



ANNÉE UNIVERSITAIRE 2023/2024  
1ÈRE SESSION D'AUTOMNE

**Parcours :** Master mention informatique  
**Épreuve :** M2 Informatique pour l'image et le son  
**Date :** 4TIS904U, Acquisition, reconstruction et modélisation  
**Date :** Lundi 15 Janvier 2024  
**Heure :** 9h  
**Durée :** 45mn  
**Documents :** Non-autorisés  
**Épreuve de :** Mme Gueorguieva, S.

Le barème est donné à titre **indicatif**.

Le sujet comporte 2 pages.

**Exercice 1 (4 pts) Reconstruction**

On considère la reconstruction de surfaces polyédriques (des maillages) à partir de nuages de points en utilisant les deux méthodes expérimentées en cours : la méthode de "Poisson" et la méthode "BPA" ("Ball-Pivoting Algorithm").

Remplir le tableau ci-dessous de sorte que chaque colonne contient les caractéristiques de la méthode respective. Les réponses attendues sont Oui/Non/Optionel.

	"Poisson"	"BPA"
(1) Les normales aux points sont-elles requises ?		
(2) Les données aberrantes (bruitées) sont-elles gérées ?		
(3) L'échantillonnage variable est-il géré ?		
(4) Les sommets de la surface reconstruite font-ils partie du nuage initial ?		
(5) La surface reconstruite est-elle toujours fermée ?		
(6) La surface reconstruite est-elle toujours lisse ?		
(7) La surface reconstruite est-elle toujours "2D-variété" ("manifold") ?		
(8) La surface reconstruite est-elle toujours orientable ?		

**Exercice 2 (4 pts) Représentation**

Soit une surface polyédrique représentée par la structure arête-ailée implémentée dans *HalfedgeDS* de **CGAL**. Donner le pseudo code d'une fonction *faceAdjacentFaces* qui produit la liste des faces voisines d'une face donnée *f*.

Ci-dessous le rappel des références qu'on peut utiliser.

HalfedgeDS

```

1  typedef typename Base::Vertex Vertex
2  typedef typename Base::Halfedge Halfedge
3  Halfedge he;
4  he.prev(); // référence sur la demi-arête qui précède he
5  he.next(); // référence sur la demi-arête qui suit he
6  he.opposite(); // référence sur la demi-arête opposée
7  he.vertex(); // référence sur le sommet incident à he
8  he.face(); // référence sur la face voisine
9  typedef typename Base::Face Face
10 Face f;
11 f.halfedge(); // référence sur une demi-arête qui appartient
12 // à la face et qui pointe vers cette face
13 f.face_begin(); // Circulateur de demi-arêtes autour de la face
14 // (dans le sens inverse des aiguilles d'une montre).

```

**Exercice 3 (6 pts) Optimisation : formule d'Euler-Poincaré**

Soit le maillage illustré sur la Fig. 1(a). On souhaite effectuer les transformations locales :

- (a) → (b) suppression d'arête ("edge collapse")
- (a) → (c) subdivision d'arête ("edge split")

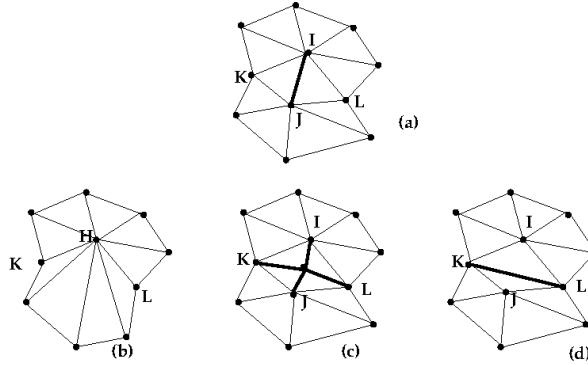


FIGURE 1 – Transformations locales

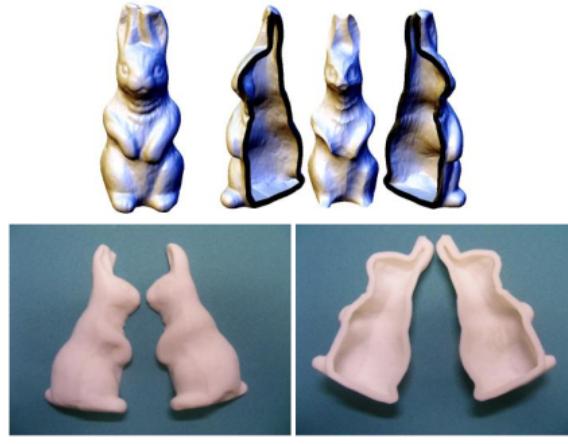


FIGURE 2 – Creuser un objet [Liu et al., 2014, Fig.21]

— (a)  $\rightarrow$  (d) échange d’arêtes (“edge swap”)

Pour chaque transformation donner la suite d’opérateurs d’Euler qui les effectue.

Opérateurs d’Euler à utiliser :

	v	e	f	h	r	s		v	e	f	h	r	s
Mev	1	1	0	0	0	0	Kev	-1	-1	0	0	0	0
Mef	0	1	1	0	0	0	Kef	0	-1	-1	0	0	0
Mvfs	1	0	1	0	0	1	Kvfs	-1	0	-1	0	0	-1
KeMr	0	-1	0	0	1	0	MeKr	0	1	0	0	-1	0
KfMrh	0	0	-1	1	1	0	MfKrh	0	0	1	-1	-1	0

#### Exercice 4 (6 pts) Application à l’impression 3D

On dispose d’une surface polyédrique qui définit le bord (la surface frontière) d’un objet 3D. On souhaite imprimer cet objet de telle sorte que son intérieur soit creux. Cela s’avère essentiel pour la réduction des coûts des matériaux, du temps d’impression et du poids réduit. Une illustration est donnée sur la Fig. 2.

Proposer une chaîne de traitements qui à partir de la surface frontière d’un objet produit le maillage correspondant à la surface frontière de l’objet creux.

Commenter les limites de ces traitements pour assurer l’imprimabilité du résultat.