

# Extension multirésolution des cartes combinatoires : application à la représentation de surfaces de subdivision multirésolution

Pierre Kraemer, David Cazier et Dominique Bechmann

LSIIT, UMR CNRS 7005, Université Louis Pasteur, Strasbourg, France

---

## Résumé

*Nous présentons dans cet article un nouveau cadre pour la représentation de maillages multirésolution. Celui-ci étend les hypercartes, modèle à base topologique général sur lequel reposent les modèles issus des cartes combinatoires. Une hypercarte multirésolution est définie comme une hiérarchie d'hypercartes imbriquées. Chaque niveau de résolution du maillage est défini par une hypercarte, bénéficiant de sa généralité et de son efficacité. A partir de ce cadre général, nous définissons les 2-cartes multirésolution et leur représentation duale, permettant de décrire la topologie de la subdivision de surfaces multirésolution.*

*Nous appliquons ici ce modèle à la représentation de surfaces de subdivision multirésolution. Les structures habituellement utilisées pour les représenter sont basées sur des quadrees, dérivant directement de la hiérarchie de faces générée par les schémas de subdivision. Ces structures souffrent d'un certain nombre de défauts : elles sont limitées aux maillages triangulaires ou carrés, et doivent être développées spécifiquement pour chacun de ces maillages; l'adaptivité génère une inconsistance topologique; la complexité en temps des requêtes d'adjacence n'y est pas optimale. Les 2-cartes multirésolution permettent de pallier ces défauts en permettant la représentation de maillages quelconques, un support consistant de l'adaptivité et une efficacité accrue des opérateurs d'adjacence.*

---

**Mots clé :** modélisation géométrique, modèle topologique, cartes combinatoires, maillages multirésolution, surfaces de subdivision

## 1. Introduction

La représentation multirésolution d'objets géométriques suscite un intérêt croissant dans de nombreux domaines de l'informatique graphique, et notamment en modélisation géométrique. Dans le cadre de la représentation par les bords, une surface est décrite par un maillage, c'est-à-dire une subdivision finie composée de sommets, d'arêtes et de faces. Dans une représentation multirésolution, cette surface n'est plus définie par un maillage unique, mais par une série de maillages imbriqués, et par les règles permettant de passer de l'un à l'autre de ces maillages. Ces maillages constituent les différents niveaux de résolution de l'objet représenté. La représentation est dite adaptative si la résolution maximale n'est pas la même pour toutes les zones de l'objet.

La représentation multirésolution trouve de nombreuses applications allant de la compression de maillages, à leur transmission progressive, en passant par l'affichage selon le point de vue ou l'édition multirésolution.

Les surfaces de subdivision multirésolution [ZSS97, Zor05] sont une des méthodes permettant de construire des surfaces multirésolution. De nombreux outils ont été proposés pour manipuler ces surfaces, comme l'insertion d'arêtes vives [BLZ00], l'application d'opérations booléennes [BKZ01] ou le copier-coller d'éléments de surface [BMBZ02].

En général, les auteurs de ces outils se focalisent sur la qualité des surfaces obtenues, et moins sur les structures de données sous-jacentes. Ceci conduit à l'utilisation de modèles qui ne sont pas des plus optimaux pour représenter les objets qu'ils manipulent. La structure de données classiquement utilisée est une forêt de quadrees (ou arbres quaternaires) dérivant naturellement de la hiérarchie de faces générée par les algorithmes de subdivision [DKP04]. Or les al-

algorithmes de subdivision ne travaillent pas tous sur le même type de maillage : certains s'appliquent à des maillages triangulaires, d'autres à des maillages polygonaux. Les structures à base de quadrees sont limitées aux maillages triangulaires ou carrés et doivent être développées spécifiquement pour chacun de ces maillages. La subdivision adaptative engendre des trous topologiques dans le maillage. De plus, elles sont limitées aux schémas de subdivision primaux, et les requêtes d'adjacence ne sont exécutées qu'en  $O(\log(n))$  (des solutions ont été proposées afin de pallier ce dernier défaut [Sch92, LS00] mais celles-ci présentent d'autres limitations, comme par exemple l'impossibilité d'une représentation adaptative).

Partant de ce constat, l'objectif de notre travail est de définir un cadre unifié et efficace pour la représentation de maillages multirésolution. Dans le contexte des surfaces de subdivision multirésolution, l'objectif est de pouvoir représenter des objets générés par le plus grand ensemble possible de schémas de subdivision, tout en maintenant la consistance topologique du maillage dans le cas adaptatif, et en améliorant l'efficacité des opérateurs d'adjacence – très sollicités par les algorithmes de subdivision.

Les cartes combinatoires [Edm60, Vin83, Tut84] sont un bon point de départ pour un tel modèle. En effet, celles-ci peuvent décrire des maillages quelconques, les requêtes d'adjacences y sont effectuées en temps constant, et les structures de données qui en dérivent sont légères. Nous définissons une carte multirésolution comme une hiérarchie de cartes emboîtées. Chaque niveau de résolution de l'objet représenté est ainsi défini par une carte combinatoire, bénéficiant de la généralité et de l'efficacité de ce modèle. Dans le cadre des surfaces de subdivision multirésolution, le modèle ainsi défini nous permet de répondre aux problèmes précédemment cités.

Cet article est organisé comme suit : dans la deuxième section, nous allons présenter le modèle des cartes combinatoires et le cadre plus général des hypercartes dans lequel elles se définissent. Dans la troisième section nous allons définir l'extension multirésolution de ce cadre. La quatrième section expose l'application des cartes multirésolution à la représentation de surfaces de subdivision multirésolution. La cinquième section est consacrée à la mise en œuvre de ce modèle en termes de structure de données, et compare celle-ci aux structures à base de quadrees. Enfin, la dernière section propose une conclusion et quelques perspectives à ce travail.

## 2. Cartes combinatoires et hypercartes

Les cartes combinatoires de dimension 2 (ou 2-cartes) et leurs extensions [Lie89, Lie91] ont fait l'objet de nombreux travaux [BD94, Duf97, CD99]. Ces modèles se définissent dans un cadre plus général qui repose sur la notion d'hypercartes [Duf91], dont nous rappelons quelques notions ci-après.

### 2.1. Les hypercartes

Les hypercartes définissent un cadre général permettant, en le contraignant, de définir des modèles à base topologique ordonnés. Ceux-ci peuvent décrire la topologie de différents types d'objets géométriques, tels que des surfaces ou des volumes, qu'ils soient ouverts ou fermés, orientés ou non, ... Cette topologie est décrite en subdivisant les objets en cellules de différentes dimensions (sommets, arêtes, faces, ...) partageant des relations d'incidence et d'adjacence. Nous nous focaliserons ici sur la représentation de maillages surfaciques.

Une hypercarte est constituée d'un ensemble de brins, reliés entre eux par des relations. Formellement, étant donné un ensemble fini de brins  $B$  et des permutations  $\alpha_0$  et  $\alpha_1$  sur cet ensemble  $B$ , une hypercarte est un triplet :

$$H = (B, \alpha_0, \alpha_1)$$

Les cellules de la subdivision ne sont pas représentées explicitement dans le modèle, mais sont définies *implicitement* par des sous-ensembles de brins. Ces sous-ensembles sont obtenus grâce à la notion d'orbite. Pour une permutation  $\sigma$  sur  $B$ , l'orbite de  $x \in B$  est le sous-ensemble de  $B$  noté  $\langle \sigma \rangle(x)$  égal à  $\{x, \sigma(x), \dots, \sigma^k(x)\}$ , où  $k$  est le plus petit entier positif tel que  $\sigma^{k+1}(x) = x$ . On voit clairement que tous les éléments de  $\langle \sigma \rangle(x)$  ont la même orbite. En pratique, appliquée à un brin  $x$  de  $B$ , l'orbite représente l'ensemble des brins atteignables depuis  $x$  par les applications successives de la permutation  $\sigma$ .

