Recovering Primitives in 3D CAD meshes

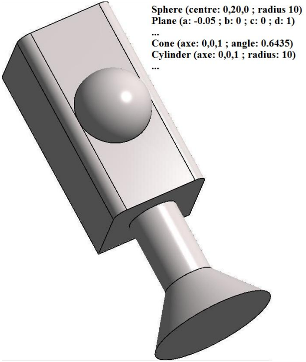
Roseline Bénière[1,2]
G. Subsol[1], G. Gesquière[3], F. Le Breton[2] and W. Puech[1]

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

January 26st
SPIE 2011

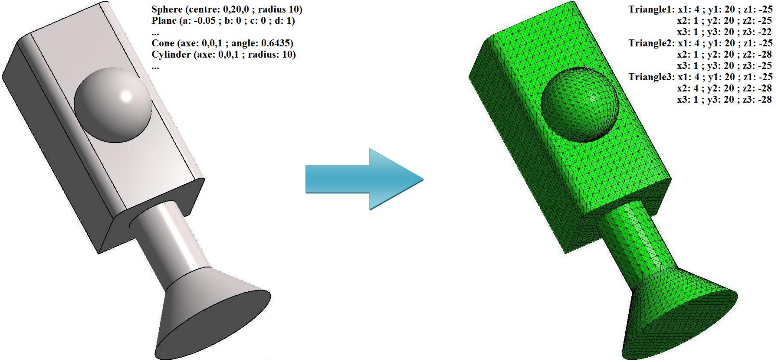
Objective

- A CAD object is usually modeled by a structured combination of primitive surfaces (Plane, Sphere, Cone, Cylinder, Splines ...)



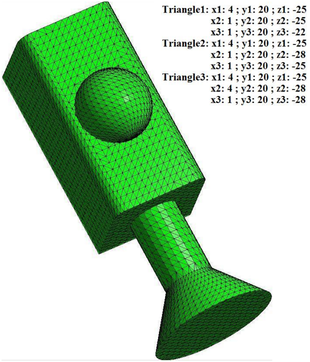
Objective

- A CAD object is usually modeled by a structured combination of primitive surfaces (Plane, Sphere, Cone, Cylinder, Splines ...)
- But a discretization into a 3D mesh is used in many cases



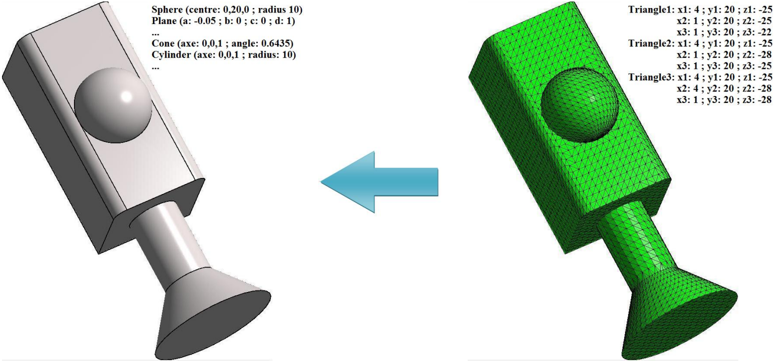
Objective

- A CAD object is usually modeled by a structured combination of primitive surfaces (Plane, Sphere, Cone, Cylinder, Splines ...)
- But a discretization into a 3D mesh is used in many cases
- And the initial continuous model can be lost or not correspond anymore



Objective

- A CAD object is usually modeled by a structured combination of primitive surfaces (Plane, Sphere, Cone, Cylinder, Splines ...)
- But a discretization into a 3D mesh is used in many cases
- And the initial continuous model can be lost or not correspond anymore
- Then a primitive extraction algorithm may be required to reconstruct a continuous representation



Overview

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

- Sunil and Pande
  Automatic recognition of features from freeform surface CAD models
  Computer-Aided Design 40(4): 502-517 2008

Overview

- Benkõ *et al.*
  Algorithms for reverse engineering boundary representation models
  Computer-Aided Design 33(11): 839-851 2001
- Sunil and Pande
  Automatic recognition of features from freeform surface CAD models
  Computer-Aided Design 40(4): 502-517 2008

Our method:

1. Definition of a local shape criterion

Overview

- Benkõ *et al.*
  Algorithms for reverse engineering boundary representation models
  Computer-Aided Design 33(11): 839-851 2001
- Sunil and Pande
  Automatic recognition of features from freeform surface CAD models
  Computer-Aided Design 40(4): 502-517 2008

Our method:

1. Definition of a local shape criterion
2. Grouping vertices into areas corresponding to one primitive type

Overview

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

- Sunil and Pande
  Automatic recognition of features from freeform surface CAD models
  Computer-Aided Design 40(4): 502-517 2008

Our method:

1. Definition of a local shape criterion
2. Grouping vertices into areas corresponding to one primitive type
3. Computation of the primitive parameters

Local shape definition

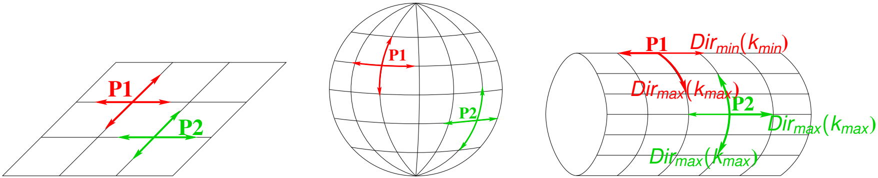
The shape definition uses

Local shape definition

The shape definition uses **Curvature**
$\Rightarrow$ points contained in Plane, Sphere, Cone or Cylinder have specific features:

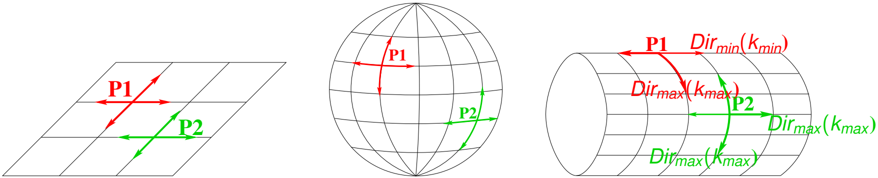
Local shape definition

The shape definition uses **Curvature**
⇒ points contained in Plane, Sphere, Cone or Cylinder have specific features:



Local shape definition

The shape definition uses **Curvature**
⇒ points contained in Plane, Sphere, Cone or Cylinder have specific features:



	k_{min}	k_{max}	$\mathbf{Dir_{min}}$	$\mathbf{Dir_{max}}$
Plane	$= 0$	$= 0$	not defined	not defined
Sphere	$= \frac{1}{Radius}$	$= \frac{1}{Radius}$	not defined	not defined
Cone/Cylinder	$= 0$	$= \frac{1}{Radius}$	$=$ generating line	not used
	$= \frac{1}{Radius}$	$= 0$	not used	$=$ generating line

Curvature computation

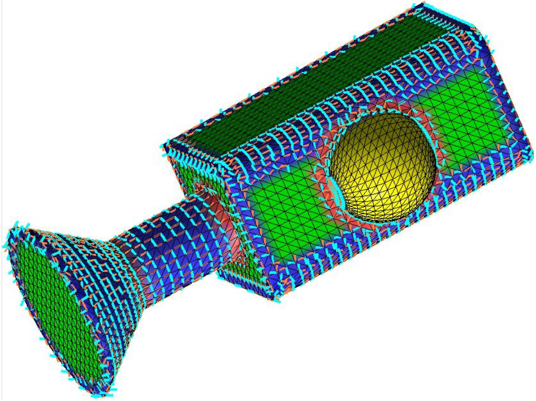
⇒ We choose a method based on **Euler formula**

Dong and Wang
Curvatures estimation on triangular mesh
Journal of Zhejiang University-Science A 6(1): 128-136
2005

Curvature computation

⇒ We choose a method based on **Euler formula**

Dong and Wang
Curvatures estimation on triangular mesh
Journal of Zhejiang University-Science A 6(1): 128-136
2005

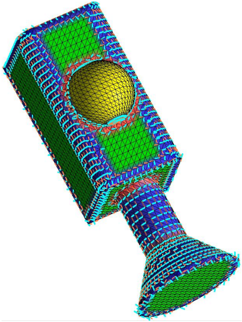


Concave point
Convex point
Plane point
Sphere point
Dir_{max}
Dir_{min}

Neighborhood ring= 1

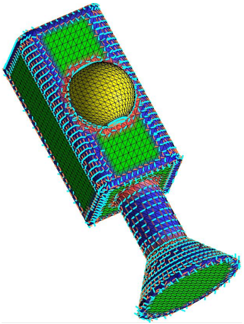
Plane/Sphere Extraction

- Compute k_{min}, k_{max}, $\mathbf{Dir_{min}}$ and $\mathbf{Dir_{max}}$



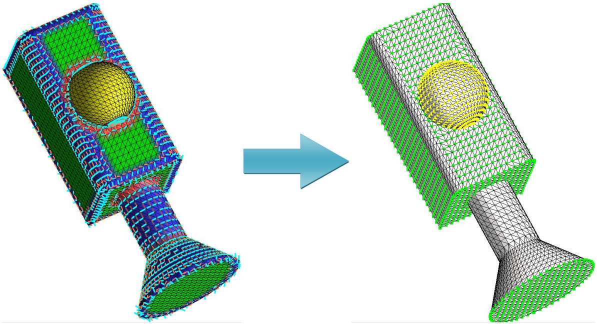
Plane/Sphere Extraction

- Compute k_{min}, k_{max}, $\mathbf{Dir_{min}}$ and $\mathbf{Dir_{max}}$
- Group all adjacent points with $k_{max} = k_{min} \approx k$ (if plane then $k = 0$)



