



# **Compression sans perte de maillages triangulaires adaptée aux applications météorologiques**

**Mémoire**

**Simon Fréchet**

**Maîtrise en génie électrique**  
Maître ès sciences (M. Sc.)

Québec, Canada

© Simon Fréchet, 2017

# **Compression sans perte de maillages triangulaires adaptée aux applications météorologiques**

**Mémoire**

**Simon Fréchet**

Sous la direction de :

Denis Laurendeau, directeur de recherche

## Résumé

La compression est un incontournable lorsque des modèles triangulaires 3D massifs doivent être transmis via un réseau de communication. La compression se doit d'être sans perte lorsque les modèles sont utilisés à des fins métrologiques. Cependant, les modèles capturés par scanners 3D contiennent généralement des artefacts de numérisation tels que la présence de trous dans le maillage, de petits regroupements distincts de triangles sous forme de surfaces ou de volumes ainsi que de singularités *non-manifold* (c.-à-d. un sommet appartenant à deux regroupement de triangles distincts). Ces aberrations rendent les techniques de compression standards inaptes à compresser sans échec le modèle.

Ce mémoire propose une extension à une technique de compression et décompression sans perte des données topologiques nommée *Edgebreaker*. Le remplissage des trous par l'addition d'un sommet, l'insertion de faces triangulaires afin de lier les îlots ainsi que la duplication des sommets *non-manifold* sont proposées comme étapes de prétraitement afin de rendre le modèle compatible avec l'approche standard d'*Edgebreaker*.

Les résultats obtenus démontrent que la solution proposée permet la compression sans perte de modèles hautement bruités à de hauts taux de compression. Les taux de compression résultants obtenus par notre approche se comparent également avec les taux observables pour des modèles sans imperfections compressés par *Edgebreaker*.

# Table des matières

RÉSUMÉ	III
TABLE DES MATIÈRES	IV
LISTE DES TABLEAUX	VI
LISTE DES FIGURES	VII
REMERCIEMENTS	X
AVANT-PROPOS	XI
<b>CHAPITRE 1.</b>	<b>INTRODUCTION ..... 1</b>
1.1.	PRÉSENTATION DE LA PROBLÉMATIQUE .....1
1.2.	MOTIVATION ET OBJECTIFS .....3
1.3.	CHAIRE DE RECHERCHE INDUSTRIELLE CRSNG-CREAFORM SUR LA NUMÉRISATION 3D : CREATION-3D .....4
1.4.	ORGANISATION DU MÉMOIRE .....4
<b>CHAPITRE 2.</b>	<b>ÉTAT DE L'ART SUR LA COMPRESSION DE MODÈLES 3D ..... 5</b>
2.1.	INTRODUCTION .....5
2.2.	VUE GÉNÉRALE SUR LA 3D .....5
2.3.	COMPRESSION ET DÉCOMPRESSION DE MODÈLES 3D .....10
2.4.	FORMATS DE FICHER .....12
2.5.	REVUE DES MÉTHODES DE COMPRESSION DE MODÈLES 3D .....14
2.5.1.	<i>Compression à taux unique (single-rate compression)</i> ..... 15
	<i>Compression des données géométriques</i> ..... 16
	<i>Compression des données de la connectivité</i> ..... 19
	<i>Visite du maillage</i> ..... 19
2.5.2.	<i>Compression progressive (progressive compression)</i> ..... 21
	<i>Compression par Laplacien</i> ..... 22
	<i>Compression par ondelettes</i> ..... 24
2.5.3.	<i>Approche choisie</i> ..... 26
<b>CHAPITRE 3.</b>	<b>DESCRIPTION DE L'ALGORITHME DE COMPRESSION ET DÉCOMPRESSION ..... 27</b>
3.1.	COMPRESSION D'UN MAILLAGE SIMPLE .....27
3.2.	COMPRESSION D'UN MODÈLE AVEC PRÉSENCE DE POIGNÉE .....31
3.3.	DÉCOMPRESSION PAR <i>WRAP&amp;ZIP</i> .....32
3.4.	RESTRICTIONS ET LIMITES DE L'ALGORITHME .....33
3.4.1.	<i>Les trous répartis à travers le maillage</i> ..... 34
3.4.2.	<i>Les points et les arêtes non-manifold</i> ..... 34
3.4.3.	<i>Présence de plusieurs surfaces et volumes indépendants</i> ..... 35
3.5.	CONCLUSION .....36

<b>CHAPITRE 4.</b>	<b>MODIFICATIONS, PRÉTRAITEMENT ET POST-TRAITEMENT DE L'APPROCHE PROPOSÉE..</b>	<b>37</b>
4.1.	INTRODUCTION .....	37
4.2.	OUVERTURE DU FICHIER SOURCE .....	39
4.3.	RÉAJUSTEMENT DES DÉCIMALES PAR QUANTIFICATION .....	39
4.4.	DÉTECTION ET CORRECTION DES POINTS <i>NON-MANIFOLD</i> .....	39
4.5.	ÉVALUATION ET RÉPARATION DES TROUS ET ÎLOTS .....	42
4.5.1.	<i>Repérage des frontières et regroupement des faces</i>	42
4.5.2.	<i>Remplissage des trous</i>	44
4.5.3.	<i>Connexion des îlots</i>	46
4.5.4.	<i>Mise-à-jour automatique des chaînes</i>	48
4.6.	FICHIER TRANSITOIRE : SOMMET OPPOSÉ .....	52
4.7.	CODAGE PAR ENTROPIE .....	53
4.8.	DÉCOMPRESSION .....	54
4.8.1.	<i>Post-traitement à Wrap&amp;Zip</i>	54
4.8.2.	<i>Comparaison des modèles pré et post compression</i>	56
4.9.	CONCLUSION .....	59
<b>CHAPITRE 5.</b>	<b>PRÉSENTATION ET ANALYSE DES RÉSULTATS.....</b>	<b>61</b>
5.1.	INTRODUCTION .....	61
5.2.	PRÉSENTATION DES MODÈLES ÉTUDIÉS.....	61
5.3.	PRÉSENTATION DES RÉSULTATS .....	65
5.3.1.	<i>Taux de compression</i>	65
5.3.2.	<i>Comparaison des résultats obtenus avec notre méthode avec ceux obtenus par l'approche de Zhu</i>	67
5.3.3.	<i>Temps de calcul</i>	69
5.3.4.	<i>Comparaison des modèles</i>	71
5.4.	ANALYSE DES RÉSULTATS ET DISCUSSION .....	72
5.5.	CONCLUSION .....	74
<b>CHAPITRE 6.</b>	<b>CONCLUSION .....</b>	<b>75</b>
<b>BIBLIOGRAPHIE</b>	<b>77</b>	

# Liste des tableaux

TABLEAU 1- LISTE DES OPÉRATEURS DE COMPRESSION ET LEURS FONCTIONS.....	28
TABLEAU 2 - TABLEAU DES DIFFÉRENTES SITUATION D'ORIENTATION DES CHÂÎNES ET DES MODIFICATIONS À APPORTER.....	50
TABLEAU 3 - DICTIONNAIRE D'ATTRIBUTION DES BITS POUR L'ENCODAGE DES OPÉRATEURS CLERS DANS LE FICHIER CLERS.TXT PROPOSÉ PAR ROSSIGNAC ET ZHU .....	54
TABLEAU 4- CARACTÉRISTIQUES ET SPÉCIFICATIONS DES MODÈLES ÉTUDIÉS .....	62
TABLEAU 5 - TAILLE DES MODÈLES AVANT ET APRÈS L'APPLICATION DU PROCESSUS DE COMPRESSION D' <i>EDGEBREAKER</i> AINSI QUE LA CONTRIBUTION DES LOGICIELS DE COMPRESSION WINZIP ET 7-ZIP AVANT ET APRÈS <i>EDGEBREAKER</i> .....	65
TABLEAU 6 - TAILLE INITIALE ET TAUX DE COMPRESSION OBTENUES AVEC LES DIFFÉRENTS OUTILS AINSI QUE DES COMBINAISONS POSSIBLES DE CEUX-CI.....	66
TABLEAU 7 - TABLEAU DE COMPARAISON DES TAUX DE COMPRESSION DES MODÈLES PRÉSENTÉS ET DES MODÈLES DE ZHU .....	68
TABLEAU 8 - TAILLE INITIALE DES MODÈLES ET TEMPS DE CALCUL TOTAUX POUR LA COMPRESSION ET LA DÉCOMPRESSION EN PLUS DU TEMPS DE CALCUL DES 4 PRINCIPALES PHASES DU PROCESSUS DE COMPRESSION.....	69

# Liste des figures

FIGURE 1 - MAILLAGE TRIANGULAIRE D'UN QUARTIER DE PARIS.....	2
FIGURE 2 - A) SYSTÈME D'AXE CARTÉSIEN; B) CHAMP SCALAIRE DE DONNÉES EN SOUFFLERIE .....	6
FIGURE 3 - ÉCHANTILLON D'UNE FACE TRIANGULAIRE DE SOMMET $ACB$ ET DE NORMALE $N$ . L'ORIENTATION DES CÔTÉS EST ILLUSTRÉE PAR LES FLÈCHES BLEUES POUR LA FACE ÉTUDIÉE ET ROUGES POUR LES FACES VOISINES.....	7
FIGURE 4 - A) DEUX TRIANGLES VOISINS PARTAGEANT UNE MÊME ARÊTE; B) FRONTIÈRE AUTOUR D'UN MAILLAGE TRIANGULAIRE. ....	8
FIGURE 5 - UN TORE DE 1 <sup>ER</sup> GENRE.....	9
FIGURE 6 - REPRÉSENTATION D'UN SOMMET NON-MANIFOLD : A) DANS UN MAILLAGE 2D; B) DANS UN MAILLAGE 3D.....	9
FIGURE 7 - MAILLAGE D'UN CUBE CONSTRUIT AVEC 12 FACES TRIANGULAIRES .....	12
FIGURE 8 - FICHER PLY D'UN CUBE (2 TRIANGLES PAR FACE) .....	13
FIGURE 9 - FICHER STL D'UN CUBE (2 TRIANGLES PAR FACE) .....	14
FIGURE 10 - SCHÉMA D'UN SYSTÈME DE COMPRESSION GÉNÉRAL.....	15
FIGURE 11 - ESTIMATION D'UN SOMMET EN SUIVANT LA RÈGLE DU PARALLÉLOGRAMME .....	17
FIGURE 12 - A) CONSTRUCTION DU SYSTÈME DE COORDONNÉES. B) DISTRIBUTION DES SOMMETS DANS LE SYSTÈME DE COORDONNÉES DE L'ESPACE MODÈLE. FIGURE TIRÉE DE [23]. ....	18
FIGURE 13 - CINQ OPÉRATEURS D'IDENTIFICATION DE TRIANGLES. LE MAILLAGE EST CONVERTI EN UNE LISTE D'OPÉRATEURS. L'EXEMPLE ILLUSTRÉ A LA LISTE : CRRRSLCRSERRELRRRCRRRE. FIGURE TIRÉE DE [1]. ....	20
FIGURE 14 - DIFFÉRENTS NIVEAUX DE DÉTAILS SUR UN MODÈLE D'ÉLÉPHANT À MAILLAGE POLYGONAL OBTENU PAR DÉCIMATION DE SOMMETS ET REMAILLAGE DANS LA ZONE LOCALE DÉCIMÉE. FIGURE TIRÉE DE [38]. ....	22
FIGURE 15 - EXEMPLE DE COMPRESSION PAR LAPLACE SUR UNE STRUCTURE SIMPLE .....	23
FIGURE 16 - COMPRESSION PROGRESSIVE PAR ONDELETTES. FIGURE TIRÉE DE [47]. ....	25
FIGURE 17 - EXEMPLE DE COMPRESSION D'UN MAILLAGE TRIANGULAIRE AVEC <i>EDGEBREAKER</i> : A) ÉVALUATION DU TRIANGLE $X$ SELON LA PORTE; B) ATTRIBUTION DE L'OPÉRATEUR $C$ ET L'INDICE 22 EST DONNÉ AU SOMMET NON-VISITÉ (OPPOSÉ À LA PORTE). LA PORTE EST PLACÉE À L'ARÊTE DE DROITE DU TRIANGLE ; C) ATTRIBUTION DE $C$ ET L'INDICE 23 EST DONNÉ AU SOMMET NON-VISITÉ ; D) AJOUT DE $R$ ; E) AJOUT DE $R$ ; F) SÉPARATION DE LA BRANCHE PAR L'OPÉRATEUR $S$ . LA PORTE SE DÉPLACE À DROITE ET L'ARÊTE GAUCHE EST MÉMORISÉE DANS LA LISTE ACTIVE ; G) FIN DE LA BRANCHE (IDENTIFIÉ PAR LES FLÈCHES NOIRES). RECHERCHE DE LA DERNIÈRE ENTRÉE DANS LA LISTE ACTIVE (CELLE IDENTIFIÉE EN FUCHSIA EN F) ET ATTRIBUTION DE LA PORTE SUR L'ARÊTE AU-DESSUS DE LA LISTE. CETTE ENTRÉE DANS LA LISTE ACTIVE EST ENSUITE SUPPRIMÉE. DÉBUT D'UN NOUVEAU CHEMIN (FLÈCHES BLEUES) ; H) SÉPARATION DU CHEMIN ; I) ATTRIBUTION DE $L$ ; J) FIN DE LA BRANCHE ET REPOSITIONNEMENT DE LA PORTE ; K) ATTRIBUTION DE $L$ ; L) LE CHEMIN SE POURSUIT JUSQU'À UNE SÉPARATION DU CHEMIN ; M) PROGRESSION DU CHEMIN JUSQU'À UNE AUTRE SÉPARATION. L'ARÊTE $\{25, 13\}$ EST PLACÉE AU-DESSUS DE LA LISTE ACTIVE. L'ARÊTE $\{24, 17\}$ EST DÉCALÉE DANS L'ORDRE DE LA LISTE ACTIVE ; N) FIN DU CHEMIN DES FLÈCHES VERTES. DÉBUT DU CHEMIN JAUNE ; O) FIN DE LA BRANCHE JAUNE ET RÉCUPÉRATION DE LA NOUVELLE PORTE SELON L'ARÊTE AU-DESSUS DE LA LISTE ACTIVE ; P) FIN DE LA COMPRESSION PAR <i>EDGEBREAKER</i> DU MAILLAGE.....	29

FIGURE 18 - ARBRE DE CONNECTIVITÉ POUR LE MAILLAGE DE LA FIGURE 17 .....	30
FIGURE 19 - ILLUSTRATION DU CHEMIN D' <i>EDGEBREAKER</i> ENTRANT DANS UNE POIGNÉE. L'ARÊTE EN FUCHSIA EST D'ABORD CONSERVÉE DANS LA LISTE ACTIVE MAIS EST ENSUITE RETIRÉE LORSQUE LA TRAVERSE FAIT LE TOUR DE LA POIGNÉE ET ATTRIBUE UN OPÉRATEUR AU TRIANGLE EN BLANC. LES POINTS ROUGES CORRESPONDENT AUX COINS CONSERVÉS DANS LE FICHIER <i>HANDLES</i> . .....	31
FIGURE 20 – REPRÉSENTATION DES OPÉRATEURS POUR LA DÉCOMPRESSION DU MAILLAGE PAR L'ALGORITHME DE DÉCOMPRESSION <i>WRAP&amp;ZIP</i> . LES FLÈCHES BLEUES DONNENT L'ORIENTATION DES CÔTÉS LIBRES N'ÉTANT PAS DES PORTES.....	32
FIGURE 21 – DEUX EXEMPLES DE DÉCOMPRESSION DE MAILLAGE OÙ LE TRIANGLE INFÉRIEUR GAUCHE CORRESPOND AU TRIANGLE DE DÉPART. LES FLÈCHES BLEUES MONTRENT L'ORIENTATION DES CÔTÉS LIBRES. A) SQUELETTE CRÉÉ À PARTIR DE L'HISTORIQUE <i>LSRSELELRE</i> ; B) SQUELETTE CRÉÉ À PARTIR DE L'HISTORIQUE <i>LCRSESESEE</i> ; C) LES ARÊTES DE R ET L NE SONT PAS JOINTES CAR LEUR ORIENTATION NE POINTE PAS À L'OPPOSÉ DU SOMMET PARTAGÉ ; D) LES ARÊTES DE C ET E SONT JOINTES CAR LEUR ORIENTATION POINTE À L'OPPOSÉ DU SOMMET PARTAGÉ. ....	33
FIGURE 22 – POINT <i>NON-MANIFOLD</i> (JAUNE) PARTAGÉ PAR DEUX ENSEMBLES DE TRIANGLES VOISINS (ENSEMBLE ROUGE ET ENSEMBLE VERT) DANS UN MAILLAGE 2D .....	35
FIGURE 23 - ORGANIGRAMME DU PRÉTRAITEMENT ET DE LA COMPRESSION POUR UN MODÈLE .....	38
FIGURE 24 - ORGANIGRAMME POUR LA DÉTECTION ET LA CORRECTION DE POINTS <i>NON-MANIFOLD</i> .....	40
FIGURE 25 - CORRECTION D'UN POINT <i>NON-MANIFOLD</i> DANS UN MAILLAGE À 8 FACES 2D. LES FACES SONT NUMÉROTÉES SELON LES CHIFFRES EN CERCLÉS. A) MAILLAGE D'ORIGINE. B) NOUVEAU MAILLAGE SUITE AUX MODIFICATIONS APPORTÉES (EN ROUGE) .....	41
FIGURE 26 – MODÈLE CONSTITUÉ DE DEUX ÎLOTS DISTINCTS À 14 FACES. DEUX REGROUPEMENTS DE FACES DANS LE MAILLAGE SONT CONSTRUITS À PARTIR DE L'APPROCHE PAR <i>SPANNING TREE</i> . LES DEUX REGROUPEMENTS DE FACES SONT GROUPE 1 = {1, 2, 7, 8, 9, 10, 11, 12, 3} ET GROUPE 2 = {4, 5, 6, 13, 14}. ....	44
FIGURE 27 - REMPLISSAGE DE TROUS PAR L'INCLUSION DE SOMMETS ET DE FACES SUR UNE SURFACE TROUÉE.....	45
FIGURE 28 - DEUX TOPOLOGIES ÉTUDIÉES POUR LA STRUCTURE DES PONTS RELIANT LES ÎLOTS A ET B. LES SOMMETS EN ROUGE ILLUSTRENT L'INFORMATION DEVANT ÊTRE CONSERVÉE. A) PONTS CONSTRUITS PAR LA CRÉATION DE 3 TRIANGLES; B) PONT CONSTRUIT PAR LA CRÉATION DE 2 TRIANGLES. ....	47
FIGURE 29 - CONSTRUCTION PROGRESSIVE D'UN PONT ENTRE LES ÎLOTS A ET B. A) DEUX SOMMETS RAPPROCHÉS SONT PRÉLEVÉS; B) LES VOISINS LE LONG DE LA CHAÎNE DE FRONTIÈRE DE CES SOMMETS SONT PRÉLEVÉS; C) CONSTRUCTION DES FACES ET VÉRIFICATION DES NORMALES AUX TRIANGLES D'ACCROCHAGE.....	48
FIGURE 30 - ILLUSTRATION D'UN CAS DE MISE-À-JOUR DE LA CHAÎNE PRIMAIRE SUITE À LA CONSTRUCTION D'UN PONT ENTRE L'ÎLOT BLANC ET GRIS. LA FLÈCHE BLEUE DÉSIGNE LE CHEMIN ET L'ORIENTATION DÉSIRÉ DE LA NOUVELLE CHAÎNE. ....	49
FIGURE 31 - CONNEXION DE 3 ÎLOTS À UNE FRONTIÈRE. A) MAILLAGE D'ORIGINE. B) CRÉATION DE 3 PONTS OÙ LE PONT 3 EST CONNECTÉ À UN TRIANGLE DU PONT 2. ....	51
FIGURE 32 – STRUCTURE DU FICHIER TRANSITOIRE <i>OV</i> ET ENSEMBLE DE 2 TRIANGLES OÙ LE SOMMET 0 A LE COIN OPPOSÉ 4 ET LE SOMMET 3 A LE COIN OPPOSÉ 0.....	52
FIGURE 33 - A) SUPPRESSION DU SOMMET S AVEC L'INDICE 4. B) MAILLAGE SUIVANT LA SUPPRESSION DU SOMMET 4. ....	56
FIGURE 34 - PROCESSUS DE COMPARAISON DES MODÈLES AVANT LA COMPRESSION ET APRÈS LA DÉCOMPRESSION .....	58

FIGURE 35 - MODÈLES UTILISÉS POUR L'ÉVALUATION DES PERFORMANCES DE L'APPROCHE ÉTUDIÉE : A) OURS; B) BOÎTE DE CONSERVE; C) SURFACE PLANE; D) TÊTE; E) CRÂNE. ....	63
FIGURE 36 - MODÈLES UTILISÉS POUR L'ÉVALUATION DES PERFORMANCES DE L'APPROCHE ÉTUDIÉE: A) ENSEMBLE DE SERVEURS SCANNÉS À L'INTÉRIEUR DU SUPERCALCULATEUR COLOSSE DE L'UNIVERSITÉ LAVAL; B) BÉBÉ; C) BLOCS D'UNE CROIX ET D'UNE MAISON; D) CIBLE DE CALIBRATION. ....	64
FIGURE 37 - GRAPHIQUE DES TAUX DE COMPRESSION DES DIFFÉRENTES COMBINAISONS D'OUTILS EN FONCTION DE LA TAILLE DES MODÈLES .....	67
FIGURE 38 - GRAPHIQUE DES TAUX DE COMPRESSION EN FONCTION DE LA TAILLE DES MODÈLES SELON L'APPROCHE PROPOSÉE ET SELON L'APPROCHE DE ZHU .....	68
FIGURE 39 - GRAPHIQUE DU TEMPS DE CALCUL DE LA COMPRESSION ET DE LA DÉCOMPRESSION EN FONCTION DE LA TAILLE DU MODÈLE ..	70
FIGURE 40 - HISTOGRAMME DU TEMPS ALLOUÉ PAR LES QUATRE PHASES PRINCIPALES DU PROCESSUS DE COMPRESSION .....	70
FIGURE 41 - COMPARAISON DU MODÈLE DE LA SURFACE PLANE AVANT LE PROCESSUS DE COMPRESSION ET APRÈS LA DÉCOMPRESSION. LA FIGURE ILLUSTRE UNIQUEMENT LE PLAN XY DU MODÈLE.....	72

## **Remerciements**

Je tiens à remercier ma famille et mes amis pour m’ avoir supporté et encouragé tout au long de mes études. Merci à Denis Laurendeau, Directeur de maîtrise, pour m’ avoir d’abord offert l’opportunité de m’initier aux défis que pose la vision artificielle puis pour son soutien et suivi lors de mon travail de Maîtrise. Merci également à Chen Xu de m’ avoir fourni un soutien technique et moral lors de mes années études.

# **Avant-Propos**

Ce mémoire se voit être une présentation des travaux de recherche de Simon Fréchet sur la compression sans perte de maillages triangulaires adaptée aux applications métrologiques. Toutes les étapes de modification du maillage sont présentées, décrites et illustrées dans ce travail. Ce travail de recherche a été accompli dans le Laboratoire de Vision et Systèmes Numériques de l'Université Laval (LVSN) de 2014 à 2017.

Simon Fréchet,

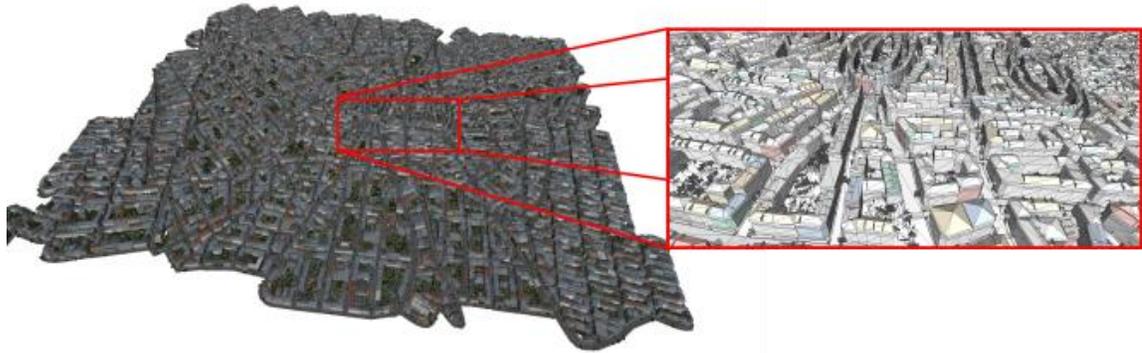
Avril 2017

## **Chapitre 1. Introduction**

### **1.1. Présentation de la problématique**

L'innovation technologique dans le domaine de la représentation bidimensionnelle (2D) et tridimensionnelle (3D) d'objets virtuels s'est développée à un rythme accéléré lors de ces dernières décennies. Le développement dans le domaine informatique rend aujourd'hui efficace la création, la génération, le maniement et la visualisation de modèles 3D. La création de modèles est couramment employée pour la modélisation de pièces mécaniques dans un environnement de développement virtuel mais également dans l'industrie du divertissement tel que dans les jeux-vidéos, le cinéma, des objets et des environnements virtuels. La génération de modèles peut être faite par numérisation, par exemple, via un scanner tenu en main ou par un système multi-caméras. Ces modèles peuvent être ensuite manipulés à la guise de l'opérateur en modifiant des attributs visuels, la géométrie, les couleurs et paramètres intrinsèques. Une hausse de l'intérêt pour l'impression 3D rend également l'utilisation des modèles 3D encore plus intéressante autant pour la réalisation de projets personnels que de l'impression de produits industriels.