Un tel modèle ne décrit que la topologie de la subdivision des objets. Il faut donc définir un modèle de plongement décrivant les données géométriques, que l'on va associer à chaque cellule du maillage, pour représenter complètement un objet. Le plongement le plus simple est le 0-plongement, où seules les 0-cellules sont plongées. C'est-à-dire que l'on associe un point 3D à chaque sommet du maillage. Le plongement des autres cellules est obtenu par interpolation linéaire du plongement des sommets. Des modèles de plongement plus évolués peuvent bien sûr être utilisés, associant par exemple des courbes aux arêtes ou des carreaux de surface aux faces.

### 2.2. Les 2-cartes

Une 2-carte est une hypercarte à laquelle on ajoute la contrainte suivante :  $\alpha_0$  est une involution sans point fixe – i.e. une permutation telle que  $\forall x \in B, \alpha_0(\alpha_0(x)) = x$  et  $\alpha_0(x) \neq x$ . Ce modèle permet de représenter la topologie de la subdivision de 2-variétés orientables fermées, i.e. de surfaces englobant des volumes. La figure 1 présente les conventions graphiques adoptées pour la représentation des brins et des relations.

Les sommets sont définis par l'orbite  $\langle \alpha_1 \rangle$ , les arêtes par l'orbite  $\langle \alpha_0 \rangle$ , et les faces par l'orbite  $\langle \alpha_0 \circ \alpha_1 \rangle$ .

La figure 2 illustre un exemple de 2-carte. Celle-ci est

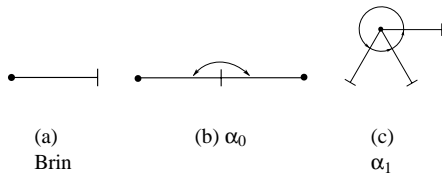


Figure 1: Conventions graphiques

composée de cinq sommets :  $\{1, 7, 5\}$ ,  $\{2, 3\}$ ,  $\{4, 6, 9\}$ ,  $\{8, 11\}$ ,  $\{10, 12\}$ ; six arêtes :  $\{1, 2\}$ ,  $\{3, 4\}$ ,  $\{5, 6\}$ ,  $\{7, 8\}$ ,  $\{9, 10\}$ ,  $\{11, 12\}$ ; et trois faces :  $\{1, 3, 6\}$ ,  $\{2, 7, 11, 10, 4\}$ ,  $\{5, 9, 12, 8\}$ .

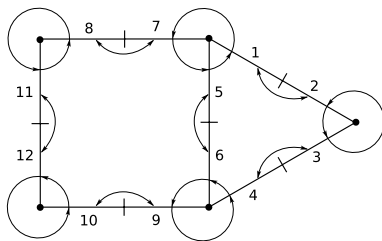


Figure 2: Exemple de 2-carte

### 2.3. Les 2-cartes duales

Etant donnée une 2-carte  $M = (B, \alpha_0, \alpha_1)$ , le triplet  $M' = (B, \phi_1, \phi_2)$ , avec  $\phi_1 = \alpha_0 \circ \alpha_1$ , et  $\phi_2 = \alpha_0$ , est aussi une 2-carte, appelée carte duale de  $M$ . La relation  $\phi_1$  est une permutation, et  $\phi_2$  est une involution sans point fixe.

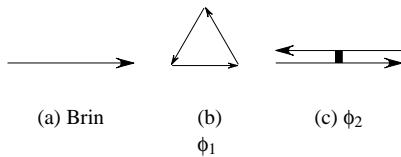


Figure 3: Conventions graphiques

La figure 3 présente les conventions graphiques adoptées pour la représentation des 2-cartes duales. Un brin est représenté par une flèche. Une séquence de brins liés par  $\phi_1$ , i.e. une orbite de  $\phi_1$ , forme une face orientée. La relation  $\phi_2$  est utilisée pour lier deux faces orientées le long d'arêtes d'orientations opposées.

Les sommets sont définis par l'orbite  $\langle \phi_1 \circ \phi_2 \rangle$ , les arêtes par l'orbite  $\langle \phi_2 \rangle$ , et les faces par l'orbite  $\langle \phi_1 \rangle$ .

La figure 4 montre le même exemple que pour les 2-cartes

primales, mais en représentation duale. Les brins sont numérotés de la même manière, et l'on retrouve donc les mêmes cellules. On voit clairement ici la face externe qui ferme la surface.

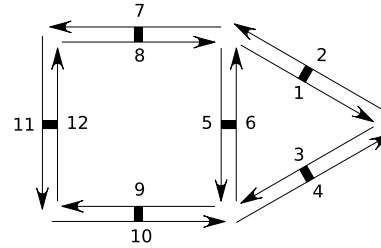


Figure 4: Exemple de 2-carte duale

### 3. Les cartes multirésolution

Nous allons commencer par présenter les hypercartes multirésolution, définies comme une extension des hypercartes présentées ci-dessus.

Nous allons ensuite présenter les mêmes exemples de restriction du modèle général que pour les hypercartes, à savoir les 2-cartes multirésolution et leur définition duale.

#### 3.1. Les hypercartes multirésolution

Les hypercartes multirésolution fournissent un cadre permettant de définir la topologie d'objets multirésolution. Chaque niveau de résolution est représenté par une hypercarte. Ceci forme un ensemble d'hypercartes s'imbriquant naturellement, les brins décrivant un niveau donné étant réutilisés dans les niveaux plus fins, tout en conservant les relations topologiques entre chaque niveau. Nous nous focaliserons ici sur les surfaces multirésolution.

Plus formellement, une hypercarte multirésolution est composée d'un ensemble de brins  $B$  et de relations entre les éléments de cet ensemble. Les brins de  $B$  sont ici indexés par le niveau de résolution auquel ils sont introduits. Les brins décrivant l'objet au niveau le plus grossier appartiennent à l'ensemble  $B^0$ . Des brins sont ajoutés à cet ensemble pour décrire le maillage de niveau supérieur (au sens "plus fin") et former l'ensemble  $B^1$ . Pour décrire chaque niveau de résolution supérieur, des brins sont ajoutés à ceux décrivant le maillage de niveau inférieur, constituant ainsi une imbrication d'ensembles.

Ainsi, dans une hypercarte multirésolution de niveau de résolution maximum  $k$ , l'ensemble de brins  $B$  est une suite  $\{B^i\}_{i \in [0, k]}$  d'ensembles telle que :

$$B^0 \subset B^1 \subset B^2 \subset \dots \subset B^k = B.$$

L'ensemble  $B^i$  est le sous-ensemble de  $B$  contenant les brins qui ont été introduits à un niveau inférieur ou égal à  $i$ .

L'ensemble  $B^i \setminus B^{i-1}$  contient les brins qui ont été introduits au niveau  $i$ . Le cardinal de  $B$  est le nombre de brins de la carte décrivant le maillage au niveau le plus fin.

Les relations topologiques entre les brins sont ici paramétrées par le niveau de résolution. Pour une permutation  $\sigma$  et pour un brin  $x \in B^i$ ,  $\sigma^i(x)$  est le brin lié à  $x$  par la relation  $\sigma$  au niveau de résolution  $i$ .