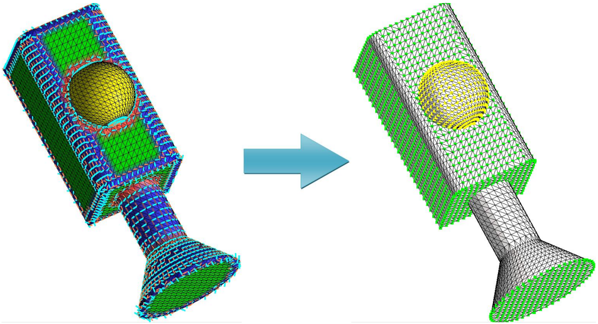
Plane/Sphere Extraction

- Compute k_{min}, k_{max}, $\mathbf{Dir_{min}}$ and $\mathbf{Dir_{max}}$
- Group all adjacent points with $k_{max} = k_{min} \approx k$ (if plane then $k = 0$)



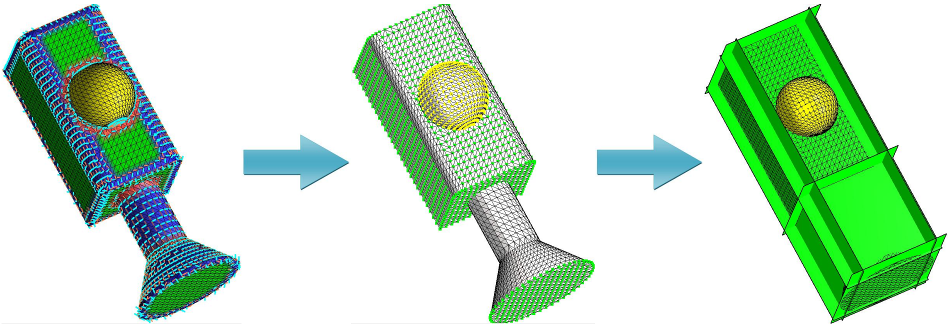
Plane/Sphere Extraction

- Compute k_{min}, k_{max}, $\mathbf{Dir_{min}}$ and $\mathbf{Dir_{max}}$
- Group all adjacent points with $k_{max} = k_{min} \approx k$ (if plane then $k = 0$)
- Approximation by a least square regression with the implicit equations



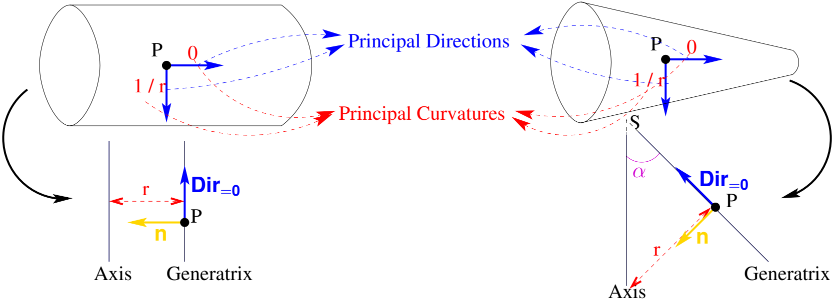
Plane/Sphere Extraction

- Compute k_{min}, k_{max}, $\mathbf{Dir_{min}}$ and $\mathbf{Dir_{max}}$
- Group all adjacent points with $k_{max} = k_{min} \approx k$ (if plane then $k = 0$)
- Approximation by a least square regression with the implicit equations



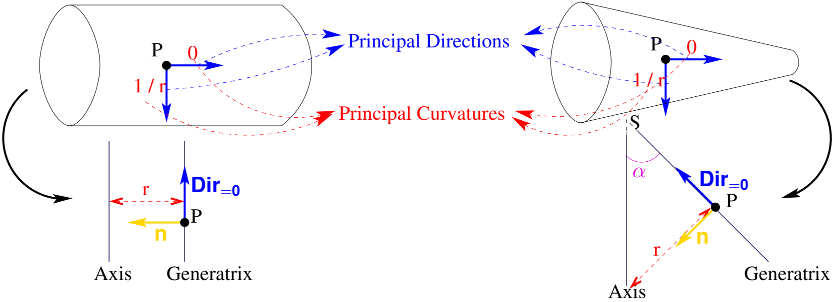
Cone/Cylinder Extraction

General features of cones and cylinders:



Cone/Cylinder Extraction

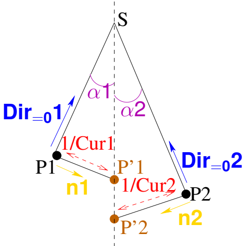
General features of cones and cylinders:



Criterion (C):

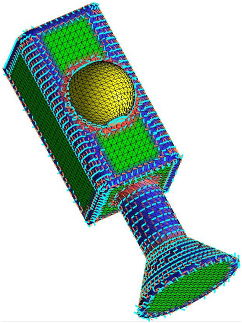
If P_1 and $P_2 \in$ same cone

$\Rightarrow \alpha_1 = \alpha_2$



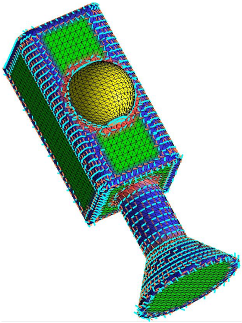
Cone/Cylinder Extraction

- Compute k_{min}, k_{max}, $\mathbf{Dir_{min}}$ and $\mathbf{Dir_{max}}$



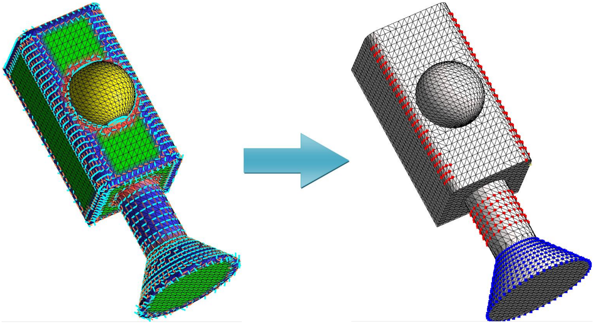
Cone/Cylinder Extraction

- Compute k_{min}, k_{max}, $\mathbf{Dir_{min}}$ and $\mathbf{Dir_{max}}$
- Group adjacent points with ($k_{min} = 0$ & $k_{max} \neq 0$) or ($k_{min} \neq 0$ & $k_{max} = 0$) and the criterion (C)



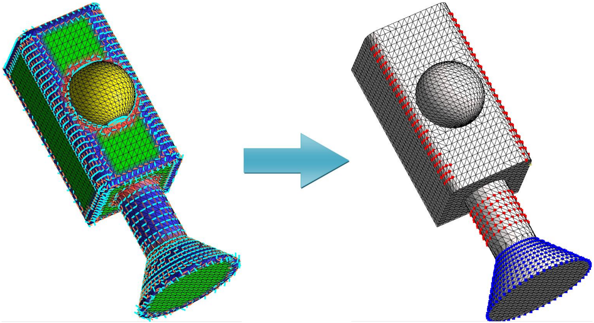
Cone/Cylinder Extraction

- Compute k_{min}, k_{max}, $\mathbf{Dir_{min}}$ and $\mathbf{Dir_{max}}$
- Group adjacent points with ($k_{min} = 0$ & $k_{max} \neq 0$) or ($k_{min} \neq 0$ & $k_{max} = 0$) and the criterion (C)



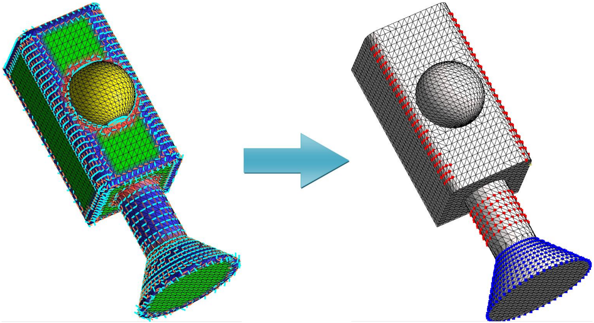
Cone/Cylinder Extraction

- Compute k_{min}, k_{max}, $\mathbf{Dir_{min}}$ and $\mathbf{Dir_{max}}$
- Group adjacent points with ($k_{min} = 0$ & $k_{max} \neq 0$) or ($k_{min} \neq 0$ & $k_{max} = 0$) and the criterion (C)
- By regression on all *pointAxis* (P') $\Rightarrow$ rotation axis



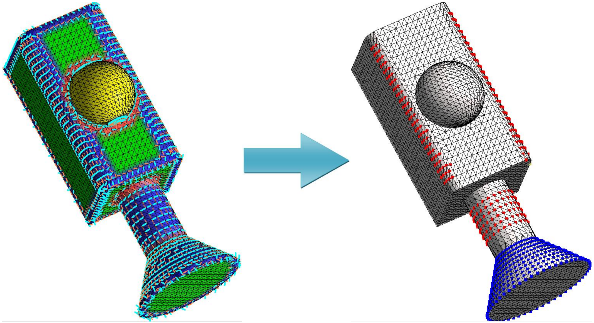
Cone/Cylinder Extraction

- Compute k_{min}, k_{max}, $\mathbf{Dir_{min}}$ and $\mathbf{Dir_{max}}$
- Group adjacent points with ($k_{min} = 0$ & $k_{max} \neq 0$) or ($k_{min} \neq 0$ & $k_{max} = 0$) and the criterion (C)
- By regression on all *pointAxis* (P') $\Rightarrow$ rotation axis
  $\Rightarrow \alpha =$ average of angles between rotation axis and $Dir_{=0}$



