

# Représentation polygonale de surfaces d'objets discrets 3D

I. Sivignon<sup>1</sup>

F. Dupont<sup>2</sup>

J.-M. Chassery<sup>1</sup>

<sup>1</sup> LIS (Laboratoire des Images et des Signaux)

961, rue de la Houille Blanche

Domaine Universitaire - BP46 38402 St Martin d'Hères Cedex

{sivignon, chassery}@lis.inpg.fr

<sup>2</sup> LIRIS (Laboratoire d'InfoRmatique en Images et Systèmes d'information)

Université Claude Bernard, Lyon 1

Bât. Nautibus, 8, bd Niels Bohr, 69622 Villeurbanne cedex FRANCE

fdupont@ligim.univ-lyon1.fr

## Résumé

*Dans cet article, nous nous intéressons au problème de la polygonalisation d'un volume discret. Ce problème a de nombreuses applications notamment en visualisation et compression. Nous présentons ici une méthode permettant de transformer un volume discret en polyèdre euclidien en utilisant la géométrie discrète. Cette transformation n'est pas réversible mais la méthode employée devrait permettre à terme d'atteindre cet objectif.*

## Mots clefs

Géométrie discrète, image 3D, modélisation, compression.

## 1 Introduction

Les images 3D sont de plus en plus présentes sur les réseaux de communication. Leur volume important rend les transferts et la visualisation difficiles. Une solution pour palier à ces problèmes est de changer le mode de description de ces volumes, en utilisant une représentation à la fois plus simple à stocker et permettant une meilleure visualisation.

La transformation réversible d'un volume discret en polyèdre euclidien possède ces propriétés. En effet, un polyèdre euclidien est décrit par ses facettes, ce qui réduit considérablement le nombre d'informations à stocker sans aucune perte d'information géométrique ni topologique dans l'objet compressé. De plus, un polyèdre euclidien est un objet simple à visualiser. Enfin, comme une transformation réversible par discrétisation est recherchée, il est possible de retrouver l'image originale exacte à partir de l'image transformée.

Plusieurs méthodes de polygonalisation d'objets discrets ont déjà été proposées, et on peut voir deux grands types de méthodes : celles utilisant la géométrie euclidienne classique pour opérer la transformation, et celles utilisant une

géométrie adaptée aux objets discrets appelée géométrie discrète. La méthode la plus utilisée actuellement pour obtenir un objet facile à visualiser est la méthode des *Marching Cubes*, qui propose un ensemble de transformation pour des configurations de voxels locales. Mais cette méthode n'offre pas de compression sans pertes puisqu'elle n'est pas réversible. Parmi les méthodes utilisant la géométrie discrète, on peut aussi distinguer deux types d'approches, l'une passant par une reconnaissance de plans discrets [1, 2], l'autre utilisant un calcul d'enveloppe convexe [3] ou bien encore un diagramme de Voronoï discret [4]. Mais aucune de ces méthodes ne vise une réversibilité de la transformation effectuée.

Nous allons présenter un algorithme permettant de décomposer une surface discrète quelconque en morceaux de plans discrets. Nous présenterons ensuite la démarche à suivre et les verrous méthodologiques à lever pour parvenir à une polygonalisation, avec à terme un objectif de réversibilité. Dans une dernière partie, nous présenterons quelques images d'objets euclidiens obtenus à partir d'objets discrets selon une polygonalisation non réversible.

## 2 Problématique et notions de base

Polygonaliser une surface discrète peut se décomposer en plusieurs étapes : tout d'abord, savoir détecter la coplanarité d'un ensemble de voxels, ensuite appliquer ce test de coplanarité sur une surface quelconque et enfin transformer chaque morceau de plan reconnu en un polygone discret.

### 2.1 Définitions

Nous nous plaçons ici dans un espace discret 3D défini comme une grille unitaire centrée sur les points à coordonnées entières. Chaque cellule est alors un cube que l'on appellera par la suite *voxel*. Les coordonnées d'un voxel sont alors les coordonnées du centre du cube correspon-

nant.