Ainsi, une hypercarte multirésolution est un triplet :

$$M = \left( B, \{\alpha_0^i\}_{i \in [0, k]}, \{\alpha_1^i\}_{i \in [0, k]} \right)$$

tel que pour tout  $i \in [0, k]$ , le triplet  $M^i = (B^i, \alpha_0^i, \alpha_1^i)$  soit une hypercarte décrivant le niveau de résolution  $i$ .

Les cellules d'un niveau de résolution  $i$  sont définies sur l'hypercarte  $M^i$  correspondante de la même manière que pour les hypercartes. Une cellule de niveau  $i$  est définie implicitement par un sous-ensemble des brins de  $B^i$ , obtenu à l'aide d'orbites.

Le plongement est lui aussi paramétré par le niveau de résolution : à chaque cellule de niveau  $i$ , avec  $i \in [0, k]$ , est associée une information géométrique. C'est-à-dire que l'on peut avoir des plongements distincts pour chaque cellule et ceci pour chaque niveau de résolution.

### 3.2. Les 2-cartes multirésolution

Une 2-carte multirésolution est une hypercarte multirésolution à laquelle on ajoute la contrainte suivante :  $\alpha_0^i$  est une involution sans point fixe – i.e. une permutation telle que  $\forall i \in [0, k], \forall x \in B^i, \alpha_0^i(\alpha_0^i(x)) = x$  et  $\alpha_0^i(x) \neq x$ . Cette contrainte est la même que celle ajoutée aux hypercartes afin de définir les 2-cartes. Ce modèle permet de représenter la topologie de maillages multirésolution décrivant des 2-variétés orientables fermées.

Chaque niveau de résolution  $i$ , avec  $i \in [0, k]$ , est représenté par une 2-carte :

$$M^i = (B^i, \alpha_0^i, \alpha_1^i)$$

Les sommets du niveau  $i$  sont définis par l'orbite  $\langle \alpha_1^i \rangle$ , les arêtes par l'orbite  $\langle \alpha_0^i \rangle$ , et les faces par l'orbite  $\langle \alpha_0^i \circ \alpha_1^i \rangle$ .

La figure 5(a) illustre un exemple de 2-carte multirésolution à un niveau  $i$ , composée de 4 sommets, 4 arêtes et 2 faces. Les figures 5(b), 5(c), 5(d) illustrent trois exemples de ce que pourrait être le niveau  $i+1$ . Dans les trois cas, les brins en orange sont ceux qui ont été introduits entre les deux niveaux – ceux appartenant à l'ensemble  $B^{i+1} \setminus B^i$ . Suivant les insertions de brins effectuées entre les deux niveaux, les relations liant des brins au niveau  $i$  ne sont pas nécessairement différentes au niveau  $i+1$ . Lorsque deux brins liés au niveau  $i$  ne le sont plus au niveau  $i+1$ , un rappel de leur liaison de niveau  $i$  est dessiné en pointillés bleus.

Dans l'exemple (b), les arêtes de la carte de niveau  $i$  sont coupées. Pour les brins de  $B^i$ , les relations  $\alpha_0^{i+1}$  ne sont pas

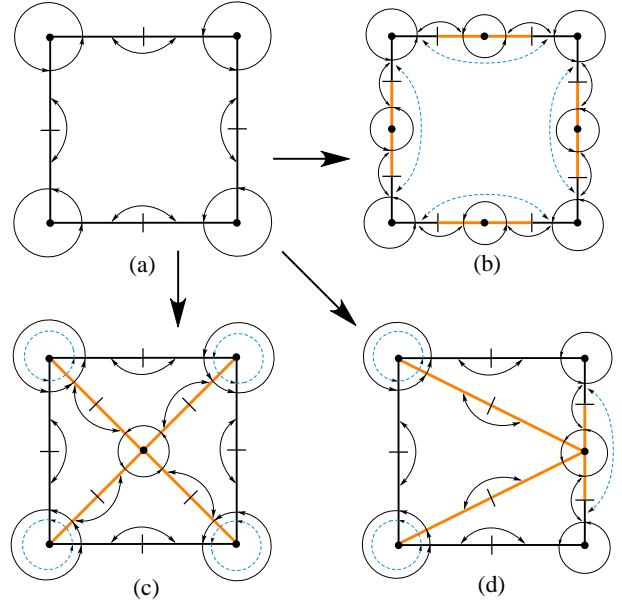


Figure 5: Exemples de 2 niveaux successifs de 2-cartes multirésolution

égales aux relations  $\alpha_0^i$ . Par contre les valences des sommets du niveau  $i$  ne changent pas, et donc pour les brins de  $B^i$ , les relations  $\alpha_1^{i+1}$  sont égales aux relations  $\alpha_1^i$ . La carte de niveau  $i+1$  est composée de 8 sommets, 8 arêtes et 2 faces.

Dans l'exemple (c), de nouvelles arêtes sont introduites entre les sommets de la carte de niveau  $i$ . Pour certains brins de  $B^i$ , les relations  $\alpha_1^{i+1}$  ne sont pas égales aux relations  $\alpha_1^i$ . Par contre les arêtes du niveau  $i$  ne sont pas coupées, et donc pour les brins de  $B^i$ , les relations  $\alpha_0^{i+1}$  sont égales aux relations  $\alpha_0^i$ . La carte de niveau  $i+1$  est composée de 5 sommets, 8 arêtes et 5 faces.

Dans l'exemple (d) une arête du niveau  $i$  est coupée et de nouvelles arêtes sont insérées entre des sommets du niveau  $i$  et un nouveau sommet du niveau  $i+1$ . Pour certains brins de  $B^i$ , les relations au niveau  $i+1$  diffèrent de celles au niveau  $i$ . On a ici 5 sommets, 7 arêtes et 4 faces.

### 3.3. Les 2-cartes multirésolution duales

Etant donnée une 2-carte multirésolution :

$$M = \left( B, \{\alpha_0^i\}_{i \in [0, k]}, \{\alpha_1^i\}_{i \in [0, k]} \right)$$

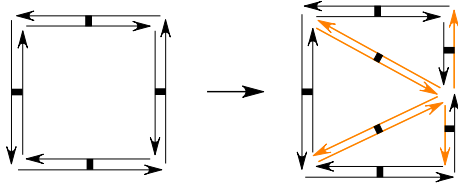
le triplet :

$$M' = \left( B, \{\phi_1^i\}_{i \in [0, k]}, \{\phi_2^i\}_{i \in [0, k]} \right)$$

avec  $\phi_1^i = \alpha_0^i \circ \alpha_1^i$  et  $\phi_2^i = \alpha_0^i$ , est aussi une 2-carte multirésolution, appelée carte duale de  $M$ . Les  $\phi_1^i$  sont des permutations, et les  $\phi_2^i$  sont des involutions sans point fixe.

Chaque niveau de résolution  $i$ , avec  $i \in [0, k]$ , est représenté par une 2-carte duale :

$$M^i = (B^i, \phi_1^i, \phi_2^i)$$



**Figure 6:** 2 niveaux successifs d'une 2-carte multirésolution duale

Les sommets du niveau  $i$  sont définis par l'orbite  $\langle \phi_1^i \circ \phi_2^i \rangle$ , les arêtes par l'orbite  $\langle \phi_2^i \rangle$ , et les faces par l'orbite  $\langle \phi_1^i \rangle$ .

