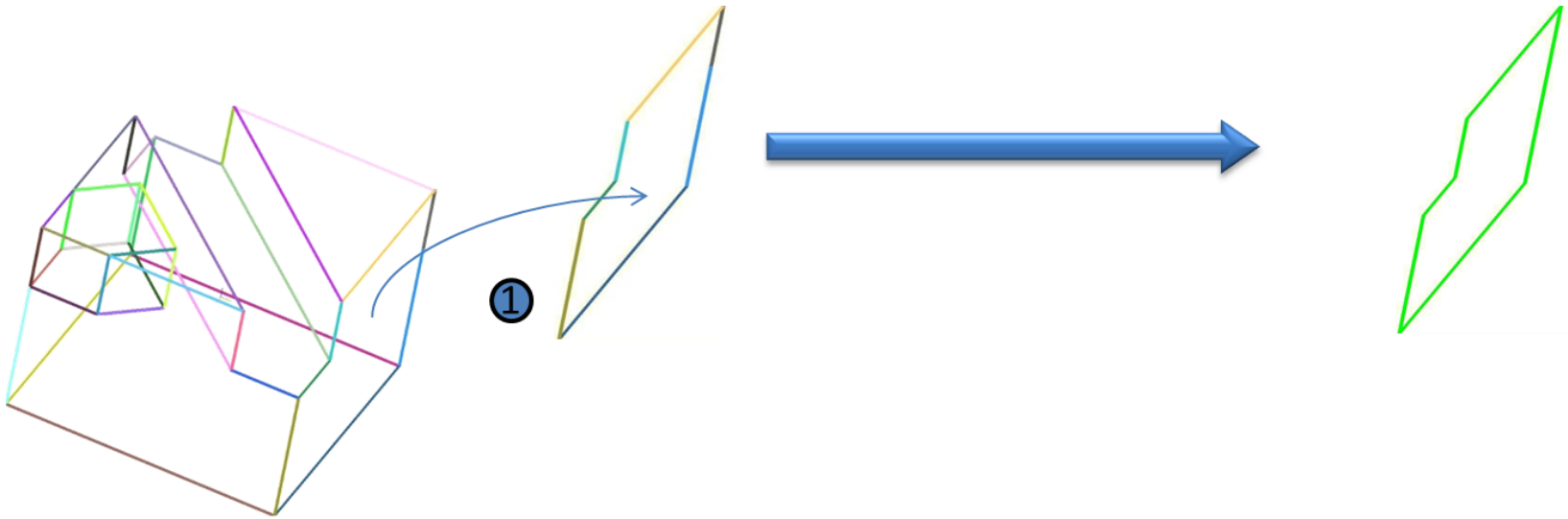
1. Only one way



Wire construction

Then the edges are assembled into wires. There is two cases:

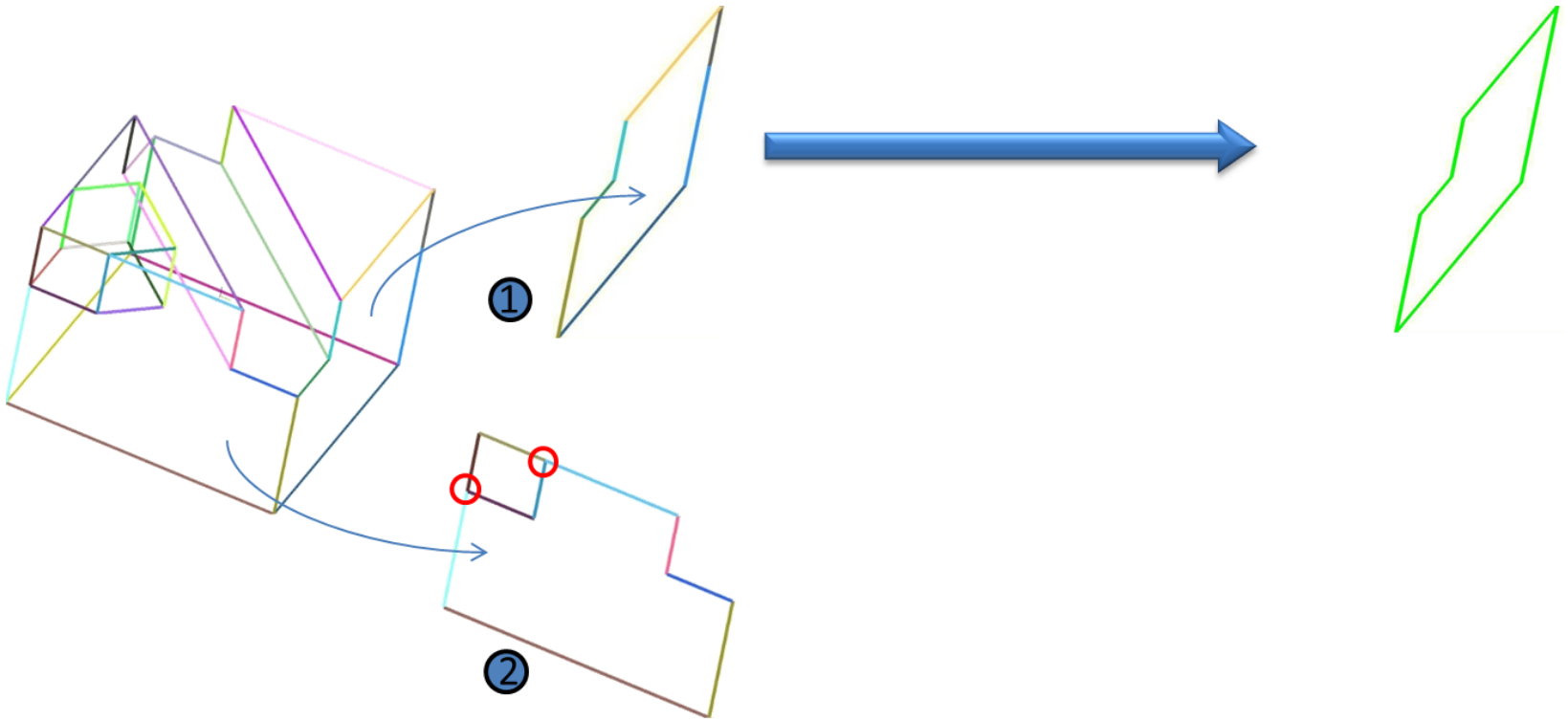
1. Only one way $\Rightarrow$ immediate construction,