Avec l'amélioration constante des performances computationnelles et de la croissance de l'espace de stockage des ordinateurs, il est maintenant possible de manipuler des modèles plus amples et plus massifs avec une meilleure précision et justesse. Que ce soit pour une étude topologique, métrologique ou purement esthétique, une résolution adéquate est impérative mais se limite par la quantité de faces et de points contenus dans un espace défini. La Fig. 1 présente un modèle 3D d'un quartier de Paris sous forme d'un maillage triangulaire.



**Figure 1** - Maillage triangulaire d'un quartier de Paris<sup>1</sup>

Les formats de base d'écriture de données brutes d'un modèle 3D dans un fichier informatique génèrent une taille trop importante pour le transfert via réseau pour les modèles à haute densité. Différentes modifications telle que la compression doivent être apportées au modèle afin de rendre son encryption moins massive pour le système et pour les réseaux de transfert. Il existe différentes techniques de compression cherchant à représenter avec le moins d'octets possible un même modèle.

En métrologie, où la précision et la justesse sont essentielles et impératives pour représenter le plus fidèlement une scène, les standards pour la qualité des résultats doivent être maximaux. Ces modèles servent généralement à faire l'inspection non-destructive de pièces c'est-à-dire de repérer les défauts physiques d'un objet sans altérer son intégrité.

---

<sup>1</sup> Modèle tiré de : <http://tf3dm.com/download-page.php?url=paris-33983>

## 1.2. Motivation et objectifs

L'objectif de ce travail consiste à compresser et décompresser des modèles à maillage 3D triangulaire massif scanné via un scanner tenu en main. À des fins métrologiques, la compression doit être faite sans perte, atteindre un ratio de compression acceptable et doit être polyvalent pour toutes les topologies d'objets. Ces contraintes limitent le choix de la méthode de compression qui sera utilisée dans le cadre de ce travail.

Il existe de nombreux avantages à compresser un modèle 3D destiné à des fins métrologiques :

- Diminuer l'espace physique associé à un modèle dans un appareil de stockage ;
- Augmenter la vitesse de transfert pour partager un modèle via une connexion réseau informatique ;
- Diminuer le coût en bande passante utilisée pour le transfert de données.

Le travail présenté dans ce mémoire est une extension d'une technique de compression de maillage triangulaire sans perte appelée *Edgebreaker* proposée par Jarek Rossignac [1]. Les améliorations apportées cherchent à permettre à l'algorithme d'opérer dans des cas problématiques pour lesquels il n'est pas prévu. Ces cas sont :

- Gérer la présence de trous dans le maillage ;
- Étendre la technique de compression pour des surfaces et volumes isolés ;
- Dupliquer les points singuliers où l'algorithme ne peut procéder régulièrement.

Un mémoire sur un sujet similaire a été présenté par Junjie Zhu [2] en 2013 à *Queen's University* dans lequel il étudie la même problématique avec une approche différente que celle proposée dans ce mémoire. Ses résultats serviront de critères de comparaison entre son approche et celle proposée dans ce travail. Nous démontrerons que les deux approches ont chacune leurs avantages et leurs inconvénients respectifs.

### **1.3. Chaire de recherche industrielle CRSNG-Creaform sur la numérisation 3D : CREATION-3D**

Le présent travail s'inscrit dans le cadre d'une chaire de recherche industrielle CRSNG-Creaform [3]. La Chaire appréhende la numérisation 3D sous un angle holistique et intégrateur en s'attaquant à des problèmes fondamentaux en vision artificielle 3D tout en mettant l'accent sur la mise au point de prototypes expérimentaux pour valider les résultats des recherches et ainsi éprouver des solutions théoriques en les soumettant à des problèmes concrets. Ces recherches s'appuient sur l'expertise acquise au cours des 30 dernières années à l'Université Laval et se font en étroite collaboration avec Creaform, entreprise spécialisée en numérisation 3D.

### **1.4. Organisation du mémoire**

Ce mémoire est structuré en six chapitres. Au premier chapitre, une mise en situation a été faite afin de présenter la problématique, les objectifs et intérêts pour ce travail faisant partie de la programmation de recherche de la Chaire CRSNG-Creaform sur la numérisation 3D. Au chapitre 2, un sommaire de l'état de l'art concernant le vocabulaire utilisé, les différents formats de fichier pour représenter un modèle et les techniques de compression de maillage sera dressé. Au chapitre 3, une description exhaustive de la méthode de compression *Edgebreaker* et de ses variantes et améliorations apportées au fil des années sera présentée. Les contraintes et limites de cette méthode seront également explorées. Les défis que ce travail aura à relever seront également présentés.

Les chapitres suivants présentent et évaluent qualitativement et quantitativement la solution proposée. Le chapitre 4 exposera le corps principal de la méthodologie employée pour résoudre et contourner les contraintes limitant *Edgebreaker* à fonctionner aisément pour des modèles en métrologie. Une évaluation et une discussion des principaux résultats d'intérêt seront réalisées au chapitre 5. Finalement, le chapitre 6 présentera une conclusion générale et des avenues possibles pour de futures recherches.

## Chapitre 2. État de l'art sur la compression de modèles 3D

### 2.1. Introduction

Ce chapitre porte sur la définition du lexique et des concepts employés en compression de modèles 3D. Tout d'abord, le lexique de base est défini, soit les termes employés régulièrement dans le domaine. Ensuite, une présentation des deux formats de fichier les plus populaires de maillage sont présentés. Les formats présentent différemment les données ce qui peut rendre un format de fichier plus lourd qu'un autre format pour un même modèle.

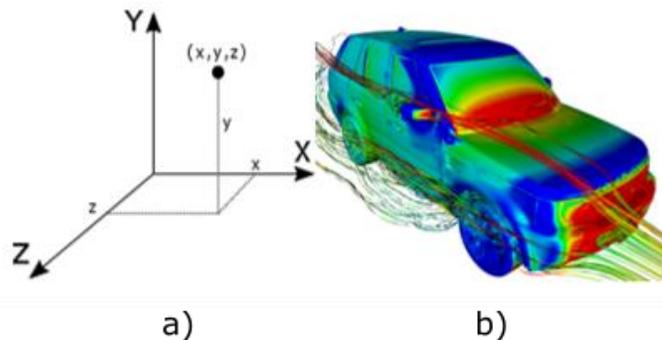
Par la suite, une analyse exhaustive des méthodes de compression et de décompression de modèles 3D est dressée afin d'exposer les raisons qui ont mené au choix de la méthode de compression utilisée dans ce travail. Les particularités découlant de cette méthode seront étudiées davantage dans ce chapitre.

### 2.2. Vue générale sur la 3D

Les représentations tridimensionnelles sont, de nos jours, amplement utilisées dans différents secteurs d'activité industrielle et personnelle. Même si les modèles varient en fonction de l'application, les bases d'une représentation 3D sont définies d'une même manière. Le squelette d'un modèle est constitué d'une série de points et de liaisons entre ceux-ci. Ces points sont connectés entre eux généralement en groupe de trois mais peuvent également être agencés en regroupement de quatre, cinq, etc. Ces regroupements créent des faces qui constituent la topologie de la surface d'un modèle. Une définition plus approfondie et complète est présentée ci-après.

Un sommet (*vertex* en anglais) est une identité sans dimension situé dans un espace. Un sommet  $s_i$  dans un espace 3D possède 3 coordonnées, soit une par base d'un système d'axes X-Y-Z (Fig. 2.a). L'ensemble des sommets  $\{s_1, s_2, \dots, s_n\} \in S$  dans  $\mathbf{R}^3$  composent le modèle. Ces sommets peuvent également contenir des informations complémentaires tels qu'une valeur de température en ce point utilisé dans des simulations de transfert thermique ou une valeur de vitesse

pour des simulations en soufflerie (Fig. 2.b). On dénote également la valence (ou le degré) d'un sommet comme le nombre d'arêtes qui lui sont reliées.



**Figure 2** - a) Système d'axe cartésien; b) Champ scalaire de données en soufflerie<sup>2</sup>

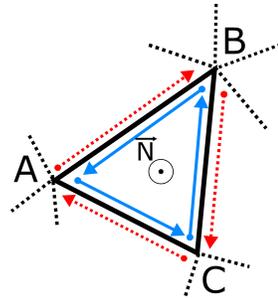
Le deuxième élément majeur d'une représentation 3D est le concept de « face ». Les faces représentent de petits éléments de surface sur un modèle. Un ensemble de faces  $\{f_1, f_2, \dots, f_n\} \in \mathbf{F}$  est nommé un maillage (*mesh* en anglais). Le type de face le plus utilisé est triangulaire car on obtient davantage de faces dans un modèle avec une certaine quantité de sommets donnés qu'avec des faces quadrilatérales. Une quantité plus élevée de faces permet une modélisation plus adéquate et représentative d'une surface. Les faces sont construites à partir des liaisons faites entre les points.

L'enveloppe convexe de deux sommets  $conv\{s_i, s_j\}$  dans un espace  $\mathbf{R}^3$  constitue une arête. Trois points liés forment les trois arêtes d'une face triangulaire et l'ensemble des arêtes  $\{a_1, a_2, \dots, a_n\} \in \mathbf{A}$  constituent la liste des arêtes du modèle. Les liens entre les points peuvent être obtenus de différentes manières telles que par la triangulation de Delaunay (qui impose un angle minimal à l'intérieur d'une face triangulaire) [4], la triangulation aléatoire ordonnée (sans l'intersection d'arêtes) [5] et autres techniques. Les faces définissent donc la connectivité de l'ensemble des points.

---

<sup>2</sup> Figure tirée de : <http://www.prweb.com/releases/cfd/simulation/prweb1891174.htm>

De plus, l'information complémentaire contenue dans les sommets peut être interpolée dans les pixels contenus à l'intérieur des faces, par exemple lors de la colorisation de celles-ci. Le degré d'une face est donné par le nombre d'arêtes qui constituent la face. Un maillage triangulaire aura des faces de degré 3 alors qu'un maillage hexagonal aura des faces de degré 6 et un maillage mixte aura des faces de degré variant de 3, 4, 5, etc. Une liste des sommets et des faces constitue donc la base de tout maillage 3D.



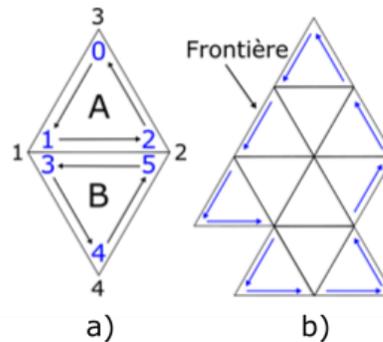
**Figure 3** - Échantillon d'une face triangulaire de sommet ACB et de normale  $\vec{N}$ .

L'orientation des côtés est illustrée par les flèches bleues pour la face étudiée et rouges pour les faces voisines.

Pour chaque face d'un modèle, un vecteur normal  $\vec{N}$  lui est associé. L'origine de ce vecteur est située au centre géométrique défini par les coordonnées  $(x, y, z)$  des coins de la face. Sa norme et orientation sont calculées par produit vectoriel des vecteurs  $\vec{AB}$  et  $\vec{AC}$  (ou des vecteurs  $\vec{BA}$  et  $\vec{BC}$  ou des vecteurs  $\vec{CA}$  et  $\vec{CB}$ ) à la Fig. 3. L'orientation de cette normale est donnée par l'ordre d'énumération des sommets constituant la face. Ces normales peuvent être notamment utilisées pour le calcul des conditions d'illumination et d'autres rendus graphiques d'une scène. Les normales servent également à s'assurer que l'ordre des points constituant les faces est uniforme à travers tout le modèle. Cette information sera pertinente dans la méthode de compression et décompression utilisée dans ce travail. À la Fig. 3, un triangle ACB a une normale  $\vec{N}$  positionnée au centre géométrique du triangle, de norme  $\|\vec{AB} \times \vec{AC}\|$  et d'orientation sortant du plan.

Chaque côté (ou arête) d'un triangle donne l'information quant à l'orientation de la face adjacente. Par exemple, à la Fig. 3, une paire de flèches (rouges et bleues) est attribuée par arête

constituant le triangle ACB. Pour deux triangles adjacents soient régulièrement orientés (c'est-à-dire que leur normale est orientée dans la même direction), il faut que ces vecteurs de demi-arêtes (*half-edge* en anglais) soient antiparallèles.



**Figure 4** - a) Deux triangles voisins partageant une même arête; b) Frontière autour d'un maillage triangulaire.

Les coins (*corner* en anglais) correspondent aux trois sommets présents dans tout triangle. Une liste de coins peut être dressée en débutant par les coins du premier triangle listé dans la liste des faces. La Fig. 4.a présente deux triangles A et B formés par les sommets  $\{1, 2, 3, 4\}$  en noir. Les coins énumérés de 0 à 5 en bleu correspondent à l'indice des coins présents dans le modèle. Chaque triangle contient 3 coins mais chaque sommet peut contenir plus qu'un seul coin comme il est montré pour les sommets 1 et 2 avec les coins  $\{1, 3\}$  et  $\{2, 5\}$  respectivement. On caractérise également un coin opposé à un sommet comme étant le coin opposé au sommet d'une face voisine. À la Fig. 4.a, le coin 0 est opposé au sommet 4 et le coin 4 est opposé au sommet 3.

Une surface se définit comme étant un ensemble de faces ayant au minimum une frontière. Une frontière existe si un triangle ne possède qu'une seule demi-arête tel qu'illustré à la Fig. 4.b. La présence de plusieurs frontières pour un seul regroupement de faces signifie qu'il y a présence de trou(s) dans le maillage.

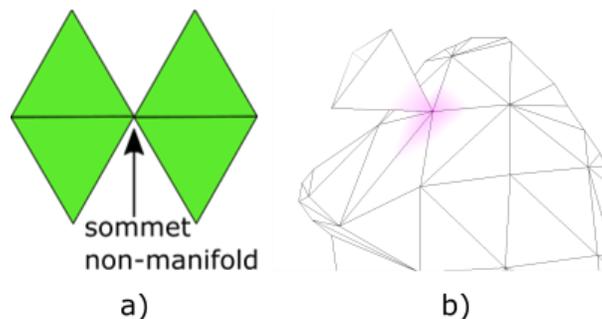
Un volume est une surface fermée où toutes les faces sont voisines d'autres faces. Cependant, un volume peut être homéomorphe à un tore (voir Fig. 5) ce qui indique la présence d'une poignée (*handle* en anglais). La quantité de poignées dans un volume définit le genre (*genus*

en anglais). Ces trous dans le volume se différencient des trous précédemment définis par l'absence de frontière dans la topologie.



**Figure 5** - Un tore de 1<sup>er</sup> genre.

Les points *non-manifold* sont des singularités dans le maillage (voir Fig. 6). Ceux-ci sont observables généralement pour des conditions instables pour l'appareil d'acquisition tels que des surfaces réfléchissantes, des changements brusques de contraste de la couleur sur le modèle scanné, des espacements restreints dont il est difficile de faire l'acquisition, etc. Généralement, ces sommets doivent être traités dans un logiciel de traitement de maillage car ils ne respectent pas les conventions prédéfinies tel qu'avoir des faces toutes orientées dans la même direction.



**Figure 6** - Représentation d'un sommet non-manifold : a) dans un maillage 2D; b) dans un maillage 3D.

### 2.3. Compression et décompression de modèles 3D

À travers les années, les modèles 3D ont évolué de telle manière qu'ils peuvent dépasser la capacité de stockage de la mémoire d'ordinateur lorsqu'on désire les manipuler. Considérant que l'on souhaite améliorer la vitesse de transfert par réseau, diminuer la consommation de bande-passante demandée tout en réduisant la quantité d'espace de stockage demandée pour ces modèles, il faut trouver une solution pour réduire la taille des modèles. Ces objectifs peuvent être atteints en compressant les modèles qui sont trop massifs à l'état brut.

La compression se partage en deux catégories : la compression avec pertes et la compression sans perte.

Il faut d'abord faire la distinction entre les données à l'état brut (*raw data* en anglais) et l'information conceptuelle contenue dans un fichier informatique. Les données à l'état brut donnent toute l'information numérique nécessaire à l'ordinateur pour interpréter le modèle. Dans un modèle 3D typique, il s'agit des coordonnées des différents sommets ainsi que des connexions entre ces sommets. L'information conceptuelle est celle que l'on perçoit en tant qu'utilisateur. L'information perçue est limitée par nos capacités sensorielles (principalement oculaire et auditive lorsque l'on travaille sur un ordinateur) mais également des capacités matérielles de l'équipement utilisé par exemple, la résolution d'un moniteur.

La compression avec perte élimine des traits et paramètres non-essentiels au contenu complet d'un fichier. Ce type de compression est principalement utilisé lors de la compression de fichiers multimédias tels que les images, les vidéos et les fichiers audio. Par exemple, la compression d'images en format JPEG diminue la taille d'une image en supprimant les fréquences jugées non-essentiels à son interprétation par la majorité des utilisateurs (i.e. en tenant compte des facteurs humains). En supprimant ces informations, on réduit la quantité d'octets nécessaires à l'encryption du fichier mais également la qualité de son contenu. Une compression trop importante induit la présence d'imperfections visibles à l'œil (ou à l'oreille pour un fichier audio) d'un utilisateur et sont qualifiées d'artefacts de compression. À des fins de traitement d'images médicales, de métrologie, ou d'autres applications scientifiques, la compression avec pertes ne peut être envisagée car elle modifie irréversiblement les paramètres du fichier.

Une compression avec perte n'induit pas nécessairement une perte d'information perçue. La compression avec perte emploie des méthodes pour diminuer le poids des fichiers principalement en alternant l'information qui ne se perçoit pas ou peu par les sens humains. Un fichier audio contenant des fréquences hors du spectre auditif perceptible à l'oreille humaine, qui se situe environ entre 31 Hz et 17.6 kHz à 60 dB, peut être ignoré sans modifier l'information sonore perçue [6].

Contrairement à la compression avec pertes, la compression sans perte conserve toutes les caractéristiques d'un fichier après la procédure de compression et de décompression. Du côté du traitement en métrologie, toute l'information contenue dans les données doit être conservée afin d'obtenir la meilleure précision possible. C'est donc en n'excluant pas de données imperceptibles aux sens sensoriels des humains qu'il est possible pour des appareils de métrologie de faire des mesures à haute précision. En conservant toute l'information, le taux de compression sans perte demeure généralement bien inférieur au taux d'une compression avec pertes.

Les techniques de compression sans perte les plus communes pour un fichier quelconque sont *Run-length encoding* (RLE), l'ensemble des techniques de compression Lempel-Ziv (qui encode en suivant un système d'appel à un dictionnaire d'échantillons répétés) et le codage par entropie d'Huffman (*Huffman entropy coding*). Certains types de compression comme les compressions de format *.rar*, *.zip* et *.7z* utilisent une combinaison de ces techniques [7].

Différentes définitions du taux de compression sont employées dans la littérature. Le taux de compression utilisé dans ce mémoire se définit comme étant le rapport de la différence du poids du fichier non-compressé  $P_i$  et la taille du fichier compressé  $P_f$  sur la taille initiale du fichier  $P_i$  comme le montre l'équation 2.1.

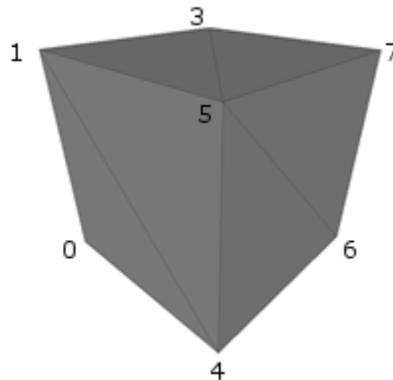
$$R[\%] = \frac{P_i - P_f}{P_i} \cdot 100 \quad (2.1)$$

Une autre méthode de comparaison entre les techniques de compression est le ratio du poids informatique d'un fichier (i.e. la taille du fichier) sur la quantité de sommets (ou parfois de faces) contenus dans le modèle à compresser (communément appelé *bit-rate*) et se mesure en bpv (*bits per vertex*).

## 2.4. Formats de fichier

Différents formats de fichier existent pour représenter un modèle 3D. Ces différents formats se différencient principalement par l'en-tête du fichier donnant les informations spécifiques à un fichier au logiciel dans lequel il sera affiché et traité. Dépendant du type d'extension, des informations complémentaires au modèle peuvent être ajoutées telles que la normale aux faces et aux sommets, la couleur, les conditions d'illumination et de transparence, les arêtes, le positionnement de la caméra, etc. Il existe deux formats populaires pour représenter les informations nécessaires à la construction d'un modèle 3D : le format PLY (*Stanford Triangle format*) et le format STL (*StereoLithography format*). À des fins de visualisation et de simplicité de compréhension, l'usage du format ASCII est prôné dans l'illustration des différences entre les deux formats. L'usage de fichiers en format binaire devrait cependant être favorisé à des fins d'utilisation générale car moins d'octets sont nécessaires à l'encryption.

Dans ce qui suit, la comparaison des méthodes de représentation est faite en analysant la structure ASCII d'un maillage de 12 faces triangulaires du cube illustré à la Fig. 7.



**Figure 7** - Maillage d'un cube construit avec 12 faces triangulaires

Le format PLY donne l'information sur les faces et les sommets de manière indépendante alors que le format STL pour chaque face les coordonnées des sommets [8]. Une structure typique

d'un fichier PLY est présentée à la Fig. 8 en format ASCII en plus de commentaires complémentaires aux lignes principales. L'information complémentaire telle que la liste des normales aux faces ou la liste des arêtes peut être ajoutée à la fin du fichier.

La structure générale d'un fichier PLY est donc : l'entête, les coordonnées (x, y, z) des sommets, les correspondances entre les sommets pour la création des faces et l'information complémentaire.

```
ply
format ascii 1.0
comment VCGLIB generated
element vertex 8
property float x
property float y
property float z
element face 12
property list uchar int vertex_indices
end_header
0.5 0.5 0.5
-0.5 0.5 0.5
0.5 -0.5 0.5
-0.5 -0.5 0.5
0.5 0.5 -0.5
-0.5 0.5 -0.5
0.5 -0.5 -0.5
-0.5 -0.5 -0.5
3 0 1 2
3 3 2 1
3 0 2 4
3 6 4 2
3 0 4 1
3 5 1 4
3 7 5 6
3 4 6 5
3 7 6 3
3 2 3 6
3 7 3 5
3 1 5 3
{format du fichier}
{Affichage ascii/binary}
{Commentaires}
{Définition de "vertex", 8 dans le fichier}
{"vertex" contient la propriété "x" en float}
{"vertex" contient la propriété "y" en float}
{"vertex" contient la propriété "z" en float}
{Définition de "face", 12 dans le fichier}
{"vertex_indices" est une liste d'entiers sans signe}
{Fin de l'en-tête}
{Liste des coordonnées des sommets
"vertex" en format x y z}
{Début de la liste des faces, face à 3 côtés}
```

**Figure 8** - Fichier PLY d'un cube (2 triangles par face)

Le deuxième format de représentation de données se fait par l'indépendance des sommets par rapport aux faces. Ce type de format (en plus du format VRML «vermal», extension .wrl et du format « Additive Manufacturing File Format », extension .amf) est principalement utilisé dans le domaine de l'impression 3D [9]. Puisque le format STL est hautement utilisé en impression 3D, des méthodes de compensation pour l'erreur induite lors de la coupe des faces par les procédés d'impression par couches ont été développées spécifiquement pour les fichiers STL [10].

L'énumération de l'information se fait par une liste de faces. Pour chaque face du modèle, les caractéristiques sont données en commençant par les composantes de la normale à la face suivie

des coordonnées x, y et z des sommets composant les facettes. La note {...}<sup>+</sup> signifie que la structure se répète pour les 9 autres faces du modèle de cube.

solid vcg	{Nom de l'objet, vcg}
facet normal 0.000000e+000 0.000000e+000 1.000000e+000	{Début de l'information lié à la première face et normale de la première face}
outer loop	{Début de l'information lié à la première face}
vertex 5.000000e-001 5.000000e-001 5.000000e-001	{Coordonnées "x y z" du premier point du triangle}
vertex -5.000000e-001 5.000000e-001 5.000000e-001	{Coordonnées "x y z" du deuxième point du triangle}
vertex 5.000000e-001 -5.000000e-001 5.000000e-001	{Coordonnées "x y z" du troisième point du triangle}
endloop	{Fin de l'information lié à la première face}
endfacet	
facet normal 0.000000e+000 0.000000e+000 1.000000e+000	{Face 2}
outer loop	
vertex -5.000000e-001 -5.000000e-001 5.000000e-001	
vertex 5.000000e-001 -5.000000e-001 5.000000e-001	
vertex -5.000000e-001 5.000000e-001 5.000000e-001	
endloop	
endfacet	
facet normal 1.000000e+000 0.000000e+000 0.000000e+000	{Face 3}
outer loop	
vertex 5.000000e-001 5.000000e-001 5.000000e-001	
vertex 5.000000e-001 -5.000000e-001 5.000000e-001	
vertex 5.000000e-001 5.000000e-001 -5.000000e-001	
endloop	
endfacet	
{...} <sup>+</sup>	{Répétition des faces}
endsolid vcg	{Fin du fichier}

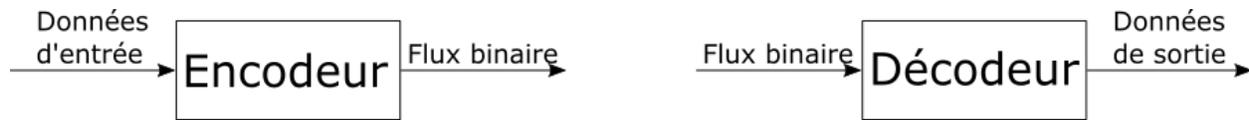
**Figure 9** - Fichier STL d'un cube (2 triangles par face)

En raison de l'information additionnelle et de la redondance des coordonnées des sommets, le fichier en format STL pèsent près de 3.16 kilobytes (KB) contrairement au même fichier en format PLY qui lui en pèse 404 bytes. Bien que le fichier STL donne plus d'information supplémentaire sur la normale pour chaque face directement, l'extraction de normales peut se faire aisément avec le format PLY grâce au produit vectoriel. Puisqu'une grande quantité de l'espace mémoire peut être économisée en représentant un modèle directement en format PLY, dans ce travail et dans la littérature, le calcul des taux de compression est basé sur l'espace alloué pour le stockage du format PLY. Si le modèle est voué à l'impression 3D, le format PLY peut être aisément convertit en format STL.

