

InvoCoder : compression de la connectivité de maillages en dimension arbitraire

Sylvain Prat
France Telecom R&D
sylvain.prat@rd.francetelecom.com

Patrick Gioia
France Telecom R&D
patrick.gioia@rd.francetelecom.com

Yves Bertrand
Laboratoire SIC
Poitiers, France
yves.bertrand@univ-poitiers.fr

Daniel Meneveau
Laboratoire SIC
Poitiers, France
daniel@sic.univ-poitiers.fr

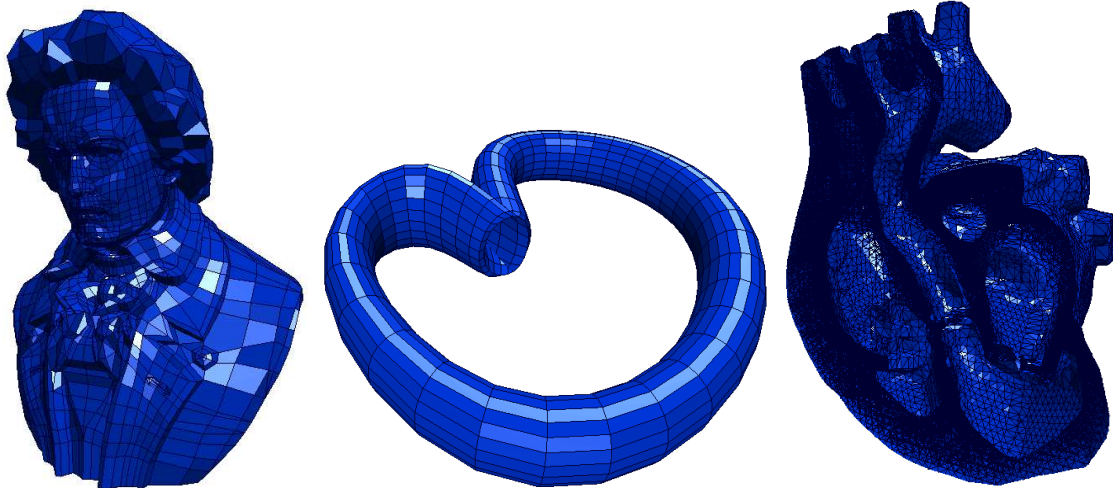


Figure 1 : diverses représentations compressées de maillages

1 Résumé

La compression de modèles géométriques est un enjeu important ces dernières années. En effet, les modèles deviennent de plus en plus volumineux, ce qui pose des problèmes de stockage sur disque ou de latence lors de transmission sur des réseaux de télécommunication. Les méthodes de compression basées connectivité remportent un franc succès puisqu'elles conservent la topologie initiale des maillages, la topologie étant nécessaire dans de nombreuses applications telles que certaines opérations de modélisation géométrique, pour la construction de structures performantes en visualisation (graphes de visibilité), pour la décimation de maillages (création de niveaux de détail), etc.

Nous proposons dans cet article la première technique de compression sans perte de la connectivité permettant de traiter des subdivisions variétés (manifold) en dimension quelconque : maillages surfaciques, volumiques, avec des cellules quelconques, avec ou sans bords, orientables ou non. Notre méthode s'appuie pour cela sur un modèle topologique appelé *carte généralisée* et généralise différentes approches de la compression basée connectivité. Nous effectuons un parcours de proche en proche des éléments de la carte généralisée en décrivant leurs connexions par des symboles afin de produire des historiques ensuite compressés par des techniques classiques. La compression des attributs géométriques est effectuée de manière indépendante en utilisant les techniques décrites dans les autres approches.

Mots clés : compression, connectivité, structure topologique, cartes généralisées

2 Introduction

La compression de modèles géométriques est l'un des problèmes majeurs posé ces dernières années. En effet, les modèles 3D issus de procédés d'acquisition automatiques tels que les scanners lasers du Digital Michelangelo Project [1] occupent toujours plus d'espace sur les périphériques de stockage ; d'autre part, la très faible bande passante disponible dans les applications de visualisation 3D au travers de réseaux ne permet pas une navigation interactive sans compression préalable.

Parmi les méthodes de compression de modèles géométriques, les approches basées connectivité sont souvent utilisées car elles conservent la topologie initiale du modèle, c'est-à-dire les relations de voisinage entre les sommets, arêtes, faces, etc. En effet, les informations topologiques sont importantes pour certaines opérations de modélisation géométrique, pour la construction des structures performantes de visualisation telles que les graphes de visibilité ou pour la décimation de maillages ou création de niveaux de détail... Dans ces méthodes, la géométrie associée à la topologie est prédite en fonction du voisinage déjà décodé et est ensuite compressée selon l'ordre de parcours imposé par la compression de la connectivité.

Il existe aujourd'hui de nombreuses méthodes permettant de compresser la topologie de modèles géométriques : cela concerne les maillages triangulaires [2], [3], [4], [5], [6], les maillages surfaciques [7], [8] puis les maillages tétraédriques [9], [10], [11] ou encore hexaédriques [12]. Cependant, aucune de ces méthodes de compression n'est directement généralisable en dimension quelconque avec des cellules arbitraires. Les structures de données et algorithmes sont en effet très spécifiques aux types de subdivisions que l'on cherche à compresser. A notre connaissance, aucune méthode ne peut encore traiter les maillages volumiques irréguliers rencontrés en géologie ou les maillages volumiques de villes constitués de bâtiments à trous collés sur un sol par exemple. De plus, la représentation des bords ou des surfaces de genre élevé soulève de nombreux problèmes.

Nous avons adopté une structure topologique appelée carte généralisée (ou G-carte) [13] pour décrire la connectivité des subdivisions. Ce choix a été motivé par deux raisons principales :

- Elle permet de représenter des variétés quelconques quelque soit leur dimension, avec ou sans bords, orientables ou non, de genre quelconque ;
- Les relations entre les éléments de base de cette structure sont homogènes sur toutes les dimensions ; cela facilite la généralisation des algorithmes.

Dans cet article, nous proposons un procédé de compression sans perte de la connectivité qui repose sur une description compacte des cartes généralisées. Nous effectuons un parcours des éléments de base de cette structure en décrivant chacun d'eux par des symboles. L'ensemble de ces symboles forme des historiques qui représentent l'information topologique obtenue lors du parcours. Nous utilisons ensuite des techniques de compression de données classiques pour réduire la taille de ces historiques, comme par exemple un codeur entropique ou à dictionnaire. Les attributs géométriques associés sont compressés lors d'une étape distincte en utilisant des méthodes existantes.

Notre méthode peut traiter l'intégralité des modèles représentables sous forme de carte généralisée, quelque soit la dimension des subdivisions et le type de cellules qui les composent. Nous pouvons ainsi traiter les maillages surfaciques non-orientables tels que des rubans de Moëbius, bouteille de Klein, des maillages volumiques de tétraèdres, d'hexaèdres mais aussi de polyèdres quelconques, qui ont ou non des bords. De plus, la structure topologique utilisée peut-être employée dans de nombreuses autres applications, comme celles citées précédemment.

Les contributions de notre travail sont notamment :

- La définition d'une méthode générique de compression de la connectivité de modèles géométriques quelque soit leur dimension (topologique) et quelque soit le type de cellules, qui s'appuie notamment sur une structure appelée carte généralisée ;
- Un traitement simplifié des bords et des surfaces de genre élevé par rapport à certaines techniques classiques de compression de la connectivité.

La partie suivante de cet article présente un bref état de l'art des différentes méthodes de compression de la connectivité, elle est suivie de quelques définitions concernant les principaux termes utilisés. Nous décrivons ensuite les principes généraux puis le détail de notre nouvelle méthode de compression. Nous donnons enfin les résultats obtenus avant de conclure.

3 Travaux antérieurs

3.1 Terminologie

Nous appelons *subdivision* d'espace (ou maillage) une décomposition de l'espace en *cellules* (sommets, arêtes, faces, volumes, etc.). Un modèle topologique permet de représenter des subdivisions, il est constitué de deux types d'information : (i) la topologie (ou connectivité) décrivant les relations d'incidence et d'adjacence entre les cellules de la subdivision ; (ii) la géométrie (ou plongements) précisant les attributs d'apparence, de forme de ces cellules (par exemple la position des sommets, la texture des volumes ou des faces, la couleur des arêtes, etc.) Dans la suite, le terme *dimension* s'applique uniquement à la topologie et non aux plongements : un maillage surfacique tel que celui de la figure 1 est de dimension 2 même si les coordonnées des sommets sont définies dans un espace à 3 dimensions. Nous

appelons *codage* le passage d'une représentation à une autre, qui peut être plus compacte mais pas nécessairement, et *décodage* l'étape inverse.