Wire construction

Then the edges are assembled into wires. There is two cases:

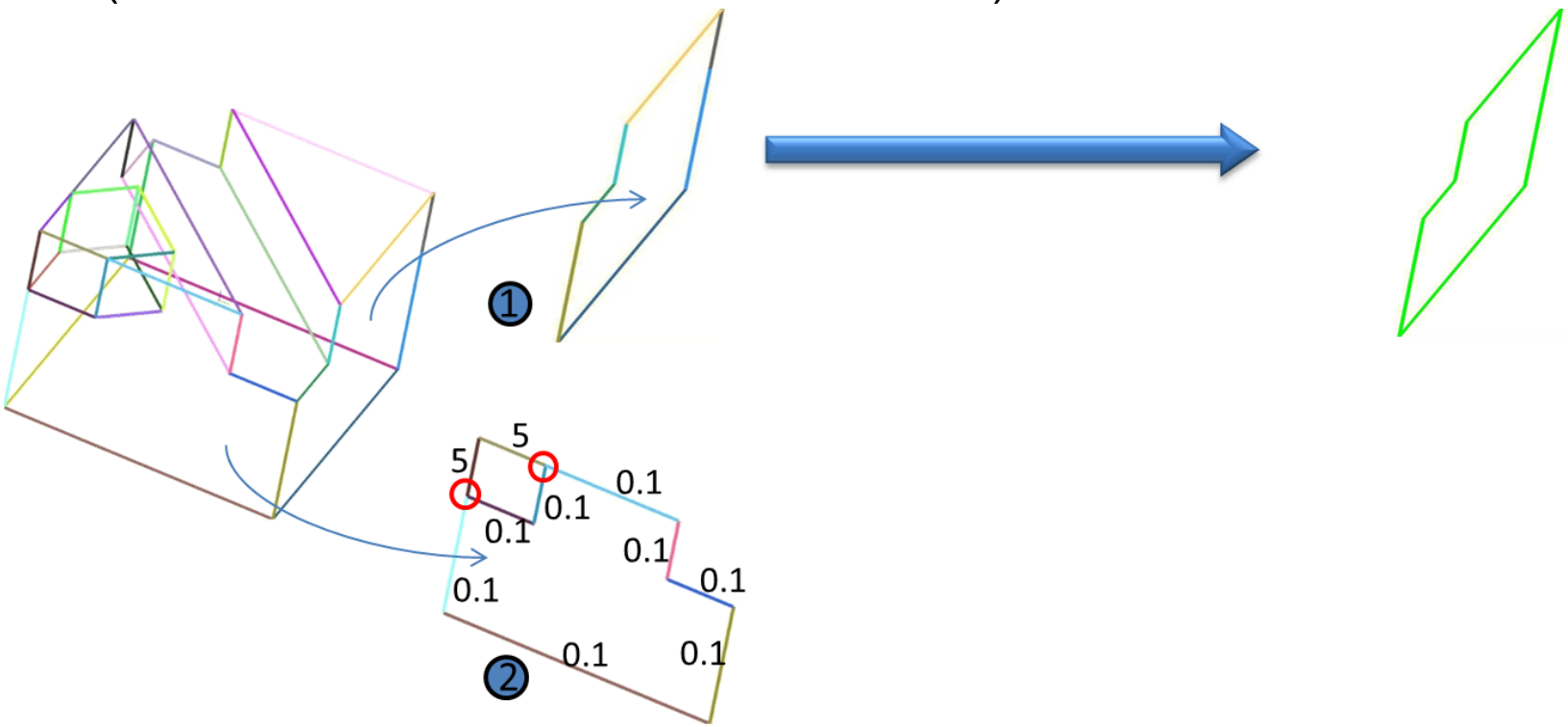
1. Only one way $\Rightarrow$ immediate construction,
2. Several possible paths



Wire construction

Then the edges are assembled into wires. There is two cases:

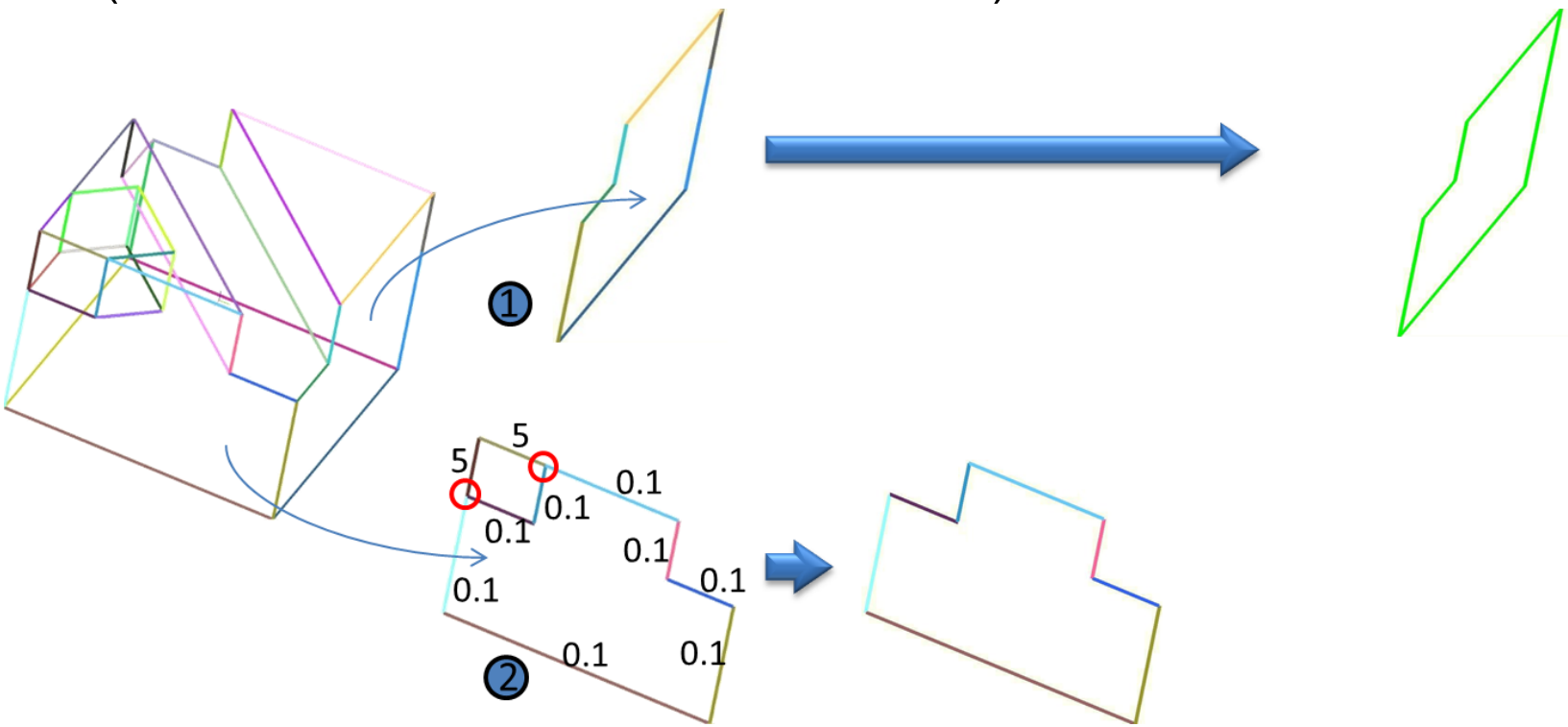
1. Only one way $\Rightarrow$ immediate construction,
2. Several possible paths $\Rightarrow$ construction using weight (= distance to Extended Point Area).



Wire construction

Then the edges are assembled into wires. There is two cases:

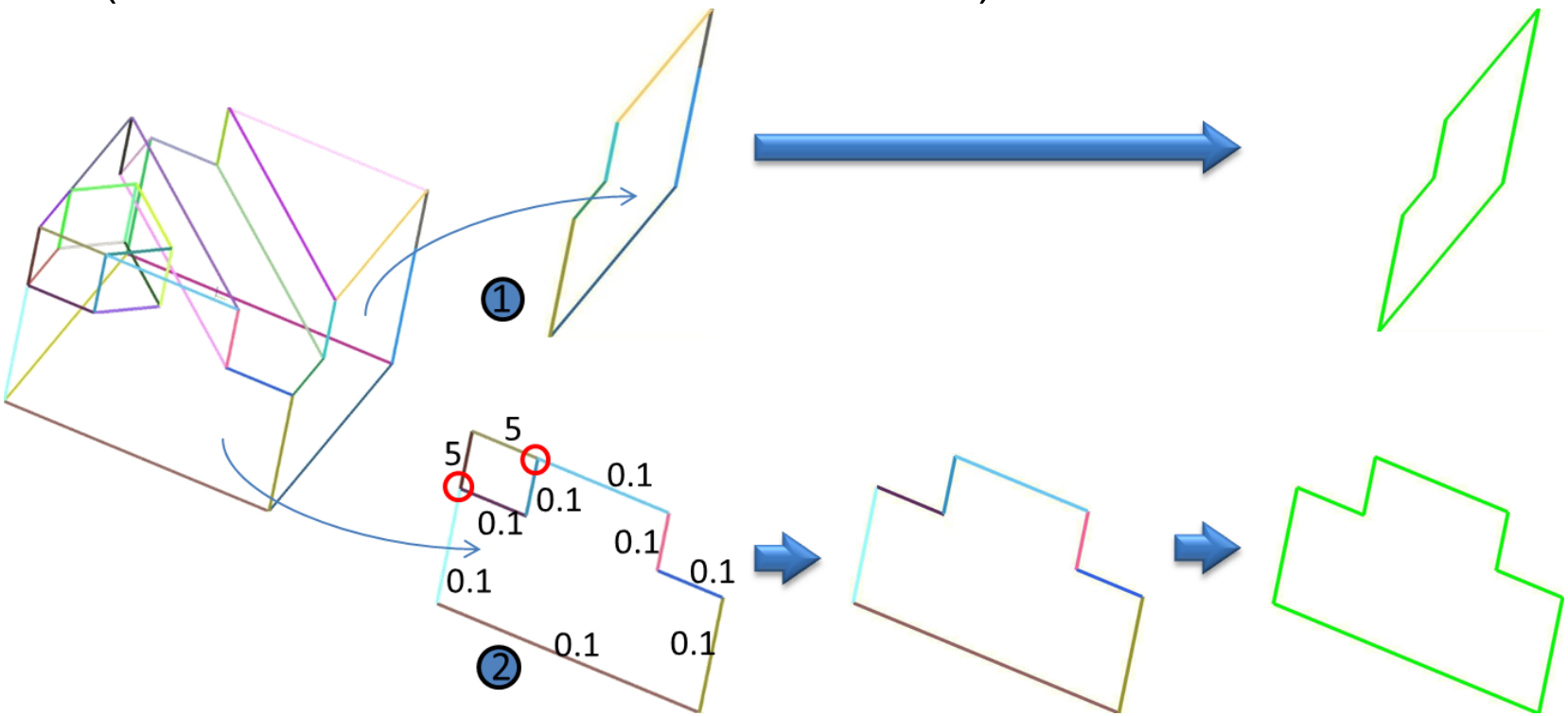
1. Only one way $\Rightarrow$ immediate construction,
2. Several possible paths $\Rightarrow$ construction using weight (= distance to Extended Point Area).