## 2.5. Revue des méthodes de compression de modèles 3D

Les systèmes de compression, voir Fig. 10, sont constitués d'un encodeur et d'un décodeur. L'encodeur convertit une série de données en un flux binaire compressé. Ce flux binaire peut ensuite être transmis ou entreposé selon le besoin. La décompression s'effectue avec un décodeur

qui effectue le processus inverse de l'encodeur et qui permet de retrouver une série de données identiques ou modifiées des données initiales.



**Figure 10** - Schéma d'un système de compression général

Il existe des techniques pour diminuer la taille des modèles comme la simplification ou le remaillage d'un modèle par suppression de faces, d'arêtes ou de sommets. Ces méthodes permettent la conversion d'un maillage complexe ou irrégulier en un maillage régulier, semi-régulier ou partiellement régulier. Le maillage d'origine peut ensuite être retrouvé par interpolation à partir des sommets et des connexions résultantes dépendant de la technique. Plusieurs approches ont été proposées pour le remaillage [11] [12] [13] [14] mais celles-ci sont souvent complexes et difficiles d'implémentation [15].

La compression peut se faire sur deux entités : les données géométriques et les données de la connectivité. Un compresseur peut exister pour la compression de l'information géométrique et un second pour l'information de la connectivité. Certains compresseurs vont compresser les deux simultanément. Les prochaines sous-sections présentent une revue des différentes méthodes selon deux approches différentes : la compression à taux unique et la compression progressive.

### **2.5.1. Compression à taux unique (single-rate compression)**

La compression à taux unique convertit l'ensemble d'un modèle en un fichier (ou un ensemble de fichiers) binaire. Une fois encodé, les fichiers ne peuvent reconstruire le modèle initial à moins d'être reçus par le décodeur approprié qui vient convertir le flux binaire afin de reconstruire le modèle d'origine avec ou sans perte.

### Compression des données géométriques

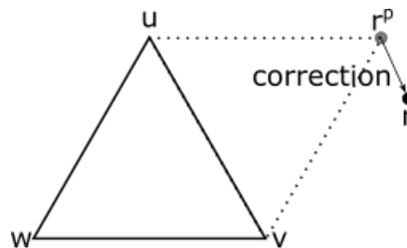
L'information géométrique est composée des composantes (x, y, z) des sommets en notation point flottant et de l'information supplémentaire quant aux normales et à la couleur. L'information géométrique est compressée par trois méthodes qui sont généralement complémentaires.

- i. En raison des limites perceptuelles de l'œil humain, la quantification (*quantization* en anglais) est couramment utilisée pour réduire la précision d'une donnée ou d'un signal analogique vers numérique. La quantification est une méthode couramment utilisée pour la compression de fichier audio et d'images. Dans le cas d'un modèle 3D, il s'agit d'ignorer les décimales non-significatives des coordonnées d'un sommet. Cette troncature dans la précision induit une perte d'information numérique mais pas nécessairement une perte d'information visuelle. Pour des applications métrologiques, il est important de conserver la précision donnée par l'appareil d'acquisition. Cependant, la quantification peut être apportée sur des valeurs en point flottant en autant que l'erreur induite par la troncature soit inférieure à l'erreur de mesure générée par l'appareil d'acquisition. Abderrahim et al. [16] ont proposé une quantification variante pour compresser l'information géométrique d'un modèle. Le niveau de quantification varie pour chaque sommet d'un modèle en quantifiant plus agressivement les sommets près les uns aux autres que les sommets seuls dans leur entourage.
- ii. L'*Entropy coding* existe sous différentes formes. Possiblement la plus populaire et la plus utilisée est le codage d'Huffman mais également le codage arithmétique (*context-based arithmetic coding*) [17]. Le codage d'Huffman attribue un nombre minimal de bits au caractère qui est le plus présent dans le document à compresser puis augmente progressivement en fonction de la redondance des caractères. L'entropie d'une séquence de **n** symboles est :

$$E(i) = -\sum_n (p_i \log_2(p_i)) \quad (2.2)$$

où le  $i^{\text{eme}}$  symbole survient avec une probabilité de  $p_i$ . Plus la valeur de l'entropie est faible, meilleure sera le taux de compression. Contrairement à l'encodage d'Huffman, le codage arithmétique encode l'entièreté d'un fichier en un seul code binaire. Les deux méthodes utilisent donc la redondance de l'information contenue dans un fichier pour créer une interprétation binaire compacte pour une compression optimale et sans perte des données [18].

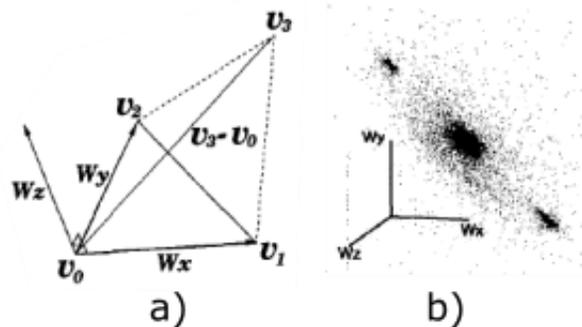
- iii. Une autre méthode qui a été introduite par Touma et Gotsman [19], puis généralisée pour des maillages non-triangulaires et plus tard optimisée par Isenburg et al. [18] [20], est de prédire la position d'un sommet à partir des coordonnées géométriques d'un triangle voisin avec la règle du parallélogramme. Supposant un triangle de sommet  $(u, v, w)$  voisin à un autre triangle partageant l'arête  $(u, v)$ , la position du sommet du triangle voisin peut être approximée par l'équation :  $r^p = v + u - w$  comme le montre la Fig. 11. L'erreur d'estimation est quantifiée et conservée. L'angle entre les deux triangles doit également être pris en compte pour un estimé plus proche de la valeur réelle car deux triangles voisins ne sont pas toujours coplanaires. Une approche proposée par Lee et al. [21] est de quantifier non-uniformément l'angle entre deux faces voisines après la prédiction en accordant davantage de précision aux angles dièdres entre deux faces et moins de précision aux angles internes des triangles. L'erreur de prédiction (qui est au voisinage de zéro et varie moins que la position des sommets) est ensuite encodée par *entropy coding* et peut être réutilisée pour retrouver les coordonnées d'origine d'un sommet.



**Figure 11** - Estimation d'un sommet en suivant la règle du parallélogramme

Bien qu'en théorie cette dernière technique semble être prometteuse, la prédiction ne peut être efficace que pour les modèles où la position des sommets est organisée et généralement uniforme à travers tous le modèle. Les maillages irréguliers doivent être irréversiblement modifiés afin que la prédiction se fasse efficacement en produisant un maillage partiellement régulier par segmentation des zones planes d'un modèle [22] ou en appliquant des techniques de remaillage [11] [12] [13] [14]. Pour des modèles destinés à l'analyse métrologique, la présence de valeurs aberrantes (*outliers* en anglais) et de singularités comme des points *non-manifold* peuvent générer des sections à maillage irrégulier.

Lee et al. [23] ont proposé un concept de compression de maillage triangulaire exclusivement basé sur l'information géométrique contenue dans le modèle en créant un nouveau système de coordonnées. Comme l'illustre la Fig. 12.a, deux des axes de ce système ( $w_x$  et  $w_y$ ) sont déterminés de manière progressive en fonction de la topologie d'un triangle (angle interne et longueur des arêtes) de sommets  $\{v_0, v_1, v_2\}$  alors que le troisième axe ( $w_z$ ) est obtenu par produit vectoriel normalisé. Un sommet  $v_3$  appartenant à un triangle voisin de  $\{v_0, v_1, v_2\}$  est représenté alors par une combinaison linéaire  $v_3 - v_0 = \alpha w_x + \beta w_y + \gamma w_z$ .



**Figure 12** - a) Construction du système de coordonnées. b) Distribution des sommets dans le système de coordonnées de l'espace modèle. Figure tirée de [23].

Lee et al. [23] ont observé une tendance des points transformés à se restreindre à un regroupement serré. Une méthode de quantification vectorielle permet de construire de petits vecteurs de reconstruction. Comme il s'agit d'une méthode de quantification, une perte irréversible de données à lieu et rend la compression avec pertes.

L'information géométrique d'un modèle est rarement constituée uniquement des coordonnées des sommets mais également de l'information sur la couleur et sur les normales. Ces deux derniers sont moins importants et peuvent alors être quantifiés plus agressivement en allouant moins de bits pour leur encodage [24].

### **Compression des données de la connectivité**

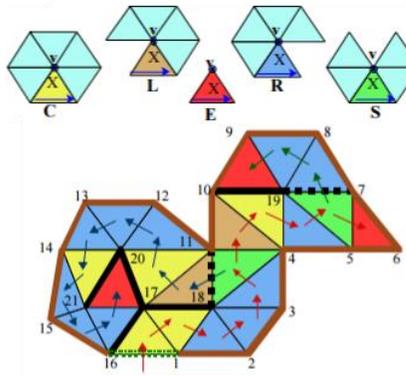
La compression de la connectivité sert à compresser l'information quant aux interactions entre les sommets et les faces. Les différentes méthodes présentées sont toutes sans perte et proposent généralement d'excellents taux de compression en dessous de 4 bits par sommet en raison des nombreux avancements et recherches dans ce domaine. Les approches qui se concentrent sur les maillages non-triangulaires [25] [26] ne seront pas étudiées dans le cadre de ce travail car pour des applications métrologiques, les maillages triangulaires offrent la meilleure résolution.

Alliez et al. [27] propose une approche inspirée de Trouma et Gotsman [19] dans laquelle un maillage triangulaire est compressé en se basant sur l'information de la valence des sommets. Un sommet est d'abord choisi et est ajouté à une liste active. Dans un ordre antihoraire, l'algorithme va encoder la valence des sommets voisins au point étudié. Lorsque la valence de tous les voisins d'un point a été encodée, le sommet dans la liste active est remplacé par le dernier sommet étudié. Un désavantage de cette méthode est que le ratio de bits par sommets est affecté par la distribution des sommets. Selon leur évaluation, les sommets doivent avoir en moyenne 6 valences afin d'obtenir un bon taux de compression. Puisque la méthode dépend hautement de la topologie du modèle, l'approche proposée n'est pas appropriée à la compression de maillage à des fins métrologiques.

### **Visite du maillage**

L'approche la plus employée en termes de compression de la connectivité est la traverse de maillage par incrément à travers les faces de la topologie.

Rossignac [1] fut le pionnier en matière de compression à taux unique sans perte de la connexion d'un modèle 3D à maillage triangulaire avec son algorithme qui fait une visite du maillage : *Edgebreaker*. L'idée générale d'*Edgebreaker* consiste à construire un chemin à travers les triangles et d'attribuer un opérateur spécifique pour chaque face visitée. Plutôt que de représenter la connexion via trois indices de sommets comme le fait le format PLY, *Edgebreaker* va représenter par un des cinq opérateurs (**C-L-E-R-S**) l'ordre des faces rencontrées tel qu'illustré à la Fig. 13. Les lettres sont attribuées selon que le sommet **v** a été visité. S'il a été visité, on évalue quel triangle du voisinage du triangle évalué l'a visité précédemment. Une fois l'opérateur attribué, on évalue le triangle voisin de droite pour les cas **C**, **L** et **S** et à la gauche pour le cas **R**. Le cas **E** signifie une fin de parcours. Le cas **S** divise le chemin actif en 2 parcours et conserve le chemin inactif en mémoire. Une fois le parcours actif terminé, l'algorithme retourne visiter les chemins auparavant inactifs.



**Figure 13** - Cinq opérateurs d'identification de triangles. Le maillage est converti en une liste d'opérateurs. L'exemple illustré a la liste : CRRRSLCRSERRELCRRRCRRRE. Figure tirée de [1].

Plusieurs suggestions ont été faites pour améliorer l'algorithme initialement proposé par Rossignac comme l'amélioration du processus de décompression [28], la compatibilité avec les modèles comportant des poignées (*handle*) [29], les maillages non-triangulaires [30] et pour optimiser l'efficacité du code pour des maillages réguliers où les valences sont généralement de degré 6 [31].

*Edgebreaker* est également employé pour compresser des maillages dynamiques en faisant intervenir l'analyse en composantes principales (ACP) comme l'a proposé Váša et al. [32]. L'approche exploite une cohérence spatiale et temporelle en appliquant l'ACP sur une matrice de trajectoire d'un point puis en représentant tous les points du modèle dans cette nouvelle base. *Edgebreaker* est ensuite appliqué pour compresser la connectivité.

Les détails de l'algorithme *Edgebreaker* seront discutés davantage dans le Chapitre 3 de ce travail. Il sera employé pour la compression de la connectivité des modèles car la méthode compresse efficacement la connectivité à un bon taux (1.5 à 2 bits par triangle). Bien que l'algorithme ne possède pas le meilleur *bit rate*, il est efficace, rapide et polyvalent pour traiter des maillages à haute densité.

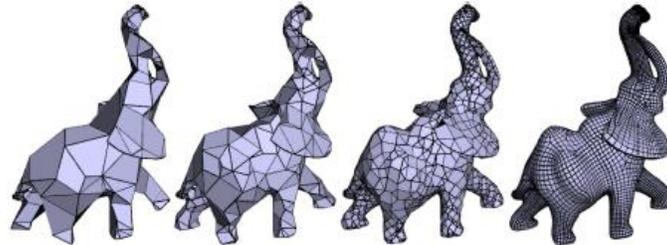
Des variations d'*Edgebreaker* ont ensuite été proposées dont *Angle-analyzer* par Lee et al. [21]. Ceux-ci ont proposé un ensemble d'opérateurs différents et un comportement différent pour définir le prochain triangle à évaluer. En effet, l'algorithme vient évaluer quelle direction prendre après avoir attribué un opérateur à une face de manière à augmenter la répétition d'un même symbole. Les séries de symboles identiques sont mieux compressées par l'*entropy coding* et permettent d'obtenir des taux de compression plus importants.

### 2.5.2. Compression progressive (progressive compression)

Contrairement à la compression à taux unique, la compression progressive donne une approximation à faible résolution du maillage d'origine à son état compressé selon différentes techniques telles que les ondelettes [16] [33] [34] [35] [36] [37], par décimation de sommets et remaillage dans la zone locale [38] [39] et globale [40] [41], en profitant des propriétés particulières de la triangulation de Delaunay [42] ou par Laplacien [43] [44].

Une illustration graphique d'une compression progressive est présentée à la Fig. 14. Une restriction qu'impose la compression progressive est la présence d'un maillage de base semi-régulier ou régulier. Les maillages irréguliers ne respectent pas les contraintes demandées pour la décimation, la prédiction ou l'interpolation de sommets car le nombre de valence des sommets est très variant. Des modifications irréversibles tel que la *coarsification* par sous-échantillonnage et

par synthèse du maillage doivent être faites sur les maillage irréguliers pour les rendre semi-réguliers ou réguliers si l'on souhaite les compresser par une approche progressive [45] [46]. Tel que mentionné, ces modifications construit une nouvelle structure de maillage différent du maillage d'origine ce qui engendre une perte d'information en plus de ne pas conserver de trace du maillage d'origine.



**Figure 14** - Différents niveaux de détails sur un modèle d'éléphant à maillage polygonal obtenu par décimation de sommets et remaillage dans la zone locale décimée. Figure tirée de [38].

Seule la compression par Laplacien compresse exclusivement l'information géométrique d'un modèle c'est-à-dire que la résolution du modèle demeure inchangée. La compression par Laplacien peut ainsi être jumelée à d'autres techniques de compression progressive afin d'atteindre de plus hauts taux de compression.

La compression progressive peut transmettre un modèle progressivement à travers un réseau de transmission un état à la fois. L'information sujette aux basses fréquences est soumise prioritairement puis le modèle se raffine progressivement pour retrouver le modèle d'origine avec certaines pertes.

### **Compression par Laplacien**

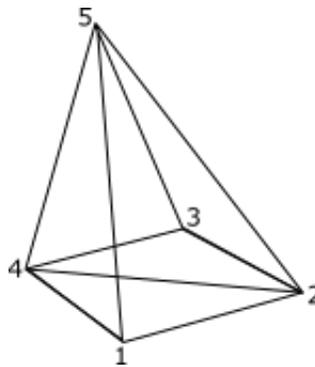
L'approche de compression progressive par bases spectrales est une extension de l'analyse par transformation de Fourier pour les maillages 3D. Les fonctions de base de Fourier pour un

signal 2D discret (fichier image) sont les vecteurs propres d'une matrice Laplacienne  $\mathbf{L}$  définie par la différence d'une matrice de valences  $\mathbf{V}$  diagonale et une matrice de connexions  $\mathbf{C}$  où  $v_i$  est la valence du point ou pixel étudié [44].

$$\mathbf{V}_{ij} = \begin{cases} v_i & i = j \\ 0 & \text{autrement} \end{cases} \quad (2.3)$$

$$\mathbf{C}_{ij} = \begin{cases} 0 & i = j \\ 1 & i \text{ et } j \text{ sont voisins} \\ 0 & \text{autrement} \end{cases} \quad (2.4)$$

Dans un quadrillage discret comme une image où les pixels constituent le quadrillage, tous les pixels qui ne constituent pas le contour ont une valence de 4 alors que ceux constituant le contour ont une valence de 2 (coins) ou 3 (côtés). L'analogie peut être faite avec les maillages où la valence correspond au nombre d'arêtes joignant un point et la matrice de connexions est faite à partir des indices des sommets joints. Un exemple de structure simple et de la matrice de valence et de connexions est donné à la Fig. 15.



$$\mathbf{V} = \begin{pmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

**Figure 15** - Exemple de compression par Laplace sur une structure simple

La matrice Laplacienne  $\mathbf{L}$  se calcule via l'équation 2.5 et les valeurs propres  $\mathbf{D}$  et vecteurs propres  $\mathbf{U}$  sont évalués avec  $\mathbf{L}$  via l'équation 2.6.

$$\mathbf{L} = \mathbf{V} - \mathbf{C} \quad (2.5)$$

$$\mathbf{L} = \mathbf{U}\mathbf{D}\mathbf{U}^T \quad (2.6)$$

$$\mathbf{V}^b = \mathbf{U}^T\mathbf{V} \quad (2.7)$$

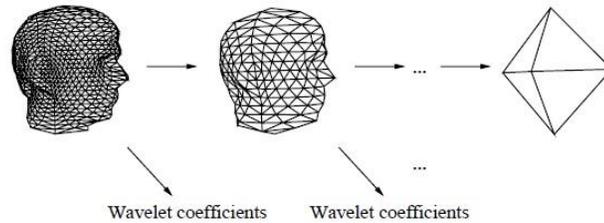
La matrice des vecteurs propres est ordonnée dans l'ordre croissant des valeurs propres associées. La matrice des vecteurs propres est ensuite utilisée pour générer de nouvelles bases orthonormales  $\mathbf{V}^b$  suivant l'équation 2.7 à partir des coordonnées cartésiennes des sommets  $\mathbf{V}$ . Lors de la transmission, les bases associées aux plus basses valeurs propres sont transmises prioritairement. L'information associée aux basses valeurs propres correspond aux basses fréquences du maillage.

Le plus grand désavantage de cette technique est sa complexité algorithmique. Pour un modèle de  $\mathbf{n}$  sommets, les fonctions de ces bases spectrales trouvées à partir des valeurs propres et des vecteurs propres doivent être calculées à partir d'une matrice  $\mathbf{n} \times \mathbf{n}$ . Ce calcul exige un temps de  $O(\mathbf{n}^3)$ . Une solution proposée par Karni et al. [43] est de subdiviser le maillage en une série de sous-maillages et d'utiliser une approche multi-résolution pour diminuer la complexité du calcul.

### Compression par ondelettes

La compression par DCT (*discrete cosine transform*) ou ondelettes (*wavelets*) est une méthode clé pour les compressions progressives avec pertes. Cette technique est principalement utilisée pour la compression de fichiers audio, vidéo et d'images et a par la suite été adaptée à la compression de maillages 3D par Eck et al. [47]. La méthode décompose un modèle en un maillage de faible résolution et conserve une série de coefficients d'ondelettes décrivant le raffinement entre l'état d'origine et l'état approximé comme le présente la Fig. 16. Ces coefficients peuvent ensuite être quantifiés et compressés par codage d'Huffman.

Considérant  $L^\infty$  comme la norme de l'erreur entre le modèle d'origine et le modèle approximé, il est possible, afin d'augmenter le taux de compression, de ne pas considérer certains coefficients en autant que l'erreur  $L^\infty$  ne dépasse pas un seuil d'erreur maximal  $\epsilon$  préalablement fixé. Si la suppression d'un coefficient rend  $L^\infty > \epsilon$ , celui-ci est conservé et l'évaluation de l'erreur  $L^\infty$  est faite sur les coefficients restants [47].



**Figure 16** - Compression progressive par ondelettes. Figure tirée de [47].

Une application intéressante de la compression par ondelettes a été proposée par Vijayaraghavan et al. [48] dans le domaine de l'imagerie médicale. L'idée générale de la méthode est de compresser un modèle 3D produit par Imagerie par Résonance Magnétique (IRM) ou par tomographie (CT et PET) par ondelettes et différents filtres et, selon les besoins, décompresser uniquement des tranches transverses 2D indépendamment du reste du modèle. Chaque tranche est associée à un coefficient d'ondelettes. L'avantage de cette technique est une réduction accrue du temps de décompression étant donné que le modèle complet ne doit pas être reconstruit dans son entièreté et que seules les tranches d'intérêts sont extraites.

L'approche de compression progressive assume toutefois que le modèle est suffisamment petit pour être chargé dans l'espace mémoire de l'ordinateur utilisé. Dans la situation où la taille du modèle est trop importante, la compression progressive ne peut pas être envisagée car tout le modèle doit être chargé afin d'obtenir toute l'information fréquentielle du modèle. En utilisant une approche de compression à taux unique et l'intervention d'un procédé de séparation comme le propose Isenburg et al. [49] dans leur travail sur la compression de maillages gigantesques, il est possible de prendre une approche par *streaming* pour séparer le maillage en plus petites sections. Ces sections peuvent alors être compressées individuellement.

### 2.5.3. Approche choisie

Dans ce travail, l'algorithme *Edgebreaker* a été choisi en raison (i) de sa polyvalence à pouvoir compresser un maillage connecté indépendamment de sa topologie, (ii) de sa rapidité d'exécution, (iii) de la disponibilité du code source et (iv) de son approche intuitive. Un module de prétraitement permettant à l'algorithme *Edgebreaker* d'être polyvalent sur toutes les conditions de maillage que l'on peut rencontrer sur un modèle destiné à des applications météorologiques est proposé. L'approche vise à altérer le maillage en corrigeant les singularités, les trous, les îlots et les volumes pouvant être générés par le scanneur avant l'application de l'algorithme de compression *Edgebreaker* classique. Les éléments listés ci-dessus seront définis au Chapitre 3.

Une étude similaire a été faite par Junjie Zhu [2] alors qu'il présente un nouveau détecteur de poignée différent que celui initialement proposé par Rossignac et al. [29] afin de diminuer la complexité algorithmique pour augmenter la vitesse de calcul. Zhu implémente également un nouvel opérateur **M** dans la liste possible **C-L-E-R-S** qui permet à *Edgebreaker* de prendre en compte les trous sans avoir à les traiter préalablement. Les îlots sont traités indépendamment du reste du modèle en appliquant l'algorithme sur chaque îlot séparément. Pour un modèle bruité, nous avons constaté expérimentalement que la quantité d'îlots devient très grande et exige donc d'appliquer à de nombreuses reprises *Edgebreaker*. L'approche que l'on propose dans ce travail vise à corriger cet effet en appliquant *Edgebreaker* le moins souvent possible en créant une seule grande surface avant l'application de l'algorithme.

Un autre point négatif à l'approche proposée par Zhu est la présence d'un nouvel opérateur. En ajoutant un nouvel opérateur **M**, la valeur de l'entropie suivant l'équation 2.8 augmente et rend la compression par entropie (utilisée conjointement avec le modèle de compression Lempel-Ziv - LZ77 lors de la compression WinZip) moins efficace [50] [7].

$$E(i) = -\sum_n (p_i \log_2(p_i)) \quad (2.8)$$

## Chapitre 3. Description de l'algorithme de compression et décompression

L'algorithme *Edgebreaker* a été introduit dans le précédent chapitre comme étant un algorithme de compression sans perte d'un maillage triangulaire fermé. Le présent chapitre est une extension plus détaillée de la description de l'algorithme faite au chapitre précédent. Une description du processus de décompression par *Wrap&Zip* y est également faite.

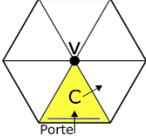
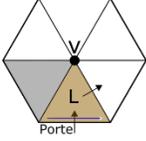
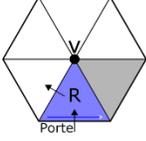
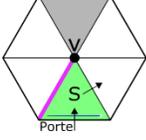
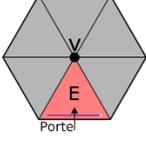
### 3.1. Compression d'un maillage simple

Tel que décrit antérieurement, *Edgebreaker* compresse uniquement la connectivité d'un maillage en traversant le maillage progressivement. La direction prise par le chemin est décidée selon des conditions prédéfinies dictées par le sommet situé en face de la porte et des triangles voisins. La porte (*gate* en anglais) correspond à l'arête par laquelle le chemin est incident et le sommet face à cette porte correspond au sommet  $v$ .

