Master Informatique, parcours IIS, 2025-2026

UE 4TIS911U "Modélisation et traitement de la géométrie"

Analyse et réparation de maillages 3D

L'objectif de ce travail est de développer des outils pour l'analyse et la réparation de certains défauts d'un maillage qui permet de le transformer dans un maillage imprimable. On pourrait s'inspirer d'outils existants dans *Meshlab* et *Blender*, illustrés sur la Fig. 3.

Exercice 1 *Analyse de la validité topologique et géométrique*

Soit les défauts d'un maillage 3D listés ci-dessous :

1. Présence d'éléments (sommets, arêtes et faces) isolés et/ou dégénérés : Fig.1
2. Sommets et arêtes non-variétés : Fig. 2
3. Auto-intersection et recouvrement d'éléments : Fig. 4b
4. Orientation cohérente des faces : Fig. 8
5. Maillages avec bords et défauts de volume fermé : Fig. 4a, Fig. 5b.

Pour trois des défauts d'un maillage 3D listés (au choix) , proposer et implémenter une méthode pour sa **détection** à partir d'un fichier **OBJ**.

Exercice 2 *Réparation de la validité topologique et géométrique*

Pour les défauts d'un maillage 3D choisis précédemment proposer et implémenter une méthode de **réparation**.

Exercice 3 *Expérimentations*

1. Construire des exemples tests pour les défauts choisis de l'execice 1 et appliquer la recherche et la réparation de ces défauts. Donner une synthèse des résultats obtenus.
2. Choisir un maillage reconstruit à partir de votre acquisition (Tdm 1).
   Appliquer la recherche et la réparation des défauts traités précédemment.
   Donner des illustrations graphiques et incorporer les tableaux récapitulatifs de la recherche et de la réparation de ces défauts avec vos outils.
3. Effectuer les mêmes recherches avec des outils existants de *Meshlab*, *Blender* et *3D Polyhedral Surface*.
   Donner une analyse comparative des résultats.

Modalités de validation des résultats obtenus :

Remise d'une archive avec le **compte rendu** sur l'ensemble des expérimentations effectuées et le **CODE SOURCE** qui comprendra les exemples tests (fichiers.OBJ) et les sources (CMakeLists+fichiers.cpp) de la recherche et la correction des défauts choisis.
Le code source doit être compilable et exécutable sur les machines de CREMI.

Date de remise : **18/12/2025**

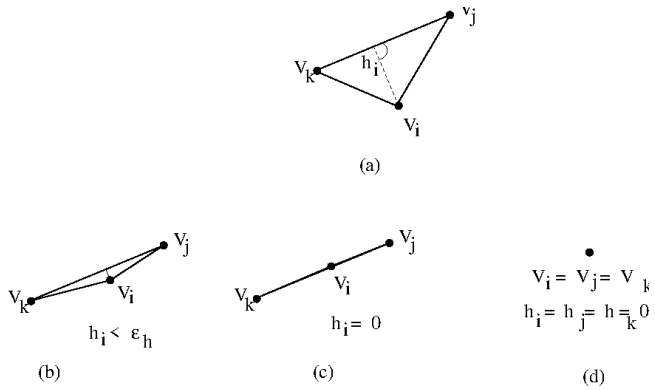
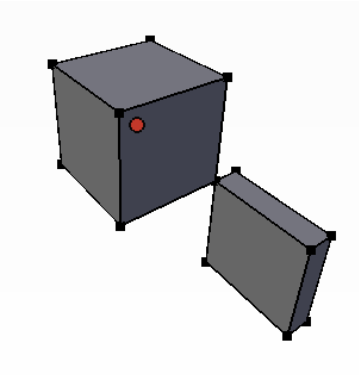
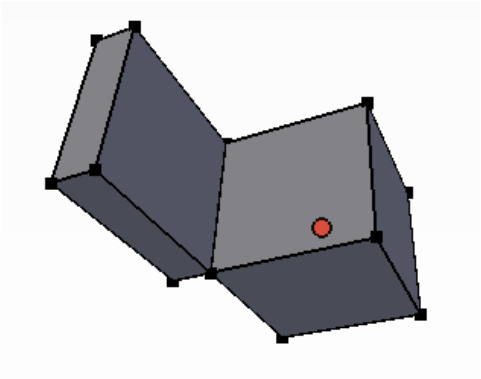