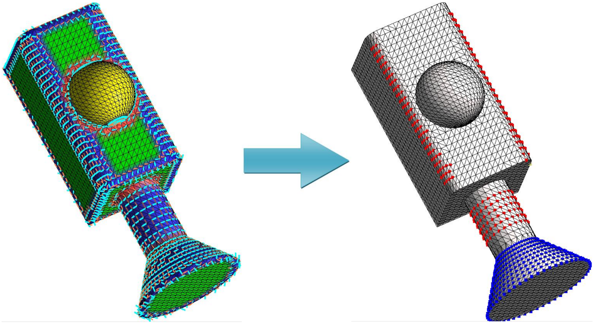
Cone/Cylinder Extraction

- Compute k_{min}, k_{max}, $\mathbf{Dir_{min}}$ and $\mathbf{Dir_{max}}$
- Group adjacent points with ($k_{min} = 0$ & $k_{max} \neq 0$) or ($k_{min} \neq 0$ & $k_{max} = 0$) and the criterion (C)
- By regression on all *pointAxis* (P') $\Rightarrow$ rotation axis $\Rightarrow \alpha =$ average of angles between rotation axis and $Dir_{=0}$
  - $\alpha = \pi \Rightarrow$ Cylinder: Average $\frac{1}{curvature} \Rightarrow$ Radius cylinder



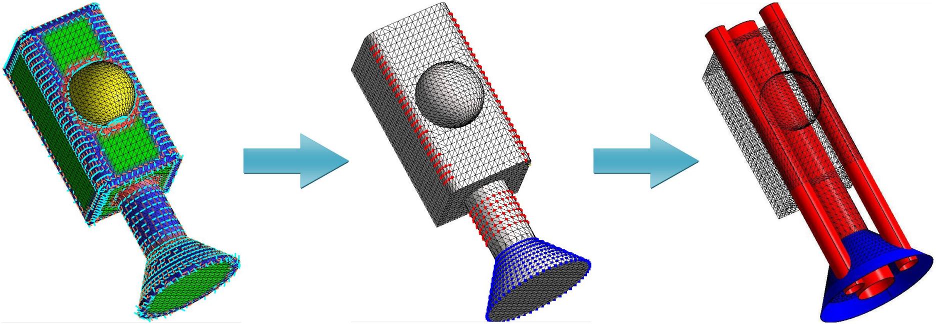
Cone/Cylinder Extraction

- Compute k_{min}, k_{max}, $\mathbf{Dir_{min}}$ and $\mathbf{Dir_{max}}$
- Group adjacent points with ($k_{min} = 0$ & $k_{max} \neq 0$) or ($k_{min} \neq 0$ & $k_{max} = 0$) and the criterion (C)
- By regression on all *pointAxis* (P') $\Rightarrow$ rotation axis $\Rightarrow \alpha =$ average of angles between rotation axis and $Dir_{=0}$
  - $\alpha = \pi \Rightarrow$ Cylinder: Average $\frac{1}{curvature} \Rightarrow$ Radius cylinder
  - $\alpha \neq \pi \Rightarrow$ Cone: $\alpha \Rightarrow$ Angle cone

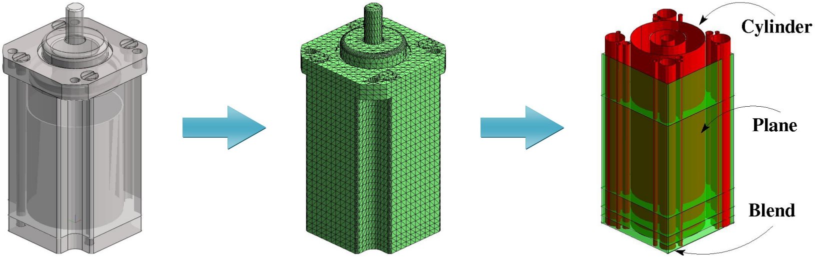


Cone/Cylinder Extraction

- Compute k_{min}, k_{max}, $\mathbf{Dir_{min}}$ and $\mathbf{Dir_{max}}$
- Group adjacent points with ($k_{min} = 0$ & $k_{max} \neq 0$) or ($k_{min} \neq 0$ & $k_{max} = 0$) and the criterion (C)
- By regression on all *pointAxis* (P') $\Rightarrow$ rotation axis $\Rightarrow \alpha =$ average of angles between rotation axis and $Dir_{=0}$
  - $\alpha = \pi \Rightarrow$ Cylinder: Average $\frac{1}{curvature} \Rightarrow$ Radius cylinder
  - $\alpha \neq \pi \Rightarrow$ Cone: $\alpha \Rightarrow$ Angle cone