Cinq opérateurs sont définis pour représenter la topologie d'un maillage : Coupe (**C**), Droite (**R**), Gauche (**L**), Séparation (**S**) et Élimination (**E**). Un seul opérateur peut être assigné par triangle présent dans le maillage. Une fois qu'un opérateur a été assigné à un triangle, l'opérateur est conservé dans une liste nommée "historique" et les sommets constituant le triangle sont conservés dans l'ordre chronologique où ils ont été visités. Une nouvelle porte est ensuite définie selon l'opérateur attribué. Le Tab. 1 présente les caractéristiques et restrictions pour l'attribution d'un opérateur à un triangle.

L'historique des opérateurs ainsi que l'ordre des sommets identifiés par des indices sont requis pour la décompression. Le même chemin que la compression est emprunté lors de la décompression. La décompression est faite selon une méthodologie qui sera décrite à la section 3.2.

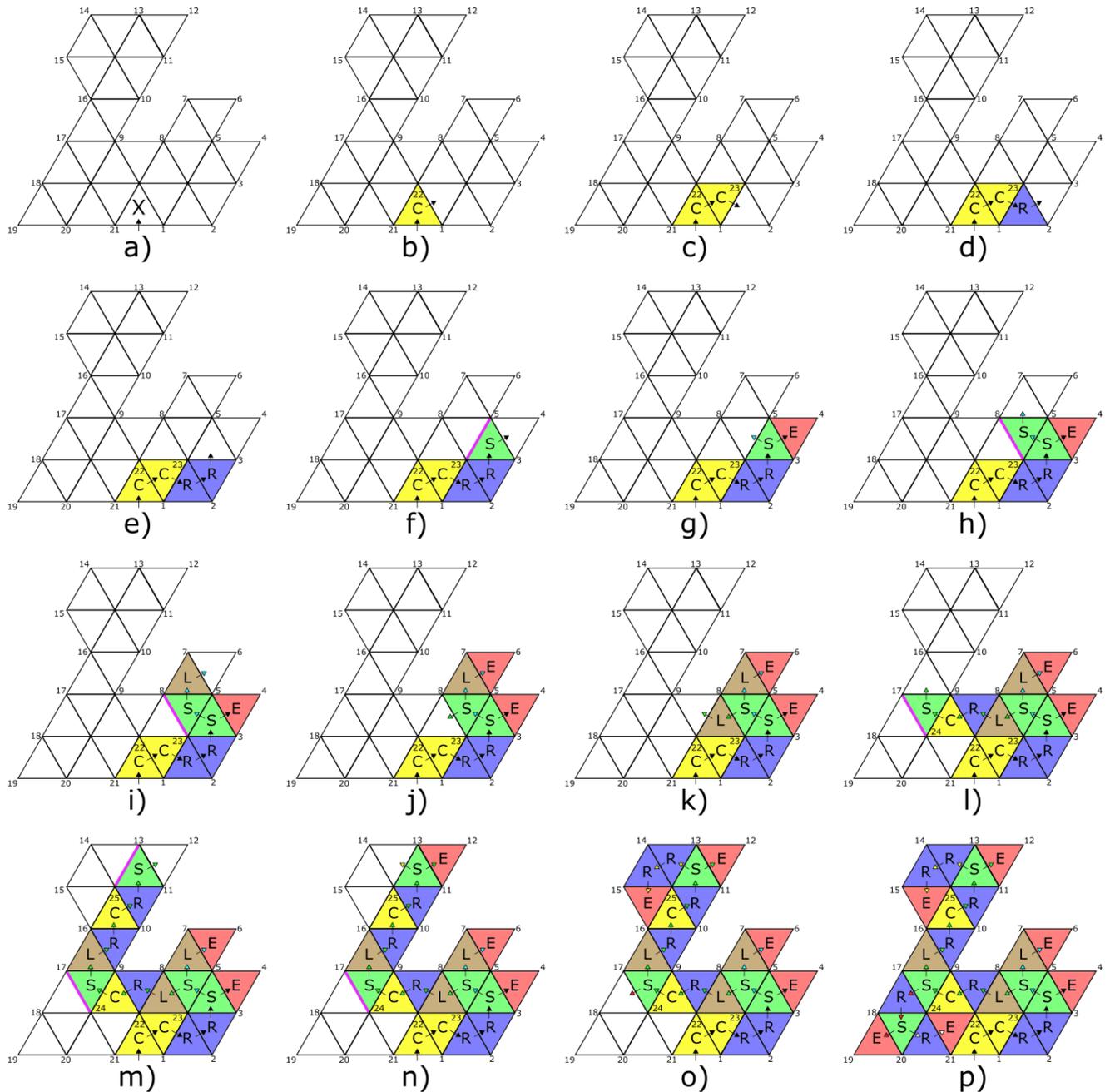
**Tableau 1-** Liste des opérateurs de compression et leurs fonctions

Opérateurs	Schémas	Voisinage	Actions
Couper ( <b>C</b> )		- <b>v</b> n'a pas été visité -Pas de triangles voisins visités	-Place <b>v</b> dans la liste des sommets visités -Attribue <b>C</b> à l'historique -Déplace la porte à droite
Gauche ( <b>L</b> )		- <b>v</b> a été visité -Triangle de gauche a été visité	-Attribue <b>L</b> à l'historique -Déplace la porte à droite
Droite ( <b>R</b> )		- <b>v</b> a été visité -Triangle de droite a été visité	-Attribue <b>R</b> à l'historique -Déplace la porte à gauche
Séparation ( <b>S</b> )		- <b>v</b> a été visité -Pas de triangles voisins visités	-Attribue <b>S</b> à l'historique -Déplace la porte à droite -Place la porte gauche au dessus de la liste active
Élimination ( <b>E</b> )		- <b>v</b> a été visité -Tous les triangles voisins ont été visités	-Attribue <b>E</b> à l'historique -La porte est déplacée à l'arête au dessus de la liste active

La liste active contient les références  $\{S_0, S_1, S_2, \dots\}$  des portes lors du processus de compression. Cette liste est construite lorsqu'un opérateur **S** est ajouté à l'historique. L'opérateur **S** est constitué de deux portes : la porte suivante (celle de droite) et la porte opposée (celle de gauche). La porte opposée du triangle ayant été associé l'opérateur **S** est placée au sommet de la liste active à la position  $S_0$  et les références déjà présentes dans la liste voient leur position décalée ( $S_0$  devient  $S_1$ ,  $S_1$  devient  $S_2$ , ...). Une fois qu'un chemin est complété, l'algorithme vérifie l'état de la liste active et définit la prochaine porte selon la référence au sommet de la liste active.

Un exemple par étapes complet de compression par *Edgebreaker* est illustré à la Fig. 17. Par souci de clarté, l'exemple est fait sur un maillage 2D (i.e. tous les triangles sont dans le même

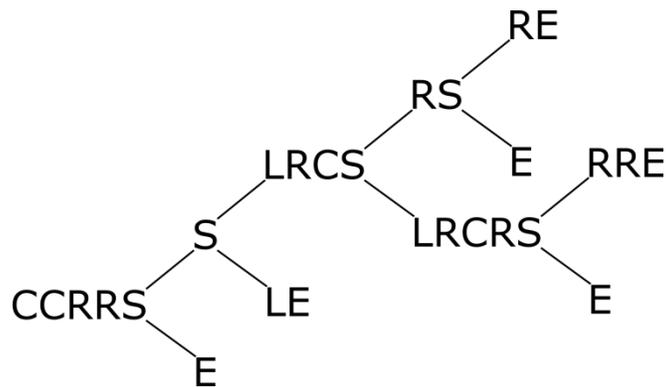
plan). On assume que les sommets formant la frontière du maillage ont déjà été visités et les indices 1 à 21 ont été attribués à ces sommets. La porte est initialisée sur l'arête {21, 1}.



**Figure 17** - Exemple de compression d'un maillage triangulaire avec *Edgebreaker* : a) Évaluation du triangle X selon la porte; b) Attribution de l'opérateur **C** et l'indice 22 est donné au sommet non-visité (opposé à la porte). La porte est placée à l'arête de droite du triangle ; c) Attribution de **C** et l'indice 23 est donné au sommet non-visité ; d) Ajout de **R** ; e) Ajout de **R** ;

f) Séparation de la branche par l'opérateur **S**. La porte se déplace à droite et l'arête gauche est mémorisée dans la liste active ; g) Fin de la branche (identifié par les flèches noires). Recherche de la dernière entrée dans la liste active (celle identifiée en fuchsia en f) et attribution de la porte sur l'arête au-dessus de la liste. Cette entrée dans la liste active est ensuite supprimée. Début d'un nouveau chemin (flèches bleues) ; h) Séparation du chemin ; i) Attribution de **L** ; j) Fin de la branche et repositionnement de la porte ; k) Attribution de **L** ; l) Le chemin se poursuit jusqu'à une séparation du chemin ; m) Progression du chemin jusqu'à une autre séparation. L'arête {25, 13} est placée au-dessus de la liste active. L'arête {24, 17} est décalée dans l'ordre de la liste active ; n) Fin du chemin des flèches vertes. Début du chemin jaune ; o) Fin de la branche jaune et récupération de la nouvelle porte selon l'arête au-dessus de la liste active ; p) Fin de la compression par *Edgebreaker* du maillage.

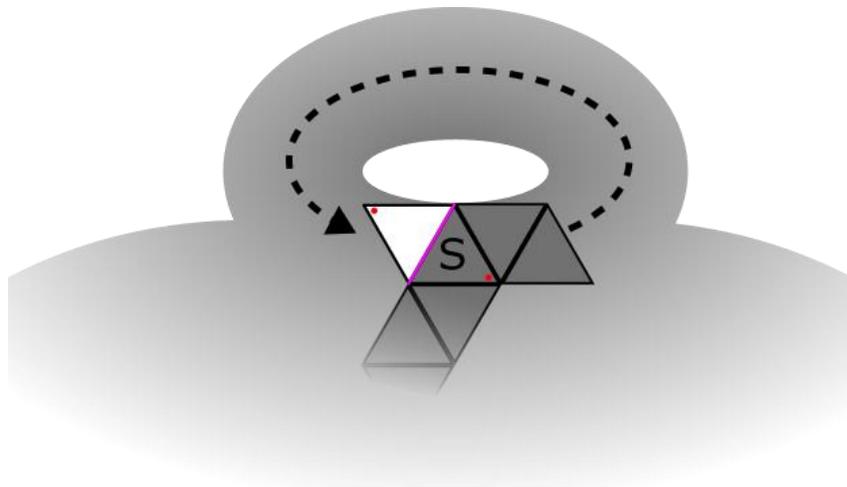
L'ensemble du procédé de compression génère ainsi un arbre de connectivité (voir Fig. 18) entre chaque face du maillage de la Fig. 17. Cet arbre correspond à l'historique **H** = CCRSESLRCSLRCSERRERSERE. Cet historique peut ensuite être compressé avec un codage par entropie tel qu'Huffman.



**Figure 18** - Arbre de connectivité pour le maillage de la figure 17

### 3.2. Compression d'un modèle avec présence de poignée

Telle que décrit à la section 2.2 du travail, une poignée est un trou dans le volume sans la présence d'une frontière (un tore par exemple). Une caractéristique spécifique de l'opérateur **S** est que le triangle voisin de gauche (illustré en blanc à la Fig. 19) ne soit pas visité avant que le chemin primaire de droite ne soit complété. Cependant, pour des maillages présentant une ou plusieurs poignées, le chemin primaire visitera inévitablement ce triangle voisin en blanc. Lorsque l'arête en fuchsia sera appelée à partir de la liste active, il y aura un problème car *Edgebreaker* aura déjà visité le triangle voisin en blanc. Afin d'éviter de passer à deux reprises sur le même triangle, Rossignac et al. [29] proposent de retirer l'arête en fuchsia de la liste active une fois qu'*Edgebreaker* visite le triangle voisin d'un triangle identifié par l'opérateur **S**. De plus, la paire de coins (illustré en rouge sur la Fig. 19) est conservée dans un fichier nommé "*handles*" ce qui permettra de connecter l'arête fuchsia du triangle blanc et gris lors de la décompression. Cette connexion est faite en repérant l'arête opposée de la paire de coins sauvegardée dans le fichier *handles* et en attribuant les mêmes indices de sommets à l'arête.

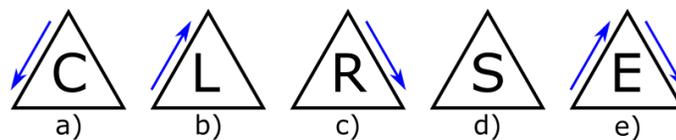


**Figure 19** - Illustration du chemin d'*Edgebreaker* entrant dans une poignée. L'arête en fuchsia est d'abord conservée dans la liste active mais est ensuite retirée lorsque la traverse fait le tour de la poignée et attribue un opérateur au triangle en blanc. Les points rouges correspondent aux coins conservés dans le fichier *handles*.

### 3.3. Décompression par *Wrap&Zip*

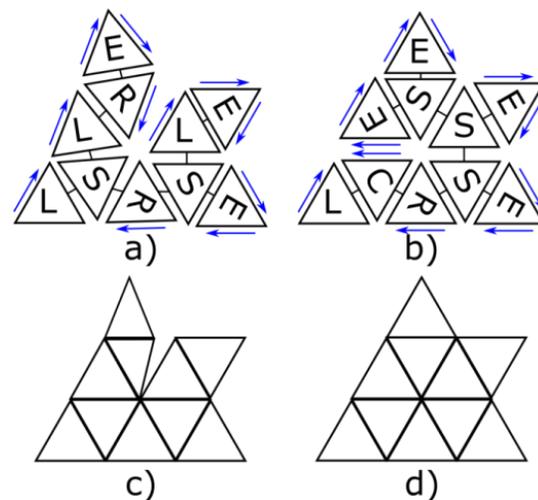
Le processus de décompression se fait essentiellement selon une méthode semblable à celle de la compression. L'algorithme procède également par traverse incrémentale du maillage. La méthode de décompression originalement proposée par Rossignac [1] était compliquée et lente en raison de son approche non-linéaire. L'algorithme proposé par Rossignac et Szymczak [28] nommé *Wrap&Zip* vient améliorer la technique initiale en introduisant une approche simple et linéaire qui permet la reconstruction de la topologie en créant un squelette de triangles interconnectés par un seul côté (*Wrap*). L'algorithme vient ensuite connecter les côtés des différents triangles afin de restituer le reste des connexions du maillage (*Zip*). Le squelette de triangles est semblable à l'arbre de connectivité illustré à la Fig. 18 via l'intermédiaire de triangles connectés.

Chaque opérateur désigné dans le processus de compression d'*Edgebreaker* est associé à un schéma prédéfini de triangles avec une orientation donnée sur un ou deux de ses côtés (voir Fig. 20). Le chemin se fait selon le même ordre que la compression et un nouveau triangle est collé à la porte du triangle actuel. La position des portes des opérateurs demeure la même que lors du processus de compression. La conservation de la porte de gauche de l'opérateur **S** dans une liste active reste également le même. L'ordre de rencontre des nouveaux sommets (opérateur **C**) est conservé et une fois que la topologie est complètement reconstruite, les coordonnées géométriques des sommets sont attribuées chronologiquement. Un squelette de triangles avec connexion simple entre les triangles est donc généré. Cette étape constitue la première phase *Wrap* de la décompression *Wrap&Zip*.



**Figure 20** – Représentation des opérateurs pour la décompression du maillage par l'algorithme de décompression *Wrap&Zip*. Les flèches bleues donnent l'orientation des côtés libres n'étant pas des portes.

L'orientation des côtés donnée par les flèches bleues (nommé également côté libre) de la Fig. 20 permet à l'algorithme de *Wrap&Zip* de décider si deux triangles voisins partagent un même côté. Si deux triangles avec deux côtés libres partagent un même sommet, l'algorithme vérifie l'orientation des côtés libres. Si ces côtés pointent à l'opposé du sommet partagé, l'algorithme joint les arêtes ensemble en identifiant le sommet partagé avec un indice identifié. Un exemple est donné à la Fig. 21 pour une situation où les côtés ne sont pas unifiés (21.a et 21.c) et lorsque les côtés sont unifiés (21.b et 21.d) par l'algorithme *Wrap&Zip*.



**Figure 21** – Deux exemples de décompression de maillage où le triangle inférieur gauche correspond au triangle de départ. Les flèches bleues montrent l'orientation des côtés libres. a) Squelette créé à partir de l'historique LSRSELELRE ; b) Squelette créé à partir de l'historique LCRSESESEE ; c) Les arêtes de R et L ne sont pas jointes car leur orientation ne pointe pas à l'opposé du sommet partagé ; d) Les arêtes de C et E sont jointes car leur orientation pointe à l'opposé du sommet partagé.

### 3.4. Restrictions et limites de l'algorithme

Bien que l'algorithme *Edgebreaker* soit suffisant pour compresser la majorité des modèles, il a essentiellement été conçu et testé pour la compression et décompression de modèles parfaits et sans défaut dans le maillage. Certains artefacts d'acquisition ne peuvent être gérés par la version actuelle de l'algorithme de compression. Ces limites sont :

- Les trous répartis à travers le maillage
- Les points et les arêtes *non-manifold*
- La présence de plusieurs surfaces et volumes indépendants (i.e. section de la triangulation non-connectés par des sommets ou des arêtes)

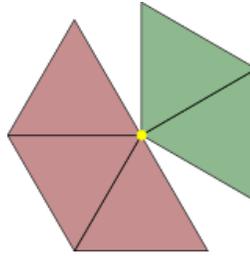
### 3.4.1. Les trous répartis à travers le maillage

La particularité de la présence de trous dans un maillage est l'absence d'une deuxième orientation pour chaque arête constituant sa frontière. Il peut arriver que le chemin d'*Edgebreaker* rencontre une arête appartenant à la frontière d'un trou. L'algorithme assume la présence d'un triangle voisin à cette arête et assume qu'il sera visité par le chemin subséquent lors de la compression du maillage. Le triangle voisin étant inexistant, l'historique final de la compression ne pourra reconstruire le modèle initial suite au processus de décompression.

### 3.4.2. Les points et les arêtes *non-manifold*

Les circonstances menant à la génération de points et arêtes *non-manifold* surviennent rarement. Ceci dit, la précision de l'appareil, l'angle de capture entre l'axe optique et la surface scannée, les conditions d'illumination de l'environnement et les caractéristiques réfléchissantes de la surface sont quelques facteurs pouvant jouer un rôle dans la création d'artefacts *non-manifold*.

Un point *non-manifold* existe si un sommet est inclus dans deux (ou plus) ensembles de triangles voisins (voir Fig. 22). Une telle configuration a un impact important dans la création du chemin lors du processus de compression d'*Edgebreaker*. Lorsque le chemin rencontre le point *non-manifold* pour la première fois, ce point est marqué comme étant visité. Cependant, lorsque le trajet arrive via l'autre ensemble de triangles, le point est déjà indexé ce qui induit une erreur d'attribution d'opérateurs lors de la compression par *Edgebreaker*.



**Figure 22** – Point *non-manifold* (jaune) partagé par deux ensembles de triangles voisins (ensemble rouge et ensemble vert) dans un maillage 2D

Une arête *non-manifold* ne peut exister dans un maillage 2D mais peut apparaître dans un maillage 3D. Cette arête est appelée dans 3 (ou plus) faces du modèle et induit les mêmes erreurs qu'un point *non-manifold*. La solution proposée au Chapitre 4 pour la correction des points s'apprête également à la correction des arêtes *non-manifold*.

### 3.4.3. Présence de plusieurs surfaces et volumes indépendants

L'algorithme de compression parcourt la topologie fermée d'une triangulation. Cependant, l'algorithme ne peut atteindre les mailles isolées d'un maillage principal (i.e. les sections de la triangulation non-connectée par des sommets et des arêtes). La conséquence directe de cette situation est une omission de l'algorithme à compresser ces surfaces non-reliées à la surface de départ. Lors de la décompression du modèle, les surfaces et volumes n'ayant pas été visités seront absents. Tout comme pour l'apparition des points *non-manifold*, de petits éléments de surfaces flottantes autour d'un modèle peuvent être construits par le scanneur dans certaines conditions. De plus, il est courant qu'un modèle contienne diverses sections disjointes ne résultant pas du bruit de l'appareil d'acquisition. Une solution proposée dans l'article original [1] est d'appeler à nouveau *Edgebreaker* pour chaque surface et volume isolés. Pour certains modèles contenant des milliers d'îlots, cette approche n'est pas optimale.

### 3.5. Conclusion

Dans ce chapitre, une description des algorithmes principaux servant de base à ce travail a été introduite, expliquée et illustrée. L'algorithme de compression qui a été sélectionné est *Edgebreaker* et l'algorithme de décompression est *Wrap&Zip*.

Pour le premier algorithme, cinq opérateurs sont créés et attribués selon des conditions de voisinage pour chaque triangle du maillage. La liste des sommets est réorganisée selon l'ordre de visite et un historique des opérateurs est construit et conservé dans un fichier texte.

L'algorithme de décompression réutilise ces informations du fichier texte pour recréer un arbre de connexions à partir de l'historique et l'indice des coordonnées des sommets. Tout comme la compression, la décompression s'effectue à l'aide de 5 opérateurs préalablement définis. L'algorithme reconstruit ensuite l'arbre des connexions à partir de l'historique et connecte ensemble les côtés ayant une orientation sortante d'un sommet commun.

Bien que l'approche initialement proposée parvienne à compresser la majorité des modèles, elle n'a cependant pas été développée pour la compression de maillages non-fermés, avec des îlots séparés, des trous et des volumes fermés indépendants. Le Chapitre 4 de ce travail présente l'approche proposée afin de rendre les algorithmes d'*Edgebreaker* et *Wrap&Zip* capable de palier aux limitations ci-dessus et le rendant polyvalent pour la compression de maillages typiques scannés pour des applications météorologiques.

# Chapitre 4. Modifications, prétraitement et post-traitement de l'approche proposée

## 4.1. Introduction

Les modifications apportées cherchent à altérer temporairement la topologie du modèle avant le processus de compression, puis à retrouver la topologie d'origine suite à la décompression. Le format de la topologie utilisé suit le format *Standford Triangle* (PLY). Il est à noter également que les algorithmes de toutes les étapes de prétraitement et de post-traitement ont été implémentés dans l'environnement Matlab 2014a.

Le code source de la compression *Edgebreaker* et de décompression *Wrap&Zip* a été implémenté en C++ sous la plateforme Visual Studio 2012 et est disponible publiquement suite à des requêtes industrielles et éducatives. L'implémentation de la version du code utilisé dans ce travail a été développé par Alla Safonova au *Georgia Institute of Technology* [51].

Le prétraitement proposé à la compression se déroule en quatre étapes :

- Réajustement des décimales
- Détection et correction des points *non-manifold*
- Remplissage des trous
- Connexion des îlots et volumes indépendants

L'organigramme de la Fig. 23 dresse un portrait général de l'organisation de la solution proposée pour le prétraitement et la compression d'un modèle 3D triangulaire. Les étapes seront décrites de manière exhaustive dans la suite de ce chapitre.

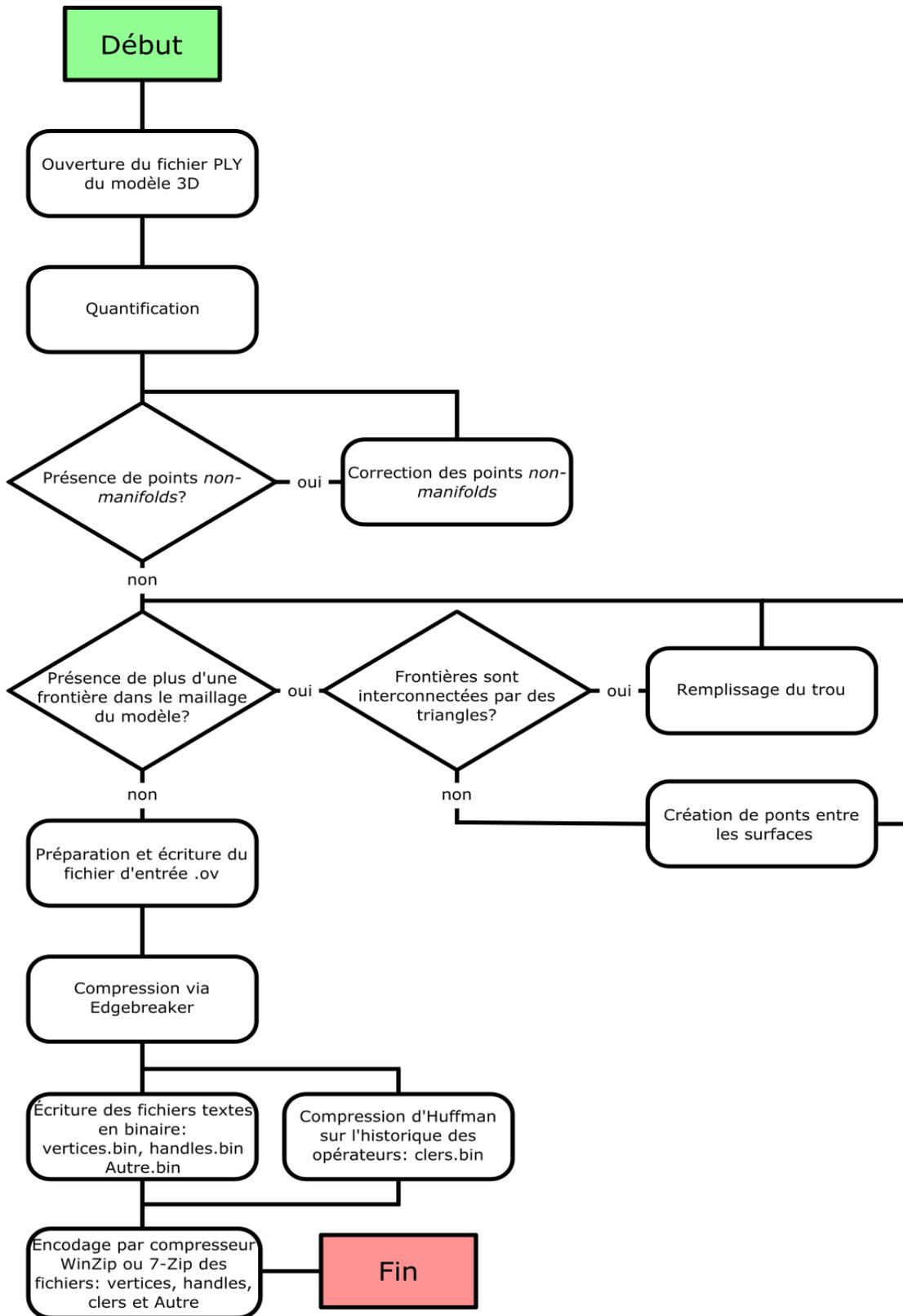


Figure 23 - Organigramme du prétraitement et de la compression pour un modèle

## 4.2. Ouverture du fichier source

L'ouverture et la lecture du fichier source en format PLY se fait par une fonction disponible publiquement développée par Pascal Getreuer [52]. Cette fonction lit et sépare l'information de connectivité et l'information géométrique contenues dans le fichier dans une variable *struct*. L'entête et les commentaires du fichier sont lus mais ne sont pas conservés.