FIGURE 1 – Maillage 3D : (a) Face triangulaire (b) Triangle "aplati" (c) Triangle dégénéré : sommets colinéaires (d) Triangle dégénéré : sommets confondus



(a) Sommet non-variété construit par fusion de sommets



(b) Arête non-variété construite par fusion de sommets et d'arêtes

FIGURE 2 – Conditions de non-variété

Références :

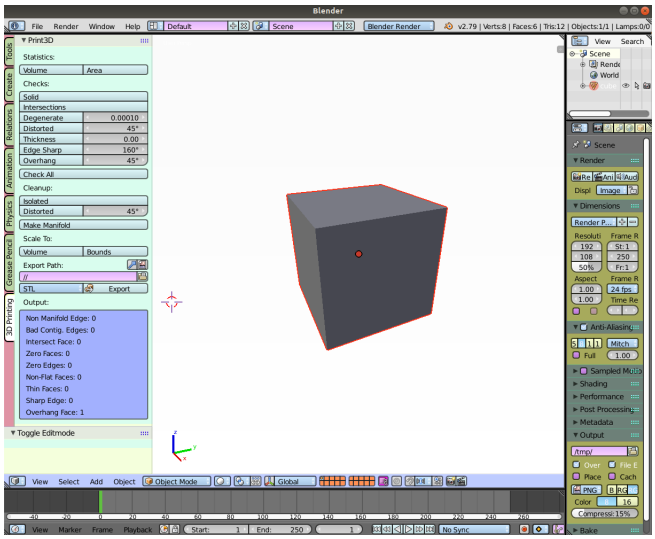
[1] "Geometric and Solid modeling : An Introduction", Christoph M. Hoffman, Morgan Kaufmann Publishers
`https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=1884&context=cstech`

[3] "From 3D models to 3D prints : an overview of the processing pipeline"
Marco Livesu, Stefano Ellero, Jonàs Martínez, Sylvain Lefebvre, Marco Attene
Comput. Graph. Forum 36(2) : 537-564 (2017)
`https://arxiv.org/pdf/1705.03811`

[4] "As-exact-as-possible repair of unprintable STL files", Marco Attene
Journal reference : Rapid Prototyping Journal (2018)
`https://arxiv.org/pdf/1605.07829`

[5] "Computational geometry in C", Joseph O'Rourke, Cambridge University Press, 2nd edition 1998

[6] "Polygons and meshes", Paul Bourke
`https://paulbourke.net/geometry/polygonmesh/`