Il est facile de définir sur cette grille 3 voisinages : deux voxels sont dits *6-voisins* s'il partagent une face, *18-voisins* s'ils partagent une arête, et *26-voisins* s'ils n'ont qu'un sommet en commun (cf Figure 1).

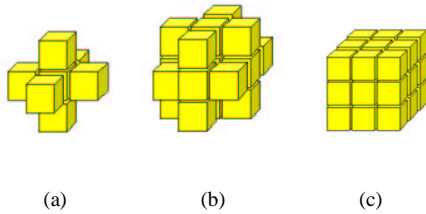


FIG. 1 – Les 3 voisinages classiques

Comme en géométrie euclidienne, il existe une définition arithmétique de plan discret.

**Définition 1** Soient  $a, b, c, r \in \mathbb{Z}$  et  $\omega \in \mathbb{N}^*$ . Un **plan discret** de vecteur normal  $(a, b, c)$  avec  $\text{pgcd}(a, b, c) = 1$ , de paramètre de translation  $r$  et d'épaisseur arithmétique  $\omega$  est l'ensemble noté  $P(a, b, c, r, \omega)$  des points  $M(x, y, z) \in \mathbb{Z}^3$  satisfaisant la double inégalité :

$$0 \leq ax + by + cz + r < \omega$$

On appelle **plan discret standard** un plan discret tel que  $\omega = |a| + |b| + |c|$ .

On appelle **plan discret naïf** un plan discret tel que  $\omega = \max(|a|, |b|, |c|)$ .

Un exemple de morceau de plan discret naïf est présenté Figure 2. L'épaisseur d'un plan détermine sa connectivité. Ainsi, les plans discrets naïfs sont les plans les plus fins sans trous, ce qui les rend particulièrement bien adaptés pour un travail sur la surface des objets discrets. En revanche, les plans standards sont des plans 6-connexes possédant des propriétés géométriques intéressantes comme nous le verrons par la suite.

## 2.2 Algorithme de reconnaissance de plans naïfs

Cette définition arithmétique a permis l'élaboration de plusieurs algorithmes efficaces de reconnaissance de plans discrets. Un tel algorithme a pour but de déterminer si les voxels d'un ensemble donné sont coplanaires, et si c'est le cas, de fournir une solution, i.e. les paramètres d'un plan euclidien dont la discrétisation (selon un schéma donné) contient l'ensemble de voxels donné.

Dans ce travail, nous utiliserons un algorithme proposé par Vittone [5] utilisant une transposition du problème dans l'espace de paramètres, ou espace dual. Cet algorithme est incrémental, et permet donc l'ajout successif de voxels lors du test de coplanarité. De plus, si les voxels sont coplanaires, il fournit non pas un plan solution, mais tous les

plans euclidiens solutions, cet ensemble étant représenté par un polyèdre dans l'espace des paramètres.

En effet, dans ce nouvel espace, un plan est représenté par un point dont les coordonnées dépendent de la normale et du paramètre de translation du plan. De même, un point dans l'espace original est représenté par un plan dans l'espace des paramètres. Ainsi, un voxel de coordonnées  $(x, y, z)$  est représenté par deux plans parallèles de normales  $(x, y, z)$  et  $(x, y, z - 1)$  si on se place dans le premier 48ième d'espace (ensemble des plans dont la normale  $(a, b, c)$  vérifie  $0 \leq a \leq b \leq c$ ). L'ensemble des plans solutions pour un ensemble de voxels donné est donc l'intersection de demi-espaces, donc un polyèdre convexe dans l'espace des paramètres. Si cette intersection est vide, cela signifie que les voxels ne sont pas coplanaires. Un exemple est présenté Figure 2. A gauche, un morceau de plan dans l'espace discret ; à droite, le polyèdre de l'ensemble des paramètres des plans euclidiens solutions.