3.2 Compression de modèles géométriques

On distingue 3 grandes familles de compression de modèles géométriques :

- La première procède par simplifications/raffinements successifs du maillage [14] : le triangle restant après simplification extrême et les différents raffinements qui permettent de reconstruire le maillage initial (éclatement de sommet la plupart du temps) sont représentés de manière compacte dans le fichier compressé ;
- La seconde s'attache à décrire la connectivité du maillage (la façon dont les éléments du maillage sont connectés entre eux) à l'aide d'un parcours, la géométrie est prédite grâce au voisinage déjà décodé ;
- Enfin la troisième procède à une analyse multi-résolution de la géométrie et à un remaillage régulier de la surface (décomposition en ondelettes [15] [16] ou par Fourier [17]).

Nous souhaitons conserver la topologie initiale des modèles (compression sans perte de la connectivité) car c'est une contrainte forte liée aux applications pratiques de notre technique : modélisation CAO, visites de villes virtuelles par exemple. Pour cette raison, nous avons écarté la troisième famille même si elle offre les meilleurs taux de compression. Parmi les deux méthodes restantes, la première offre la progressivité mais les taux de compressions sont très insuffisants par rapport à ceux de la seconde famille. C'est pourquoi nous privilégions cette seconde catégorie.

Nous supposons dans la suite que le modèle géométrique est composé d'une seule et unique composante connexe sans perte de généralité (il suffit de reproduire le schéma de codage/décodage pour chacune des composantes connexes). La quasi-totalité des techniques que nous citons imposent que les modèles soient des variétés topologiques. En pratique cette contrainte peut être contournée en dupliquant les cellules singulières des maillages non-variétés ou en effectuant un traitement local [18]. Par exemple dans le cas surfacique, une arête doit être incidente à deux faces au maximum. Si ce n'est pas le cas, l'arête est dupliquée afin de se ramener au cas variété.

Nous évoquons d'abord le cas des surfaces et en particulier les maillages triangulaires avant de décrire les généralisations pour les cas volumiques qui nous permettent de dégager un principe commun à toutes ces méthodes. Ensuite nous parlerons du codage de la géométrie proprement dite.

3.2.1 Cas surfacique

Taubin et Rossignac [2] ont développé l'une des premières méthodes de compression de maillages triangulaires utilisant la connectivité, dénommée « Topological Surgery ». Un découpage selon certaines arêtes du maillage dit « en pelure d'orange » permet d'obtenir un arbre des faces du maillage (i.e. chaque face est un nœud du graphe et les faces partageant une arête sont liées par une arête dans le graphe) ainsi qu'un arbre des sommets (ces deux représentations sont duales mais nécessaires). Ce découpage revient à chercher un arbre couvrant minimal du graphe des faces et permet ainsi de créer des bandes de triangles les plus longues possibles. Le procédé de compression consiste ensuite à représenter au mieux l'arbre des sommets et l'arbre des faces de la pelure : pour chaque branche de l'arbre, on indique sa longueur, si l'extrémité de la branche est une feuille ou un autre nœud de branchement et s'il y a une autre branche partant du nœud courant (sauf dans le cas de l'arbre des faces puisqu'il est binaire). La recherche de l'arbre couvrant minimal étant un problème NP-complet, différentes heuristiques ont été utilisées et un parcours en spirale semble donner les meilleurs résultats car il produit des branches plus longues et moins nombreuses. L'efficacité de ce parcours pour la compression est à l'origine du développement de nouvelles approches.

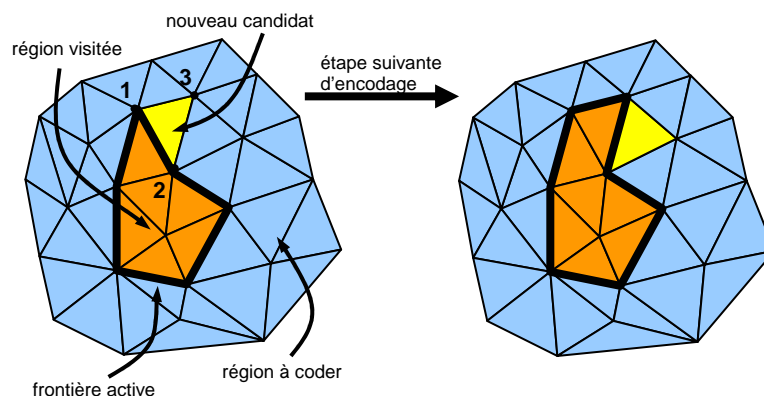


Figure 2 : parcours de proche en proche

Ces nouvelles approches suivent toutes un principe commun. Lors du processus de codage ou décodage de la connectivité, un maillage peut être décomposé en plusieurs régions : une région « visitée » qui comprend l'ensemble des éléments déjà encodés et une ou plusieurs régions (en fonction de la topologie du maillage) contenant les éléments restant. La frontière entre ces régions est appelée frontière active (ou « active-list », « cut-border »...). Le parcours du maillage est appelé croissance de région (« region-growing ») car chaque nouveau candidat à l'encodage/décodage est incident à la frontière active; cela fait grandir la région « visitée » comme le montre la figure 2. Un symbole est enregistré pour décrire chaque nouvel élément (triangles ici) en fonction de ses relations d'incidence par rapport à la frontière active. L'ensemble de ces symboles forme un historique ensuite compressé via des méthodes classiques de compression de données comme par exemple un codage entropique.

Par exemple, dans EdgeBreaker [4] (maillages triangulaires), il existe 5 symboles différents décrivant différentes configurations d'incidence du triangle courant par rapport à la frontière active. Ce triangle partageant déjà deux sommets avec la frontière active le long d'une des arêtes de cette frontière (sommets 1 et 2 sur la figure 2), le symbole enregistré concerne le troisième sommet du triangle :

- soit il se trouve hors de la frontière : écriture d'un symbole C signifiant l'introduction d'un nouveau sommet
- soit il se trouve dessus : symbole L s'il est immédiatement à gauche sur la frontière, R si c'est immédiatement à droite ou S si c'est ailleurs sur la frontière (la frontière active est découpée en deux parties qui seront traitées l'une après l'autre)
- soit ces 3 sommets forment le dernier triangle à traiter sur la frontière active : symbole E

Ces différents symboles sont compressés à l'aide d'un code à taille variable (un peu comme les codes de Huffman) : le symbole C largement plus fréquent n'occupe qu'un bit contre trois pour les autres codes. Il n'est pas nécessaire d'associer un indice pour identifier le sommet concerné sur la frontière active pour le cas S. En effet, un pré-traitement lors du décodage permet de retrouver ce sommet par comptage du nombre de symboles entre les paires (S,E) qui se correspondent. Cela garantit ainsi un coût de stockage linéaire en fonction du nombre de sommets pour des maillages simples. En revanche, pour la gestion des bords ou des surfaces de genre non-nul (homéomorphes à des tores à x trous), deux symboles supplémentaires M et M' sont nécessaires et associés à des indices de sommets, rendant ainsi le coût de stockage non linéaire. En pratique, ils ont peu d'influence sur la taille du fichier compressé car le genre ou le nombre de bords de la surface sont considérés comme faibles. Dans [3], une approche similaire est décrite, mieux adaptée à une utilisation interactive.

Dans [5] (maillages triangulaires également), les symboles correspondent aux valences des nouveaux sommets introduits. Plus exactement, un sommet arbitraire est choisi comme point de départ du parcours du maillage et pour chaque nouveau sommet rencontré sa valence v est enregistrée puis ses v arêtes incidentes sont visitées dans le sens des aiguilles d'une montre. Les arêtes sont traitées une à une et introduisent (pour certaines du moins) un nouveau sommet qui est ajouté dans la frontière active pour un traitement ultérieur. Lorsque toutes les arêtes issues d'un sommet sont traitées, nous procédons de même pour le sommet suivant dans la liste active, etc. Puisque les valences tendent à être régulières dans les maillages usuels (6 en moyenne), elles sont ensuite compressées par un codeur arithmétique. Bien que cet algorithme semble simple, deux cas spéciaux surviennent parfois. Deux symboles spéciaux appelés "division" et "fusion" ("split" et "merge") correspondent respectivement à une division du parcours en deux branches (lorsqu'il y a des triangles déjà traités parmi ceux incidents au sommet courant) et à une fusion de deux branches en une (lié au genre). Ces deux symboles sont associés à un indice permettant d'identifier le sommet incriminé sur la frontière, rendant la complexité en espace non linéaire par rapport au nombre de sommets. Pour représenter les bords, un faux sommet est introduit permettant de construire une face virtuelle pour boucher chaque bord : ils sont marqués de façon spéciale pour pouvoir être identifiés parmi les autres au moment du décodage.

Dans [6], une heuristique est proposée pour réduire le nombre de divisions et les indices associés, car ce sont les éléments les plus coûteux à stocker dans la représentation compressée. La liste active n'est plus traitée comme une file d'attente mais la priorité des sommets dépend du nombre d'arête restant à traiter, les plus petits nombres sont traités d'abord, ceci afin de limiter la formation de concavités sur la frontière active, principales responsables de l'apparition de division. Dans [7] et [8], cette dernière méthode est généralisée aux maillages polygonaux grâce à l'ajout des degrés des faces dans la représentation (i.e. le nombre de sommets des faces). Dans [19], Gotsman établit que quelque-soit l'heuristique utilisée, le nombre de divisions ne peut pas être négligé car l'entropie des valences seules est inférieure à l'entropie pour les maillages triangulaires ou plus généralement surfaciques, ce qui contredit le théorème de Tutte ([20], [21]).