CAD results: Motor

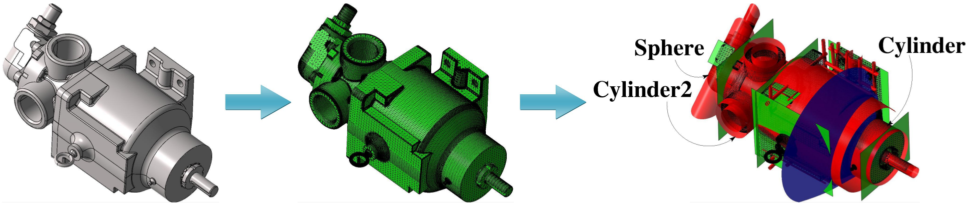


22,482 triangles $\Longrightarrow$ extraction of 26 planes + 25 cylinders

		Original Values	*MotorMesh*
Cylinder	Axis (x;y;z)	0;0;1	0;0;0.999
	Radius	33.5	33.499
Blend	Axis (x;y;z)	0;0;1	0;0;0.999
	Radius	7	7.079
Plane	Coefficients (a;b;c;d)	0;0.024;0;1	0;0.024;0;1

http://shapes.aimatshape.net/viewgroup.php?id=1242

CAD results: Pump



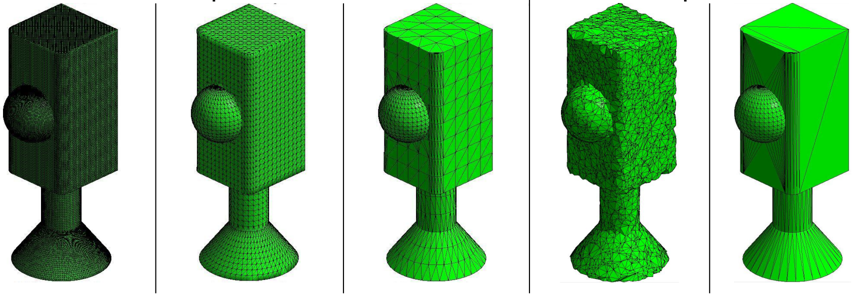
158,746 triangles $\Longrightarrow$ extraction of
9 planes + 1 sphere + 1 cone + 13 cylinders + 31 blends

		Original Values	*PumpMesh*
Cylinder1	Axis (x;y;z)	1;0;0	0.999;-0.011;-0.004
	Radius	49.669	49.669
Cylinder2	Axis (x;y;z)	0;1;0	-0.006;1;0.002
	Radius	34.162	34.175
Sphere	Center (x;y;z)	409.175;367.654;515.722	409.167;367.682;515.780
	Radius	23.114	23.088