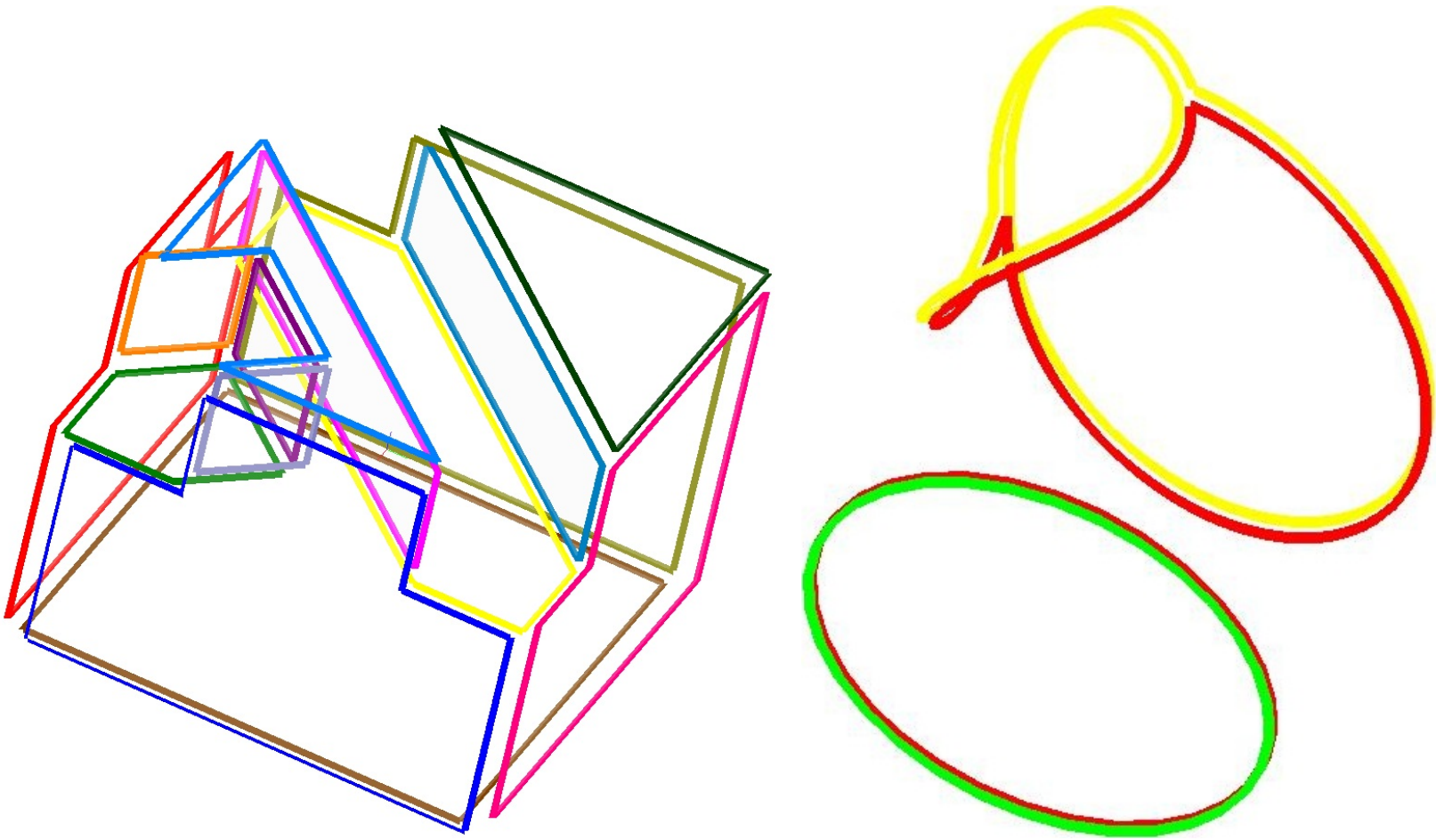
Wire construction

Then the edges are assembled into wires. There is two cases:

1. Only one way $\Rightarrow$ immediate construction,
2. Several possible paths $\Rightarrow$ construction using weight (= distance to Extended Point Area).