3.2.2 Généralisation au cas volumique

Dans [9], une technique spécialisée pour les maillages tétraédriques est décrite, procédant à la construction d'un arbre de tétraèdres avec des opérations de recollage qui permettent de reconstituer la topologie originale, un peu de la même manière que Topological Surgery [2] pour les maillages triangulaires. L'approche décrite dans [10] généralise celle de EdgeBreaker [3] : des symboles permettent de décrire les différentes configurations de connexions des tétraèdres lors d'un parcours de proche en proche de la subdivision. Dans [11], le bord du maillage tétraédrique (une surface triangulée) est d'abord compressé à l'aide d'une méthode surfacique, puis chaque tétraèdre incident à la surface active

est décrit par un symbole en se rapprochant du centre du maillage. Les différents symboles sont compressés également par Huffman. Pour les maillages hexaédriques, Isenburg [12] propose d'utiliser le degré des arêtes pour décrire la topologie de ces maillages, de la même manière que dans [5] pour le cas surfacique.

3.2.3 Codage de la géométrie

La topologie est ensuite utilisée pour prédire la géométrie, en particulier la position de chaque nouveau sommet en fonction de celle des sommets déjà codés/décodés. Chaque sommet est d'abord quantifié (uniformément) puis décrit par un terme correctif par rapport à la prédiction effectuée ; cela a pour conséquence de limiter l'amplitude des valeurs à coder et de les rendre plus régulières et par conséquent plus faciles à compresser ensuite. Dans [2], cette prédiction est effectuée grâce à l'arbre des sommets, par extrapolation par rapport aux positions des 2 ou 3 ancêtres directs du sommet courant. Les termes correctifs après prédiction sont ensuite compressés par codage entropique (plus un symbole est fréquent, moins il occupe de bits dans le fichier résultant). Dans [5], une prédiction « en parallélogramme » est effectuée : le sommet prédit forme un parallélogramme avec les sommets du triangle codé directement adjacent sur la frontière active. Pour les maillages volumiques, des variantes inspirées de la prédiction parallélogramme sont utilisées. Quelquefois d'autres types d'attributs géométriques sont pris en compte également ([22], [23]).

En résumé, les grandes étapes des méthodes de compression de la connectivité de l'état de l'art sont :

- Un parcours de proche en proche des éléments de la subdivision ;
- Par chacun des éléments rencontrés, un symbole décrit sa connectivité, c'est-à-dire la manière de le connecter avec ses voisins précédemment codés/décodés (relations d'incidence avec la frontière active) ;
- L'ensemble de ces symboles est ensuite compressé par une méthode de type Huffman ou un codeur arithmétique ;
- Les sommets sont codés au fur et à mesure de leur visite lors du parcours, en prédisant leur position en fonction de celle des voisins (fonction linéaire des ancêtres, prédiction parallélogramme...) et en compressant l'erreur résiduelle.

Le problème majeur de ces méthodes concerne la généralisation au cas volumique général (maillages constitués de polyèdres quelconques rencontrés en géologie) ou en dimension quelconque. Les structures de données et algorithmes sont en effet très spécifiques aux classes de modèles traités. De plus, la gestion des surfaces de genre non nul ou avec beaucoup de bords apparaît comme un cas particulier qui introduit un coût de stockage assez pénalisant. Par conséquent, le genre et le nombre de bords sont souvent considérés comme négligeables par rapport au nombre de sommets. Enfin pour les maillages non orientables, la seule solution apportée est la coupure de certaines arêtes afin de rendre la surface orientable mais cela crée des bords et détruit une partie de l'information topologique.

3.3 Structures topologiques

Les modèles géométriques issus de modeleurs sont généralement représentés sous forme indexée (formats VRML, PLY, OBJ, STL, les modèles volumiques issus de modeleurs CAO). Ces représentations sont formées d'une liste des sommets (triplets de coordonnées flottantes) ainsi que d'une liste de primitives (triangles, faces, tétraèdres, hexaèdres) décrites par des n-uplets des indices des sommets qui les constituent. Ces représentations sont très pauvres du point de vue topologique car certaines informations d'adjacence ou d'incidence entre primitives ne sont pas directement accessibles (adjacence entre les faces d'un modèle surfacique notamment). Néanmoins, on peut en extraire la connectivité complète grâce à des techniques de hachage sur les indices des primitives.

Des structures topologiques sont donc fréquemment utilisées pour représenter les modèles car elles permettent un accès rapide aux informations d'adjacence et d'incidence permettant notamment d'effectuer la compression. Il existe de nombreux modèles topologiques permettant de représenter diverses classes de subdivisions (2D, 3D, orientables ou non, avec ou sans bords, variété ou non) : ce sont les graphes d'incidence, les modèles basés arête, ou les modèles ordonnés.

Les structures à base d'arête (Half-edge [24] en particulier) sont fréquemment employées lors de la compression de modèles surfaciques mais elles ne sont pas utilisables au-delà de la dimension 2. Dans le cas volumique, Isenburg [7] utilise une structure Spin-Edge (structure d'arête pour les maillages volumiques), alors que dans [10] ou [9], des graphes d'incidence entre les triangles de la frontière active sont préférés pour les maillages de tétraèdres. Etant donné la spécificité de ces structures qui ne peuvent représenter qu'une classe limitée de subdivisions, la généralisation des méthodes existantes est très difficile. A l'heure actuelle, il n'existe donc pas de méthode pour compresser des maillages en dimension quelconque avec des cellules arbitraires.

Nous avons adopté une structure topologique appelée carte généralisée (ou G-carte) [13] pour représenter la connectivité des subdivisions. Ce choix a été motivé par deux raisons principales :

- Elle permet de représenter des variétés quelconques quelque-soit leur dimension, y compris avec bords, non-orientables ;

- Les relations entre les éléments de base de cette structure sont homogènes sur toutes les dimensions ; cela facilite la généralisation des algorithmes.

Le processus de codage de notre méthode repose principalement sur l'utilisation de cette structure et reproduit le schéma de codage classique rencontré dans l'état de l'art, à savoir un parcours de proche en proche de la structure et la description des relations d'incidence et d'adjacence à l'aide de symboles lors du parcours. Grâce aux possibilités de représentation de cette structure, notre méthode peut compresser tous types de modèles géométriques variétés, c'est-à-dire des maillages en dimension quelconques orientables ou non, avec ou sans bords, et notamment :

- tous les maillages surfaciques, quelque-soit le genre de la surface, le nombre de bords, y compris les surfaces non-orientables comme la bouteille de Klein par exemple.
- tous les maillages volumiques : maillages de tétraèdres ou d'hexaèdres mais aussi de polyèdres quelconques, pouvant avoir des trous (genre non nul), etc. Cela inclut par exemple les modèles utilisés en géologie, les modèles volumiques de villes et sol.

4 Définitions

Une carte généralisée (ou G-carte) [13] est un modèle topologique permettant de représenter des variétés (manifold) orientables ou non, avec ou sans bords, en dimension quelconque. C'est un modèle ordonné (défini par [25]) : elle est ainsi constituée d'un seul type d'élément de base appelé *brin* et de relations d'incidence –connexité– entre ces éléments. Dans cette représentation, les cellules (sommets, arêtes, faces..., de dimensions respectives 0, 1, 2,...) ne sont pas représentées explicitement, mais elles sont déduites grâce aux relations entre les brins. Les brins peuvent être vus comme des cellules de dimension -1 .

Une carte généralisée de dimension N (N -G-carte) est définie de la manière suivante : $G=(B,\alpha_0,\alpha_1,\alpha_2,\dots,\alpha_N)$ est un $(N+2)$ -uplet, B est l'ensemble des brins et les α_i sont les $N+1$ relations entre les brins. Ces relations sont des permutations qui vérifient les propriétés suivantes :

1. $\forall 0 \leq i \leq N, \alpha_i$ est une *involution* (dite simple), c'est-à-dire que $\alpha_i^2(b) = b$.
2. $\forall i, j, 0 \leq i < i+2 \leq j \leq N, \alpha_i \alpha_j(b)$ est aussi une involution (dite composée).