http://www.vikingpump.com/en/engineering/3dmodels.html

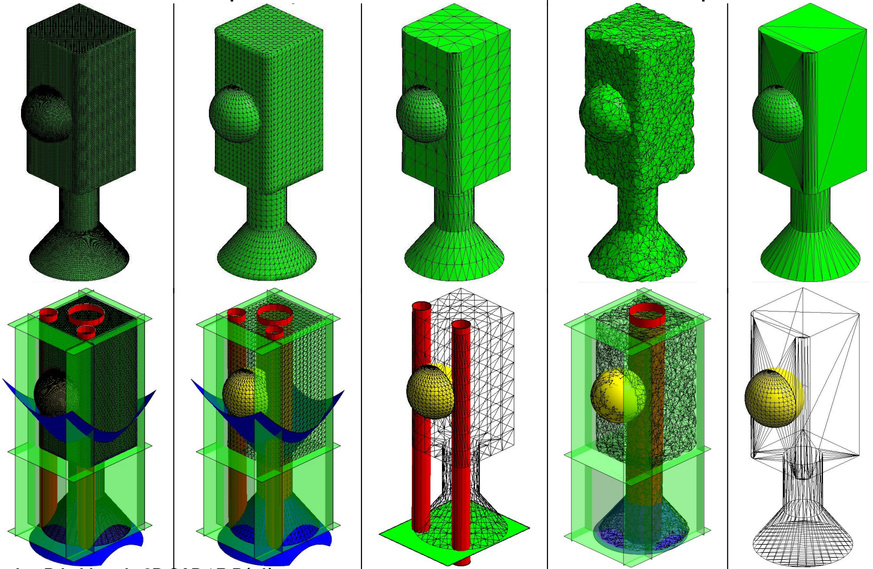
Influence of the discretization

Isotropic Meshes | Anisotropic Meshes



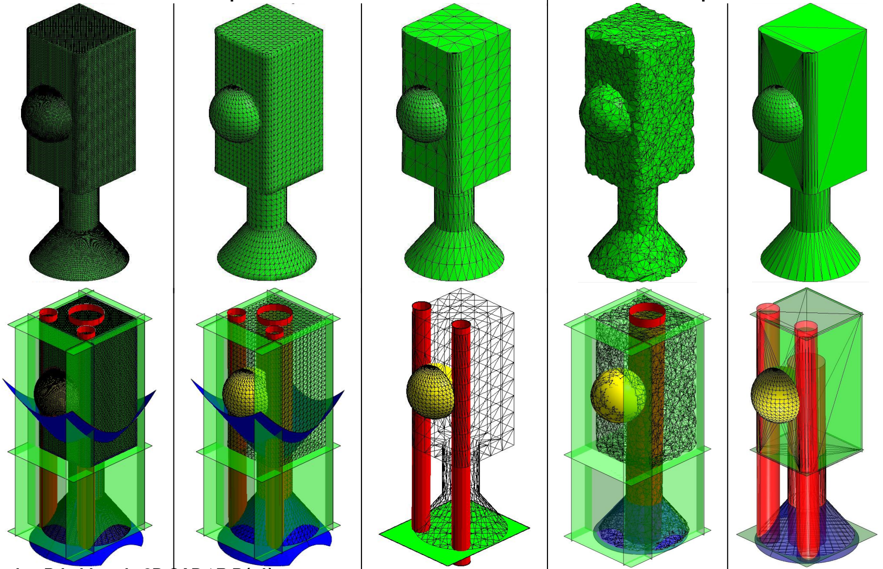
Influence of the discretization

Isotropic Meshes | Anisotropic Meshes

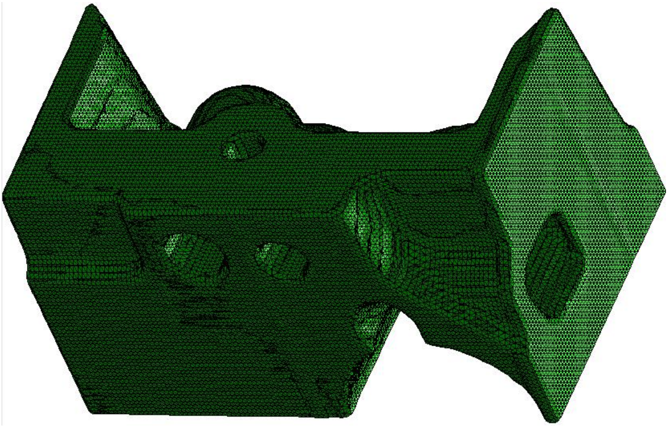


Influence of the discretization

Isotropic Meshes | Anisotropic Meshes

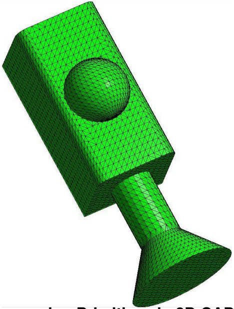


Scanned Mesh



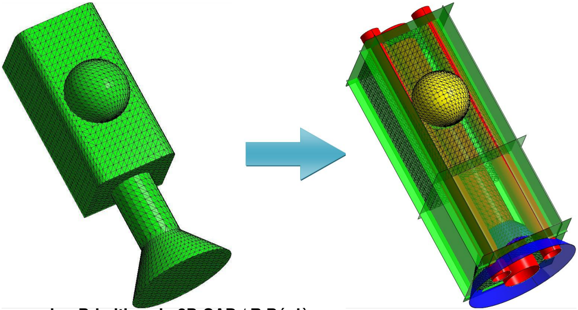
Conclusion

Our method takes a mesh



Conclusion

Our method takes a mesh $\Longrightarrow$ extract geometrical primitives

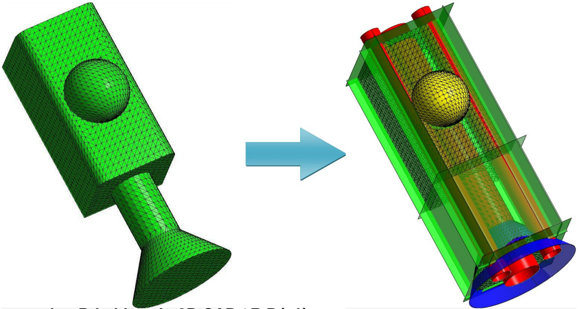


Conclusion

Our method takes a mesh $\Longrightarrow$ extract geometrical primitives

Future Work:

- Improve the cylinder/cone parameter computation (approximation)