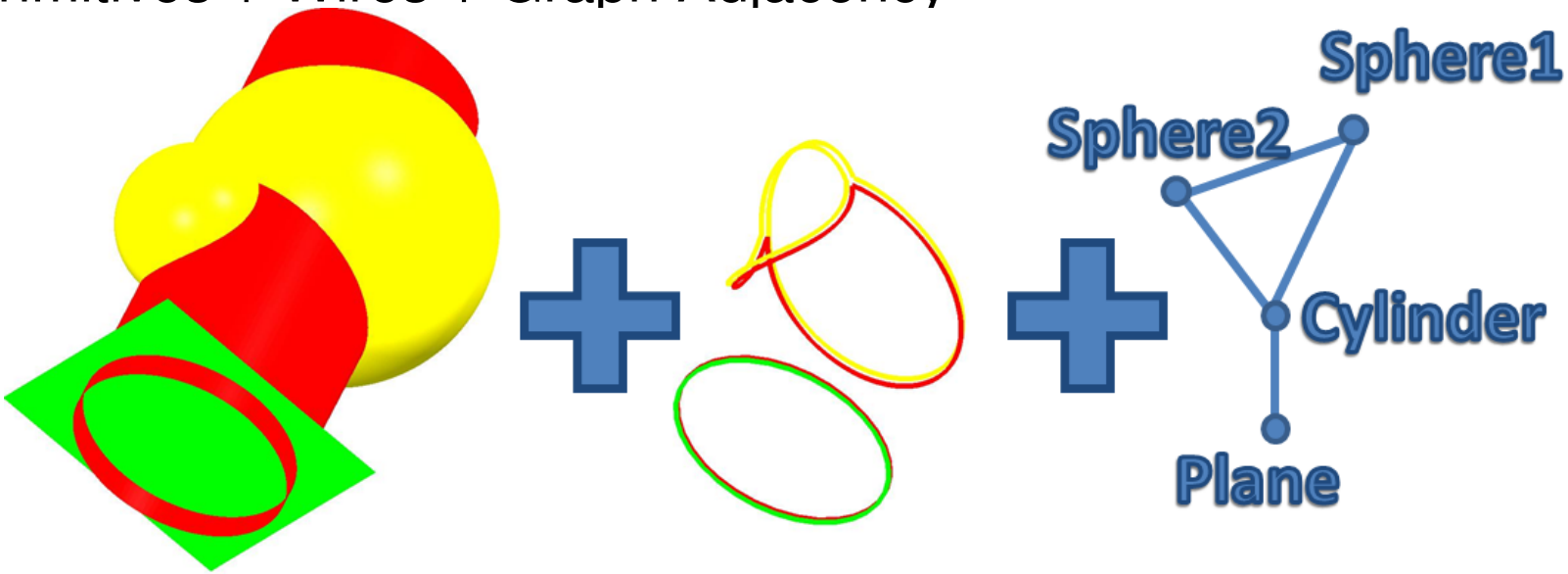


Wire construction



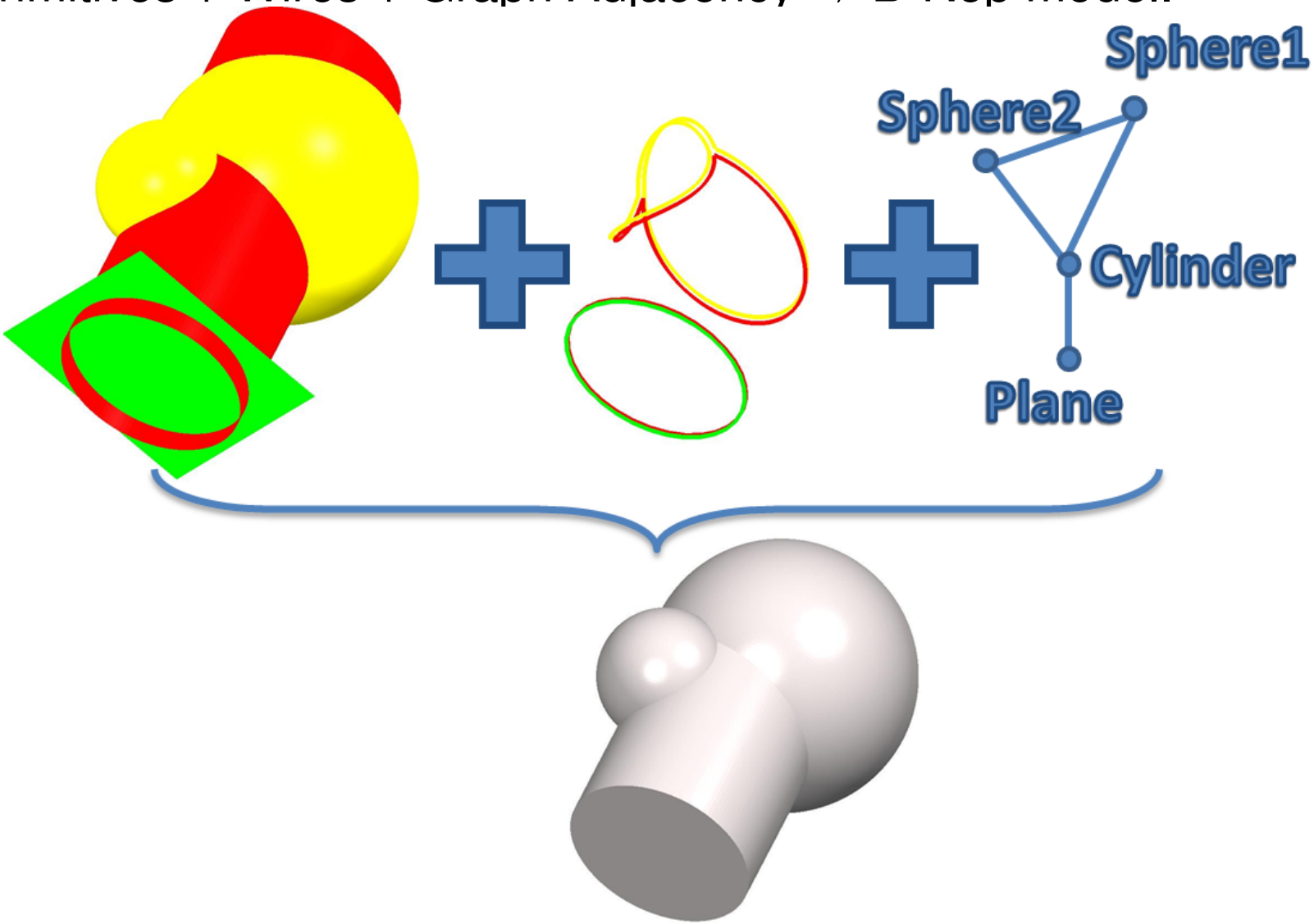
B-Rep model creation

Primitives + Wires + Graph Adjacency

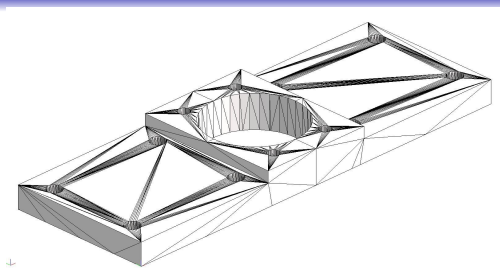


B-Rep model creation

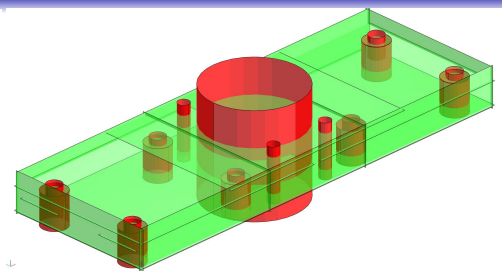
Primitives + Wires + Graph Adjacency ⇒ B-Rep model.