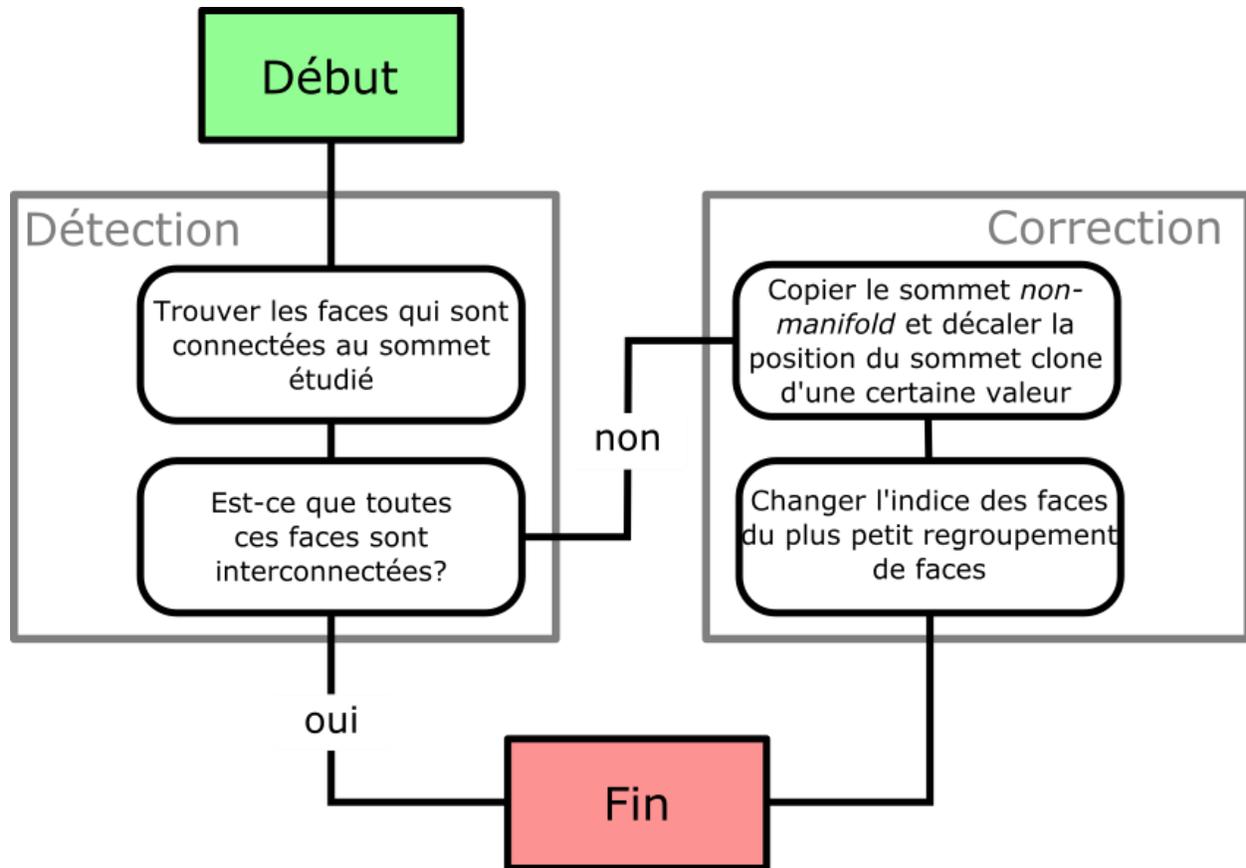
## 4.3. Réajustement des décimales par quantification

Le réajustement des décimales se fait selon la quantification décrite au Chapitre 2. Cette étape restreint le nombre de décimales affecté à la précision de la valeur numérique des coordonnées des sommets. Considérant que l'appareil d'acquisition (MetraScan 3D et GoScan de Creaform) donne une erreur minimale de 64  $\mu\text{m}$  [53], on se permet de ne conserver que les 6 premières décimales des coordonnées des sommets. Les décimales suivantes sont considérées non-significatives et peuvent être rejetées sans altérer la précision du modèle. Moins de bits sont donc nécessaires pour représenter les valeurs numériques du modèle.

L'erreur de précision des coordonnées des sommets demeure influencée uniquement par l'appareil d'acquisition. Conserver toutes les décimales stockées dans le fichier PLY peut générer une perte de précision lors des opérations algorithmiques car le code *Edgebreaker* n'a pas été développé pour traiter un grand nombre de décimales.

## 4.4. Détection et correction des points *non-manifold*

La détection et la correction des points *non-manifold* se fait en deux sections distinctes tel qu'illustré à la Fig. 24. L'algorithme de détection selon l'approche de Jacobson [54] est vectorisé, ce qui permet de réduire la complexité algorithmique et d'augmenter la vitesse d'exécution.

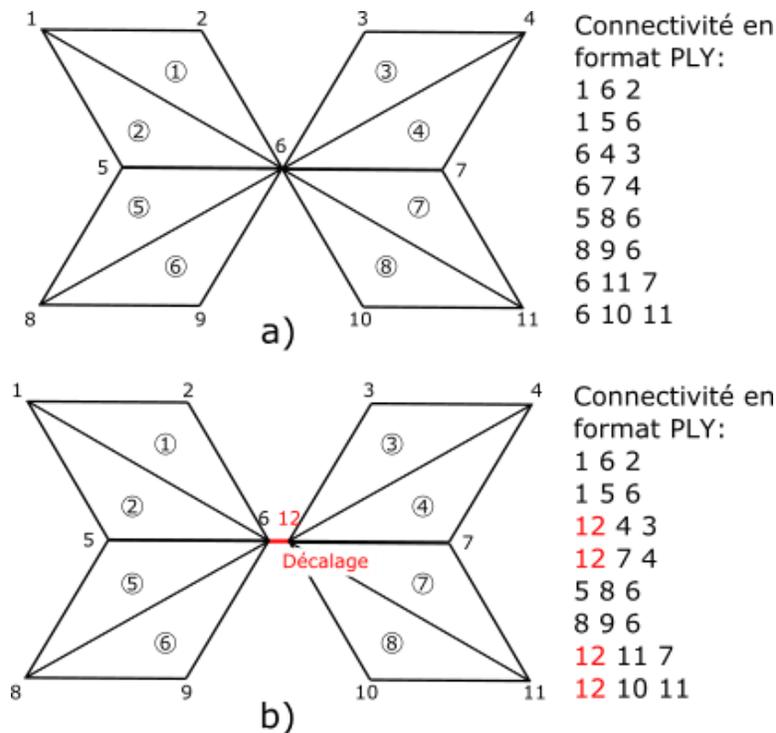


**Figure 24** - Organigramme pour la détection et la correction de points *non-manifold*

L'approche proposée par Jacobson évalue tous sommets présents dans le modèle. Pour chaque sommet, l'algorithme extrait les faces qui sont connectées. Par la suite, l'algorithme évalue si ces faces sont interconnectées en regroupant les faces qui partagent une arête avec une face voisine. Suite à cette étape, s'il existe 2 regroupements de triangles distincts (i.e. une série de faces interconnectées), alors le sommet étudié est caractérisé comme étant *non-manifold* et une correction doit être apportée pour permettre à *Edgebreaker* de fonctionner correctement.

La correction que nous proposons duplique le sommet et décale la position du sommet *non-manifold* repéré lors de l'étape précédente d'une certaine valeur prédéterminée (voir Fig. 25). Cette valeur de décalage est arbitraire mais doit être la même lors du processus de décompression. Une fois le sommet dupliqué, l'algorithme modifie son indice à la valeur  $N+1$  où  $N$  correspond au

nombre total de sommets dans le maillage. La Fig. 25 illustre les opérations de duplication, décalage et renumérotation d'un sommet non-manifold.



**Figure 25** - Correction d'un point *non-manifold* dans un maillage à 8 faces 2D. Les faces sont numérotées selon les chiffres encadrés. a) Maillage d'origine. b) Nouveau maillage suite aux modifications apportées (en rouge)

Pendant le processus de décompression, les coordonnées du sommet ajouté seront lues et le sommet sera décalé à l'inverse et repositionné à sa position d'origine. Selon les besoins de l'utilisateur, l'algorithme peut reconstruire le maillage d'origine en recréant la structure *non-manifold* ou conserver un maillage *manifold* sans décalage du sommet dupliqué. Le point est donc dupliqué et a les mêmes coordonnées que son clone.

## 4.5. Évaluation et réparation des trous et îlots

L'évaluation et la réparation des trous et des îlots correspond à la plus grande contribution de ce mémoire à l'algorithme d'origine d'*Edgebreaker*. Comme il a été discuté au Chapitre 2, Zhu [2] propose l'intégration d'un nouvel opérateur **M** à la liste des opérateurs **CLERS**. L'intégration d'un nouvel opérateur affecte cependant l'entropie du fichier sortant. Les encodeurs standards tels que WinZip, 7-Zip, WinRar, etc. sont davantage efficaces lorsque l'entropie est faible afin de profiter des patterns et des répétitions des données du fichier à compresser.

La solution proposée dans ce travail cherche à compresser des modèles 3D sans l'inclusion d'un nouvel opérateur. Pour ce faire, la topologie doit être modifiée afin de remplir les trous avec des faces et de connecter tous les îlots ensemble de manière à ce que le modèle ne soit constitué que d'une seule grande surface et de volumes indépendants. La surface obtenue est ensuite fermée suite à l'inclusion d'un sommet nommé "*dummy vertex*".

### 4.5.1. Repérage des frontières et regroupement des faces

L'approche implémentée étudie les connexions entre les regroupements des faces et les différentes frontières du maillage. Une analyse de ces connexions permettra de faire la détection et différencier les frontières des îlots et les frontières des trous.

Les différentes frontières sont d'abord extraites du modèle en repérant les côtés n'ayant pas d'orientation inverse (côté inclus une seule fois dans l'énumération des sommets dans la liste des faces). On dénotera les sommets constituant les frontières d'un îlot ou d'un trou par "chaîne de frontière". Ces sommets sont identifiés facilement par la méthode suivante :

1. Trier les indices des sommets de toutes les faces (formés par une paire de sommets **S**) et comparer si le côté **j** a un côté opposé **j\*** où  $C_j = \{S_i, S_{i+1}\}$  et  $C_{j^*} = \{S_{i+1}, S_i\}$  où **i** est l'indice du sommet.
2. Conserver uniquement les côtés qui ne sont pas en paire.
3. Construire une chaîne de côtés qui définit la frontière puis extraire les sommets de telle sorte que  $\{S_k, S_{k+1}\}, \{S_{k+1}, S_{k+2}\}, \dots, \{S_{k+n}, S_k\}$  où **k** est l'indice des sommets

composant la frontière et  $n$  est le nombre total de sommets dans cette chaîne de frontière. Lorsque le premier sommet est répété à la fin de la chaîne, la chaîne est complétée.

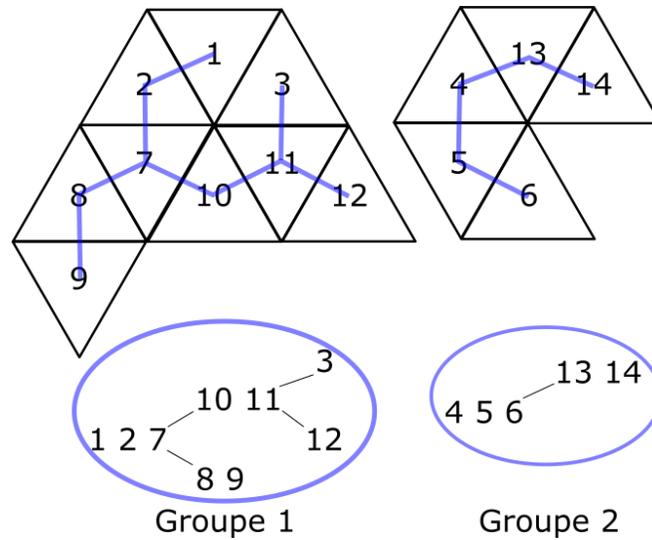
4. Retirer les sommets qui se répètent de cette chaîne afin d'avoir une liste ordonnée et orientée : Chaîne de frontière =  $\{S_k, S_{k+1}, S_{k+2}, \dots, S_{k+n}\}$

La surface associée à la chaîne ayant le plus grand nombre de sommets est considérée comme étant l'îlot principal alors que les autres îlots sont catégorisés comme étant secondaires.

Un deuxième élément important pour la différenciation entre les îlots et les trous est l'interconnexion entre les faces. Des regroupements de faces devront donc être créés afin d'identifier si deux chaînes de frontière font partie du même regroupement de faces indiquant ainsi qu'une frontière définit un trou et l'autre la frontière externe de l'îlot.

Pour générer ces regroupements de faces, on construit un *spanning tree* en traversant le maillage via les côtés des faces commençant par la première face énumérée dans le fichier PLY (voir Fig. 26). Ce *spanning tree* se fait en visitant le triangle de droite au triangle étudié. Si le triangle de droite a déjà été visité, le triangle gauche est visité. Si ce dernier a également été visité, on revient aux triangles précédemment visités et les voisins sont visités selon la même logique. L'ensemble des faces connectées à cette première face est considéré comme un groupe distinct de triangles.

Une fois que le premier *spanning tree* est complété, on génère un second en commençant par une face n'ayant pas été visitée par les autres *spanning tree*. Une fois que tous les *spanning tree* ont été générés, on obtient un nombre  $h$  (nombre d'îlots et de volumes isolés) de groupes de faces  $\mathbf{G}$ . Ces groupes sont formés de l'indice des faces qui sont interconnectées comme l'illustre la Fig. 26. Sur cette figure,  $\mathbf{G}_1 = \{1, 2, 7, 8, 9, 10, 11, 12, 3\}$  et  $\mathbf{G}_2 = \{4, 5, 6, 13, 14\}$ .

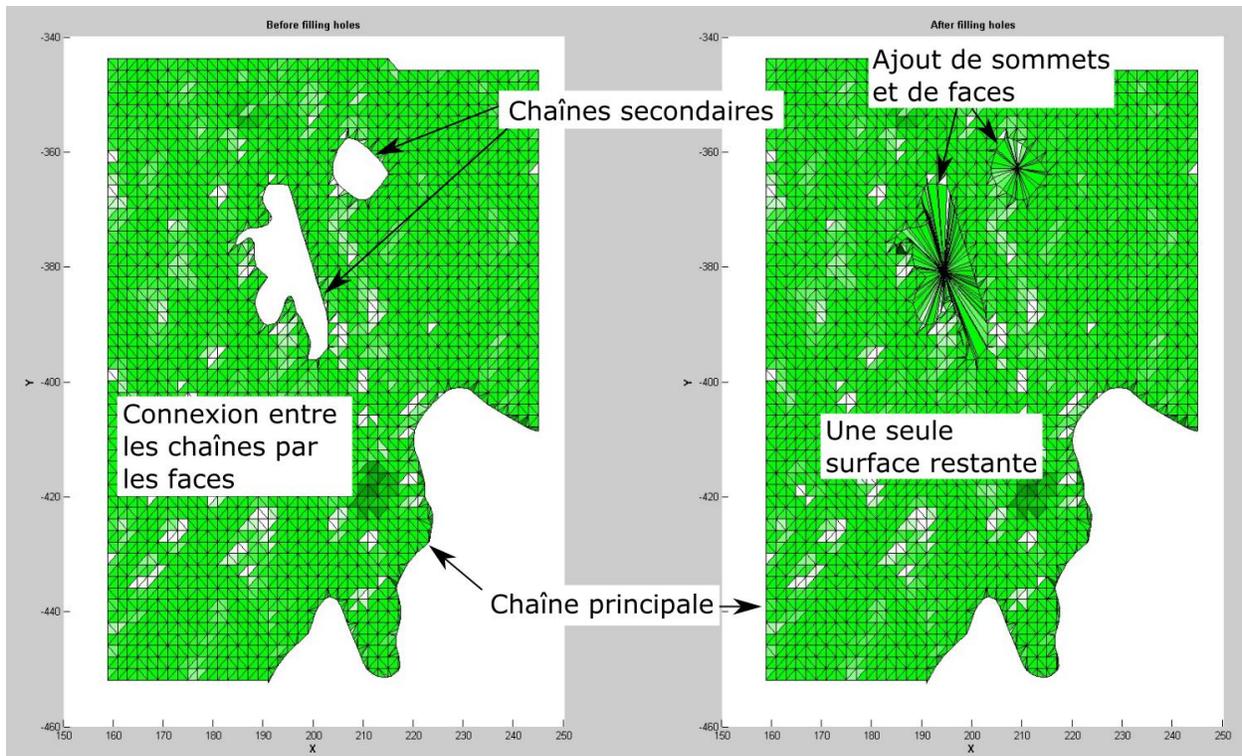


**Figure 26** – Modèle constitué de deux îlots distincts à 14 faces. Deux regroupements de faces dans le maillage sont construits à partir de l'approche par *spanning tree*. Les deux regroupements de faces sont Groupe 1 = {1, 2, 7, 8, 9, 10, 11, 12, 3} et Groupe 2 = {4, 5, 6, 13, 14}.

Si une chaîne  $C(S_i) \in G(F(S_i))$  alors les sommets de la chaîne  $C$  font parties du groupe de faces  $G$  où  $F$  est une face et  $S_i$  un sommet quelconque du maillage. Si cette condition est satisfaite pour plus d'une chaîne, alors la plus petite chaîne correspond à un trou.

#### 4.5.2. Remplissage des trous

Tel que discuté au Chapitre 3, le maillage envoyé au compresseur *Edgebreaker* doit être un volume complètement fermé. Les trous doivent donc être remplis par l'intégration de nouvelles faces. Lorsqu'un trou a été détecté, un sommet est ajouté en son centre et  $N$  faces sont créées et connectées à la chaîne de frontière du trou à partir de ce nouveau sommet, où  $N$  est le nombre de sommets de la frontière comme l'illustre la Fig. 27.



**Figure 27** - Remplissage de trous par l'inclusion de sommets et de faces sur une surface trouée

L'ajout d'un sommet augmente l'espace requis pour l'encodage de l'information géométrique ( $3 \times 32$  bits avant encodage par entropie) en plus de la nécessité de conserver ses coordonnées dans un fichier "Autre". En conservant sa position, le sommet peut être retrouvé puis supprimé afin de reconstruire le maillage d'origine. L'ajout des faces pour le remplissage du trou peut être représenté par  $N+1$  bits dans le meilleur des cas. Cette situation survient si les triangles sont compressés par la séquence d'opérateurs : CCC...CR où le nombre d'opérateurs correspond au nombre de faces  $N$  et que le lexique de compression de l'encodage par entropie (discuté à la fin de ce chapitre) attribue le nombre minimal de bits pour les opérateurs  $C$  et  $R$  comme c'est le cas dans ce travail.

### 4.5.3. Connexion des îlots

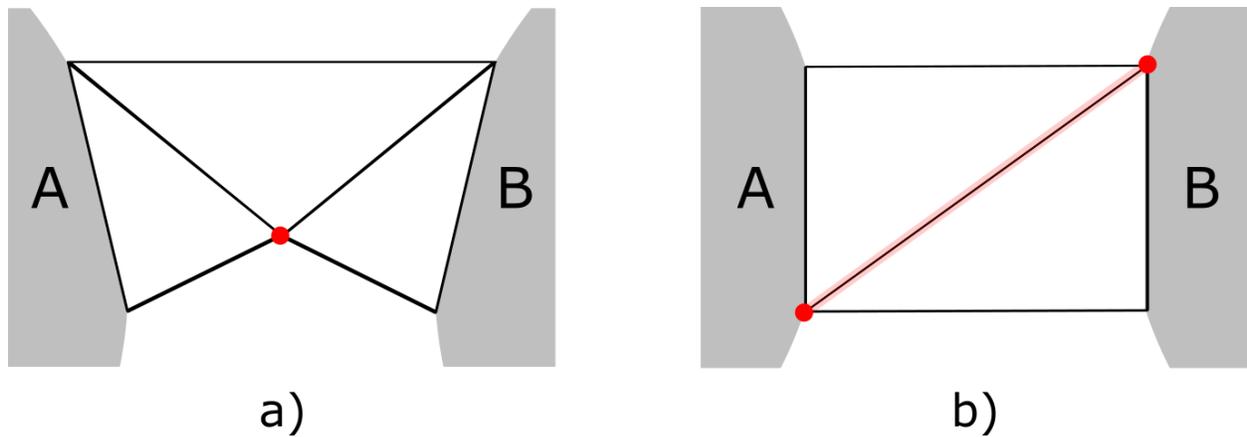
La prochaine étape de l'approche proposée est l'interconnexion de tous les îlots restants. L'approche de Zhu [2] pour corriger cette problématique est de fermer chaque surface flottante par l'ajout d'un sommet et de connecter ce sommet aux frontières, puis de compresser chaque îlot individuellement par *Edgebreaker*. Ceci implique qu'*Edgebreaker* est appelé pour chaque îlot présent dans le maillage et qu'un nombre imposant de fichiers sortants sont produits pour la compression d'un modèle.

L'approche proposée du présent travail fait appel à *Edgebreaker* un nombre minimal de fois en connectant tous les îlots. L'avantage est que le compresseur est appelé une seule fois pour l'ensemble des surfaces isolées maintenant interconnectées.

On caractérise un îlot secondaire comme étant un groupe de triangles connectés à une frontière (chaîne) et que cette frontière n'est pas la plus longue (en terme de quantité de sommets) du modèle. La connexion des îlots secondaires se fait itérativement en construisant un ensemble de triangles (nommé pont dans ce travail) afin de lier les îlots entre eux et avec l'îlot principal.

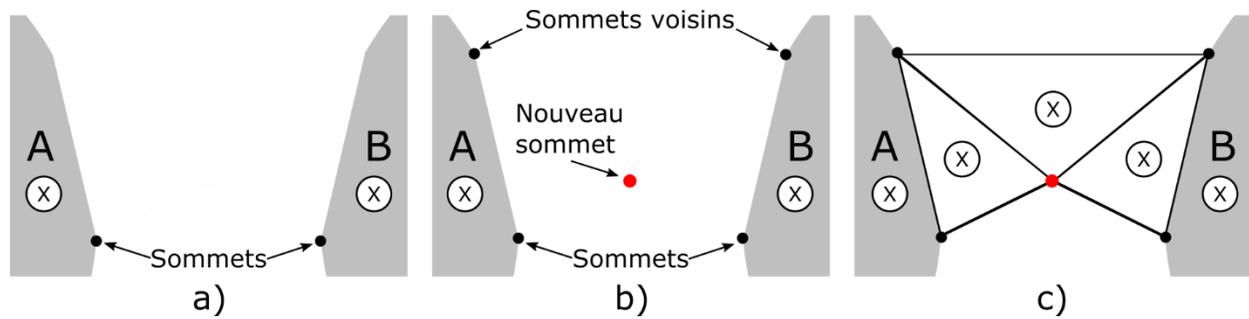
Deux topologies considérées sont illustrées à la Fig. 28. La topologie à 3 faces (Fig. 28.a) requiert un nombre supérieur de bits pour encoder l'information topologique que la structure à 2 faces (Fig.28.b). Cependant, il est nécessaire de conserver de l'information géométrique afin de retrouver les ponts suite à la compression et la décompression. L'information topologique ne permet pas de retrouver les ponts car elle est altérée après le passage d'*Edgebreaker*.

La structure à 3 faces requiert uniquement la sauvegarde des coordonnées d'un seul point (en rouge sur la Fig. 28.a). En retrouvant ce point et en retirant tous les triangles qui lui sont connectés, le pont est effacé. La structure à 2 faces demande à ce que deux sommets soient conservés (en rouge sur la Fig. 28.b). Ces points faisant partie du maillage d'origine doivent être conservés et les faces du pont sont effacées en retirant l'arête reliant les deux sommets. La topologie à 3 faces est donc favorisée car un seul sommet doit être conservé (3x32 bits), contrairement à 2 pour le pont à 2 faces (6x32 bits).



**Figure 28** - Deux topologies étudiées pour la structure des ponts reliant les îlots A et B. Les sommets en rouge illustrent l'information devant être conservée. a) Ponts construit par la création de 3 triangles; b) Pont construit par la création de 2 triangles.

Tel qu'illustré sur l'organigramme de la Fig. 23, le processus de connexion des îlots et de remplissage de trous commence par l'évaluation des chaînes de frontière. Ces chaînes sont d'abord différenciées selon qu'elles définissent un trou ou une frontière par la méthode développée à la section 4.5.1. Une fois que les frontières ont été identifiées, on trouve la paire de sommets sur deux chaînes de frontière qui partagent la plus petite distance entre les chaînes (voir Fig. 29). Sur chaque chaîne, on prélève le sommet voisin (gauche ou droite). Les coordonnées du nouveau point médian du pont sont calculées comme étant la moyenne des coordonnées des 4 points d'accrochage du pont. Les faces sont ensuite construites et on vérifie que les normales des triangles d'accrochage ont la même orientation. Si un triangle d'accrochage n'a pas la même orientation que les faces de l'îlot, on choisit le sommet voisin opposé au premier sommet prélevé.