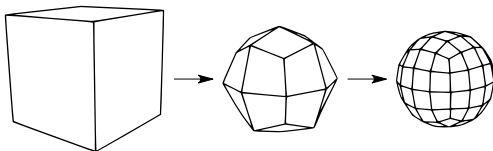
La figure 6 illustre en représentation duale le même exemple qu'aux figures 5(a) et 5(d). Des brins sont insérés entre les deux niveaux  $i$  et  $i + 1$  (ceux représentés en gris), et pour certains brins de  $B^i$  les relations au niveau  $i + 1$  sont différentes de celles au niveau  $i$ .

#### 4. Application aux surfaces de subdivision multirésolution

Nous allons voir dans cette section comment les modèles que nous avons définis dans la section précédente permettent de représenter efficacement des surfaces de subdivision multirésolution. Nous allons dans un premier temps introduire les principes des surfaces de subdivision ainsi que de leur extension multirésolution.

##### 4.1. Définitions

Le principe des surfaces de subdivision est de définir une surface comme la limite d'une séquence de maillages raffinés successivement. Un schéma de subdivision définit la manière d'appliquer ce raffinement au maillage (figure 7). On peut distinguer ici deux étapes : d'abord, la topologie du maillage est raffinée, puis de nouvelles informations géométriques sont calculées à partir du maillage de départ et associées au maillage obtenu. Différents schémas génèrent des surfaces aux propriétés différentes.



**Figure 7:** 2 itérations de subdivision sur un cube

Le processus de calcul de la géométrie d'un schéma de subdivision est habituellement classé en deux familles, appelées interpolante et approximante. Dans les deux cas,

les informations géométriques associées aux sommets sont calculées localement pour chaque sommet en appliquant un masque sur les positions des sommets voisins dans le maillage de départ. Dans un schéma approximant, les positions de tous les sommets du nouveau maillage sont calculées et se rapprochent de la surface limite. Dans un schéma interpolant, les sommets du maillage de départ se situent déjà sur la surface limite, et seules les positions des nouveaux sommets insérés sont calculées.

Le processus de raffinement de la topologie d'un schéma de subdivision est également classé en deux familles, appelées primale et duale. Nous allons voir dans la suite les différences entre ces schémas, et comment les modèles que nous avons définis dans la section précédente permettent de les supporter.

Le principe de l'extension multirésolution des surfaces de subdivision est de conserver tous les maillages intermédiaires du processus de subdivision – appelés alors niveaux de résolution – et d'introduire des vecteurs de détail entre ces niveaux. Ces vecteurs expriment la différence en chaque sommet entre sa position au niveau  $i$  et celle résultant de la subdivision du niveau  $i - 1$ .

Une forte connexion existe entre les surfaces multirésolution et les ondelettes, et en particulier les deux opérations d'analyse et de synthèse. L'analyse calcule les positions des sommets du maillage de niveau  $i - 1$  en appliquant un filtre moyenneur sur le maillage de niveau  $i$  et calcule par la même occasion les vecteurs de détail. La synthèse reconstruit les données au niveau  $i$  en subdivisant le maillage de niveau  $i - 1$  et en y ajoutant les vecteurs de détail.

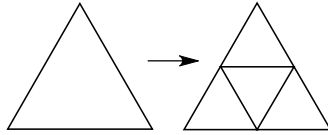
Deux méthodes existent pour générer des surfaces de subdivision multirésolution : la méthode "grosier vers fin" et la méthode "fin vers grosier". La première démarre d'un objet grosier (qui constituera le niveau de résolution minimum de l'objet), et construit les niveaux fins en appliquant un schéma de subdivision. La deuxième part d'un objet fin (qui constituera le niveau de résolution maximum de l'objet) et construit les niveaux grossiers en appliquant le processus d'analyse – dans ce cas, le maillage de départ doit avoir une connectivité de subdivision, i.e. la même connectivité que s'il avait été généré par un schéma de subdivision.

Une fois un tel objet construit, le maillage peut être édité à des niveaux de résolution différents : une édition à un niveau grosier entraîne une déformation globale, en conservant les informations fines grâce aux vecteurs de détail; une édition à un niveau fin entraîne une déformation plus locale de l'objet. A chaque édition les niveaux plus fins sont mis à jour par le processus de synthèse, et les niveaux plus grossiers par le processus d'analyse.

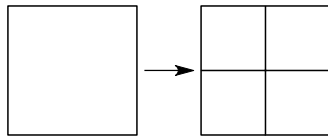
##### 4.2. Schémas primaux

Dans un schéma de subdivision primal, ce sont les faces du maillage qui sont subdivisées. Ceci est réalisé en cou-

par leurs arêtes et en reliant les sommets ainsi créés afin de former de nouvelles faces. Les schémas de Loop [Loo87], de Catmull-Clark [CC78] ou le schéma Butterfly [DLG90, ZSS96] sont des exemples de schémas primaux. Ceux-ci travaillent sur des maillages de type différents : le schéma de Loop (illustré à la figure 8a) ou le Butterfly sont basés sur des maillages triangulaires, celui de Catmull-Clark (illustré à la figure 8b) sur des maillages quelconques (après un pas de subdivision, toutes les faces sont des quadrilatères).



(a) Loop



(b) Catmull-Clark

**Figure 8:** *Subdivision primale*

Les surfaces de subdivision multirésolution issues de tous ces schémas peuvent être représentées par des 2-cartes multirésolution. Chaque niveau de résolution est alors représenté par une 2-carte permettant de décrire des maillages quelconques.

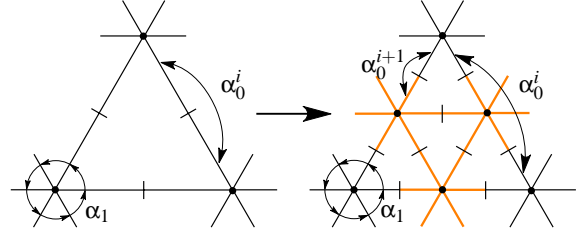
Lors d'une étape de subdivision d'un schéma primal, la valence des sommets existants dans le maillage initial n'est pas modifiée. Les seules opérations effectuées sont des sections d'arêtes et des insertions d'arêtes, ces dernières ayant lieu uniquement entre des nouveaux sommets. Cela revient à dire qu'aucun brin n'est inséré dans une permutation  $\alpha_1$  entre deux niveaux, et donc qu'une fois un brin inséré dans la carte à un niveau  $i$ , le brin auquel il est lié par  $\alpha_1$  ne change pas aux niveaux supérieurs. Formellement, cela signifie que  $\forall x \in B^i \setminus B^{i-1}, \alpha_1^j(x) = \alpha_1^i(x)$ , pour  $i < j \leq k$ .

Ainsi la topologie d'une surface de subdivision multirésolution peut être modélisée par une 2-carte multirésolution spécialisée en :

$$M = (B, \{\alpha_0^i\}_{i \in [0, k]}, \alpha_1)$$

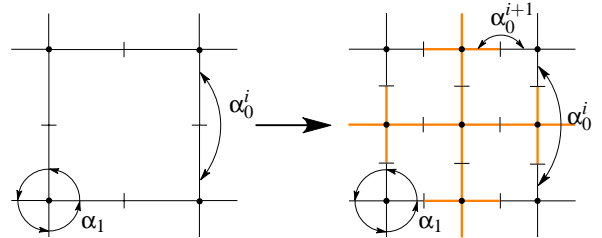
Chaque niveau de résolution est alors représenté par la 2-carte  $M^i = (B^i, \alpha_0^i, \alpha_1|_{B^i})$  où  $\alpha_1|_{B^i}$  est la restriction de  $\alpha_1$  aux éléments de  $B^i$ .