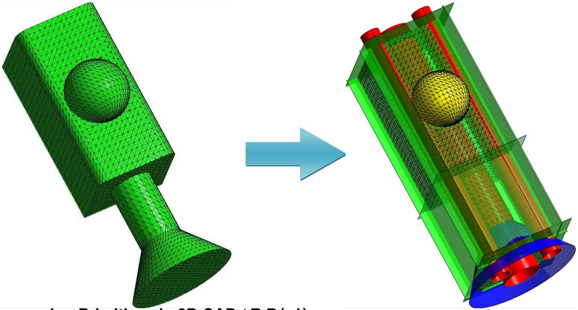


Conclusion

Our method takes a mesh $\Longrightarrow$ extract geometrical primitives

Future Work:

- Improve the cylinder/cone parameter computation (approximation)
- Extract new primitive types (revolution surfaces ...)

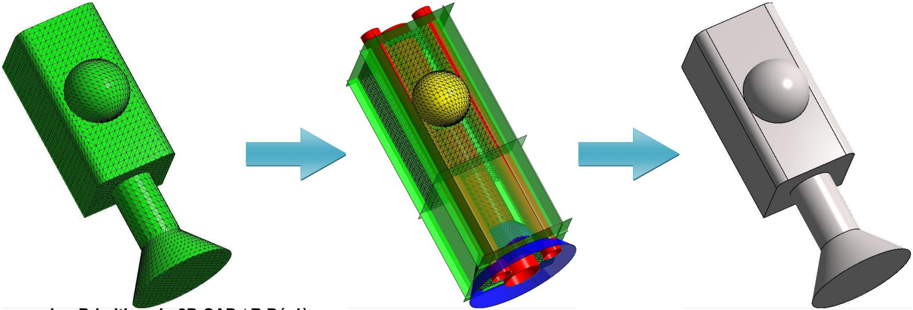


Conclusion

Our method takes a mesh $\Longrightarrow$ extract geometrical primitives

Future Work:

- Improve the cylinder/cone parameter computation (approximation)
- Extract new primitive types (revolution surfaces ...)
- Trim the primitives and reconstruct the object (topology)

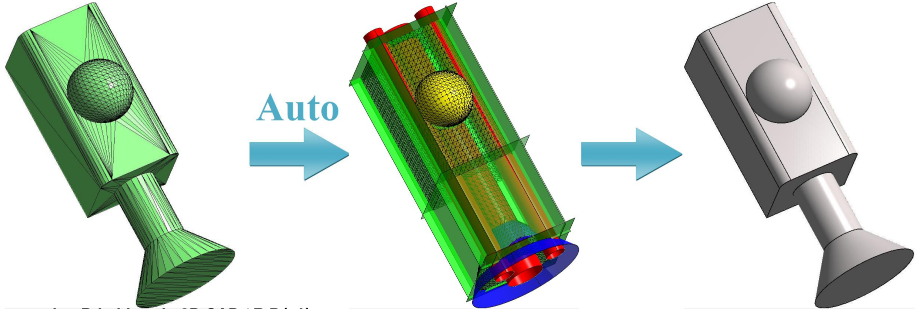


Conclusion

Our method takes a mesh $\Longrightarrow$ extract geometrical primitives

Future Work:

- Improve the cylinder/cone parameter computation (approximation)
- Extract new primitive types (revolution surfaces ...)
- Trim the primitives and reconstruct the object (topology)
- Add an automatic segmentation step (sparse meshes)

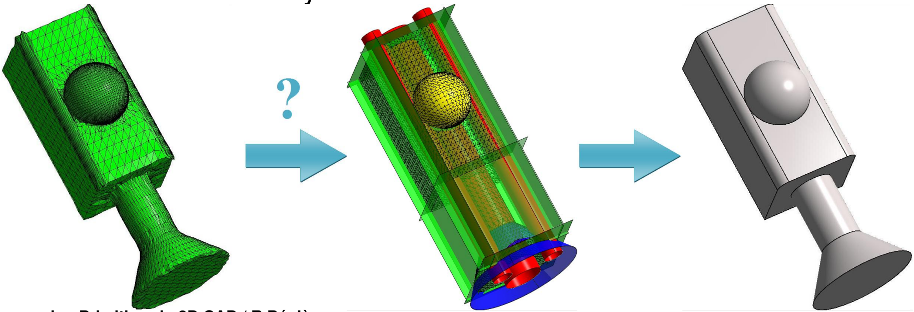


Conclusion

Our method takes a mesh $\Longrightarrow$ extract geometrical primitives

Future Work:

- Improve the cylinder/cone parameter computation (approximation)
- Extract new primitive types (revolution surfaces ...)
- Trim the primitives and reconstruct the object (topology)
- Add an automatic segmentation step (sparse meshes)
- Deal with noisy meshes



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,
Recovering Primitives in 3D CAD meshes, SPIE, *San Francisco*, 2011