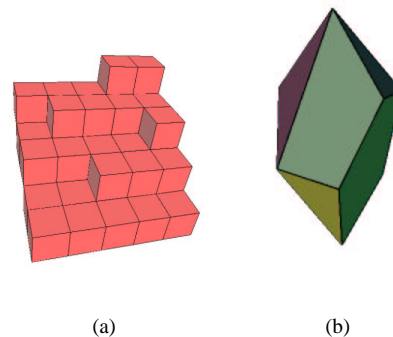


FIG. 2 – Un ensemble de voxels coplanaires et le polyèdre des solutions dans l'espace des paramètres. Le plan représenté a pour paramètres  $(1, 3, -5, 0)$ .

## 3 Algorithme de segmentation

Il reste maintenant à utiliser l'algorithme de reconnaissance de plans discrets présenté dans la section précédente sur la surface d'un objet discret quelconque. L'algorithme de segmentation proposé ici est basé sur un test de coplanarité permettant d'assurer une forme régulière aux morceaux de plans reconnus.

Pour ce faire, on utilise des structures basiques des plans discrets appelées *tricubes*. Soit  $P$  un plan discret du premier 48ième d'espace. Alors un tricube de  $P$  est un ensemble de 9 voxels voisins tels que la projection de ce tricube sur le plan  $(Oxy)$  est un carré de taille 3 :  $T(i, j) = \{(x, y, z) \mid i \leq x < i + 3, j \leq y < j + 3\}$ .

Il existe 40 tricubes différents et tout plan discret naïf est composé de tricubes [6]. Notre algorithme impose que chaque voxel d'un morceau de plan reconnu appartienne à un tricube. Ainsi, chaque voxel d'un morceau de plan discret a au moins 3 de ces 8 voisins dans ce même plan.

### Algorithme 1 : Segmentation d'une surface discrète en plans discrets

- 1: nbplans = 0
- 2: Pour les 6 faces de l'objet
- 3: Pour chaque voxel frontière de l'objet
- 4: Si ce voxel V n'a jamais été traité alors
  - origine = V
  - nbplans ++
  - mettre V dans la liste Atraiter
  - Tant que Atraiter est non vide
    - Choisir un voxel V' dans cette liste
    - Si V' est centre d'un tricube du plan courant alors
      - Ajouter V' au plan courant
      - Mettre ces 8 voisins dans la liste Atraiter
    - Sinon
      - Si V' = V alors nbplans -- et vider Atraiter

FIG. 3 – Algorithme de segmentation

Cet algorithme, présenté Figure 3 traite successivement les 6 directions d'un objet qui correspondent aux 6 faces d'un cube. L'origine de chaque morceau de plan est un voxel qui n'a jamais été traité et qui n'appartient donc à aucun plan. Cela assure que l'algorithme termine. Les morceaux de plans reconnus peuvent avoir des voxels communs : cela permet de limiter l'influence des choix des voxels graines des plans.

Nous avons maintenant une décomposition de la surface discrète d'un objet en morceaux de plans discrets. Pour passer de cette décomposition à un polyèdre euclidien, plusieurs étapes sont nécessaires.

## 4 Extraction d'une solution euclidienne et compression

### 4.1 Une première solution

Pour des objets convexes, il est maintenant possible d'obtenir un polyèdre euclidien à partir des morceaux de plans discrets reconnus. En effet, on a pour chaque morceau de plan l'ensemble des normales des plans euclidiens dont la discrétisation contient le morceau de plan discret. Ainsi, en calculant l'intersection des demi-espaces définis par les normales des plans euclidiens solutions, on obtient une approximation de l'objet discret.

Cette transformation n'est pas réversible : en effet, si on est sûr qu'une partie de chaque plan euclidien se discrétise dans l'objet, on ne contrôle pas ce qui se passe lors du calcul des intersections entre deux demi-espaces. Par conséquent, rien ne nous permet d'assurer que les arêtes et sommets du polyèdre obtenu par cette méthode se discrétisent dans l'objet discret original.