La figure 9 illustre un détail de maillage triangulaire aux

**Figure 9:** *Loop avec une 2-carte multirésolution*

niveaux  $i$  et  $i + 1$  d'une subdivision de Loop. Les brins en orange sont ceux introduits entre les deux niveaux – i.e. ceux appartenant à  $B^{i+1} \setminus B^i$ .

La figure 10 illustre de la même manière un détail de maillage carré aux niveaux  $i$  et  $i + 1$  d'une subdivision de Catmull-Clark.

**Figure 10:** *Catmull-Clark avec une 2-carte multirésolution*

L'adaptativité consiste dans les schémas primaux à ne plus subdiviser une face à partir d'un niveau donné. Ceci implique que certaines arêtes du maillage ne sont plus coupées. Formellement, si l'arête du brin  $x$  n'est plus coupée à partir du niveau de résolution  $i$ , cela signifie que  $\alpha_0^j = \alpha_0^i$ , pour  $i < j \leq k$ . La figure 11 illustre un détail d'un maillage triangulaire aux niveaux  $i$  et  $i + 1$ . Entre ces deux niveaux, seule la face centrale est subdivisée. Dans la 2-carte représentant le niveau  $i + 1$ , les relations  $\alpha_0$  des brins appartenant à des arêtes n'ayant pas été coupées sont égales à celles du niveau précédent, et les triangles n'ayant pas été subdivisés deviennent ici simplement des quadrilatères. Grâce à la représentation implicite des cellules, le maillage est donc ici toujours topologiquement consistant – aucun trou n'apparaît dans la surface.

La figure 12 illustre deux niveaux  $i$  et  $i + 1$  d'une subdivision de Catmull-Clark adaptative avec un quadtree (à gauche) et avec une 2-carte multirésolution (à droite). Avec le quadtree on voit clairement le trou topologique qui est introduit, alors qu'avec la 2-carte multirésolution, la face de gauche devient simplement une face à cinq côtés.

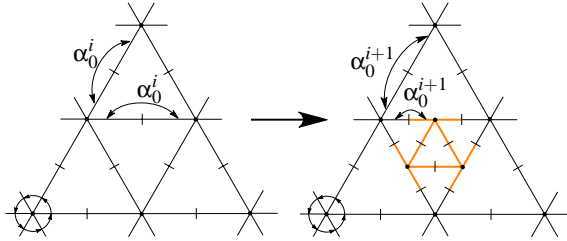


Figure 11: Loop avec une 2-carte multirésolution adaptative

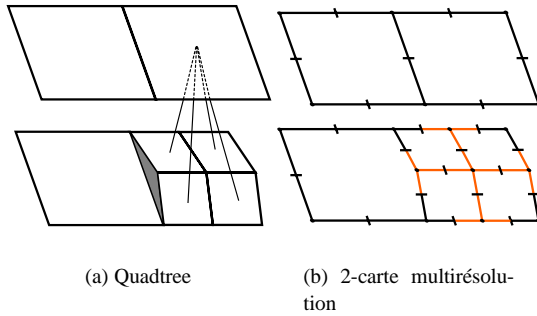


Figure 12: Catmull-Clark avec un quadtree et avec une 2-carte multirésolution

### 4.3. Schémas duaux

Dans un schéma de subdivision dual, ce sont les sommets du maillage qui sont subdivisés. Les faces du maillage de départ sont réduites, et les trous ainsi créés sont comblés avec de nouvelles faces. Les schémas de Doo-Sabin [Doo78, DS78] (illustré à la figure 13) ou le schéma Mid-Edge [PR97] sont des exemples de schémas duaux. Ceux-ci travaillent sur des maillages quelconques.

Les 2-cartes multirésolution duales peuvent représenter des surfaces de subdivision multirésolution issues de tous ces schémas.

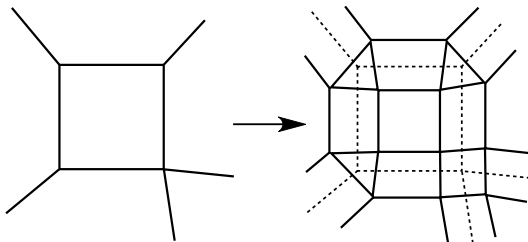


Figure 13: Subdivision duale

Lors d'une étape de subdivision d'un schéma dual, le nombre de côtés des faces existantes dans le maillage initial n'est pas modifié. Les seules opérations effectuées sont

des insertions de faces. Cela revient à dire qu'aucun brin n'est inséré dans une permutation  $\phi_1$  entre deux niveaux, et donc qu'une fois un brin inséré dans la carte à un niveau  $i$ , le brin auquel il est lié par  $\phi_1$  ne change pas aux niveaux supérieurs. Formellement, cela signifie que  $\forall x \in B^i \setminus B^{i-1}, \phi_1^j(x) = \phi_1^i(x)$ , pour  $i < j \leq k$ .

Ainsi la topologie d'une surface de subdivision multirésolution duale peut être modélisée par une 2-carte multirésolution duale spécialisée en :

$$M = (B, \phi_1, \{\phi_2^i\}_{i \in [0, k]})$$

Chaque niveau de résolution est alors représenté par la 2-carte duale  $M^i = (B^i, \phi_1|_{B^i}, \phi_2^i)$  où  $\phi_1|_{B^i}$  est la restriction de  $\phi_1$  aux éléments de  $B^i$ .

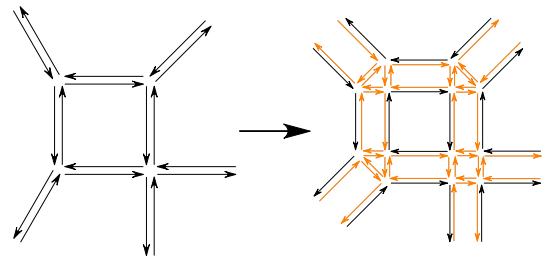


Figure 14: Doo-Sabin avec une 2-carte multirésolution duale

La figure 14 illustre un détail de maillage aux niveaux  $i$  et  $i + 1$  d'une subdivision de Doo-Sabin. Les brins en orange sont ceux introduits entre les deux niveaux – ceux appartenant à  $B^{i+1} \setminus B^i$ . La figure 15 illustre un exemple d'objet obtenu en utilisant le schéma de Doo-Sabin.

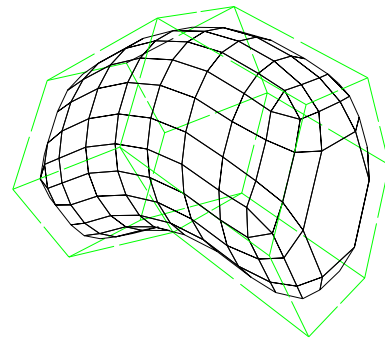


Figure 15: Exemple de 2-carte multirésolution duale

## 5. Mise en œuvre et comparaison

Dans cette section nous allons présenter un exemple de structure de données qui implante le modèle des 2-cartes multirésolution que nous venons de définir. Nous nous focalisons ici sur la représentation primale, mais la version duale s'implante aussi simplement.

Nous comparerons ensuite cette structure avec les forêts de quadrees, classiquement utilisées pour la représentation de surfaces de subdivision multirésolution adaptatives, aussi bien en complexité en temps qu'en besoins en mémoire.

### 5.1. Structure de données

Une 2-carte multirésolution consiste en un ensemble de brins. Comme nous l'avons vu dans la définition du modèle, l'ensemble de brins  $B$  est une suite  $\{B^i\}_{i \in [0,k]}$  d'ensembles telle que  $B^0 \subset B^1 \subset B^2 \subset \dots \subset B^k = B$ . Les brins peuvent donc être stockés dans un tableau de listes. La liste située dans la  $i^{\text{eme}}$  case du tableau contient l'ensemble de brins  $B^i \setminus B^{i-1}$  – i.e. les brins insérés au niveau de résolution  $i$ .