**Figure 29** - Construction progressive d'un pont entre les îlots A et B. a) Deux sommets rapprochés sont prélevés; b) Les voisins le long de la chaîne de frontière de ces sommets sont prélevés; c) Construction des faces et vérification des normales aux triangles d'accrochage.

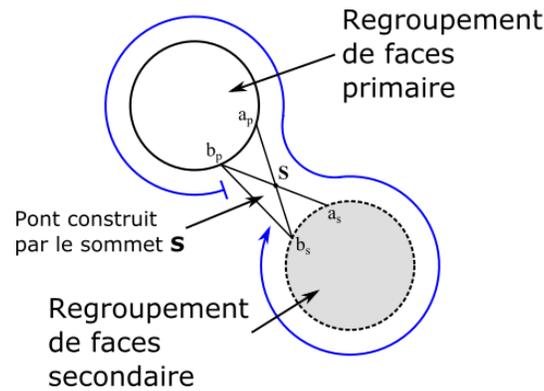
#### 4.5.4. Mise-à-jour automatique des chaînes

Lorsque 2 îlots sont connectés, les 2 chaînes de frontière sont fusionnées en une seule. Il serait possible de recalculer au complet les frontières du maillage à partir de l'approche développée à la section 4.5.1. Cependant, une mise-à-jour diminue considérablement la complexité et le temps de calcul. La solution proposée vient implanter la chaîne secondaire (2<sup>e</sup> îlot) dans la liste de la chaîne primaire (1<sup>er</sup> îlot). L'orientation de la chaîne ainsi que les sommets de départ et de fin des chaînes doivent être pris en compte.

Le lexique utilisé dans cette section est le suivant :

- L'indice **p** signifie que la position des sommets appartient à la chaîne primaire et l'indice **s** qu'il appartient à la chaîne secondaire.
- $a_p$  et  $b_p$  définissent la position des sommets voisins appartenant à la même chaîne de frontière primaire  $C_p = \{S_1, S_2, \dots, S_{a_p}, S_{b_p}, \dots, S_n\}$  ou  $C_p = \{S_1, S_2, \dots, S_{b_p}, S_{a_p}, \dots, S_n\}$  selon la situation.
- Le regroupement de faces primaires est illustré par un disque blanc (voir Fig. 30).
- Le regroupement de faces secondaires est illustré par un disque gris (voir Fig. 30).

La Fig. 30 illustre la chaîne résultante désirée. On cherche à mettre à jour la chaîne primaire afin qu'elle ait la structure suivante :  $C_{\text{mise-à-jour}} = \{Sb_p, \dots, Sa_p, S, Sa_s, \dots, Sb_s\}$ . La technique d'approche doit être invariante à l'orientation des chaînes primaires et secondaires et invariante de la disposition du pont.



**Figure 30** - Illustration d'un cas de mise-à-jour de la chaîne primaire suite à la construction d'un pont entre l'îlot blanc et gris. La flèche bleue désigne le chemin et l'orientation désiré de la nouvelle chaîne.

Les étapes suivantes décrivent le processus de mise-à-jour des chaînes :

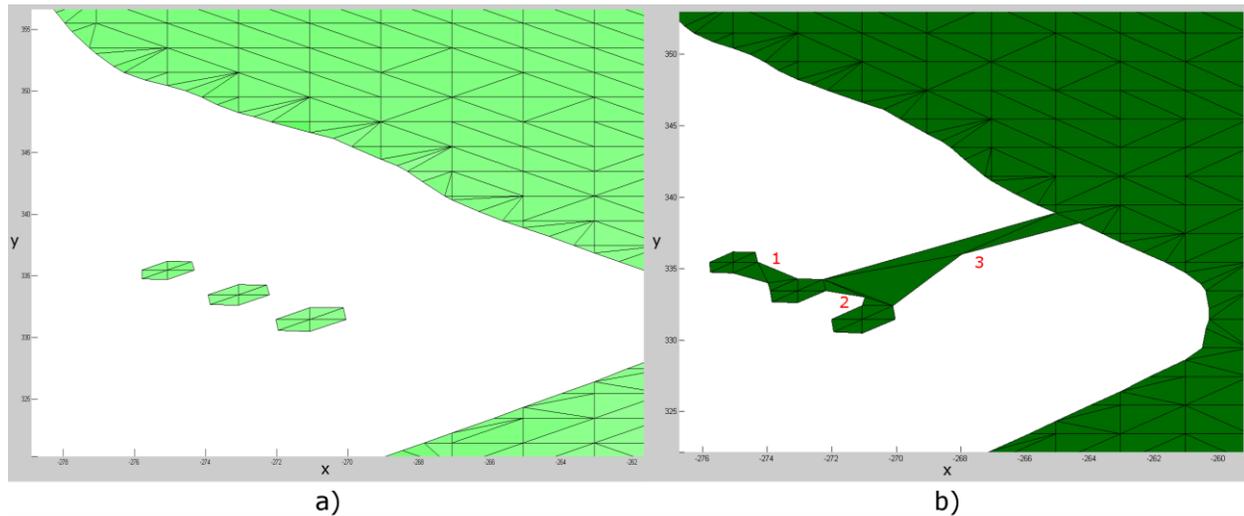
1. Détermination de l'orientation de la chaîne primaire selon deux cas :  $a_p < b_p$  ou  $a_p > b_p$
2. Ajustement de l'orientation en permutant l'ordre de la chaîne primaire au besoin.
3. Réorganiser la chaîne primaire en débutant par le sommet  $Sb_p$  et en finissant par le sommet  $Sa_p$ , i.e.  $C_{p'} = \{Sb_p, \dots, Sa_p\}$ .
4. Inclure le sommet du pont  $S$  dans la chaîne.
5. Refaire les étapes 1 et 2 pour la chaîne secondaire.
6. Réorganiser la chaîne secondaire en débutant par le sommet  $Sa_s$  et en finissant par le sommet  $Sb_s$ , i.e.  $C_{s'} = \{Sa_s, \dots, Sb_s\}$ .
7. Écrire la chaîne résultante à partir des chaînes réorganisées, i.e.  $C_{\text{mise-à-jour}} = \{C_{p'}, S, C_{s'}\} = \{\{Sb_p, \dots, Sa_p\}, S, \{Sa_s, \dots, Sb_s\}\}$ .

Le Tab. 2 illustre toutes les situations possibles d'orientation des chaînes et du pont unissant deux îlots. Les flèches bleues représentent l'orientation désirée en fonction de la disposition du pont (selon les standards définis à la Fig. 30) alors que les flèches noires représentent l'orientation actuelle de la chaîne de l'îlot.

**Tableau 2** - Tableau des différentes situation d'orientation des chaînes et des modifications à apporter

Chaîne de frontière primaire		Chaîne de frontière secondaire	
$a_p < b_p$	$a_p > b_p$	$a_s < b_s$	$a_s > b_s$
Aucun changement apporté à l'orientation de la chaîne primaire	Changement d'orientation de la chaîne primaire	Changement d'orientation de la chaîne secondaire	Aucun changement apporté à l'orientation de la chaîne secondaire

La mise-à-jour des frontières permet également la connexion de pont sur un pont ayant été précédemment construit. La Fig. 31 présente 3 îlots annexes à une surface. Les îlots sont d'abord joints par la construction des ponts 1 et 2. Ces 3 îlots et les deux ponts sont ensuite considérés comme étant un îlot puis joints à la surface annexe via le pont 3. Cette connexion à un autre pont est permise grâce à la mise-à-jour des frontières qui a été faite sur les 3 îlots.



**Figure 31** - Connexion de 3 îlots à une frontière. a) Maillage d'origine. b) Création de 3 ponts où le pont 3 est connecté à un triangle du pont 2.

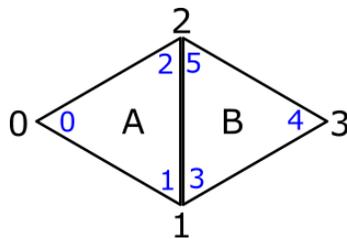
Lorsque tous les îlots ont été connectés, une seule chaîne de frontière reste. Un sommet nommé "*dummy vertex*" est ajouté, pour fermer la surface, à une position quelconque (une position n'étant pas déjà occupée par un autre sommet) et des faces sont créées selon la même méthode utilisée pour le remplissage des trous. La position du point est également ajoutée dans le fichier "*Autre*".

Pour ce qui a trait aux volumes fermés distincts (i.e. un regroupement de faces n'ayant pas de frontière), aucune modification n'est apportée à ces régions du modèle. La topologie respectant déjà les critères d'acceptabilité d'*Edgebreaker* n'a pas à être modifiée et la compression peut être appliquée directement. Lorsque le premier appel à *Edgebreaker* est fait, on vérifie que toutes les faces du modèle ont été visitées. Les faces n'ayant pas été visitées appartiennent à ces volumes fermés. *Edgebreaker* est ensuite rappelé pour tous les regroupements trouvés. Pour chaque appel supplémentaire d'*Edgebreaker*, une nouvelle série de fichiers *clers*, *vertices* et *handles* est générée.

## 4.6. Fichier transitoire : Sommet opposé

L'appel de la fonction de compression *Edgebreaker* demande qu'un certain nombre de paramètres d'entrée lui soient fournis. Parmi ceux-ci, on retrouve l'adresse informatique de la position de l'exécutable du code source de compression, le type de maillage (*manifold*), le coin de départ d'où débutera la traverse du maillage (coin lié au sommet ayant le plus grand nombre de faces connectées, généralement le *dummy vertex*), le format des fichiers de sortie (ASCII ou Binaire) ainsi qu'un fichier sommet opposé. Ce fichier de sommets et des coins opposés permet de fournir l'information de la connectivité ainsi que l'information géométrique. Ce fichier a une extension **OV** (*opposite vertex* en anglais). Le format du fichier transitoire contient la liste des sommets et de leur coin opposé dans le triangle voisin et est construit selon la structure montrée à la Fig. 32.

<b>T</b>		Nombre total de triangles
<b>S1</b>	<b>O1</b>	S1: Sommet 1 du 1 <sup>er</sup> triangle; O1: Coin opposé au sommet 1
<b>S2</b>	<b>O2</b>	S2: Sommet 2 du 1 <sup>er</sup> triangle; O2: Coin opposé au sommet 2
<b>S3</b>	<b>O3</b>	S3: Sommet 3 du 1 <sup>er</sup> triangle; O3: Coin opposé au sommet 3
<b>S4</b>	<b>O4</b>	S4: Sommet 1 du 2 <sup>e</sup> triangle; O4: Coin opposé au sommet 1
(...)		(...)
<b>S</b>		Nombre total de sommets
<b>X1 Y1 Z1</b>		Coordonnées (x,y,z) du sommet 1
<b>X2 Y2 Z2</b>		Coordonnées (x,y,z) du sommet 2
<b>X3 Y3 Z3</b>		Coordonnées (x,y,z) du sommet 3
(...)		(...)



**Figure 32** – Structure du fichier transitoire OV et ensemble de 2 triangles où le sommet 0 a le coin opposé 4 et le sommet 3 a le coin opposé 0

L'ordre d'appel des sommets pour la création du fichier entrant dans l'exécutable de compression est le même que celui du fichier PLY d'origine. Comme il a été discuté dans le Chapitre 3, l'algorithme *Edgebreaker* tient compte de l'ordre de rencontre des sommets dans sa traverse du maillage lors de la compression afin de réattribuer les bonnes coordonnées aux bons sommets lors de la décompression par *Wrap&Zip*. L'ordre de rencontre n'est généralement pas le même que l'ordre initial dans le fichier PLY. Cette approche a comme effet de changer l'ordre d'appel ainsi que les indices des sommets dans la liste de faces. Le modèle après décompression a les mêmes informations géométriques et topologiques que le modèle initial mais écrites différemment.

Un autre fichier de transition OV est construit suite à la décompression par *Wrap&Zip*. Celui-ci permet de retourner l'information géométrique et topologique à Matlab afin de supprimer les ajouts (sommets et faces) qui ont été faits au maillage d'origine.

Le code pour le convertisseur utilisé pour passer de PLY à OV et inversement est disponible publiquement et a été développé et implémenté par Gabriel Peyré [55] dans Matlab.

## 4.7. Codage par entropie

Tel que mentionné au Chapitre 2, le codage d'Huffman attribue un nombre minimal de bits au caractère qui a la fréquence d'apparition la plus élevée. Le dictionnaire proposé par Zhu [2] sera favorisé par rapport au dictionnaire proposé par Rossignac [1]. Le dictionnaire de Zhu sera appliqué uniquement à la chaîne des caractères présents dans le fichier des opérateurs d'*Edgebreaker* clers.txt. L'historique sera maintenant associé à l'extension .bin.

L'information géométrique des sommets initiaux et ajoutés suite au remplissage des trous et suite à la construction des ponts est traitée par la compression WinZip et 7-Zip. La compression par 7-Zip permet d'obtenir de meilleurs taux de compression que le format WinZip. Cependant, la compression WinZip est la plus utilisée et l'utilitaire le plus répandu dans les ordinateurs. Les résultats présentés au Chapitre 5 seront compressés et comparés suite à l'application de ces deux outils de compression.

**Tableau 3** - Dictionnaire d'attribution des bits pour l'encodage des opérateurs CLERS dans le fichier `clers.txt` proposé par Rossignac et Zhu

Historique CLERS	Rossignac	Zhu
C	0	0
L	110	1111
E	111	1110
R	101	10
S	100	110

La compression par entropie d'Huffman tire avantage des répétitions des données. Cependant, le dictionnaire proposé par Rossignac ne tient pas compte des fréquences d'apparition des opérateurs dans le fichier. Dans son dictionnaire, Rossignac attribue 3 bits pour l'encodage des opérateurs **L**, **E**, **R** et **S**. Représenter ces opérateurs avec un nombre égal de bits suppose qu'ils ont une fréquence d'apparition égale. Cependant, en pratique, les fréquences d'apparition ne sont pas équilibrées.

Suite à la compression des modèles présentés au Chapitre 5, la fréquence d'attribution a été évaluée. En moyenne, l'opérateur **C** représente ~50% des opérateurs attribués dans l'historique établi par la traverse du maillage d'un modèle par *Edgebreaker*. Cette haute fréquence est principalement causée par l'attribution inévitable d'un opérateur **C** pour chaque sommet présent dans le modèle. L'opérateur **R** représente ~43%, les opérateurs **E** et **S** ~3% et l'opérateur **L** ~1%. L'opérateur **R** est le second opérateur le plus appelé en raison du procédé de traverse d'*Edgebreaker*. En procédant au triangle voisin de droite, les probabilités sont plus élevées d'être voisin d'un triangle ayant été visité précédemment. L'opérateur **R** est souvent appelé suivant un ou plusieurs opérateurs **C**.

## 4.8. Décompression

### 4.8.1. Post-traitement à *Wrap&Zip*

*Wrap&Zip* a besoin, pour la décompression, des fichiers de l'historique (`clers`), des fichiers des coordonnées des points (*vertices*) et des arêtes visitées deux fois par *Edgebreaker* (*handles*). Cependant, une étape préliminaire de décompression d'Huffman doit être faite d'abord sur les

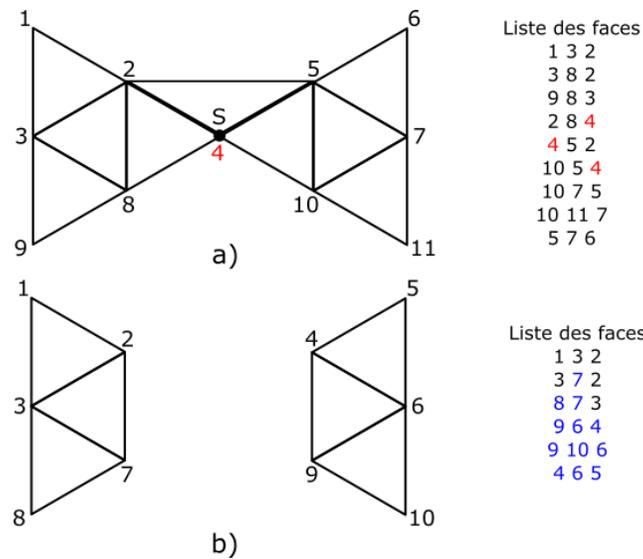
fichiers donnant l'historique afin qu'il puisse être lu par l'algorithme de traverse de maillage de *Wrap&Zip*. Le dictionnaire d'Huffman, étant constant pour tout modèle, sera utilisé pour décompresser l'historique **CLERS**.

Tel que mentionné précédemment dans ce chapitre, les sommets ajoutés afin de remplir les trous, connecter les îlots ainsi que pour fermer la dernière frontière doivent être supprimés suivant l'opération *Wrap&Zip*. Cette étape de suppression des sommets se fait après avoir retrouvé, via *Wrap&Zip*, le maillage qui avait été compressé par *Edgebreaker*. L'ordre des sommets n'étant plus le même, un scan doit être fait dans la liste des coordonnées pour trouver la position du point ajouté dans la liste. La méthodologie est la même pour les 3 scénarios.

La suppression des points indésirables se fait donc selon la procédure suivante :

1. Trouver la position **P** du point ajouté **i** dans la liste des sommets en comparant les coordonnées  $(x, y, z)$  avec les coordonnées  $(x_i, y_i, z_i)$  des sommets. Supprimer ce point lorsqu'il est trouvé.
2. Dans la liste des connectivités, chercher toutes les faces joignant le sommet à la position **P**. Supprimer ces faces.
3. Dans la liste des connectivités, chercher l'indice des sommets supérieur à l'indice **P**.
4. Diminuer l'indice de ces sommets par 1.

La Fig. 33 illustre le processus de suppression de points. Dans cet exemple, le sommet **S** a l'indice 4 et les faces contenant cet indice dans la liste sont enlevées. La liste des faces est mise-à-jour en réduisant d'une unité la valeur des indices supérieurs à 4.



**Figure 33** - a) Suppression du sommet S avec l'indice 4. b) Maillage suivant la suppression du sommet 4.

#### 4.8.2. Comparaison des modèles pré et post compression

La compression sans perte exige qu'aucune information ne soit perdue ou égarée en cours de traitement. L'information peut cependant prendre une forme différente comme c'est le cas suivant la compression par *Edgebreaker*. Afin de valider que le processus de compression est sans perte, le contenu géométrique et topologique doit être comparé au modèle avant la compression et après la décompression. Des étapes de traitement doivent être faites étant donné qu'*Edgebreaker* change l'ordre d'appel des sommets en fonction de l'ordre de rencontre lors de la visite du maillage. La Fig. 34 illustre le processus de comparaison des modèles.

L'information géométrique est facilement comparable en organisant en ordre ascendant les coordonnées (x, y, z) de la liste des sommets. En comparant chaque ligne des coordonnées du modèle initial et du modèle final, on peut aisément conclure s'il y a eu une altération de la précision géométrique.

L'information topologique est plus complexe à comparer. L'ordre des sommets ayant été alterné lors de la compression, les indices utilisés dans la liste des faces sont différents et doivent

être mis à jour pour comparer la connectivité des modèles. La topologie des modèles initiaux et finaux est convertie sur les mêmes bases en utilisant les nouveaux indices de sommets (suite au triage géométrique).

Les topologies sont triées et la comparaison de la connectivité peut être faite tel que montré à la Fig. 34. Tel qu'illustré dans cette figure, le modèle avant compression (1) ainsi que le modèle après compression (2) voit leurs coordonnées être triées en ordre croissant (3). Cependant, une fois ce tri effectué, l'indice des sommets a été alterné par rapport aux indices originaux. Afin de pouvoir comparer la topologie des modèles, l'information topologique des modèles (1) et (2) devra être mise à jour à partir des nouvelles positions obtenues à l'étape (3). Suite à cette étape, on obtient le modèle avant compression suivant le remplacement des indices (4) et le modèle après décompression suivant le remplacement des indices (5). À cette étape, tous les sommets du modèle ont les mêmes indices. Il ne reste plus qu'à effectuer un triage de la topologie en ordre ascendant afin que les deux modèles soient exprimés dans les mêmes bases (6). S'il n'existe pas de différence entre les deux modèles dans la géométrie ou dans la topologie, les modèles sont considérés comme étant identiques.

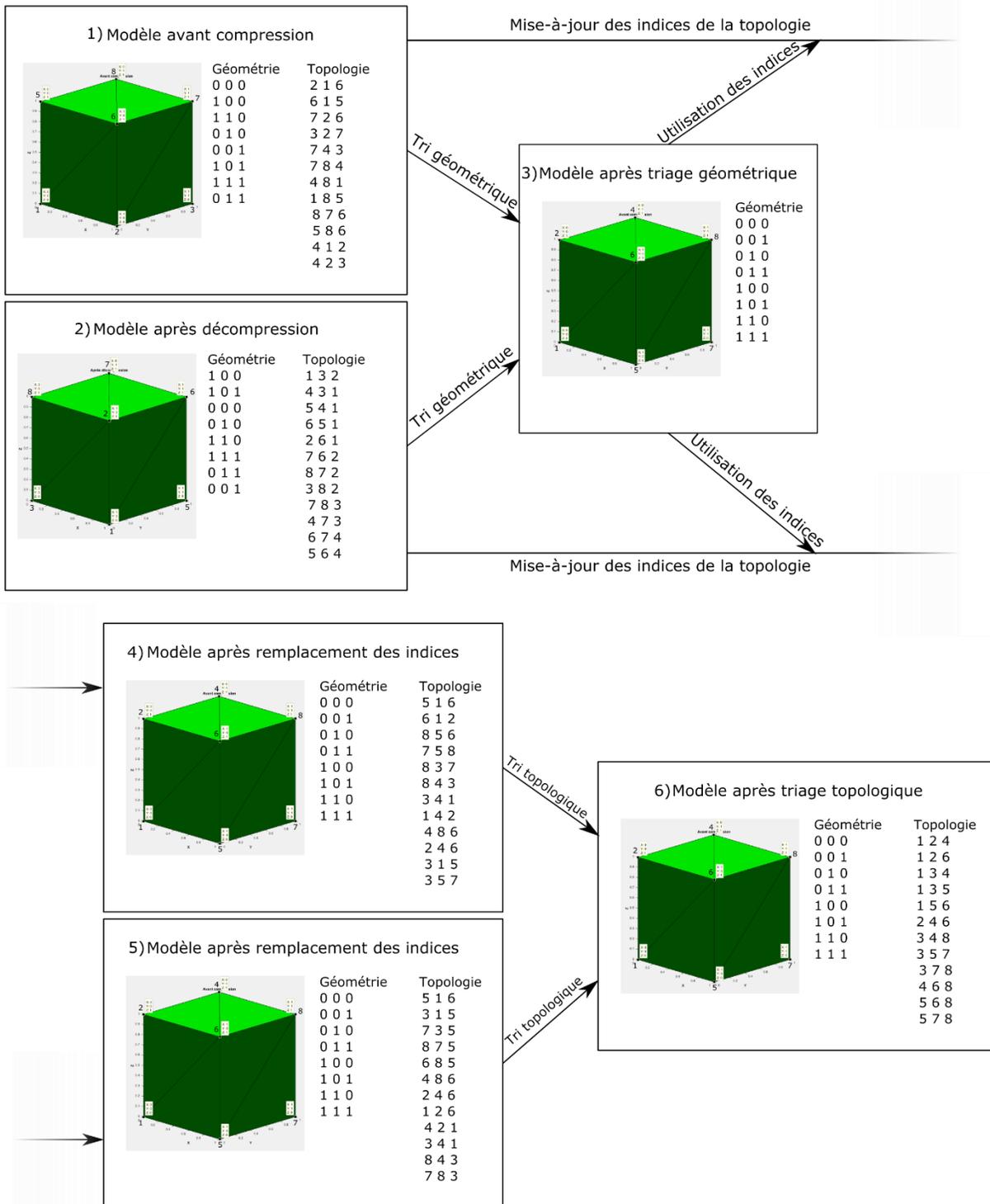


Figure 34 - Processus de comparaison des modèles avant la compression et après la décompression

## 4.9. Conclusion

Le Chapitre 4 a présenté la démarche générale et l'implémentation des différentes approches et techniques pour la compression des modèles triangulaires présentant des défauts. La solution proposée est une méthode alternative au travail de Zhu [2].

Une première modification ayant été apportée à l'algorithme d'origine d'*Edgebreaker* est la quantification de la précision de l'information géométrique du modèle. En ne conservant que les décimales supérieures à la précision de l'appareil de numérisation 3D, on élimine des décimales non-significatives tout s'assurant de ne pas perdre la précision métrologique des données.

Plusieurs spécificités rendent le maillage incompatible à la compression *Edgebreaker*. Les points *non-manifold* ont d'abord été repérés par l'évaluation des faces partageant le point *non-manifold* puis en repérant les interconnexions entre ces faces voisines. Lorsque le point a été désigné comme étant *non-manifold*, ses coordonnées sont dupliquées puis décalées d'une certaine valeur et une reconnexion des faces est faite.

Grâce à notre approche, les trous sont comblés et les îlots sont connectés afin d'être compatible avec les restrictions posées par le design de l'algorithme de compression. Le processus proposé dans ce travail est sensiblement le même pour la correction des trous et des îlots en introduisant un sommet puis en le connectant à des faces. Une fois ces artefacts corrigés, la surface restante est fermée via l'insertion d'un *dummy vertex*. L'information géométrique de tous les sommets ajoutés en plus de la valeur du décalage du sommet dupliqué dans la correction des points *non-manifold* a été sauvegardée et transmise dans le fichier "Autre". Lorsqu'un modèle compressé doit être transmis sur un réseau de communication, les fichiers transmis sont les suivants :

- Vertices\*.bin : Donne les coordonnées (x, y, z) de tous les points du maillage.
- Handles\*.bin : Paire de coins opposés suite à la présence d'une poignée dans la topologie.
- Clers\*.bin : Historique des opérateurs selon l'attribution faite par *Edgebreaker*. Fichier ensuite converti en binaire par codage d'Huffman.
- Autre\*.bin : Liste des coordonnées (x, y, z) des points à supprimer lors du processus de décompression (trous, ponts et *dummy vertex*)

\* : Valeur numérique dénotant l'ordre d'appel d'*Edgebreaker*. *Edgebreaker* est appelé autant de fois qu'il y a de volumes indépendants dans le modèle.