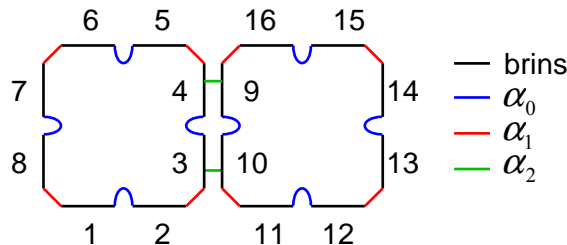


Figure 3 : exemple de 2-G-carte

b	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
alpha_0(b)	2	1	4	3	6	5	8	7	10	9	12	11	14	13	16	15
alpha_1(b)	8	3	2	5	4	7	6	1	16	11	10	13	12	15	14	9
alpha_2(b)	1	2	10	9	5	6	7	8	4	3	11	12	13	14	15	16

Figure 4 : tableau des involutions

La figure 3 représente la carte généralisée correspondant à deux carrés collés le long d'une arête. Les brins y sont représentés par des demi-arêtes et les involutions par des liens de couleur. La figure 4 représente les valeurs des involutions correspondantes.

La première propriété lie les brins deux à deux sur chaque dimension i excepté dans le cas où le brin est incident à un bord. Dans l'exemple précédent, les brins 4 et 10 sont liés entre eux sur la dimension 2, alors que le brin 5 est incident à un 2-bord (bord d'une surface) : $\alpha_2(4)=9, \alpha_2(9)=4, \alpha_2(5)=5$. La deuxième propriété correspond intuitivement à l'hypothèse de variété topologique (manifold), c'est-à-dire que les i -cellules doivent être cousues, collées entre elles via des $(i-1)$ -cellules de même forme. Dans l'exemple précédent, les faces (2-cellules) sont adjacentes par une arête (1-cellule) définie par les brins 3,4 et 9,10.

Les orbites notées $\langle \alpha_{i_1}, \alpha_{i_2}, \dots, \alpha_{i_n} \rangle (b)$ définissent l'ensemble des brins obtenus par toutes les compositions possibles des involutions α_{i_k} (aussi appelés groupes de permutation). Par exemple, $\langle \alpha_1, \alpha_2 \rangle (2) = \{2, 3, 10, 11\}$ est l'orbite correspondant au sommet incident au brin 2.

Les i -cellules (sommets, arêtes, faces, ...) sont définies par des orbites particulières constituées de toutes les involutions sauf α_i . Intuitivement, l'involution α_i permet de sortir d'une i -cellule donnée et de rentrer dans la i -cellule directement adjacente par le brin considéré. Ainsi, en sautant toutes les involutions des brins de la cellule sans en sortir (par α_i), on obtient tous les brins constitutifs de la cellule. Par exemple, $\langle \alpha_0, \alpha_1 \rangle (2)$ est l'orbite face (2-cellule) associée au brin 2, c'est-à-dire la face incidente au brin 2.

Vu de manière constructive, l'opération de *couture* permet de lier deux cellules de dimension n (orbite $\langle \alpha_0, \dots, \alpha_{n-1} \rangle$) le long de cellules de dimension $n-1$ incidentes à leur bord (orbite $\langle \alpha_0, \dots, \alpha_{n-2} \rangle$) par α_n . Ces cellules de dimension $n-1$ doivent avoir la même forme, ceci est garanti pour la propriété 2 des cartes généralisées.

Les *plongements* sont souvent associés à l'un des brins composant l'orbite du plongement. Par exemple, pour définir les coordonnées tridimensionnelles des sommets, seul un brin de chaque orbite sommet ($\langle \alpha_1, \alpha_2 \rangle$ dans une 2-G-carte) est plongé. Sur la figure 3, on peut par exemple associer des coordonnées tridimensionnelles aux brins 1, 2, 6, 4, 14, 12.

Le principal intérêt des cartes généralisées est une définition homogène des différentes dimensions quelque-soit le type de subdivision. Ainsi, il est plus facile de concevoir des algorithmes génériques. La structure de données utilisée pour représenter une carte généralisée est représentée sur la figure 5.

```

struct Dart {
    Dart* alpha[N+1]; // les N+1 involutions
    int marks; // marques booléennes utilisées pour les parcours
    list<Attributes*> attr; // attributs géométriques associés au brin
};

struct GMap {
    list<Dart*> darts; // liste de tous les brins ou des composantes
    connexes
};

```

Figure 5 : structure G-carte

Pour chaque brin d'une carte généralisée, le tableau alpha contient N+1 pointeurs représentant les images du brin pour les N+1 involutions. L'attribut marks permet l'étiquetage : chaque brin peut être marqué de façon à identifier un état (statut) de ce brin pendant un algorithme en utilisant l'un des bits de cet attribut (32 bits en tout). En particulier, les marques servent lors des parcours ou des traitements à signifier si les brins ont déjà été visités ou non. Enfin, les attributs géométriques portés par le brin sont stockés dans la liste attr.

La structure G-carte peut contenir soit une liste de tous les brins (facilite l'effacement et le démarquage des brins), soit une liste des composantes connexes (moins gourmand en espace mémoire mais aussi plus coûteux à mettre à jour lors des décousures).

5 Principe général

Notre procédé de compression consiste à construire des historiques de symboles lors d'un parcours de la carte généralisée. Ces symboles représentent les relations d'incidence et d'adjacence entre les cellules du maillage. Nous décrivons le voisinage de chaque brin rencontré lors du parcours, c'est-à-dire la valeur des involutions pour ce brin sur chacune des dimensions. Ensuite, les historiques produits sont compressés par des techniques classiques de compression de données, en l'occurrence un codeur par dictionnaire de type LZW [26]. Pour notre algorithme, aucune supposition n'est faite sur la topologie des subdivisions, en particulier sur le genre des subdivisions dans le cas surfacique et sur le nombre de bords.

L'algorithme se déroule de la façon suivante :

- Un parcours de proche en proche est réalisé sur la carte généralisée afin de décrire chacun des brins la constituant

- Pour chaque brin, chacune des involutions va être décrite par un symbole pour différencier différentes configurations possibles de connexion entre les brins
- L'ensemble des symboles générés lors du parcours forme des historiques (représentation intermédiaire compacte)
- Les historiques sont ensuite compressés à l'aide de techniques classiques de compression de données (codage par dictionnaire)
- Pour chaque type de plongements, un parcours indépendant est effectué afin de réaliser le codage de chaque attribut géométrique via les méthodes citées dans l'état de l'art, c'est à dire quantification, prédiction et compression du résidu

6 InvoCoder

Pour commencer, nous décrivons la production des historiques qui représentent la connectivité du modèle obtenue lors du parcours. Ensuite nous expliquons comment effectuer le décodage de ces historiques afin de reconstituer le modèle initial au niveau du décodeur ainsi que la compression proprement dite des historiques. Enfin, nous abordons le traitement de la géométrie associée au modèle.

6.1 Génération des historiques

Dans un premier temps, nous construisons des historiques qui décrivent la topologie ou connectivité du modèle : ils sont obtenus par un parcours des brins de la carte généralisée en explicitant leurs relations d'incidence, c'est-à-dire les images des brins par les involutions α_i .

6.1.1 Parcours

Soit G_N une carte généralisée de dimension N ($N+1$ involutions) composée de n brins. Lors du parcours, les brins rencontrés en attente de traitement sont stockés dans $N+1$ files d'attente notées L_i . La file L_0 correspond aux brins rencontrés pour lesquels l'involution α_0 n'a pas encore été traitée, la file L_1 pour les brins pour lesquels l'involution α_1 n'a pas encore été traitée, etc.

Lors de l'initialisation de l'algorithme, toutes les files sont vides. Nous choisissons un brin arbitraire de la composante connexe pour le départ du parcours ; il est ajouté dans la file L_0 . L'algorithme se déroule de la manière suivante : tant que les files ne sont pas vides, un brin est choisi en tête de la file de rang le plus faible, c'est-à-dire d'abord dans L_0 puis sinon dans L_1 , etc. Nous effectuons un traitement sur le brin courant puis une fois ce traitement terminé, le brin est inséré dans la liste L_{i+1} (si $i < N$), et ainsi de suite. Lorsqu'un brin est passé successivement dans toutes les files, son codage est terminé, il disparaît alors des files.

6.1.2 Historiques

Le traitement du brin choisi consiste à décrire une relation d'incidence entre ce brin et l'un de ses voisins, c'est-à-dire à générer un symbole qui décrit la connectivité du brin courant. A chaque file L_i correspond un historique H_i dans lequel sont inscrites les relations d'incidence entre les brins pour l'involution α_i . L'information stockée dans les historiques H_i permet de reconstituer la topologie initiale du maillage lors du décodage.

Lors du parcours et du traitement des brins pour constituer les historiques, nous procédons au marquage des brins par une étiquette $V(b)$ correspondant à un statut « visité », et à leur numérotation $N(b)$ dans l'ordre imposé par le parcours. Initialement, seul le brin choisi arbitrairement est marqué et numéroté 0. Ensuite, chaque nouveau brin découvert par α_i est marqué, un numéro lui est attribué et il est ajouté à son tour dans la liste L_0 afin d'être traité ultérieurement lors du parcours. D'autre part, pour un brin donné, les involutions α_i sont traitées dans l'ordre croissant.