Chaque brin contient des pointeurs vers d'autres brins, matérialisant ainsi les relations topologiques. Le lien  $\alpha_1$  est directement représenté par un pointeur vers un autre brin. Les liens  $\alpha_0^i$  sont stockés dans un tableau de pointeurs. Comme un brin n'a pas de relations à un niveau de résolution inférieur à son propre niveau d'insertion, ce tableau ne stocke que les relations nécessaires. Soit  $a$  le niveau d'insertion du brin  $x$  ( $x \in B^a \setminus B^{a-1}$ ) et soit  $k$  le niveau de résolution maximum : le brin  $x$  ne stocke que  $k - a$  relations  $\alpha_0$ . Ainsi l'accès au brin lié à  $x$  par la relation  $\alpha_0^i$  est donné en suivant le pointeur situé dans la case  $i - a$  du tableau de relations.

L'information géométrique est attachée directement aux brins. Nous utilisons ici un simple 0-plongement (i.e. seules les 0-cellules sont plongées). Un brin plongé est un brin qui contient un pointeur vers un point 3D. Tous les brins d'un même sommet pointent vers le même point 3D. Comme le plongement est également paramétré par le niveau de résolution (un sommet reçoit un plongement quand il est inséré, puis peut changer de plongement à chaque niveau supérieur), les pointeurs vers le plongement sont également stockés dans un tableau. De même que pour les liens  $\alpha_0$ , un brin ne stocke que le nombre nécessaire de pointeurs vers des points 3D. Soit  $a$  le niveau d'insertion du brin  $x$  et soit  $k$  le niveau de résolution maximum : le brin  $x$  ne stocke que  $k - a$  pointeurs vers des points 3D. Ainsi l'accès au plongement de niveau  $i$  du sommet du brin  $x$  est donné en suivant le pointeur situé dans la case  $i - a$  du tableau.

Le listing 5.1 donne une exemple de ce que pourrait être l'implantation des 2-cartes multirésolution en C++.

Comme nous l'avons précisé plus haut, les cellules de la subdivision sont représentées implicitement à chaque niveau  $i$  par des sous-ensembles de brins de  $B^i$  construits à l'aide d'orbites. Ces parcours d'orbite peuvent être simplement implantés par des itérateurs. Le code correspondant à un itérateur de face peut être par exemple présenté dans le listing 5.2. Cet itérateur est construit en fonction d'un brin donné et du niveau de résolution voulu. Il est évidemment nécessaire que le brin ait été inséré dans la carte à un niveau inférieur au niveau demandé. Deux itérateurs de même

---

#### Listing 5.1

---

```
class CarteMultiresolution
{
    vector<list<Brin>> carte;
};

class Brin
{
    vector<Brin*> a0;
    Brin *a1;
    vector<Point3D*> em0;
};
```

---

niveau initialisés avec des brins appartenant à la même orbite sont considérés comme égaux. Les itérateurs correspondant aux autres cellules de la subdivision sont implantés de la même manière.

---

#### Listing 5.2

---

```
class FaceIterateur
{
    Brin *first, *current;
    int niv;

    FaceIterateur(Brin *b, int n)
    {
        first = current = b;
        niv = n;
    }

    void next()
    {
        current =
            current->alpha0(niv)->alpha1();
    }

    bool end()
    {
        return current==first;
    }
};
```

---

Un itérateur de cellule peut être utilisé au sein d'un objet correspondant à une cellule et disposant de méthodes pouvant agir sur cette cellule. Un tel objet peut être implanté tel que présenté dans le listing 5.3. Ce type d'objet n'existe que le temps d'effectuer un traitement sur la cellule.

Ces structures ont été implantées au sein d'un modèleur. La figure 16 illustre des exemples d'objets obtenus par subdivision adaptative avec le schéma de Catmull-Clark. Les objets peuvent être édités à n'importe quel niveau de résolution. L'adaptativité est guidée automatiquement durant l'édition par un critère de courbure : une face n'est subdivisée à un niveau donné que si l'angle formé avec ses faces voisines

**Listing 5.3**

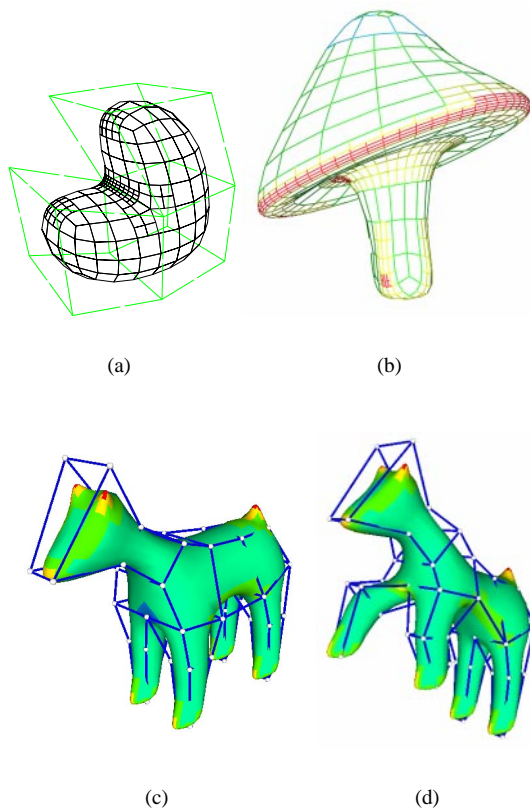
```

class Face
{
    FaceIterateur f;

    Face(Brin *b, int n) : f(b,n)
    {
    }

    void affiche();
    Point3D normale();
    void subdivise();
};

```

**Figure 16:** Exemple d'application

dépasse un certain seuil; si l'angle repasse sous le seuil fixé, la face est à nouveau simplifiée.

Nous avons introduit une restriction – non nécessaire dans la définition du modèle – qui interdit à deux faces adjacentes d'avoir plus d'un niveau de différence. Bien que ceci engendre un peu plus de subdivision que nécessaire autour des régions de forte courbure, cela a plusieurs avantages. D'une part, cela permet d'avoir toujours des masques de subdivision pleins, c'est-à-dire qu'un sommet de niveau  $i$  dispose toujours de tous ses voisins du niveau  $i - 1$  nécessaires au calcul de sa position, évitant ainsi le calcul temporaire des positions des sommets qui manqueraient. D'autre part, cela permet de trianguler le maillage obtenu bien plus facilement en vue d'un affichage, en réduisant le nombre de cas à traiter.

**5.2. Comparaison****Complexité en temps**

Dans le cadre des surfaces de subdivision, les requêtes d'adjacence entre sommets sont une des opérations les plus courantes. Celles-ci sont nécessaires aussi bien pour l'exécution de l'opérateur de synthèse que pour celui d'analyse.

Dans une hypercarte multirésolution, les requêtes d'adjacence sont exécutées en temps constant, quel que soit le niveau de résolution considéré. En effet, les cellules adjacentes sont récupérées directement en suivant un nombre donné de pointeurs. Ce qui est vrai pour les hypercartes multirésolution est vrai également pour les 2-cartes multirésolution. Par exemple, au niveau de résolution  $i$ , visiter tous les sommets voisins d'un sommet défini par le brin  $x$  peut être effectué tel que présenté dans le listing 5.4 : on parcourt simplement l'orbite du sommet de  $x$  grâce à l'itérateur de sommet; pour chaque brin de cette orbite, on atteint le sommet voisin en traversant l'arête correspondante, ceci en suivant simplement l'involution  $\alpha_0$  du niveau correspondant.