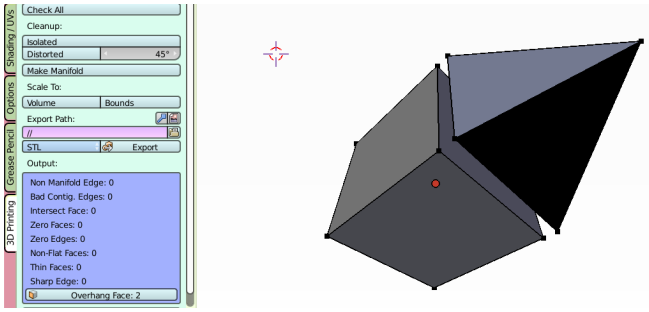


(a) Wavefront OBJ, importation

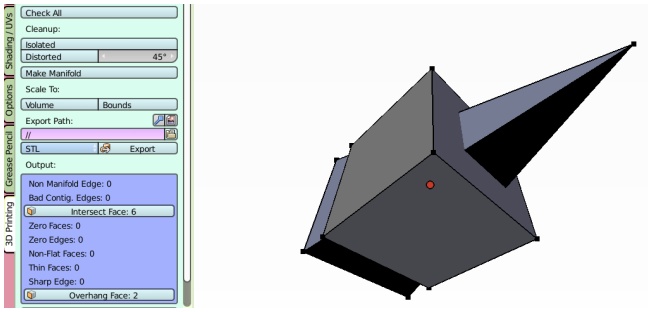
```
# Blender v2.79 (sub 0) OBJ File: ''
# www.blender.org
mtllib blender-cube.mtl
o cube
v -10.000000 -10.000000 -10.000000
v 10.000000 -10.000000 -10.000000
v 10.000000 -10.000000 10.000000
v -10.000000 -10.000000 10.000000
v 10.000000 10.000000 -10.000000
v 10.000000 10.000000 10.000000
v -10.000000 10.000000 -10.000000
v -10.000000 10.000000 10.000000
vn 0.0000 -1.0000 0.0000
vn 1.0000 0.0000 0.0000
vn 0.0000 1.0000 0.0000
vn -1.0000 0.0000 0.0000
vn 0.0000 0.0000 -1.0000
vn 0.0000 0.0000 1.0000
usemtl None
s off
f 1//1 2//1 3//1 4//1
f 2//2 5//2 6//2 3//2
f 5//3 7//3 8//3 6//3
f 7//4 1//4 4//4 8//4
f 7//5 5//5 2//5 1//5
f 4//6 3//6 6//6 8//6
```

(b) Wavefront OBJ, exportation

FIGURE 3 – Blender : analyse d'un maillage 3D construit à l'importation d'un fichier au format Wavefront OBJ



(a) Maillage avec deux composantes et volume non-connexe



(b) Maillage avec deux composantes et auto-intersections

FIGURE 4 – Blender, "3D Printing, Check all"

Annexe A : Rappels sur les fondements des représentations par frontières (BRepr)

Orientation d'une face

Le choix de l'orientation d'une face définit la direction de sa normale $\vec{n}$.

Une face dont le bord est connexe, composé par un contour fermé d'arêtes, est illustrée sur la Fig. 6.

Une face dont le bord n'est pas connexe, composé d'un contour externe englobant l'intérieur de la face et un contour interne englobant un trou, est donnée sur la Fig. 7.

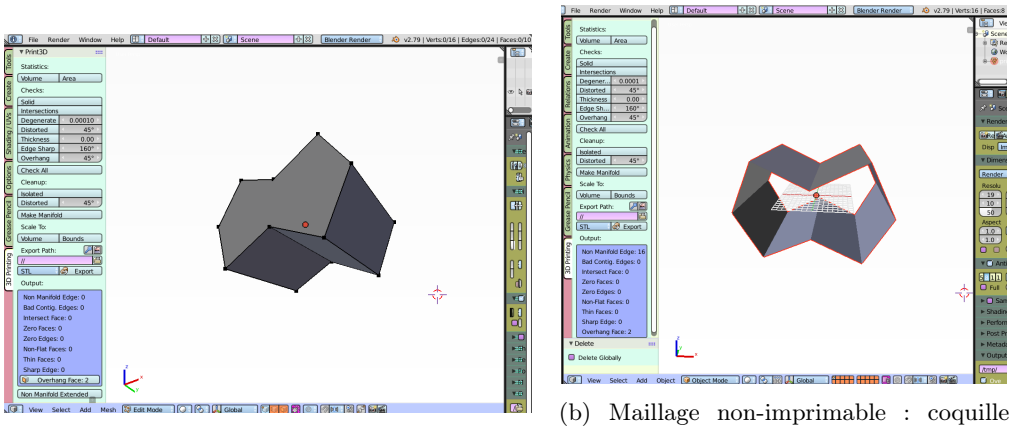
L'orientation d'un contour d'arêtes est choisie de telle sorte qu'en parcourant les arêtes selon cette orientation, l'intérieur de la face est toujours du même côté. Par la suite on fixera le côté sur la "gauche" comme illustré sur les exemples.

Orientation d'un maillage

Un maillage connexe peut être orienté de manière cohérente si chaqu'une de ses arêtes est voisine à au plus deux faces.

Si l'arête est voisine à deux faces, elle doit être incluse avec des directions opposées dans chaqu'une de ces deux faces.