Les symboles inscrits dans les historiques H_i sont déterminés de la manière suivante. Soit b' le brin image par l'involution α_i du brin courant b ($b' = \alpha_i(b)$), nous distinguons plusieurs configurations d'incidence entre b et b' (cf. figure 6) :

- Si $b' = b$, le brin constitue un bord et par conséquent le symbole \mathbb{B} (border) est émis ;
- Si le brin image b' n'a pas été rencontré précédemment lors du parcours (i.e. $V(b')$ est faux), le symbole \mathbb{N} (new) est émis, et on ajoute ce nouveau brin dans la file L_0 pour traitement ultérieur (marquage et numérotation) ;
- En revanche s'il a déjà été rencontré, le symbole w (waiting) est émis en y associant un numéro $N(b')$ permettant d'identifier le brin correspondant ;
- Si l'information de connectivité est redondante (voir plus loin), aucun symbole n'est émis (noté - dans les exemples)

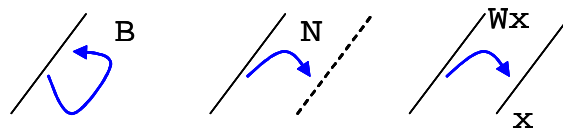


Figure 6 : symboles

La numérotation sert notamment à constituer l'indice associé au code w . L'historique global $H = \{H_0, H_1, \dots, H_N\}$ est construit de façon incrémentale lors du traitement de chaque brin.

La figure 7 illustre un exemple simple de codage en dimension 1 (polyligne) comprenant deux involutions α_0 et α_1 . Chaque ligne décrit : (à gauche) le numéro du brin choisi lors du parcours, le rang i de l'involution α_i en cours de traitement et le symbole qui y est associé, et (à droite) les files d'attente. Un brin arbitraire est choisi comme point de départ pour le codage de la composante connexe. Il est numéroté 0, marqué par V et il est inséré dans la liste L_0 . Ensuite, itérativement, nous choisissons un brin parmi l'une des listes L_i et nous le traitons. Le brin 0 est choisi sur L_0 lors de la première itération, et puisque son image par l'involution α_0 n'a pas encore été rencontrée (le brin image n'est pas marqué), un symbole \mathbb{N} est inscrit dans l'historique H_0 . Le brin 0 est ensuite déplacé vers la liste L_1 en attendant de décrire son image par α_1 . Lors de ce traitement, le nouveau brin découvert est numéroté 1, marqué par V , puis est ajouté dans L_0 . Ce brin est choisi lors de la deuxième itération. L'information d'incidence par α_0 de ce brin 1 étant redondante par rapport au brin 0, aucun symbole n'est donc émis pour le brin 1 (-) et il est déplacé à son tour dans L_1 . Ensuite vient le traitement du brin 0 pour α_1 (choisi sur la liste L_1) qui produit un symbole \mathbb{N} dans l'historique H_1 et découvre un nouveau brin numéroté 2, et ainsi de suite. Un peu plus tard lors du parcours, le choix se porte sur le brin 3 sur L_1 : son image par α_1 (brin 5) a déjà été visitée donc un symbole w_5 est inscrit dans H_1 (5 correspondant au numéro du brin incident à 3 par α_1). A la fin du codage, 2 historiques H_0 et H_1 ont été générés.

brin(i) : symbole	$L_0 = \{ \}$	$L_1 = \{ \}$
0(0) : \mathbb{N}	$L_0 = \{1\}$	$L_1 = \{0\}$
1(0) : -	$L_0 = \{ \}$	$L_1 = \{0, 1\}$
0(1) : \mathbb{N}	$L_0 = \{2\}$	$L_1 = \{1\}$
2(0) : \mathbb{N}	$L_0 = \{3\}$	$L_1 = \{1, 2\}$
3(0) : -	$L_0 = \{ \}$	$L_1 = \{1, 2, 3\}$
1(1) : \mathbb{N}	$L_0 = \{4\}$	$L_1 = \{2, 3\}$
4(0) : \mathbb{N}	$L_0 = \{5\}$	$L_1 = \{2, 3, 4\}$
5(0) : -	$L_0 = \{ \}$	$L_1 = \{2, 3, 4, 5\}$
2(1) : -	$L_0 = \{ \}$	$L_1 = \{3, 4, 5\}$
3(1) : w_5	$L_0 = \{ \}$	$L_1 = \{4, 5\}$
4(1) : -	$L_0 = \{ \}$	$L_1 = \{5\}$
5(1) : -	$L_0 = \{ \}$	$L_1 = \{ \}$

$H_0 = [\mathbb{N}, \mathbb{N}, \mathbb{N}]$
 $H_1 = [\mathbb{N}, \mathbb{N}, w_5]$

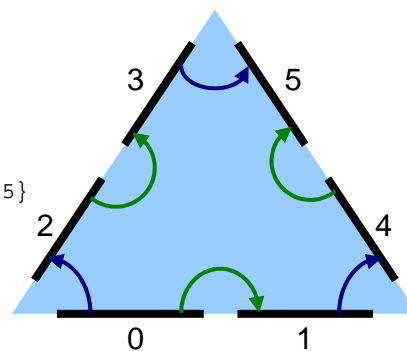


Figure 7 : exemple en dimension 1

6.1.3 Détection des redondances

Dans l'exemple précédent, nous avons constaté la présence de redondances d'informations pour lesquelles nous n'émettons pas de symbole. Sur la figure 7, elles apparaissent sous la forme d'un faux symbole (noté $-$). Ces redondances sont la conséquence de la propriété 1 des G-cartes (i.e. $\alpha_i^2(b) = b$). Au delà de la dimension 1, la propriété 2 intervient également pour créer un autre type de redondance. Sur la Figure 8, les redondances sont représentées par des flèches barrées. Pour le premier cas, les involutions étant des liens bidirectionnels entre brins, nous ne conservons qu'une direction sur les deux. Pour le second, l'information de bord est déjà connue pour le brin 1 par déduction par rapport au brin 0. Enfin dans le troisième, nous déduisons la présence d'un brin 3 dès l'instant où l'existence du brin 2 est confirmée : il s'agit du cas inverse du second c'est-à-dire qu'il n'y a pas de bord à cet endroit. Pour détecter ces redondances, nous utilisons une étiquette $E_i(b)$ pour savoir s'il faut inscrire un symbole dans l'historique H_i pour l'involution α_i d'un brin b donné. Initialement, toutes les étiquettes $E_i(b)$ sont positionnées pour tous les brins.

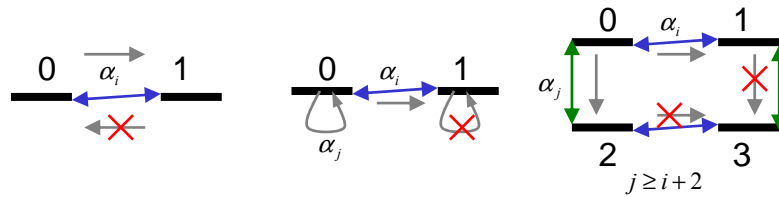


Figure 8 : les 3 cas de redondance

La mise à jour des étiquettes se fait itérativement pendant le codage et le décodage de la subdivision. Nous définissons l'orbite d'origine $O_O = \langle \alpha_0, \dots, \alpha_{i-2} \rangle(b)$ et l'orbite de destination $O_D = \langle \alpha_0, \dots, \alpha_{i-2} \rangle(b')$ lorsque $b' \neq b$. Ces deux orbites sont homéomorphes d'après la propriété 2 des G-cartes. Ainsi, il n'est pas nécessaire de décrire la connectivité entre les brins de l'orbite destination, puisque la connectivité est la même que celle de l'orbite origine. La mise à jour des étiquettes s'effectue de la manière suivante :

- N : l'orbite destination n'a pas encore été visitée, mais elle a la même forme (même connectivité) que l'orbite origine : par conséquent nous mettons à zéro les étiquettes E_k avec $k = \{0, \dots, i-2, i\}$ pour les brins de l'orbite destination et les étiquettes E_i pour les brins de l'orbite origine. Les brins de l'orbite destination sont également marqués par V et numérotés selon un ordre semblable à celui qui est effectué lors du parcours et insérés dans le même ordre dans la liste L_{i-1} ;
- B : l'orbite origine constitue un bord de la subdivision, nous masquons par conséquent les étiquettes E_i des brins de l'orbite origine pour éviter de dupliquer l'information de bord ;
- w : l'orbite destination a déjà été décrite (puisque les brins ont déjà été visités pour les dimensions inférieures à i), les étiquettes E_i des brins de l'orbite origine et de l'orbite destination sont donc mises à zéro.

La figure 9 illustre le codage d'un modèle en dimension 2 (surface). Notez les redondances pour l'involution α_2 pour les brins impairs et le traitement des brins 4, 5, 6, 7 en particulier.