Des algorithmes similaires peuvent être écrits par exemple pour visiter les faces adjacentes à une face d'un niveau de résolution  $i$  donné (par exemple dans le but de calculer les angles entre la normale d'une face et celles de ses voisines) : pour cela on parcourt l'orbite face et pour chaque brin de cette orbite, on arrive sur un brin de sa face voisine en suivant simplement le lien  $\alpha_0$  du niveau voulu.

Dans une forêt de quadrees, les requêtes de voisinage entre faces au sein d'un quadree sont résolues en remontant dans l'arbre jusqu'au parent commun des deux faces partageant une arête. Le voisinage entre les différents quadrees de la forêt est résolu explicitement au niveau d'un ensemble de racines qui constitue le niveau 0 de l'objet. Ces requêtes sont exécutées en  $O(\log(n))$ , avec  $n$  la profondeur de l'arbre, c'est-à-dire le niveau de résolution maximum. Même si en pratique  $\log(n)$  n'atteint pas une valeur très élevée, celle-ci n'est pas 1, et ces opérations étant les plus utilisées pour la mise à jour de l'objet durant l'édition d'un maillage, cette amélioration n'est pas négligeable.

**Listing 5.4**

```

class SommetIterateur
{
    Brin *first, *current;
    int niv;

    SommetIterateur(Brin *b, int n)
    {
        first = current = b;
        niv = n;
    }

    void next()
    {
        current = current->alpha();
    }

    bool end()
    {
        return current==first;
    }
};

class Sommet
{
    SommetIterateur s;

    Sommet(Brin *b, int n) : s(b,n)
    {
    }

    void visite_voisins()
    {
        while(!s.end())
        {
            faire quelquechose avec
                Sommet(s.current->alpha0(s.n), s.n);
            s.next();
        }
    }
}

```

**Complexité en espace**

Nous comparons ici le coût mémoire d'une 2-carte multirésolution et celui d'une forêt de quadrees triangulaires. Comme aucune formulation générale ne peut être donnée dans le cas adaptatif et que la subdivision régulière est le cas le pire pour les besoins en mémoire, nous effectuons cette comparaison dans le cas régulier.

Soit  $|B|$  le nombre total de brins d'une 2-carte multirésolution.  $|B|$  est égal au nombre de brins nécessaires à la description de l'objet au niveau de résolution maximum. Soit  $b_0$  le nombre de brins décrivant le niveau de résolution 0, et soit  $k$  le niveau de résolution maximum. Comme dans les schémas de subdivision primaires présentés ci-dessus le nombre

de brins est multiplié par 4 à chaque pas de subdivision, on a :

$$|B| = b_0 \cdot 4^k \quad (1)$$

Pour calculer le coût total de l'information topologique, il nous faut compter le nombre de pointeurs stockés par tous les brins.

Pour la relation  $\alpha_0$ , il faut additionner les tailles des tableaux de pointeurs contenus dans chaque brin. La taille de ce tableau est fonction du niveau d'insertion du brin. On note que 3/4 des brins ont été insérés au niveau de résolution maximum et n'ont donc qu'une seule liaison  $\alpha_0$ ; 3/4 des autres brins, i.e. 3/16, ont deux liaisons  $\alpha_0$ ; ... Formellement, pour  $i$  entre 1 et  $k$ , il y a  $|B| \cdot \frac{3}{4^i}$  brins dont le tableau a  $i$  éléments. Les brins décrivant le niveau 0 ont  $k+1$  éléments dans leur tableau.

Le nombre total d'éléments dans les tableaux de liaisons  $\alpha_0$  de tous les brins est donc, après remplacement de  $|B|$  en utilisant l'équation (1) :

$$b_0 \cdot (k+1) + b_0 \cdot 3 \cdot \sum_{i=1}^k i \cdot 4^{k-i}$$

Pour la relation  $\alpha_1$ , les choses sont beaucoup plus simples. En effet, chaque brin ne dispose que d'une liaison  $\alpha_1$ . En d'autres termes, il y a en tout  $|B|$  ou  $b_0 \cdot 4^k$  pointeurs stockés.

L'information géométrique est attachée aux brins par un tableau de pointeurs vers des points 3D. Dans le cas où on utilise un schéma de subdivision approximant, le plongement est paramétré par le niveau de résolution (celui-ci étant modifié à chaque niveau). La taille du tableau de pointeurs de points 3D a exactement la même taille que celui de liaisons  $\alpha_0$ .

Le nombre total de pointeurs stockés par tous les brins est donc :

$$b_0 \cdot 4^k + 2 \cdot \left( b_0 \cdot (k+1) + b_0 \cdot 3 \cdot \sum_{i=1}^k i \cdot 4^{k-i} \right) \quad (2)$$

La somme de l'expression (2) peut être identifiée à la série entière :  $\sum_{n \geq 0} n \cdot x^n = \frac{x}{(1-x)^2}$ , définie pour  $|x| < 1$ . Après avoir sorti les termes constants de la somme, on substitue ici  $n$  par  $i$  et  $x$  par  $\frac{1}{4}$ . La somme peut alors être exprimée comme suit (en négligeant les termes de la série tels que  $i > k$ ) :

$$\sum_{i=1}^k i \cdot 4^{k-i} \simeq 4^k \cdot \frac{\frac{1}{4}}{(1-\frac{1}{4})^2}$$

L'expression (2) se simplifie alors en (en omettant les termes négligeables) :

$$\frac{33}{9} \cdot b_0 \cdot 4^k \quad (3)$$

Les quadrees stockent cinq pointeurs par nœud pour l'information topologique : quatre vers les fils et un vers le

père. Les racines des quadrees stockent sept pointeurs : quatre vers les fils, et trois pour les adjacences entre les racines. Pour l'information géométrique, trois pointeurs vers des points 3D sont stockés dans chaque nœud, ce qui fait donc un total de dix pointeurs par racine, et huit pointeurs pour chacun des autres nœuds. Soit  $f_0$  le nombre de faces du maillage de niveau 0, et  $k$  le niveau de résolution maximum. Comme le nombre de faces est multiplié par 4 à chaque pas de subdivision, le nombre total de pointeurs stockés est :

$$10 * f_0 + 8 * f_0 * \sum_{i=1}^k 4^i \quad (4)$$

La somme de l'expression (4) peut être identifiée à la forme suivante :  $\sum_{i=0}^n x^i = \frac{x^{n+1} - 1}{x - 1}$ . La somme peut alors être exprimée comme suit ( $i$  commençant ici à 1, on retranche le terme tel que  $i = 0$ ) :

$$\sum_{i=1}^k 4^i = \frac{4^{k+1} - 1}{3} - 1$$

L'expression (4) se simplifie alors en (en omettant les termes négligeables) :

$$\frac{32}{3} \cdot f_0 \cdot 4^k \quad (5)$$

On peut maintenant calculer le ratio entre le coût d'une 2-carte multirésolution (3) et celui d'une forêt de quadrees (5). Dans un maillage triangulaire, il y a trois brins par faces. On a alors  $f_0 = \frac{b_0}{3}$ . On obtient donc le ratio suivant :

$$\frac{\frac{33}{9} \cdot b_0 \cdot 4^k}{\frac{32}{9} \cdot b_0 \cdot 4^k} = \frac{33}{32} \quad (6)$$

On voit que notre modèle ne nécessite environ que 3% d'espace mémoire de plus qu'une forêt de quadrees. Si on prend en compte le coût de stockage des points 3D, qui est équivalent pour les deux structures, ce ratio est encore plus faible.