Si l'arête est voisine à une seule face, elle appartient au bord du maillage. Son orientation doit



(a) Maillage imprimable

(b) Maillage non-imprimable : coquille sans épaisseur

FIGURE 5 – Blender : test d'imprimabilité ("Add-on : 3D Printing")

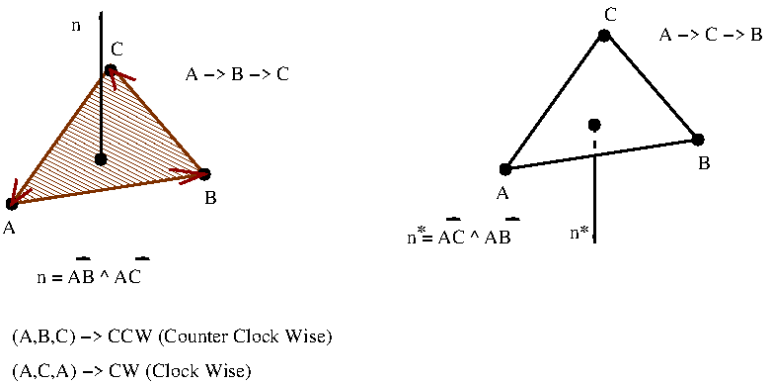


FIGURE 6 – Orientation d'une face de bord connexe

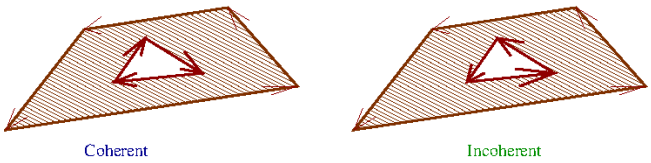


FIGURE 7 – Orientation d'une face de bord non connexe

être telle que quand on parcourt l'arête, l'intérieur de la face est situé à gauche.
Un exemple de maillage orienté de manière cohérente est donné sur la Fig.8(gauche).
Un exemple d'orientation non-cohérente est illustré sur la Fig.8(droite).

Si le maillage est composé par plusieurs composantes connexes, une orientation cohérente peut être définie si l'ensemble des composantes connexes constitue le bord d'un volume fermé. Dans ce cas l'orientation des composantes est choisie de telle sorte que les normales aux faces pointent vers l'extérieur du volume.

Validité topologique

Pour qu'un maillage soit une variété topologique le voisinage d'un point de ce maillage doit être de l'un des trois types ci-dessous selon si :

— Le point est confondu avec un sommet : Fig. 9(gauche)
— Le point est sur une arête : Fig. 12(gauche)
— Le point appartient à l'intérieur d'une face : Fig. 15

Dans le cas contraire on parle de maillage non-variété.

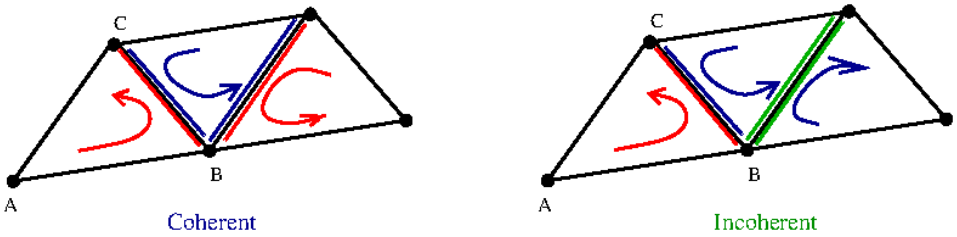


FIGURE 8 – Orientation d'un maillage

Un maillage qui est une variété topologique sans bords définit un maillage imprimable.

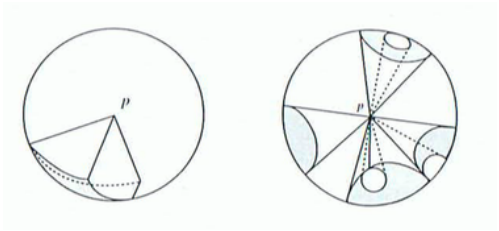


FIGURE 9 – Voisinage d'un sommet appartenant à : (gauche) 2D variété (droite) Non-variété