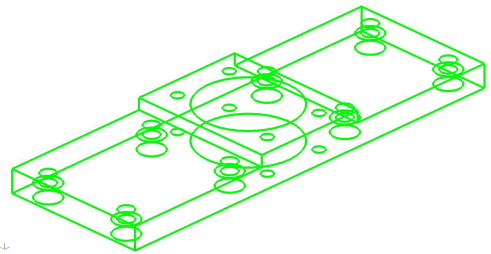
Results and computation time



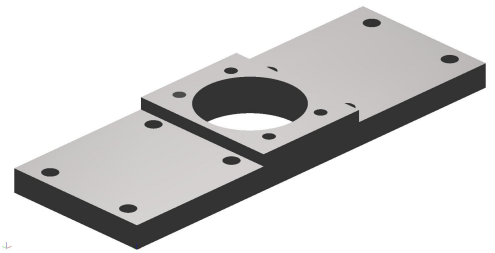
Mesh



Primitives



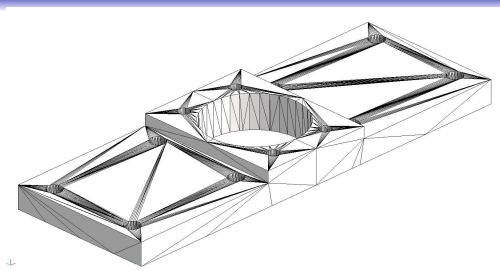
Wires



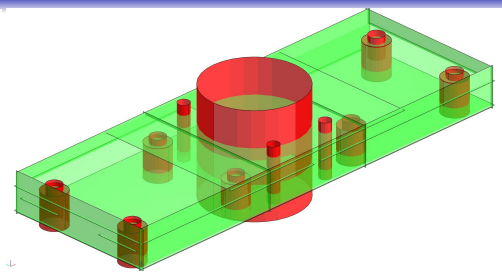
B-Rep model

Nb Triangles	Nb Primitives	Time
3,220	39	2 s

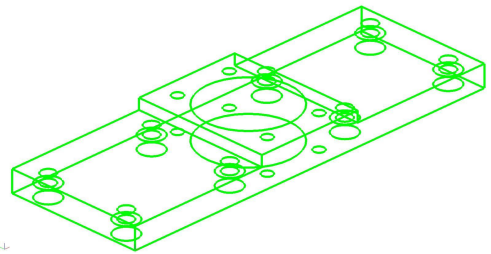
Results and computation time



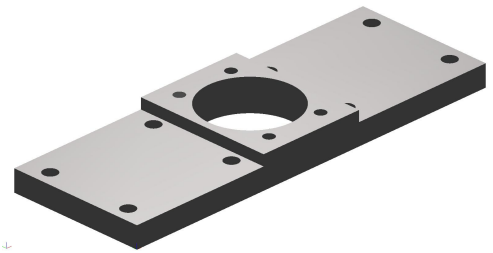
Mesh



Primitives

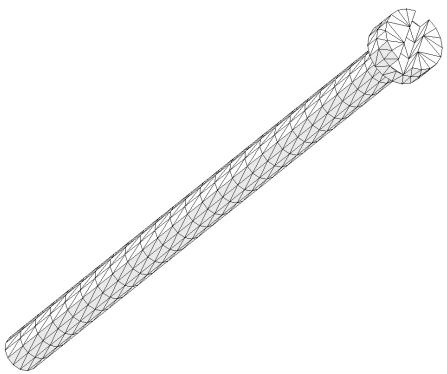


Wires

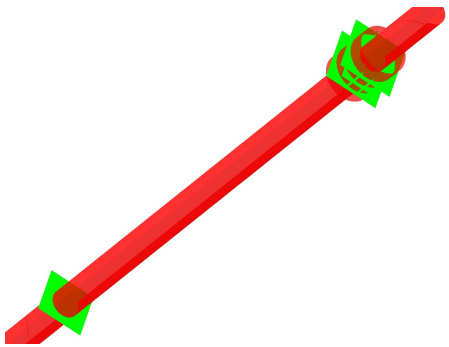


B-Rep model

Nb Triangles	Nb Primitives	Time
3,220	39	2 s



Mesh



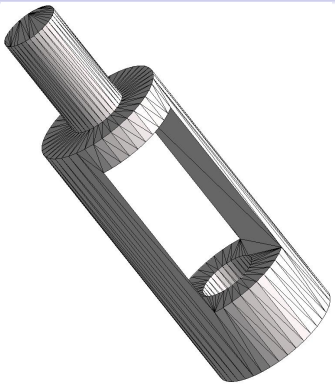
Primitives

Wires

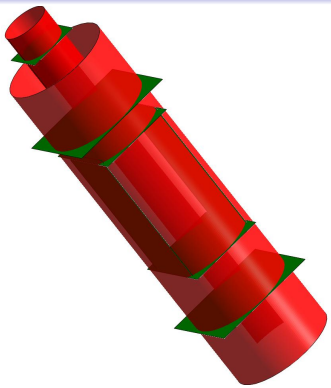
B-Rep model

Nb Triangles	Nb Primitives	Time
854	9	4 s

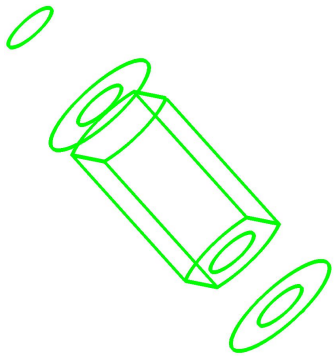
Results and comparison



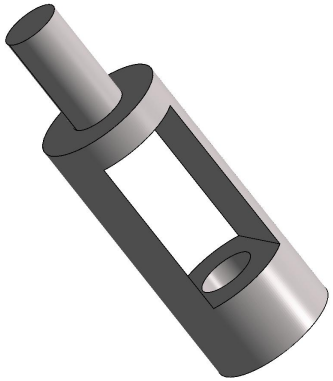
Mesh



Primitives

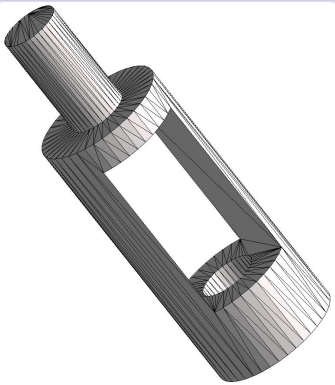


Wires

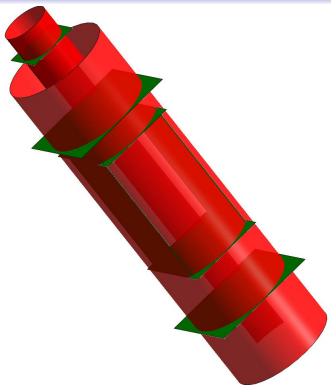


B-Rep model

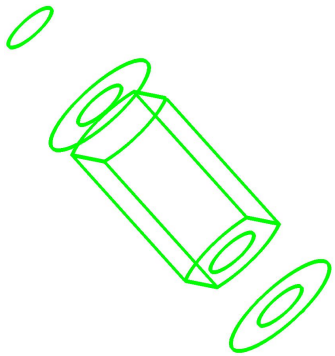
Results and comparison



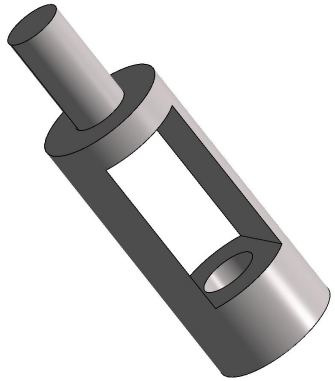
Mesh



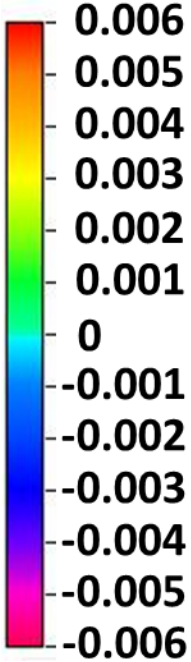
Primitives

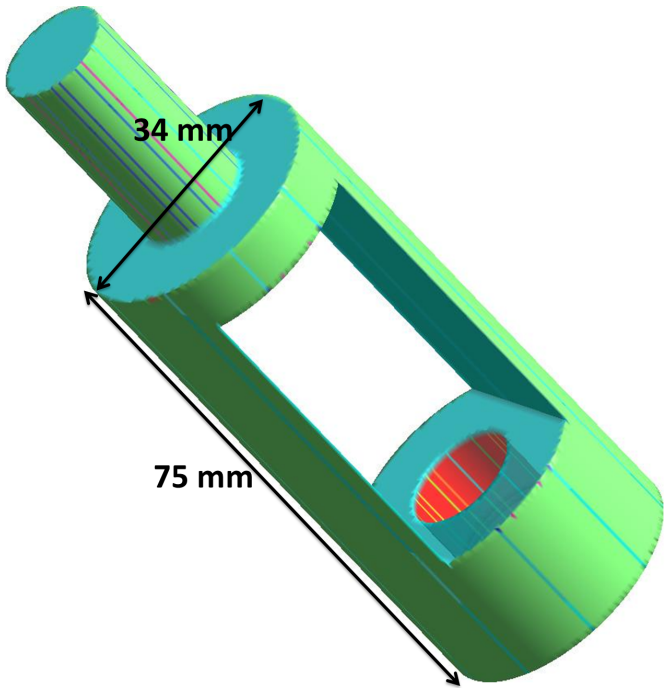


Wires



B-Rep model





Distance

Minimum	Mean	Maximum
0.000	≈0.000	0.052

Conclusion and Perspectives

In this paper: a new approach to construct a consistent topology from a mesh and the corresponding set of primitives.

Conclusion and Perspectives

In this paper: a new approach to construct a consistent topology from a mesh and the corresponding set of primitives.

Perspective 1: improve the primitive extraction

Imprecision in the primitives parameters computation raises many problem in the topology reconstruction step (intersection computation, wire construction...)

Conclusion and Perspectives

In this paper: a new approach to construct a consistent topology from a mesh and the corresponding set of primitives.