0(0):N	0(2):B	11(0):-
1(0):-	1(2):-	8(1):-
0(1):N	2(2):B	9(1):W11
2(0):N	3(2):-	10(1):-
3(0):-	4(2):N	11(1):-
1(1):N	6(0):-	5(2):-
4(0):N	7(0):-	6(2):-
5(0):-	6(1):N	7(2):-
2(1):-	8(0):N	8(2):B
3(1):W5	9(0):-	9(2):-
4(1):-	7(1):N	10(2):B
5(1):-	10(0):N	11(2):-

H0=[N,N,N,N,N]
H1=[N,N,W5,N,N,W11]
H2=[B,B,N,B,B]

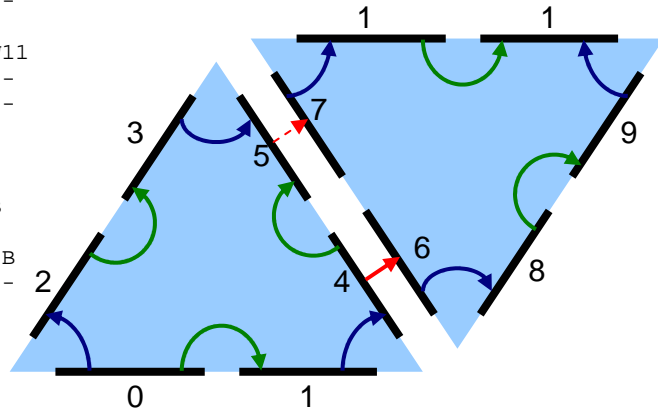


Figure 9 : exemple en dimension 2

6.2 Décodage des historiques

Le décodage est symétrique à la phase de codage. L'algorithme de décodage effectue un parcours identique, en construisant les brins au fur et à mesure du parcours et en les numérotant/marquant de la même façon que lors du codage. Les symboles issus des historiques permettent de reconstituer la connectivité initiale du modèle. De la même façon que pour le codage, les étiquettes E_i servent à déterminer s'il faut lire un symbole dans l'historique H_i . Un traitement différent est effectué en fonction du symbole lu dans un historique H_i :

- B : les étiquettes E_i de l'orbite origine sont masquées
- N : l'orbite origine courante est dupliquée pour constituer l'orbite destination, puis les orbites origine et destination sont cousues par α_i . Les brins de l'orbite destination sont numérotés (de la même façon que pour le codage) et sont ajoutés dans L_{i-1} . Les étiquettes E_k des brins créés sont toutes positionnées, puis les étiquettes E_k avec $k = \{0, \dots, i-2, i\}$ pour les brins de l'orbite destination et les étiquettes E_i pour les brins de l'orbite origine sont mises à zéro.
- w : les étiquettes E_i de l'orbite origine et de l'orbite destination sont mises à zéro puis ces deux orbites sont cousues par α_i .

Reprenons l'exemple de la figure 9. Initialement, le brin 0 est créé et marqué, toutes ses étiquettes sont positionnées puis il est mis en attente dans L_0 . A la première itération, ce brin est choisi, nous lisons un code N dans α_0 provoquant la création de 1 avec toutes les étiquettes positionnées excepté E_0 qui est inséré dans L_0 . Aucune information n'est lue dans l'historique H_0 pour le brin 1 car l'étiquette E_0 est à zéro. Pour l'involution α_1 , le brin 0 provoque la création de 2 (avec toutes les étiquettes positionnées sauf E_1) grâce au symbole N lu dans H_1 , puis le brin 2 provoque la création de 3 par α_0 (toutes les étiquettes positionnées sauf E_0), etc. Le reste du décodage se déroule de manière similaire et dans le même ordre que lors du codage.

6.3 Compression de l'historique

Le mécanisme de compression est réalisé en deux étapes :

- Prédiction des indices pour les symboles w
- Compression des symboles et des indices

6.3.1 Prédiction des indices

Nous avons vu que pour chaque symbole w, un indice lui est associé afin d'identifier le brin à coudre. Nous avons constaté empiriquement que le nombre de symboles w n'est absolument pas négligeable : il représente environ 50% des symboles dans certains historiques. De plus, la plage de valeurs des indices possibles étant très grande ($O(n)$), ils sont difficilement compressibles.

Nous avons donc mis en place un système de prédiction des indices basé sur le voisinage déjà codé/décodé. L'idée consiste à effectuer un parcours de l'orbite $\langle \alpha_i, \alpha_{i-1} \rangle$, afin de trouver un brin dont l'involution α_i n'a pas encore été décrite (ce brin appartient par conséquent à la liste L_i du fait du parcours effectué). Plus précisément, à partir du brin initial, nous traversons alternativement les involutions α_{i-1} et α_i et le parcours s'arrête sur le premier brin dont l'involution α_i n'a pas été décrite (étiquette E_i à zéro).

L'indice associé au symbole w devient alors la différence (delta) entre l'indice réel du brin à coudre et l'indice du brin prédit lors de ce parcours. Sur l'exemple de la figure 10, nous observons les parcours appliqués respectivement pour l'involution α_1 et l'involution α_2 résultant tous deux à un indice nul, c'est-à-dire une bonne prédiction.

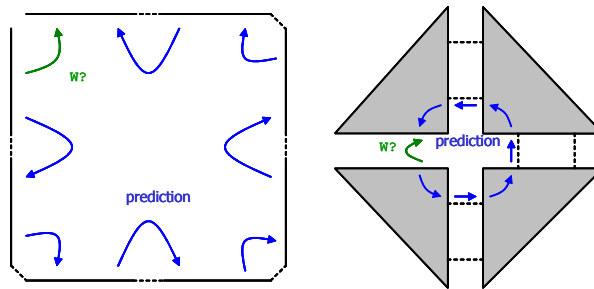


Figure 10 : prédiction des indices

6.3.2 Représentation compressée des symboles

Pour compresser les historiques, nous avons décidé d'utiliser un compresseur par dictionnaire (par exemple LZW [26]). Ce type de méthode fonctionne par constitution d'un dictionnaire de mots à partir des symboles au fur et à mesure de la compression : les motifs fréquemment rencontrés dans les séquences de symboles se voient attribuer un code court qui remplacera le motif lors de ses prochaines occurrences. Le choix de ce type de méthode est motivé par l'observation de longues séquences répétitives de symboles dans les subdivisions testées. D'autre part, nous avons remarqué que les fréquences des symboles N et $w0$ sont relativement semblables sur certaines involutions ce qui suggère qu'un codage entropique ne serait pas très adapté pour une compression efficace.

Avant la compression proprement dite, nous établissons un codage en octets de nos symboles et de l'indice associé au symbole w : le nombre d'octet du code est variable en fonction de cet indice. Dans la plupart des cas, un seul octet suffit : ses 2 bits de poids faible servent à indiquer le symbole rencontré et les 6 bits restant de l'octet servent à indiquer l'indice associé au code w . Nous avons choisi d'attribuer la valeur 0 pour B et 1 pour N . Si les 6 bits restants suffisent à représenter intégralement l'indice, le symbole w est codé par la valeur 2, sinon, il prend la valeur 3 indiquant que des octets supplémentaires sont utilisés pour représenter le reste de l'indice. Les 6 bits restant de l'octet permettent de stocker une partie de l'indice (ses 6 bits de poids faible). Pour chacun des octets suivants, le bit de poids fort indique la présence d'octets supplémentaires pour définir le reste de l'indice et les 7 autres bits représentent une nouvelle partie de l'indice.

Une fois ce codage effectué, nous compressons les représentations obtenues grâce à la librairie Zlib (<http://www.gzip.org/zlib/>) pour produire un fichier binaire. Les résultats obtenus par la prédiction et la compression sont donnés dans la section 7.

6.4 Codage et décodage de la géométrie

Après avoir décrit notre système de compression de la topologie, nous présentons dans cette partie le codage/décodage des attributs de forme (géométrie) de la structure : coordonnées de sommets, couleurs, courbures d'arêtes ou de faces, textures, etc. Il s'agit de traiter les différents attributs de manière à ce que le décodeur les rencontre dans le même ordre que le codeur. Etant donné que la topologie de la subdivision est entièrement connue, il suffit pour cela d'effectuer un parcours identique de la structure dans les deux cas.

Nous effectuons un parcours distinct pour chaque type d'attribut : un parcours des sommets pour les coordonnées tridimensionnelles, un parcours des faces pour les textures, etc. Chaque parcours débute par le brin initial (brin 0) du codage de la topologie afin de synchroniser le décodeur des attributs par rapport au codeur. Un tampon de données distinct est rempli pour chaque type d'attribut géométrique. Les schémas de codage prédictif décrits dans [4] ou [5]

peuvent par exemple être utilisés pour le codage des sommets : les données inscrites dans le tampon des sommets seront alors les résidus après prédiction.