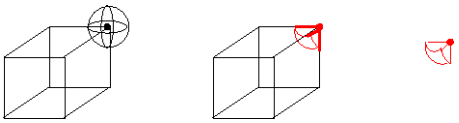


FIGURE 10 – Voisinage d'un sommet appartenant à une 2D variété

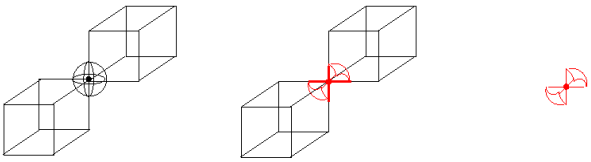


FIGURE 11 – Voisinage d'un sommet appartenant à une "non-variété"

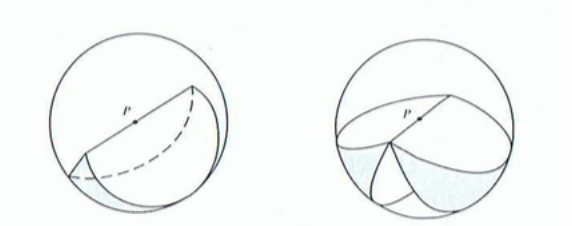


FIGURE 12 – Voisinage d'un point sur une arête appartenant à : (gauche) 2D variété (droite) Non-variété

Pour approfondir on pourrait se référer à [1] et [2].
Pour l'analyse et la réparation des maillages on pourrait utiliser [3] et [4].

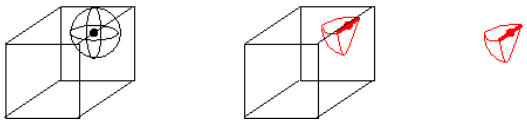


FIGURE 13 – Voisinage d'un point sur une arête appartenant à une 2D variété

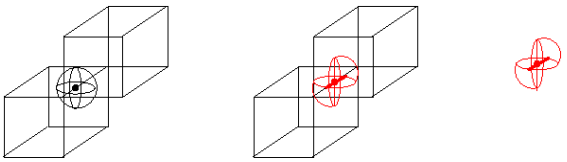


FIGURE 14 – Voisinage d'un point sur une arête appartenant à une "non-variété" 2D variété

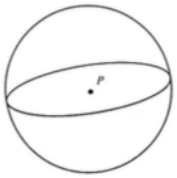


FIGURE 15 – Voisinage d'un point appartenant à une face d'un maillage 2D variété

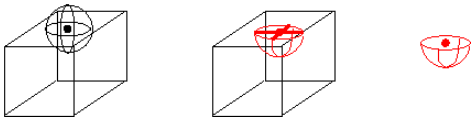


FIGURE 16 – Voisinage d'un point sur une face appartenant à une 2D variété

Annexe B

1. Conditions de **non-variété (NonManifold) (NM)**
   Pour tester si un sommet v est un sommet non-variété, il suffit de tester que l'ensemble de faces voisines à f ne forme pas un cycle fermé, pour un sommet appartenant à l'intérieur du maillage, ou cycle ouvert, pour un sommet appartenant au bord du maillage). Voir algorithme 1.
   Pour tester si une arête e est une arête non-variété il suffit de compter le nombre de faces voisines. Pour les arêtes non-variété ce nombre est supérieur à 2. Voir algorithme 2.
2. **Éléments dégénéres**
   Une arête dégénérée est de longueur 0.
   L'aire d'une face dégénérée est 0.
   Soit f une face définie par la suite de sommets $V_0, V_1, \ldots, V_{n-1}, V_n, V_0 = V_n$.
   Soit O le centre de gravité, $O = \frac{\Sigma_{i=0}^{n-1} V_i}{n}$
   Soit A l'aire de f.
   $$A = \|\Sigma_{i=0}^{n-1} \frac{\vec{OV_i} \times \vec{OV_{i+1}}}{2}\| \quad (1)$$
3. **Intersections et recouvrements**
   De par la définition d'un maillage, deux faces sont soit disjointes soit ont un sommet ou une arête en commun. Ces relations de voisinages sont explicitement représentées dans le modèle BRep.
   Dans le cadre du travail demandé on recherchera des sommets, arêtes et faces dont le