## Chapitre 5. Présentation et analyse des résultats

### 5.1. Introduction

Les résultats présentés dans ce chapitre ont été obtenus par deux scanners Creaform (Metrascan et GoScan) au Laboratoire de Vision et Systèmes Numériques de l'Université Laval. Les scans ont ensuite été sauvegardés en fichiers de format PLY binaire puis ont été traités par l'implémentation décrite au Chapitre 4 dans Matlab 2014a et Visual Studio 2012. Les données obtenues ont été calculées par un système d'exploitation Windows 64-bits avec 32.0 GB de RAM et un CPU Intel® Core™ i7-3930K @ 3.20 GHz.

Différentes variables sont impliquées dans l'analyse des performances de l'approche choisie. La taille finale des fichiers compressés est influencée par la quantité de sommets et de faces du modèle d'origine mais également par la quantité de sommets ajoutés suite à la réparation des différents artefacts avant l'utilisation d'*Edgebreaker*.

### 5.2. Présentation des modèles étudiés

Les modèles présentent des caractéristiques différentes qui permettent d'évaluer si la solution proposée est versatile et englobe toutes les spécificités de maillages pouvant être numérisés par un appareil d'acquisition tels les scanners 3D tenus en main de Creaform ou autres scanners. Les modèles initiaux étudiés se limitent cependant à une liste des sommets et des faces. Les informations supplémentaires pouvant faire partie du fichier telles que la colorisation des faces et sommets, les normales aux faces et sommets ainsi que les arêtes ne font pas partie des fichiers étudiés. Les Fig. 35 et 36 présentent les 9 modèles qui ont été utilisés pour faire la validation de l'approche proposée. À noter que le modèle de la Fig. 35.e est un modèle de crâne obtenu disponible publiquement sur Internet<sup>3</sup> et produit par un scanner 3D inconnu. Les propriétés des

---

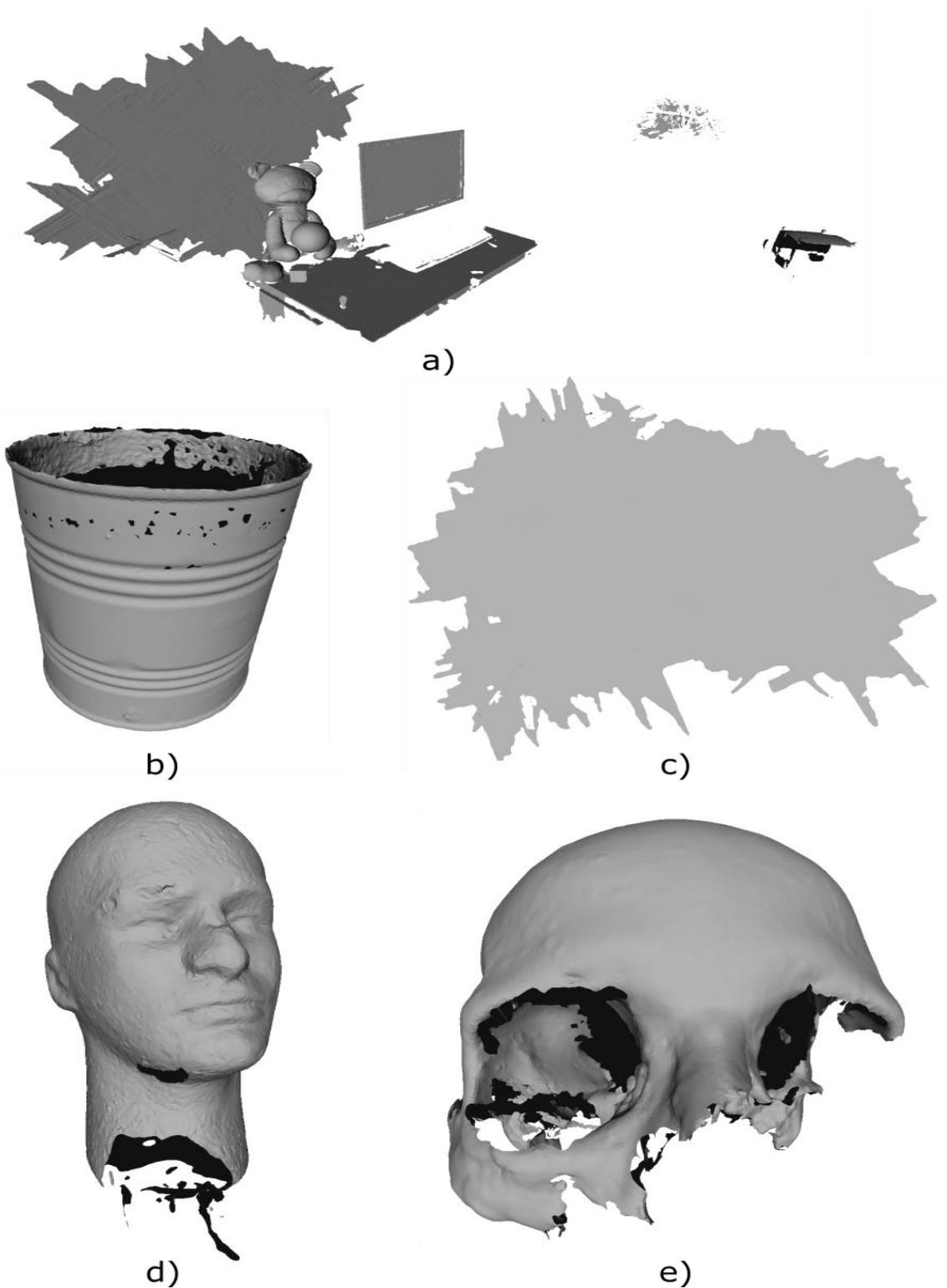
<sup>3</sup> Disponible à l'adresse suivante : <http://people.sc.fsu.edu/~jburkardt/data/ply/ply.html>

modèles sont listées au Tab. 4. Le ratio S/F correspond au ratio du nombre de sommets sur le nombre de faces.

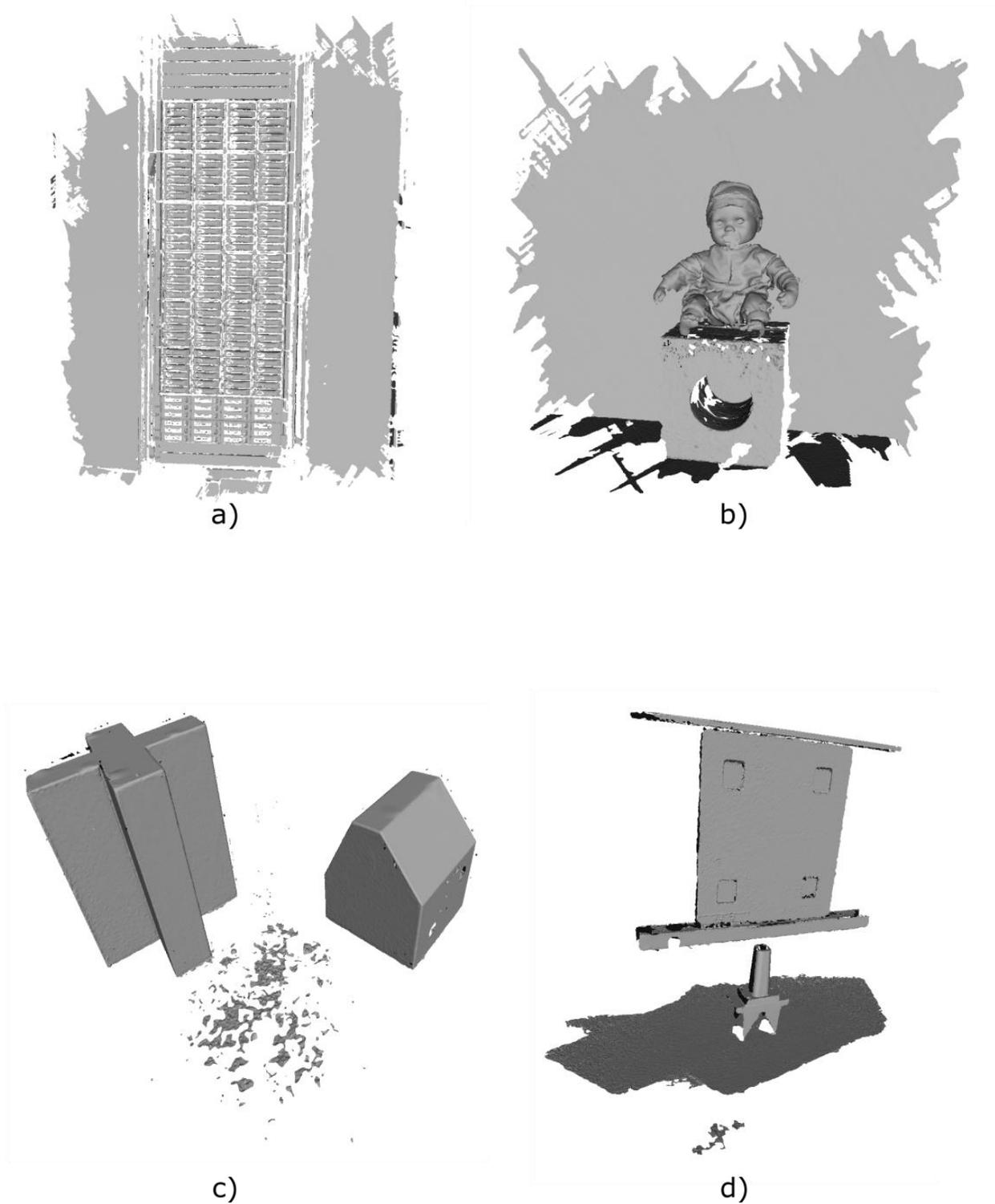
**Tableau 4-** Caractéristiques et spécifications des modèles étudiés

Modèle	Taille initiale [KB]	Sommets	Faces	Ratio S/F	Îlots	Trous	Sommets <i>non-manifold</i>
Bébé	6,962	195,077	368,268	0.53	156	41	1
Cible	10,722	295,760	571,491	0.52	9	103	10
Blocs	7,029	194,040	374,493	0.52	403	15	1
Ours	11,207	312,500	594,287	0.53	271	45	2
Serveurs	17,876	526,153	922,326	0.57	1074	321	4
Tête	1,109	30,347	59,296	0.51	17	3	1
Crâne	4,721	132,913	248,811	0.53	10	12	4
Plan	4,640	127,267	247,944	0.51	23	4	0
Boîte	1,438	39,435	76,843	0.51	7	15	0

Les modèles ont été sélectionnés afin d’avoir un échantillon varié selon la quantité de faces et de sommets, mais également selon nombre d’artefacts présents dans deux modèles ayant sensiblement la même taille.



**Figure 35** - Modèles utilisés pour l'évaluation des performances de l'approche étudiée : a) Ours; b) Boîte de conserve; c) Surface plane; d) Tête; e) Crâne.



**Figure 36** - Modèles utilisés pour l'évaluation des performances de l'approche étudiée: a) Ensemble de serveurs scannés à l'intérieur du Supercalculateur Colosse de l'Université Laval; b) Bébé; c) Blocs d'une croix et d'une maison; d) Cible de calibration.

## 5.3. Présentation des résultats

### 5.3.1. Taux de compression

Le Tab. 5 présente les résultats avant et suivant la compression par *Edgebreaker* et ayant été prétraités avec notre approche. Tous les fichiers utilisés pour l'évaluation des tailles de fichier sont encodés en format binaire car leur taille est près de 2 fois moins élevée que pour le format ASCII. Le Tab. 6 permet de comparer les taux de compression conformément aux données présentées au Tab. 5. Il est à noter également que les modèles récupérés suite à la décompression sont identiques aux modèles non-compressés à l'exception de la réparation des points *non-manifold* des modèles.

**Tableau 5** - Taille des modèles avant et après l'application du processus de compression d'*Edgebreaker* ainsi que la contribution des logiciels de compression WinZip et 7-Zip avant et après *Edgebreaker*

Modèle	Taille avant Edgebreaker [KB]			Taille après Edgebreaker [KB]		
	Sans compression	WinZip	7-Zip	Edgebreaker	Edgebreaker + WinZip	Edgebreaker + 7-Zip
Bébé	6,962	2,778	2,146	5,172	1,464	995
Cible	10,722	3,792	2,885	7,840	1,995	1,285
Blocs	7,029	3,097	2,527	5,169	1,397	1,001
Ours	11,207	4,476	3,493	8,284	2,281	1,583
Serveurs	17,876	7,644	5,991	13,971	4,080	3,046
Tête	1,109	472	380	805	214	160
Crâne	4,721	2,347	1,649	3,348	1,498	1,195
Plan	4,640	1,608	1,185	3,370	878	560
Boîte	1,438	599	473	1,044	279	210

**Tableau 6** - Taille initiale et taux de compression obtenues avec les différents outils ainsi que des combinaisons possibles de ceux-ci

Modèle	Taille initiale [KB]	Taux de compression [%]				
		WinZip	7-Zip	Edgebreaker	Edgebreaker + WinZip	Edgebreaker + 7-Zip
Bébé	6,962	60.1	69.2	25.7	79.0	85.7
Cible	10,722	64.6	73.1	26.9	81.4	88.0
Blocs	7,029	55.9	64.0	26.5	80.1	85.8
Ours	11,207	60.1	68.8	26.1	79.6	85.9
Serveurs	17,876	57.2	66.5	21.8	77.2	83.0
Tête	1,109	57.4	65.7	27.4	80.7	85.6
Crâne	4,721	50.3	65.1	29.1	68.3	74.7
Plan	4,640	65.3	74.5	27.4	81.1	87.9
Boîte	1,438	58.3	67.1	27.4	80.6	85.4
Moyenne	-	58.8	68.2	26.5	78.7	84.7

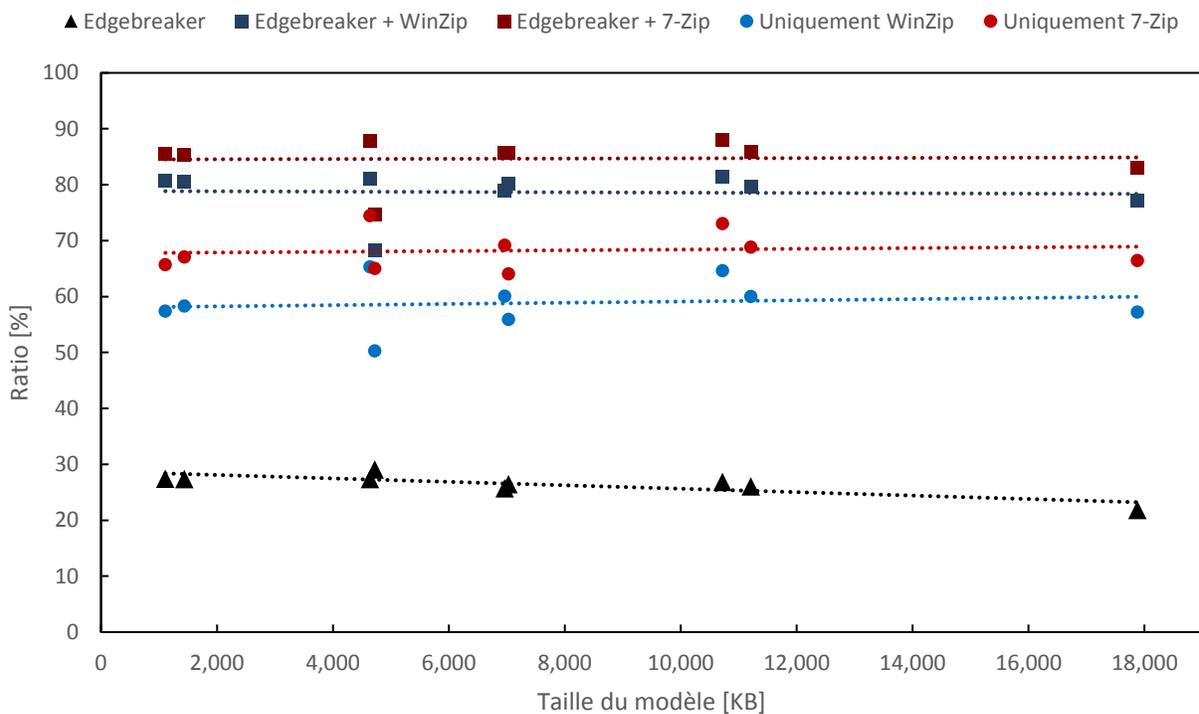
En analysant les Tab. 5 et 6, on constate qu'*Edgebreaker* permet d'obtenir un taux de compression inférieur à ce que WinZip et 7-Zip seul permette d'obtenir mais en combinant l'utilisation de notre approche d'*Edgebreaker* avec les 2 types de compression. La combinaison des deux méthodes ne se traduit pas nécessairement par une addition des taux de compression mais tend à s'y approcher.

À partir des taux de compression obtenus au Tab. 6 et en se servant des caractéristiques intrinsèques des modèles compressés présentées au Tab. 4, il est possible de constater une certaine corrélation entre la quantité d'artefacts présent dans le modèle et le taux de compression obtenu par la compression *Edgebreaker* en raison de l'ajout de faces pour corriger ces artefacts. De plus, *Edgebreaker* ne peut pas compresser l'information géométrique des modèles. Pour des modèles contenant un plus haut ratio S/F aura un taux de compression plus faible suite à une compression *Edgebreaker*. Ces deux facteurs expliquent la variation du taux de compression pour les modèles étudiés principalement pour le modèle *Serveurs* qui a le plus haut ratio S/F des modèles et donc le plus faible taux de compression par *Edgebreaker*.

Une donnée intéressante dans les taux calculés est la faible performance des algorithmes de compression WinZip et 7-Zip sur le modèle de crâne comparativement aux autres modèles. Cette différence est causée par un nombre supérieur de décimales aux coordonnées géométriques du modèle. Puisque cette information n'influence pas la compression *Edgebreaker*, le taux obtenu

est semblable au taux des autres modèles. Cependant, ces décimales supplémentaires ajoutent plus d'information pour WinZip et 7-Zip ce qui réduit le taux de compression.

Les taux de compression suivent l'équation 2.1 :  $T[\%] = \frac{P_i - P_f}{P_i} \cdot 100$  où  $P_i$  est la taille du fichier avant compression et  $P_f$  est la taille du fichier suivant la compression. Plus le taux est élevé, meilleure est la compression.



**Figure 37** - Graphique des taux de compression des différentes combinaisons d'outils en fonction de la taille des modèles

### 5.3.2. Comparaison des résultats obtenus avec notre méthode avec ceux obtenus par l'approche de Zhu

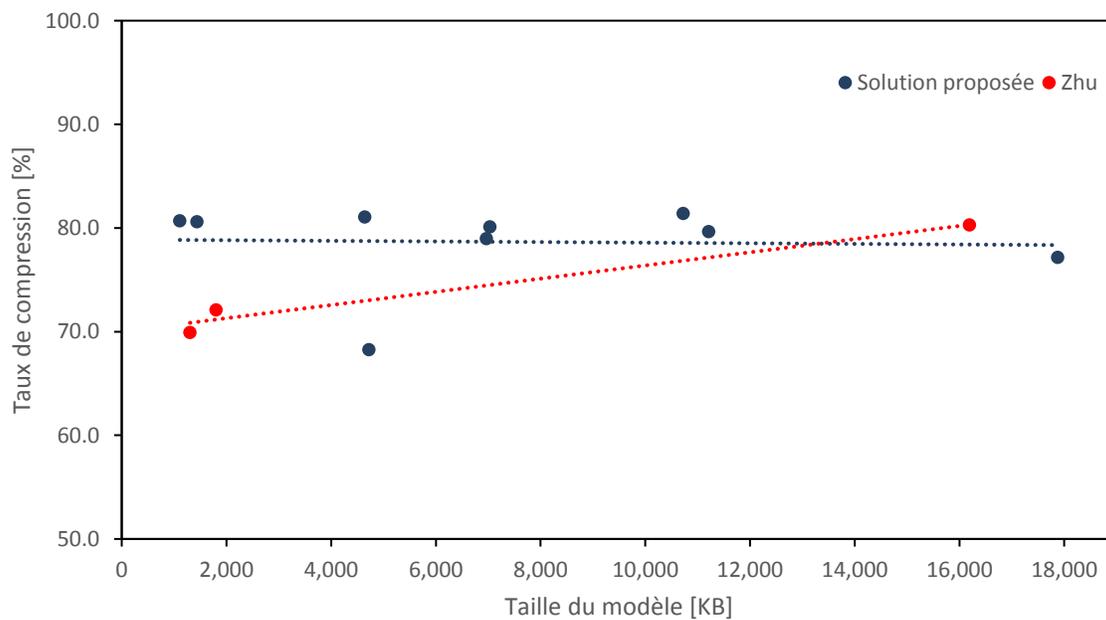
À des fins de comparaison, seulement les modèles de Zhu aillant les mêmes caractéristiques (information uniquement sur les sommets et les faces) que les modèles présentés dans ce travail sont comparés ainsi que le nombre de décimales sur l'information géométrique. Le Tab. 7 et la Fig. 38 présentent une comparaison des performances des deux approches. Les taux de compression de Zhu présentés dans son mémoire [2] sont convertis (100 - le taux de

*compression de Zhu*) afin de concorder avec la définition utilisée dans ce travail suivant l'équation 2.1. Les taux illustrés sont résultants de l'application d'*Edgebreaker* puis de WinZip.

**Tableau 7** - Tableau de comparaison des taux de compression des modèles présentés et des modèles de Zhu

Modèle	Taille initiale [KB]	Taux de compression [%]
Bébé	6,962	79.0
Cible	10,722	81.4
Blocs	7,029	80.1
Ours	11,207	79.6
Serveurs	17,876	77.2
Tête	1,109	80.7
Crâne	4,721	68.3
Plan	4,640	81.1
Boîte	1,438	80.6
Bunny	1,304	69.9
Horse	1,800	72.1
Dragon	16,192	80.3

La section inférieure du tableau correspond aux modèles de Zhu



**Figure 38** - Graphique des taux de compression en fonction de la taille des modèles selon l'approche proposée et selon l'approche de Zhu

Il est difficile d'affirmer si l'approche proposée dans ce travail donne de meilleurs résultats que ceux de Zhu. Pour tous les modèles étudiés dans cette section, les taux de compression sont contenus dans la même région (70%-80%). Avec un plus grand échantillonnage de modèles, il serait plus facile de conclure si une méthode se démarque davantage des deux approches.

### 5.3.3. Temps de calcul

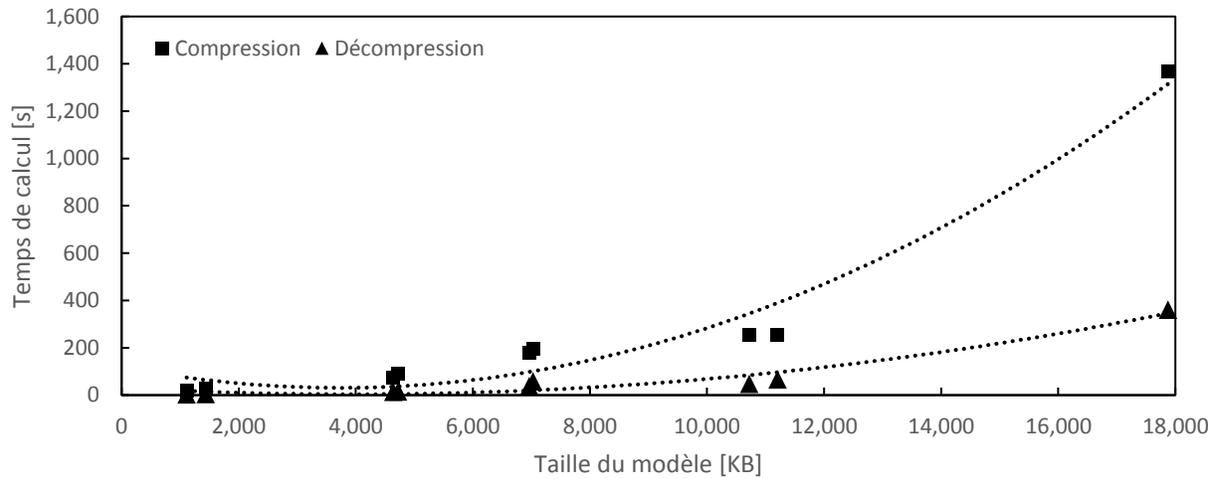
Les temps de calcul ont également été comptabilisés et sont présentés au Tab. 8, à la Fig. 39 et à la Fig. 40. Les temps de compression et de décompression sont globaux, ce qui implique que le temps de calcul de la compression inclut le prétraitement effectué par notre approche et le processus d'*Edgebreaker* et que le temps de décompression inclut le processus *Wrap&Zip* et le post-traitement également effectué par notre approche.