Nous utilisons un parcours d'orbite pour ordonner les attributs : de la même façon que pour le codage de la topologie, nous utilisons une file d'attente L pour mémoriser les brins qui doivent être traités, et nous procédons au marquage des cellules visitées (i.e. l'ensemble des brins qui composent l'orbite de l'attribut correspondant) à l'aide d'une étiquette V pour éviter de traiter plusieurs fois un même attribut. Le parcours s'effectue de manière similaire à celui de la topologie : un brin initial est inséré dans la file (ici le brin 0), puis à chaque itération nous retirons un brin de la file et s'il n'est pas marqué, nous marquons tous les brins de l'orbite attribut, nous codons/décodons l'attribut correspondant (quantification, prédiction, etc.) puis nous ajoutons les brins voisins de l'orbite dans la file.

Par souci de simplicité d'implémentation et parce que le codage des plongements n'est pas l'enjeu essentiel de notre technique, nous avons utilisé uniquement un quantificateur scalaire uniforme sans aucune prédiction pour la compression des sommets : lors d'un premier parcours nous calculons la boîte englobante des sommets et le pas de quantification, et lors d'un deuxième parcours nous procédons à la quantification de chaque sommet, la valeur quantifiée étant écrite dans le tampon de sortie. Néanmoins l'utilisation de la quantification vectorielle adaptative [27] nous semble très prometteuse car suffisamment générale pour s'adapter à de nombreux types d'attributs géométriques.

7 Résultats et discussion

Dans cette partie, nous présentons les résultats obtenus par notre méthode concernant deux principaux aspects : la qualité de la prédiction des indices et les taux de compression atteints. Pour produire les résultats, nous avons testé plusieurs types de modèles avec la même implémentation de notre méthode :

- un corpus de fichiers VRML classiques (surfaiques) utilisés pour comparer les diverses méthodes de compression surfaique
- un corpus de maillages hexaédriques et tétraédriques
- des maillages surfaiques réguliers de carrés : $N \times N$ carrés collés le long de leurs arêtes
- des maillages volumiques réguliers de cubes : $N \times N \times N$ cubes collés le long de leurs faces
- des maillages surfaiques non orientables : ruban de Moëbius, bouteille de Klein

Quelques résultats en terme de compression et de prédiction apparaissent sur les figures 11, 12, 13 et 14. Les champs représentés sont le nombre de brin (D), le nombre de sommets (V), le nombre de faces (F) ou de polyèdres (T pour les tétraèdres ou H pour les hexaèdres), la taille de la topologie compressée en octets ($topo$), la taille de la géométrie en octets (geo), le temps d'encodage en millisecondes géométrie comprise ($time$), le nombre de bits par brin (BpD) et par sommet ou polyèdre (BpV , BpH), les résultats pour d'autres méthodes lorsque disponibles ainsi que le ratio correspondant, le nombre de symboles totaux (nS) et le pourcentage de mauvaises prédictions (BP) par rapport au nombre de symboles w . Les données temporelles ont été mesurées sur un AMD Athlon XP 2200+ avec 1Go de RAM. Nous avons utilisé le paramètre par défaut pour le compromis rapidité/taux de compression de la librairie Zlib.

A titre de comparaison, nous donnons également les résultats obtenus dans [8] pour le cas surfaique (parmi les meilleurs résultats actuels), ceux de [12] pour les maillages hexaédriques et ceux de [10] pour les maillages tétraédriques. Dans tous les cas, les caractéristiques topologiques (nombre de bords, genre, orientabilité, nombre de cellules) des maillages décodés sont exactement identiques à celles des maillages initiaux.

Les résultats concernant la qualité de prédiction sont bons : 100% de bonnes prédictions pour les cas réguliers (maillages réguliers de carrés et de cubes notamment) et moins de 2% d'erreurs dans le cas contraire. Les erreurs de prédictions sont liées à des irrégularités topologiques locales ou à la présence de bords : ce type de problème est similaire à celui des « divisions » dans les méthodes de compression par valences.

name	D	V	F	topo	geo	time	BpD	BpV	KAD02		nS	BP
									(BpV)	Invo/KAD02		
teapot	9916	1189	1290	742	5379	110	0.599	4.992	1.22	4.09	12396	1.0%
galleon	18932	2373	2384	1377	11015	187	0.582	4.642	2.09	2.22	23677	4.4%
shark	20480	2560	2562	1490	11548	219	0.582	4.656	0.83	5.61	25601	1.4%
beethoven	21308	2655	2812	1903	12172	219	0.714	5.734	2.30	2.49	26777	1.9%
sandal	21716	2636	2953	1626	12114	204	0.599	4.935	2.27	2.17	27334	2.5%
triceratops	22656	2832	2834	1602	12772	250	0.566	4.525	1.28	3.54	28321	1.9%
cupie	24016	2984	3032	1857	13596	234	0.619	4.979	1.78	2.80	30026	3.0%
cow_poly	24660	2904	3263	1565	13096	265	0.508	4.311			30826	1.8%
cessna	30600	3745	3927	2953	17161	328	0.772	6.308	2.57	2.45	38261	4.3%
al	31004	3618	4175	2810	16869	281	0.725	6.213	2.50	2.49	38776	3.5%
tommygun	32340	4171	3980	2516	19862	281	0.622	4.826	2.30	2.10	40464	8.7%
cow	34824	2904	5804	1813	13096	391	0.416	4.994	1.76	2.84	43531	1.2%
moebius	96	24	12	73	136	16	6.083	24.333			133	7.7%
klein	7200	900	900	275	4078	78	0.306	2.444			9001	0.7%

Figure 11 : modèles surfaciques

name	D	V	T	topo	geo	time	BpD	BpV	BpT	BPV ?	Invo/?	nS	BP
random	301728	2000	12572	11213	9028	3250	0,297	44,852	7,135			289362	2,4%
silva13k	310464	2896	12936	9303	13060	3265	0,240	25,699	5,753	15,12	1,70	298914	1,0%
heart_tetra	5950176	50906	247924	168705	229133	86125	0,227	26,512	5,444			5721135	0,9%

Figure 12 : modèles tétraédriques

name	D	V	H	topo	geo	time	BpD	BpV	BpH	IA03		nS	BP
										(BpH)	Invo/IA03		
bump2	57072	1665	1189	1043	7521	609	0,146	5,011	7,018	2,1	3,34	59903	0,08%
fru	209280	5124	4360	2206	23086	2344	0,084	3,444	4,048	0,98	4,13	218726	0,00%
hanger	8208	398	171	311	1819	78	0,303	6,251	14,550	5,3	2,75	8764	0,55%
hutch	392256	8790	8172	2740	39583	4406	0,056	2,494	2,682	0,31	8,65	409192	0,01%
mdg-1b	178080	4510	3710	2160	20323	1922	0,097	3,831	4,658	0,77	6,05	186259	0,02%
shaft	330384	9218	6883	5147	41509	3672	0,125	4,467	5,982	1,7	3,52	346362	0,05%
test	114528	3198	2386	1196	14419	1187	0,084	2,992	4,010	0,87	4,61	120040	0,01%
warped	384000	9261	8000	1282	41703	4218	0,027	1,107	1,282	0,18	7,12	401208	0,00%

Figure 13 : modèles hexaédriques

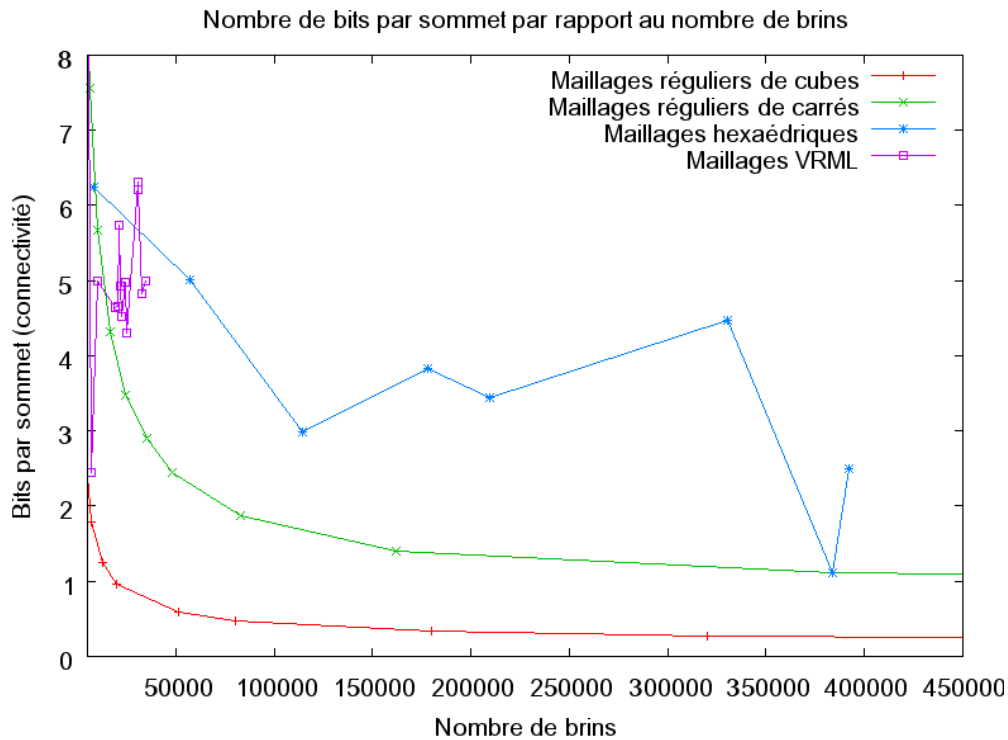


Figure 14 : courbes de BPV en fonction du nombre de brins

En terme de compression, notre méthode est un peu moins performante que les meilleures méthodes actuelles : nous observons un facteur de taille allant de 2 à 3 sur les maillages surfaciques (soulignons que notre implémentation traite ces maillages en dimension 3), un facteur 1 à 3 pour les maillages tétraédriques et un facteur 2 à 8 pour les maillages

hexaédriques. Néanmoins, notre méthode reste performante en regard de la généralité qu'elle apporte au niveau du codage de la connectivité. En effet, il est toujours possible de créer des méthodes plus efficaces sur un corpus de fichiers donné, mais aucune méthode n'est pour l'instant capable de traiter autant de cas que la notre : en plus des maillages surfaciques et volumiques réguliers à base de tétraèdres ou hexaèdres, nous pouvons également traiter les maillages volumiques quelconques, avec des cellules arbitraires, ou même des maillages variétés en dimension supérieure. Notre méthode offre par conséquent un bon compromis généralité/performance.