Algorithm 1 Check for NM vertex v

Require:
$F_{adj}(v)$ list of faces adjacent to v
$F_{adj}(f)$ list of faces adjacent to f, $f \in F_{adj}(v)$
Ensure: $True$ if v is NM vertex, $False$ otherwise
for all $f \in F_{adj}(v)$ **do**
mark f as UNUSED
end for
$f \leftarrow f \in F_{adj}(v)$ such that f is marked as UNUSED
if $f \neq$ NULL **then**
repeat
mark f as USED
$f_{nxt} \leftarrow$ NULL
for all $f_{adj} \in F_{adj}(f)$ **do**
if $f_{adj} \in F_{adj}(v)$ such that f_{adj} is marked as UNUSED **then**
$f_{nxt} \leftarrow f_{adj}$
break
end if
end for
$f \leftarrow f_{nxt}$
until $f \neq$ NULL
end if
for all $f \in F_{adj}(v)$ **do**
if f is marked as UNUSED **then**
return True
end if
end for
return False

Algorithm 2 Check for NM edge e

Require: $F_{adj}(e)$ list of faces adjacent to e
Ensure: $True$ if e is NM edge, $False$ otherwise
return len$(F_{adj}(e)) > 2$

plongement géométrique introduit une intersection ou recouvrement. Ainsi pour chaque sommet v on cherchera à vérifier s'il ne coincide pas avec un autre sommet du modèle (**vv**), s'il n'appartient pas à une arête (**ve**) ou à une face (**vf**). Ces tests sont formalisés par les algorithmes 3, 4 et 5. On pourrait s'en inspirer aussi pour écrire les tests d'íntersection et recouvremenet de deux arêtes (**ee**), d'une arête et d'une face (**ef**) et de deux faces (**ff**).
On remarquera que l'ordre d'application des tests suit une dimension incrémentale des éléments testés. Le but étant que pour l'ensemble des voisinages **v{V}, v{E}, v{F}**, **e{E}, e{F}**, et **f{F}** on détecte la présence d'interections et recouvrements dans des éléments du maillage qui ne sont pas codés explicitement par le modèle BRep sous-jacent.

Algorithm 3 Check for vv vertex vertex intersection

Require:
$v \in V$, $w \in V$, $v \neq w$, V the set of all mesh vertices
ϵ precision
Ensure: $True$ if v is multiple vertex , $False$ otherwise
return dist(v,w) $<= \epsilon$

Algorithm 4 Check for ve vertex edge intersection

Require:
$v \in V$, V the set of all mesh vertices
e, $e \in \{E \setminus E_{adj}(v)\}$, E the set of all mesh edges, $E_{adj}(v)$ the set of all edges adjacent to v
ϵ precision
Ensure: $True$ if there existe ve intersection , $False$ otherwise
if $vv(v, start(e), \epsilon) == True$ or $vv(v, end(e), \epsilon) == True$ **then**
return $True$
end if
if v on $\vec{e}$ **then**
$v = e(\vec{t_v})$
$start(e) = e(\vec{t_{start}})$
$end(e) = e(\vec{t_{end}})$
if $t_{start} < t_v < t_{end}$ **then**
return $True$
end if
end if
return $False$

4. **Bords**
Un maillage avec bords ne définit pas un volume fermé. Pour tester si un maillage est sans bords il suffit de tester que chaque arête du maillage appartient aux contours d'exactement deux faces, avec des orientations opposées.

Le calcul des caractéristiques géométriques ci-dessous est simplifié.
Pour approfondir on pourrait se référer à [5].

Algorithm 5 Check for vf vertex face intersection

Require:
$v \in V$, V the set of all mesh vertices
$f \in F \setminus F_{adj}(v)$, F the set of all mesh faces, $F_{adj}(v)$, the set of faces adjacent to v
ϵ precision
Ensure: $True$ if there existe vf intersection , $False$ otherwise
for all $w \in \partial f$ **do**
if $vv(v, w, \epsilon) == True$ **then**
return $True$
end if
end for
for all $e \in \partial f$ **do**
if $ve(v, e, \epsilon) == True$ **then**
return $True$
end if
end for
if v on f **then**
if $v \in f$ **then**
return $True$
end if
end if
return $False$