Les images présentées Figures 4 et 5 représentent les résultats obtenus pour des objets synthétisés simples, en donnant l'objet discret de départ et l'objet euclidien obtenu après transformation. La solution euclidienne choisie pour chaque morceau de plan discret est simplement le ba-

rycentre des sommets du polyèdre des solutions quand aucun sommet n'est solution, et un sommet solution sinon.

On voit que pour chacun des objets présentés ici, la solution obtenue est conforme à ce que l'on pouvait espérer.

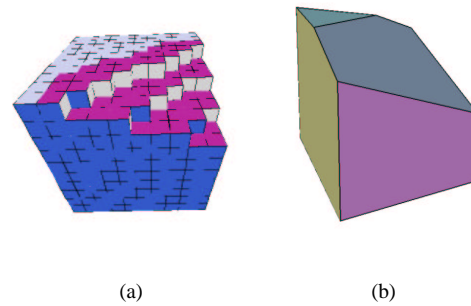


FIG. 4 – Exemple d'un cube chanfreiné : à gauche l'objet discret après reconnaissance des plans discrets ; à droite, l'objet euclidien obtenu

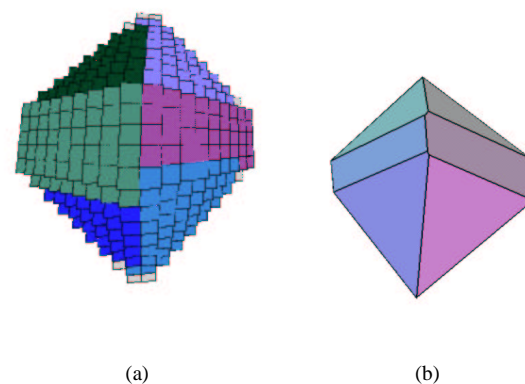


FIG. 5 – Exemple d'une pyramide à 6 côtés

La Figure 6 présente les résultats obtenus pour une sphère de rayon 20. On a vu dans la section précédente que l'algorithme proposé laisse des choix quant au sens de parcours des voxels de la surface de l'objet. Le schéma de propagation des plans discrets est aussi dépendant de ces choix. Les deux images présentées ici montrent les résultats obtenus pour deux schémas de propagation différents : à gauche, les plans sont étendus de manière quasi concentrique par rapport au voxel origine du plan ; à droite, le schéma de propagation est plus linéaire. On constate que le premier sens de parcours permet de mieux restituer les symétries de la sphère que le second. Une étude plus complète de l'influence des sens de parcours est faite dans [7].

### 4.2 Application à la compression

Pour stocker un objet discret, chacun des voxels de l'objet est nécessaire. Pour stocker la représentation euclidienne

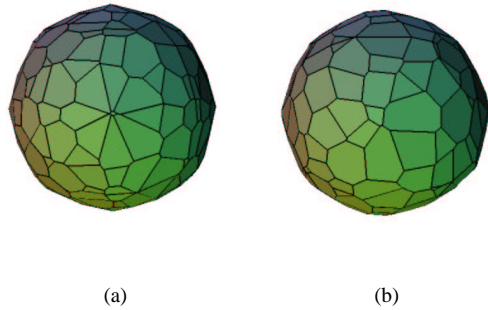


FIG. 6 – Exemple d'une sphère de rayon 20 pour deux sens de parcours différents

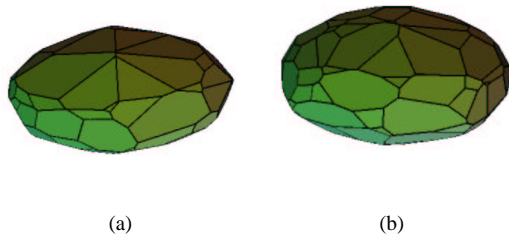


FIG. 7 – Exemple de deux ellipses de paramètres (15,10,8) (a) et (20,15,10) (b)