Le principal problème de notre méthode réside essentiellement dans le codage par dictionnaire qui n'est efficace que sur des longues séquences de symboles. Parmi les améliorations qui pourraient être apportées, nous pourrions citer :

- initialiser le dictionnaire avant le codage avec des motifs répétitifs souvent rencontrés (dictionnaire prédéfini constitué de suites de \mathbb{N} , de \mathbb{B} et de w_0) ;
- initialiser le dictionnaire par analyse d'une petite portion du maillage et stocker le dictionnaire dans la représentation compressée
- utiliser un véritable modèle statistique (chaînes de Markov par exemple)
- améliorer la prédiction en prenant en compte d'autres critères locaux
- influencer l'ordre du parcours par des heuristiques (comme dans [8])

Nous avons également intégré un codeur arithmétique adaptatif au lieu du codeur par dictionnaire. Pour obtenir des valeurs numériques assez régulières, nous procédons au comptage des occurrences successives d'un même symbole. Ce type de technique se rapproche par conséquent des méthodes de compression par valences. Néanmoins, les résultats obtenus par cette méthode ne sont pas aussi bons que ceux obtenus par le codeur à base de dictionnaire. Cela suggère que les techniques de codage par valences sont moins performantes si le type de maillages à compresser n'est pas connu à l'avance, c'est-à-dire si les maillages ont une connectivité quelconque a priori.

8 Conclusions et perspectives

Nous avons présenté une nouvelle méthode de compression sans perte de la connectivité de maillages en dimension quelconque et avec des cellules arbitraires. Cette méthode s'appuie sur une représentation appelée carte généralisée afin de produire une représentation intermédiaire de la connectivité du modèle sous forme d'historiques construits lors d'un parcours de proche en proche des brins de la structure. Cette représentation intermédiaire est compressée à l'aide d'un codage par dictionnaire pour former la représentation finale de la connectivité du modèle. Les attributs géométriques (coordonnées des sommets notamment) sont traités de façon indépendante à l'aide de techniques classiques telles que la quantification et la prédiction.

Les résultats sont encourageants malgré des taux de compression (légèrement) inférieurs à d'autres méthodes plus spécialisées. L'avantage déterminant de notre méthode est donc sa généralité. Grâce à notre technique, la compression de la topologie de maillages volumiques irréguliers est rendue possible, ceci étant particulièrement intéressant pour la modélisation du sous-sol en géologie ou la représentation (volumique) d'environnements urbains avec sol. Les perspectives de ce travail sont l'amélioration du codage de la géométrie en y intégrant une quantification vectorielle notamment.

Les applications futures de ce travail sont la transmission de scènes urbaines sur des réseaux et le calcul de visibilité au niveau du sol lors de la navigation en utilisant la topologie de la scène. Notre technique permet de réduire la bande passante nécessaire pour transmettre les données géométriques sur le réseau et par conséquent la latence, tout en véhiculant les informations topologiques nécessaires au calcul de visibilité sur la machine client.

9 Remerciements

Nous tenons à remercier Alla Sheffer pour les modèles bump2, fru et shaft, Steven Owen pour mdg-1, Scott Mitchell pour hanger et hutch, la NASA pour blunt, Bruno Notrosso pour proto (spx). Le modèle heart_tetra provient du Computational Visualization Center (University of Texas at Austin).

10 Bibliographie

- [1] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, "The Digital Michelangelo Project: 3D Scanning of Large Statues", SIGGRAPH 2000, Computer Graphics Proceedings, 2000.
- [2] G. Taubin and J. Rossignac, "Geometric compression through topological surgery," *ACM Transactions on Graphics*, vol. 17, pp. 84-115, 1998.
- [3] S. Gumhold and W. Strasser, "Real Time Compression of Triangle Mesh Connectivity," *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, vol. 32, pp. 133-140, 1998.
- [4] J. Rossignac, "Edgebreaker: Connectivity Compression for Triangle Meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, pp. 47-61, 1999.
- [5] C. Touma and C. Gotsman, "Triangle Mesh Compression", Proceedings of Graphics Interface '98, Vancouver, 1998.
- [6] P. Alliez and M. Desbrun, "Valence-Driven Connectivity Encoding for 3D Meshes", EG 2001 Proceedings, 2001.
- [7] M. Isenburg, "Compressing Polygon Mesh Connectivity with Degree Duality Prediction", Graphics Interface '02 Conference Proceedings, Calgary, Alberta, 2002.
- [8] A. Khodakovsky, P. Alliez, M. Desbrun, and P. Schröder, "Near-optimal connectivity encoding of 2-manifold polygon meshes", Graphical Models, 2002.
- [9] A. Szymczak and J. Rossignac, "Grow and fold: compression of tetrahedral meshes", Proceedings of the fifth ACM symposium on Solid modeling and applications, Ann Arbor, Michigan, United States, 1999.
- [10] S. Gumhold, S. Guthe, and W. Strasser, "Tetrahedral mesh compression with the cut-border machine", Proceedings of the conference on Visualization '99, San Francisco, California, United States, 1999.
- [11] C.-K. Yang, T. Mitra, and T.-c. Chiueh, "On-the-Fly rendering of losslessly compressed irregular volume data", IEEE Visualization 2000, 2000.
- [12] M. Isenburg and P. Alliez, "Compressing hexahedral volume meshes", Graphical Models, 2003.
- [13] P. Lienhardt, "N-Dimensional Generalized Combinatorial Maps and Cellular Quasi-Manifolds," *International Journal of Computational Geometry and Applications*, vol. 4, pp. 275-324, 1994.
- [14] H. Hoppe, "Efficient implementation of progressive meshes," *Computers and Graphics*, vol. 22, pp. 27-36, 1998.
- [15] M. Lounsbery, T. D. DeRose, and J. Warren, "Multiresolution analysis for surfaces of arbitrary topological type," *ACM Transactions on Graphics*, vol. 16, pp. 34-73, 1997.
- [16] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, "Multiresolution Analysis of Arbitrary Meshes," *Computer Graphics*, vol. 29, pp. 173-182, 1995.
- [17] Z. Karni and C. Gotsman, "Spectral compression of mesh geometry," *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 279-286, 2000.
- [18] A. Guézic, G. Taubin, F. Lazarus, and B. Horn, "Cutting and Stitching: Converting Sets of Polygons to Manifold Surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, pp. 136-151, 2001.
- [19] C. Gotsman, "On the Optimality of Valence-Based Connectivity Coding.," *Computer Graphics Forum*, vol. 22, pp. 99-102, 2003.
- [20] W. T. Tutte, "A census of planar triangulations," *Canad. J. Math.*, vol. 14, pp. 21-38, 1962.
- [21] W. T. Tutte, "A census of planar maps," *Canad. J. Math.*, vol. 15, pp. 249-271, 1963.
- [22] M. Isenburg and J. Snoeyink, "Compressing the property mapping of polygon meshes", Graphical Models, 2002.
- [23] M. Isenburg and J. Snoeyink, "Compressing Texture Coordinates with Selective Linear Predictions", Proceedings of Computer Graphics International'03, 2003.
- [24] M. Mantyla, *Introduction to Solid Modeling*: W. H. Freeman and Co., 1988.
- [25] E. Brisson, "Representing geometric structures in d dimensions: topology and order", Proceedings of the fifth annual symposium on Computational geometry, Saarbruchen, West Germany, 1989.
- [26] T. A. Welch, "A technique for high-performance data compression," *IEEE Computer*, pp. 8-19, 1984.
- [27] P. H. Chou and T. H. Meng, "Vertex Data Compression through Vector Quantization", IEEE Transactions on Visualization and Computer Graphics, 2002.