**Tableau 8** - Taille initiale des modèles et temps de calcul totaux pour la compression et la décompression en plus du temps de calcul des 4 principales phases du processus de compression

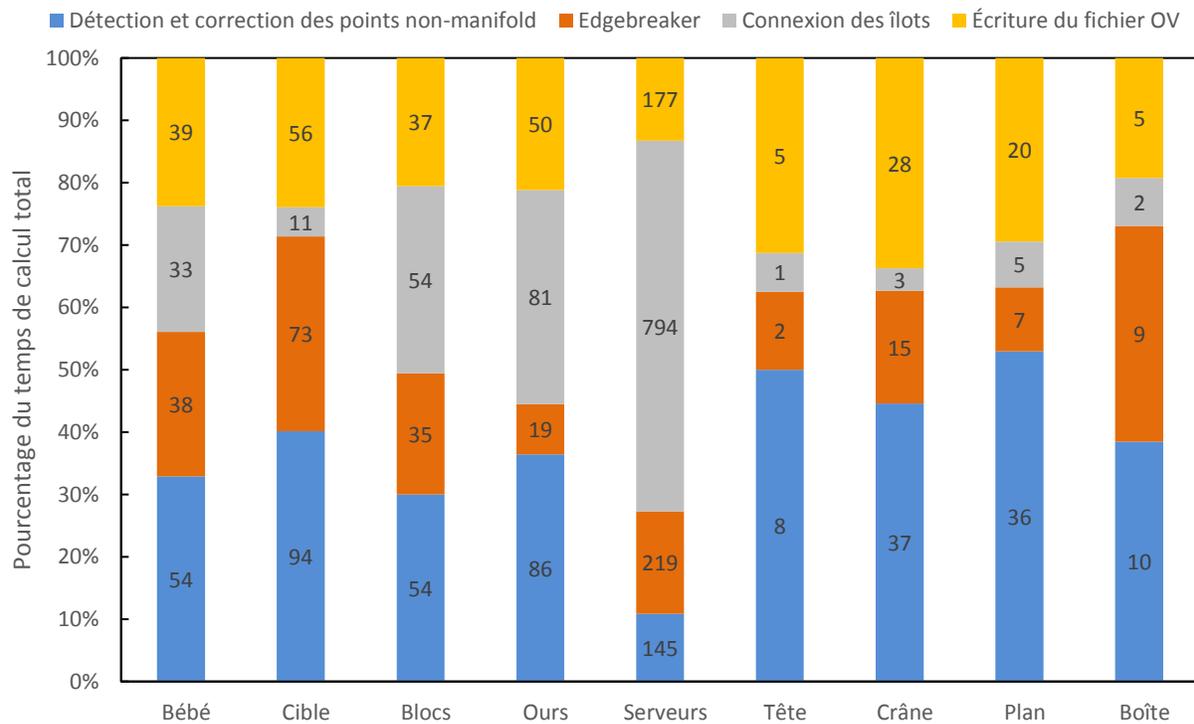
Modèle	Taille initiale [KB]	Temps de calcul globaux [s]		Détection et correction des points non-manifold [s]	Edgebreaker [s]	Connexion des îlots [s]	Écriture du fichier OV [s]
		Compression*	Décompression**				
Bébé	6,962	178	38	54	38	33	39
Cible	10,722	254	47	94	73	11	56
Blocs	7,029	196	59	54	35	54	37
Ours	11,207	255	67	86	19	81	50
Serveurs	17,876	1,368	361	145	219	794	177
Tête	1,109	18	3	8	2	1	5
Crâne	4,721	90	15	37	15	3	28
Plan	4,640	73	14	36	7	5	20
Boîte	1,438	28	4	10	9	2	5

\*Le temps de calcul pour la compression inclut le prétraitement et Edgebreaker

\*\*Le temps de calcul de décompression inclut le post-traitement et Wrap&Zip



**Figure 39** - Graphique du temps de calcul de la compression et de la décompression en fonction de la taille du modèle



**Figure 40** - Histogramme du temps alloué par les quatre phases principales du processus de compression

De par l'étude du Tab. 8 et de la Fig. 39, on constate que les temps de calcul enregistrés sont beaucoup trop élevés et croissent de manière non-linéaire en fonction de la taille des modèles. Les courbes de tendance pour la compression et la décompression illustrées suivent une interpolation polynomiale de 2<sup>e</sup> degré. Cette tendance peut s'expliquer par l'utilisation de la fonction `dsearchn()` dans Matlab qui a une complexité de  $O(n^2)$ . Cette fonction compare deux ensemble de points et calcul la distance géométrique entre tous les points de deux ensembles de points distincts. Dans le cadre de notre approche, cette fonction est utilisée pour trouver le couple de sommets les plus rapprochés dans deux périmètres d'îlots différents. Pour les modèles comportant un grand nombre d'îlots, cette fonction devient un facteur important dans le temps de calcul.

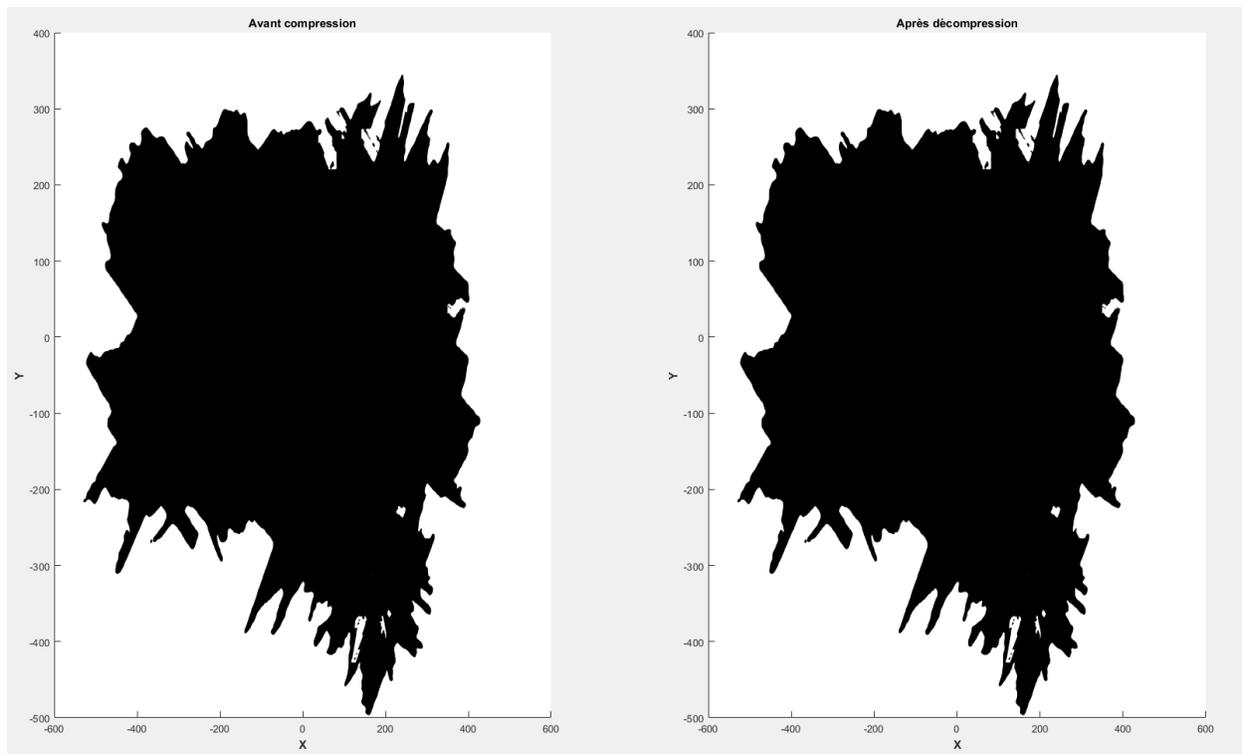
Le Tab. 8 affiche également le temps de calcul des 4 principales phases du processus de compression. Le temps pris par les sections moins importantes (en termes de temps de calcul) n'est pas affiché et explique la différence entre le temps total de la compression et la somme du temps de calcul des 4 phases. Ces phases moins importantes sont, par exemple, l'ouverture des fichiers, la quantification, l'encodage par Huffman.

D'autre part, l'étude de la Fig. 40 révèle que le temps de calcul total de la compression est largement influencé par les caractéristiques spécifiques des modèles. Les modèles contenant un grand nombre d'îlots verront leur temps de calcul s'accroître en raison de l'approche non-linéaire de la solution proposée d'où les variations observées selon les modèles. On observe également qu'une portion non-négligeable du temps de calcul total est attribuable à la détection des points *non-manifold*. Une attention particulière devra être portée à l'approche présentement utilisée de manière à trouver une méthode plus optimisée.

#### 5.3.4. Comparaison des modèles

Une fois compressés puis décompressés, les modèles sont comparés par la méthode décrite à la section 4.8.2. Tous les modèles ayant été présentés à la section 5.2 ayant été compressés et décompressés par notre méthode sont identiques aux modèles initiaux à l'exception des sommets

clonés pour ne pas reconstruire des singularités *non-manifold*. La Fig. 41 illustre le modèle de la surface plane avant la compression puis après la décompression.



**Figure 41** - Comparaison du modèle de la surface plane avant le processus de compression et après la décompression. La figure illustre uniquement le plan XY du modèle.

## 5.4. Analyse des résultats et discussion

Une première observation que l'on peut faire suite à l'analyse du Tab. 5 et Tab. 6 est que le taux de compression est maximisé lorsque les fichiers sont compressés par la combinaison *Edgebreaker* avec 7-Zip. Bien qu'*Edgebreaker* ne permet d'atteindre en moyenne un taux de compression de 26.5%, sa combinaison avec WinZip ou 7-Zip permet d'atteindre des taux moyens de 78.7% pour WinZip et de 84.7% pour 7-Zip alors que la compression sans l'application d'*Edgebreaker* n'atteint que des taux moyens de 58.8% pour WinZip et 68.2% pour 7-Zip. La pertinence de l'ajout d'*Edgebreaker* aux compresseurs classiques est nécessaire afin d'atteindre un taux maximal.

Une seconde observation qui est visible principalement à la Fig. 37 est une variance faible des taux de compression en fonction de la taille des modèles. Une variation non-négligeable est visible pour le modèle Plan et Crâne alors que leur taille est très semblable. La compression par *Edgebreaker* est meilleure pour le Crâne (29.1%) que pour le Plan (27.4%) en raison du nombre de faces supérieur. L'impact des artefacts ne peut pas expliquer ce taux compte tenu du nombre similaire d'artefacts (26 et 27 respectivement).

Bien que la taille des fichiers ait un certain impact sur le taux de compression, le taux est davantage restreint par le ratio de la quantité de sommets sur la quantité de faces (ratio S/F) présents dans le modèle car l'information géométrique est seulement compressée par les algorithmes utilisés par WinZip et 7-Zip. Le modèle ayant le plus haut ratio S/F présente le plus faible taux de compression fait par notre version d'*Edgebreaker* (modèle Serveurs à 21.8%). De meilleurs taux seraient probablement obtenus si l'information géométrique était compressée par l'une des méthodes mentionnées dans la revue de la littérature tel que la prédiction par parallélogramme de Touma et Gotsman [19] ainsi que par l'approche proposée par Isenburg et al. [18] [20]. L'information géométrique pourrait également être compressée par la méthode de Lee et al. [23] avec leur définition d'un nouvel espace de représentation de données (l'espace modèle).

Une comparaison des performances de l'approche de ce travail avec celle de Zhu ne permet pas de conclure qu'une approche est plus performante que l'autre. Ces comportements ne sont pas surprenants compte tenu que la même méthode de compression (*Edgebreaker*) est utilisée dans les deux travaux. Dans les deux approches, les taux de compression oscillent entre 70% et 80%. Une part de la différence entre les taux de compression peut également être attribuable à la version de WinZip utilisée.

Bien que le temps de calcul n'était pas un objectif du travail, une brève évaluation a démontré que le temps de calcul pour la compression est beaucoup plus long que la décompression tel que le montre la Fig.39. Contrairement aux temps de compression de 14 secondes obtenus par l'implémentation de Zhu [2] sur un modèle de 2 millions de triangles, l'approche proposée prends près de 1400 secondes pour la compression d'un modèle de 922 000 triangles. La décompression se fait plus rapidement car la grande partie de ce calcul se fait en C++ via Visual Studio, un langage plus performant qu'en Matlab. Ce temps accru de la compression par rapport à la décompression

s'explique également par les étapes de prétraitement où les artefacts sont corrigés mais également par l'écriture du fichier *Opposite-Vertex* (OV). Un moyen de diminuer ce temps de calcul serait d'implanter l'entièreté du code en C++ dans Visual Studio contrairement en Matlab comme il est le cas actuellement. En plus de pouvoir profiter d'une approche d'optimisation développée spécifiquement en C++, il ne serait plus nécessaire d'écrire un fichier de transition OV; ce fichier étant utilisé pour transmettre l'information géométrique et topologique du modèle. L'optimisation des processus de connexion des îlots et de détection des points *non-manifold* doit également être effectué.

## 5.5. Conclusion

Les résultats présentés dans ce chapitre permettent de confirmer la fiabilité et la versatilité des techniques de prétraitement apportées à l'algorithme d'origine *Edgebreaker* en corrigeant les singularités et spécificités dans la topologie qui sont incompatibles avec le processus de compression d'origine.

Une évaluation des différentes combinaisons possibles de logiciels de compression a permis de conclure que la combinaison d'*Edgebreaker* avec 7-Zip permet d'atteindre des taux de compression jusqu'à 88% de la taille du modèle initial. Cependant, il est difficile de conclure si l'approche proposée compresse mieux les modèles que celle de Zhu car trop de facteurs comme la quantité d'artefacts, la quantité de sommets, la version de WinZip, etc. peuvent influencer les taux de compression.

Une attention particulière est portée aux temps de calcul pour la compression. Les temps affichés au Tab. 8 et Fig. 39 présentent des temps beaucoup trop imposants pour utiliser aisément l'algorithme sur des modèles très gros. Ces temps ont le potentiel d'être diminués si l'entièreté du code source est implanté en C++.

## Chapitre 6. Conclusion

Dans un contexte du développement d'outils servant aux traitements métrologiques, ce travail se penche sur la compression de la taille informatique de modèles triangulaires 3D. Plus précisément, le travail présente une addition à la version actuelle de compression par *Edgebreaker* et de décompression par *Wrap&Zip* en traitant des artefacts communs pouvant apparaître sur des maillages triangulaires numérisés par un scanneur 3D. Tel que mentionné dans ce travail, *Edgebreaker* ne peut compresser les modèles qui ne sont pas contenus en un seul regroupement de faces fermé.

Des étapes de prétraitement ont été ajoutées à *Edgebreaker* afin que les scans bruts obtenus par un scanneur 3D puissent être compressés sans perte de l'information géométrique et topologique. Les quatre principales causes menant *Edgebreaker* à l'échec dans le traitement de maillages triangulaires sont les singularités non-manifold, les trous, les îlots ainsi que les volumes indépendants. Afin de corriger les artefacts non-manifold, un sommet est dupliqué et déplacé d'un certain incrément. Les coordonnées du point dupliqué sont sauvegardées dans un fichier nommé "Autre" et qui est utilisé lors de la décompression. Les coordonnées de ce sommet sont ensuite recalées afin de retrouver sa position d'origine. Les trous sont ensuite détectés selon les connexions entre les chaînes de frontière et les regroupements de face, puis un sommet est ajouté au centre géométrique du trou et des faces additionnelles ferment le trou. Les coordonnées des sommets sont également ajoutées au fichier "Autre". Une fois les trous fermés, le modèle n'est constitué que de surfaces et de volumes flottants. Des ponts sont construits de manière à obtenir une seule grande surface à laquelle un *dummy vertex* et des faces sont ajoutées pour fermer le modèle. La topologie des ponts a été sélectionnée afin de permettre à *Edgebreaker* d'atteindre les îlots avec un nombre minimal de bits rajoutés. Les coordonnées des sommets créant les ponts et celles du *dummy vertex* sont ajoutées à la liste des sommets du fichier "Autre". Lors de la décompression, ces points sont supprimés pour retrouver ainsi la topologie d'origine.

Les ajouts à l'algorithme de compression ont été testés sur un échantillon varié de modèles présentant différentes caractéristiques problématiques à la version d'origine d'*Edgebreaker*. Les résultats obtenus atteignent des taux de compression comparable aux résultats de Zhu [2] pour des conditions similaires.

Les prochaines étapes pertinentes au développement de l'approche, en plus d'incorporer un algorithme de compression de l'information géométrique efficace et d'optimiser le code source dans le but de réduire le temps de compression, seraient de compresser et de transmettre les informations complémentaires du modèle. Ces informations peuvent être les normales aux faces et aux sommets ou la colorisation RGB et RGBA.

## Bibliographie

- [1] J. Rossignac, Edgebreaker: Connectivity compression for triangle mesh, IEEE Transactions on Visualization and Computer Graphics 5, 1999, pp. 47-61.
- [2] J. Zhu, Lossless triangle mesh compression, Queen's University, 2013.
- [3] N.D., Chaire de recherche industrielle CRSNG-Creaform sur la numérisation 3D : Creation-3D, Vol. 1 de 2 [https://www.ulaval.ca/fileadmin/ulaval\\_ca/Images/recherche/Documents/chaire-partenariat/chaire-recherche-industrielle-CRSNG-Creaform-numerisation-creation-3D.pdf](https://www.ulaval.ca/fileadmin/ulaval_ca/Images/recherche/Documents/chaire-partenariat/chaire-recherche-industrielle-CRSNG-Creaform-numerisation-creation-3D.pdf), CRSNG - NSERC, 2011.
- [4] D. Lee et B. Schachter, Two algorithms for constructing a Delaunay triangulation, vol. 9, International Journal of Computer & Information Sciences, 1980, pp. 219-242.
- [5] M. Sharir et E. Welzl, Random triangulations of planar point sets, ACM éd., 22nd Annual Symposium Computer Geometry (SoCG), 2006, pp. 273-281.
- [6] L. L. Jackson, R. S. Heffner et H. E. Heffner, Free-field audiogram of the Japanese macaque (*Macaca fuscata*), vol. 106, Journals of the Acoustical Society of America, 1999, pp. 3017-3023.
- [7] Z. Shelby et C. Bormann, 6LoWPAN: The wireless embedded internet, Wiley Series on Communications Networking & Distributed Systems. John Wiley & Sons, 2011.
- [8] P. Bourke, «PLY - Polygon File Format,» [En ligne]. Available: <http://paulbourke.net/dataformats/ply/>. [Accès le 13 mai 2016].
- [9] M. Vatani, A. R. Rahimi, F. Brazandeh et A. Sanati nezhad, Distance Analysis for Layer Manufacturing, vol. 3, World Academy of Science, Engineering and Technology:

International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering, 2009.

- [10] X. Qu et B. Stucker, A 3D surface offset method for STL-format models, vol. 9, Rapid Prototyping Journal, 2003, pp. 133-141.
- [11] G. Turk, Re-tiling polygonal surfaces, Computer graphics, 1992, pp. 55-64.
- [12] W. J. Schroeder, J. A. Zarge et W. E. Lorensen, Decimation of triangle meshes, Computer Graphics, 1992, pp. 65-70.
- [13] J. Li et C.-C. Jay Kuo, Embedded coding of 3D graphic models, vol. 1, ICIP '97, 1997, pp. 57-60.
- [14] S. Valette, R. Chaine et R. Prost, Progressive lossless mesh compression via incremental parametric refinement, vol. 28, Eurographics Symposium on Geometry Processing, 2009, pp. 1301-1310.
- [15] K. Köse, 3D model compression using image compression based methods, Master of science thesis. BilKent University, 2007.
- [16] Z. Abderrahim, E. Techini et M. S. Bouhleb, Progressive compression of 3D objects with an Adaptive quantization, vol. 10, Computational Geometry, 2013, pp. 504-511.
- [17] I. H. Witten, M. N. Radford et J. G. Cleary, Arithmetic coding for data compression, vol. 6, Communications of the ACM 30, 1987, pp. 520-540.
- [18] M. Isenburg et P. Alliez, Compressing polygon mesh geometry with parallelogram prediction, IEEE Visualization, 2002, pp. 141-146.
- [19] C. Tuma et C. Gotsman, Triangle mesh compression, Graphics Interface '98, 1998, pp. 26-34.

- [20] M. Isenburg, P. Lindstrom et J. Snoeyink, Lossless compression of predicted floating-point geometry, vol. 37, *Computer-Aided Design*, 2005, pp. 869-877.
- [21] H. Lee, P. Alliez et M. Desbrun, Angle-analyzer: A triangle-quad mesh codec, vol. 21 (3), *EUROGRAPHICS 2002*, 2002.
- [22] A. Szymczak, J. Rossignac et D. King, Piecewise regular meshes: Construction and compression, *Graphics Models*, 2003.
- [23] E.-S. Lee et H.-S. Ko, Vertex data Compression for triangular meshes, *Computer Graphics and Applications*, 2000, pp. 225-234.
- [24] M. Deering, Geometry compression, *Computer Graphics*, 1995, pp. 13-20.
- [25] Y. Liu et E. Wu, Connectivity compression for non-triangular meshes by context-based arithmetic coding, *ACM SIGGRAPH GRAPHITE 2006*, 2006, pp. 417-424.
- [26] B. Kronrod et C. Gotsman, Efficient coding of non-triangular mesh connectivity, vol. 63, *Graphical Model*, 2001, pp. 263-275.
- [27] P. Alliez et M. Desbrun, Valence-driven connectivity encoding for 3D meshes, vol. 20, *EUROGRAPHICS*, 2001, pp. 480-489.
- [28] J. Rossignac et A. Szymczak, Wrap&Zip decompression of the connectivity of triangle meshes compressed with Edgebreaker, *Computational Geometry XX*, 1999, pp. 1 - 15.
- [29] J. Rossignac, H. Lopes, A. Safanova, G. Tavares et A. Szymczak, Edgebreaker: A simple compression scheme for surfaces with handles, vol. 27, *International Journal of Computers and Graphics*, 2003, pp. 553-567.
- [30] D. King, J. Rossignac et A. Szymczak, Connectivity compression for irregular quadrilateral meshes, *GVU Tech Report GVV-GIT-99-36*, 2000.

- [31] A. Szymczak, D. King et J. Rossignac, An Edgebreaker-based efficient compression scheme for regular meshes, vol. 20, Computational Geometry, 2001, pp. 53-68.
- [32] L. Váša et V. Skala, CoDDyaC: Connectivity driven dynamic mesh compression, 3DTV-CON: The True Vision - Capture, Transmission and Display of 3D Video, IEEE, 2007, pp. 1-4.
- [33] A. Khodakovsky, P. Schröder et W. Sweldens, Progressive geometry compression, In SIGGRAPH, 2000, pp. 271-278.
- [34] C. Roudet, F. Dupont et A. Baskurt, Semi-regular 3D mesh progressive compression and transmission based on an adaptive wavelet decomposition, SPIE Electronic imaging - Wavelet applications in industrial processing, 2009.
- [35] C. Zhao, H. Sun et K. Qin, Progressive wavelets compression for 3d models, Computer Graphics International, 2010.
- [36] S. Valette et R. Prost, Wavelet-based progressive compression scheme for triangular meshes: Wavemesh, vol. 10, IEEE Transactions on Visualization and computer graphics, 2004.
- [37] H. Benoit-Cattin, A. Baskurt, F. Turjman et R. Prost, 3D medical image coding using separable 3D wavelet decomposition and lattice vector quantization, vol. 59, Signal Processing, 1997, pp. 139-153.
- [38] A. Maglo, C. Courbet, P. Alliez et C. Hudelot, Progressive compression of manifold polygon meshes, vol. 36, Computers & Graphics, 2012, pp. 349-359.
- [39] H. Hoppe, Progressive Meshes, In SIGGRAPH, 1996, pp. 99-108.
- [40] F. Caillaud, V. Vidal, F. Dupont et G. Lavoué, Progressive compression of arbitrary textured meshes, vol. 35, Pacific Graphics, 2016.

- [41] K. Mamou, C. Dehais, F. Chaieb et F. Ghorbel, Shape approximation for efficient progressive mesh compression, vol. 1, IEEE 17th International Conference on Image Processing, 2010, pp. 3425-3428.
- [42] Y.-S. Kim, D.-G. Park, H.-y. Jung et H.-G. Cho, An improved TIN compression using Delaunay triangulation, vol. 15, International Journal of GIS, 2001, pp. 255-269.
- [43] Z. Karni et C. Gotsman, Spectral compression of mesh geometry, SIGGRAPH '00, 2000.
- [44] Z. Karni et C. Gotsman, 3D mesh compression using fixed spectral bases, Graphics Interface, 2001, pp. 1-8.
- [45] P. Alliez, G. Ucelli, C. Gotsman et M. Attene, Recent Advances in Remeshing of Surfaces, Springer, 2005.
- [46] P. Alliez et C. Gotsman, Recent advances in compression of 3D meshes, Symposium Multiresolution in Geometric modeling, 2003.
- [47] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery et W. Stuetzle, Multiresolution analysis of arbitrary meshes, Computer Graphics, 1995, pp. 173-182.
- [48] T. Vijayaraghavan et K. Rajan, Image coding of 3D volume using wavelet transform for fast retrieval of 2D images, vol. 153, IEEE Image Signal Process, 2006.
- [49] M. Isenburg et S. Gumhold, Out-of-core compression for gigantic polygon meshes, SIGGRAPH, 2003.
- [50] P. Deutsch, DEFLATE Compressed data format specification version 1.3. RFC 1951, Internet Engineering Task Force, 1996.
- [51] A. Safonova, *Edgebreaker: 3D Compression Software*, Georgia, 2001.
- [52] P. Getreuer, *PLYREAD*, MathWorks, 2004.

[53] N.D., «Creaform 3D, Ametek,» [En ligne]. Available: <http://www.creaform3d.com/en>.

[54] A. Jacobson, «is\_vertex\_nonmanifold.m,» 2015. [En ligne]. Available: [https://github.com/alecjacobson/gptoolbox/blob/master/mesh/is\\_vertex\\_nonmanifold.m](https://github.com/alecjacobson/gptoolbox/blob/master/mesh/is_vertex_nonmanifold.m).

[55] G. Peyré, «Convertisseur PLY-OV,» 2008. [En ligne]. Available: [https://github.com/fethallah/geodesics4meshes/blob/master/matlab/toolbox/write\\_ov.m](https://github.com/fethallah/geodesics4meshes/blob/master/matlab/toolbox/write_ov.m).