d'un objet, les 4 paramètres (vecteur normal + paramètre de translation) de chaque facette suffisent. Comme la transformation discret/euclidien n'est pas réversible, cette compression n'est pas sans perte, mais l'objet compressé garde une grande partie des caractéristiques géométriques et topologiques de l'objet original. Ainsi, cette méthode de compression sera bien sûr surtout efficace pour des objets polygonaux (cube, pyramide, etc), puisque le nombre de facettes sera assez faible dans ce cas, et indépendant de la taille de l'objet. Par exemple, stocker un cube nécessitera le stockage de  $6 \times 4 = 24$  entiers quelque soit le côté du cube.

Pour une sphère de rayon 20 par exemple, on a trouvé un polyèdre avec 269 facettes. Il faut donc stocker  $269 \times 4 = 1076$  entiers pour l'objet compressé. En revanche, si on considère que l'on stocke l'objet discret dans un cube de côté 40, l'espace mémoire nécessaire est de  $40^3 = 64000$  bits. Ainsi, le taux de compression obtenu dépend entièrement du codage des entiers : par exemple, si on considère un codage sur 4 octets, l'objet compressé demandera  $1076 \times 16 = 17216$  bits en mémoire, soit 27% de la place utilisée pour l'objet discret.

## 5 Conclusion

Le but de ce travail est de transformer un objet discret quelconque en polyèdre euclidien. Nous avons présenté ici

les premières briques pour arriver à un tel résultat. Tout d'abord, nous avons présenté un algorithme permettant de décomposer une surface discrète en morceaux de plans discrets. Nous savons que cette décomposition fournit des morceaux de plans avec des propriétés de régularité sur la forme et de recouvrement. Cet algorithme utilise un algorithme de reconnaissance de plans discret fournissant l'ensemble de solutions pour chaque morceau de plan reconnu. Nous avons dans un deuxième temps construit une première solution au problème pour les objets convexes, en donnant quelques indices sur les taux de compression que l'on peut espérer par une telle méthode.

Ce travail ouvre de nombreuses perspectives : tout d'abord élaborer un algorithme permettant d'obtenir une première solution pour un objet quelconque ; ensuite modifier cette solution pour obtenir une transformation réversible, et enfin voir si on peut définir un polyèdre discret dont les facettes soient des polygones discrets.

## Références

- [1] I. Debled-Rennesson et J.-P. Reveillès. An incremental algorithm for digital plane recognition. Dans Springer Verlag, éditeur, *DGCI'94*, pages 207–222, September 1994.
- [2] L. Papier et J. Françon. Polyhedrization of the boundary of a voxel object. Dans Springer Verlag, éditeur, *DGCI'99*, volume 1568 de *Lect. Notes of Comp. Sci.*, pages 425–434, 1999.
- [3] C.E. Kim et I. Stojmenović. On the recognition of digital planes in three dimensional space. *Pattern Recognition Letters*, 32 :612–618, 1991.
- [4] J. Burguet et Rémy Malgouyres. Strong thinning and polyhedrization of the surface of a voxel object. Dans Springer Verlag, éditeur, *DGCI'00*, volume 1953 de *Lect. Notes. of Comp. Sci.*, pages 222–234, Décembre 2000.
- [5] Joëlle Vittone. *Caractérisation et reconnaissance de droites et de plans en géométrie discrète*. Thèse de doctorat, Université Joseph Fourier, Grenoble, France, 1999.
- [6] J. Vittone et J.-M. Chassery. Coexistence of tricubes in digital naive planes. Dans Springer Verlag, éditeur, *DGCI'97*, volume 1347, pages 99–110, 1997.
- [7] I. Sivignon, F. Dupont, et J.-M. Chassery. Decomposition of a 3d discrete object surface into discrete plane pieces. *Algorithmica - Special Issue on Shape Algorithms*, Accepté pour publication.