Perspective 1: improve the primitive extraction

Imprecision in the primitives parameters computation raises many problem in the topology reconstruction step (intersection computation, wire construction...)
$\Rightarrow$ Find and add constraints (tangency or parallel to or...) to robustify the reconstruction process.

Conclusion and Perspectives

In this paper: a new approach to construct a consistent topology from a mesh and the corresponding set of primitives.

Perspective 1: improve the primitive extraction

Imprecision in the primitives parameters computation raises many problem in the topology reconstruction step (intersection computation, wire construction...)
$\Rightarrow$ Find and add constraints (tangency or parallel to or...) to robustify the reconstruction process.

Perspective 2: accelerate the wire construction step

The weight computation for each edge and the wire construction take a lot of time.

Conclusion and Perspectives

In this paper: a new approach to construct a consistent topology from a mesh and the corresponding set of primitives.

Perspective 1: improve the primitive extraction

Imprecision in the primitives parameters computation raises many problem in the topology reconstruction step (intersection computation, wire construction...)
$\Rightarrow$ Find and add constraints (tangency or parallel to or...) to robustify the reconstruction process.

Perspective 2: accelerate the wire construction step

The weight computation for each edge and the wire construction take a lot of time.
$\Rightarrow$ Use existing algorithms to find the optimal path in a valued graph.

Thank you for your attention

QUESTIONS?

Site: www.lirmm.fr/~beniere
Mail: roseline.beniere@lirmm.fr
C4W site: www.c4w.com

Roseline Bénière, G. Subsol, G. Gesquière, F. Le Breton and W. Puech, *Topology Reconstruction for B-Rep Modeling from 3D Mesh in reverse Engineering Applications*, SPIE, *San Francisco*, 2012

LIRMM

C4W Custom CAD Software