Utiliser un schéma de subdivision interpolant rend ce ratio plus avantageux pour les 2-cartes multirésolution que pour les quadrees. En effet, comme les sommets sont situés sur la surface limite dès leur insertion, l'information de plongement n'a plus besoin d'être paramétrée par le niveau de résolution. Les tableaux de pointeurs de points 3D contenus dans chaque brin se réduisent donc à un seul pointeur. Dans ce cas, le ratio entre 2-cartes multirésolution et quadrees est de  $\frac{30}{32}$ . Notre modèle nécessite ici environ 6% d'espace mémoire en moins.

## 6. Conclusion

Les hypercartes multirésolution définies dans cet article offrent un nouveau cadre général pour la représentation de maillages surfaciques multirésolution. Ce cadre est défini

comme une extension des hypercartes, notion générale sur laquelle reposent les modèles basés sur les cartes combinatoires.

Les hypercartes multirésolution ont été ici spécialisées pour définir les 2-cartes multirésolution et leur représentation duale. Ces dernières ont été ici appliquées à la représentation de surfaces de subdivision multirésolution. Dans ce cadre, elles apportent un certain nombre d'avantages par rapport aux structures basées sur des quadrees usuellement utilisées. Les cartes multirésolution permettent, au sein d'un même modèle, la représentation de surfaces générées par un grand nombre de schémas de subdivision. La capacité à représenter des maillages polygonaux est un avantage supplémentaire dans le cas adaptatif, où la coexistence de différents niveaux de résolution produit des faces non-triangulaires ou non-carrées, qui ne sont pas supportées par les structures classiques et y engendrent alors des trous topologiques. Les requêtes d'adjacence sont effectuées plus efficacement – celles-ci étant exécutées en temps constant quel que soit le niveau de résolution. De plus, les besoins en mémoire ne sont pas plus importants.

Le cadre défini ici amène de nombreuses perspectives. Tout d'abord, toujours dans le cadre des surfaces de subdivision, la généralité des 2-cartes multirésolution permet l'utilisation au sein d'un même objet d'algorithmes de subdivision différents. Ceci peut être intéressant dans la mesure où chacun de ces schémas converge vers des surfaces aux propriétés différentes.

Le modèle des 2-cartes multirésolution peut également être appliqué à la représentation de surfaces multirésolution non générées par des algorithmes de subdivision, mais par des opérateurs d'éclatement et de fusion de sommets [KCVS98]. Ceci permet, dans la méthode "fin vers grossier", de se passer de la connectivité de subdivision pour le maillage de départ.

Enfin, le traitement que nous avons ici appliqué aux hypercartes peut sans effort être étendu aux hypercartes de dimension  $n$ . Ceci implique que tous les modèles ordonnés définis comme des restrictions des hypercartes de dimension  $n$  peuvent décrire une version multirésolution des objets appartenant à leur domaine de représentation. Par exemple, l'extension multirésolution des 3-cartes, modèle décrivant des subdivisions de volumes, permet de décrire des volumes multirésolution. Ceux-ci pourraient être utilisés par exemple en visualisation volumique ou dans le cadre de simulations.

## Références

- [BD94] BERTRAND Y., DUFOURD J.: Algebraic specification of a 3D-modeler based on hypermaps. *CVGIP : Graphical Models and Image Processing. Vol. 56*, Num. 1 (1994), 29–60.
- [BKZ01] BIERMANN H., KRISTJANSSON D., ZORIN D.:

- Approximate boolean operations on free-form solids. In *Proceedings of SIGGRAPH'01* (2001), pp. 185–194.
- [BLZ00] BIERMANN H., LEVIN A., ZORIN D.: Piecewise smooth subdivision surfaces with normal control. In *Proceedings of SIGGRAPH'00* (2000), pp. 185–194.
- [BMBZ02] BIERMANN H., MARTIN I., BERNARDINI F., ZORIN D.: Cut-and-paste editing of multiresolution surfaces. *ACM Trans. Graph.* Vol. *Special issue for SIGGRAPH conference 21*, Num. 3 (2002), 312–321.
- [CC78] CATMULL E., CLARK J.: Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*. Vol. 10, Num. 6 (1978), 350–355.
- [CD99] CAZIER D., DUFOURD J.: A formal specification of geometric refinements. *Visual Computer*. Vol. 15 (1999), 279–301.
- [DKP04] DEFLORIANI L., KOBBELT L., PUPPO E.: A survey on data structures for level-of-detail models. *Advances in Multiresolution for Geometric Modelling, Series in Mathematics and Visualization* (2004), 49–74.
- [DLG90] DYN N., LEVIN D., GREGORY J. A.: A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. Graph.* Vol. 9, Num. 2 (1990), 160–169.
- [Doo78] DOO D.: A subdivision algorithm for smoothing down irregularly shaped polygons. In *Proceedings on Interactive Techniques in Computer Aided Design* (1978), pp. 157–165.
- [DS78] DOO D., SABIN M.: Analysis of the behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*. Vol. 10, Num. 6 (1978), 356–360.
- [Duf91] DUFOURD J.: Formal specification of topological subdivisions using hypermaps. *Computer-Aided Design*. Vol. 23, Num. 2 (1991), 99–116.
- [Duf97] DUFOURD J.: Algebras and formal specifications in geometric modeling. *The Visual Computer* (1997).
- [Edm60] EDMONDS J.: A combinatorial representation for polyhedral surfaces. In *Notices Amer. Math. Soc.* (1960), vol. 7.
- [KCVS98] KOBBELT L., CAMPAGNA S., VORSATZ J., SEIDEL H.-P.: Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of SIGGRAPH'98* (1998), pp. 105–114.
- [Lie89] LIENHARDT P.: Subdivision of n-dimensional spaces and n-dimensional generalized maps. In *5th ACM Conf. on Computational Geometry* (Saarbrücken, Germany, 1989), pp. 228–236.
- [Lie91] LIENHARDT P.: Topological models for boundary representation: a comparison with n-dimensional generalized maps. *Computer-Aided Design*. Vol. 23, Num. 1 (1991), 59–82.
- [Loo87] LOOP C.: *Smooth Subdivision Surfaces Based on Triangles*. Master's thesis, University of Utah, 1987.
- [LS00] LEE M., SAMET H.: Navigating through triangle meshes implemented as linear quadtrees. *ACM Trans. Graph.* Vol. 19, Num. 2 (2000), 79–121.
- [PR97] PETERS J., REIF U.: The simplest subdivision scheme for smoothing polyhedra. *ACM Trans. Graph.* Vol. 16, Num. 4 (1997).
- [Sch92] SCHRACK G.: Finding neighbors of equal size in linear quadtrees and octrees in constant time. *CVGIP: Image Understanding*. Vol. 55, Num. 3 (1992), 221–230.
- [Tut84] TUTTE W.: Graph theory. In *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1984, ch. 21.
- [Vin83] VINCE A.: Combinatorial maps. *Journal of Combinatorial Theory* (1983), 1–21.
- [Zor05] ZORIN D.: Modeling with multiresolution subdivision surfaces. *Tutorial Eurographics* (2005).
- [ZSS96] ZORIN D., SCHRÖDER P., SWELDENS W.: Interpolating subdivision for meshes with arbitrary topology. In *Proceedings of SIGGRAPH'96* (1996), pp. 189–192.
- [ZSS97] ZORIN D., SCHRÖDER P., SWELDENS W.: Interactive multiresolution mesh editing. In *Proceedings of SIGGRAPH'97* (1997), pp. 259–268.