

**THÈSE**

présentée à

**l'Université Louis Pasteur, Département d'Informatique  
Laboratoire des Sciences de l'Image, de l'Informatique et de la Télédétection,  
UMR CNRS-ULP 7005**

par

**Pierre KRAEMER**

pour l'obtention du grade de

**Docteur de l'Université Louis Pasteur de Strasbourg  
Mention Sciences  
Spécialité Informatique**

---

**Modèles topologiques pour la multirésolution**

---

Soutenue le **12 novembre 2008** devant la commission d'examen composée de :

**M. Bruno LÉVY** ..... *Rapporteur externe*  
Directeur de recherches à l'INRIA Lorraine

**M. Pascal LIENHARDT** ..... *Rapporteur externe*  
Professeur à l'Université de Poitiers

**M. Leif KOBELT** ..... *Examineur*  
Professeur à l'Université d'Aix-La-Chapelle, Allemagne

**M. Jean-François DUFOURD** ..... *Rapporteur interne*  
Professeur à l'Université Louis Pasteur de Strasbourg

**Mme Dominique BECHMANN** ..... *Directrice de thèse*  
Professeure à l'Université Louis Pasteur de Strasbourg

**M. David CAZIER** ..... *Examineur*  
Maître de conférences à l'Université Louis Pasteur de Strasbourg



# TABLE DES MATIÈRES

---

<b>Introduction</b>	<b>1</b>
1 Contexte et objectif . . . . .	1
2 Contributions . . . . .	3
2.1 Surfaces de subdivision . . . . .	4
2.2 Maillages progressifs . . . . .	5
3 Plan du mémoire . . . . .	5
<b>I Modèle topologique multirésolution</b>	<b>7</b>
<b>1 Modèles de représentation multirésolution</b>	<b>9</b>
1.1 Introduction . . . . .	9
1.2 Modèles topologiques standards . . . . .	10
1.2.1 Généralités . . . . .	11
1.2.2 Cartes combinatoires . . . . .	17
1.3 Modèles multirésolution . . . . .	32
1.3.1 Généralités . . . . .	32
1.3.2 Maillages progressifs . . . . .	33
1.3.3 Multi-Triangulation . . . . .	37
1.3.4 Quadrees . . . . .	39
1.3.5 Pyramides de cartes . . . . .	42
1.4 Conclusion . . . . .	44
<b>2 Extension multirésolution des cartes combinatoires</b>	<b>47</b>
2.1 Introduction . . . . .	47

2.2	Extension multirésolution . . . . .	48
2.2.1	Ensembles de brins . . . . .	48
2.2.2	Relations indexées . . . . .	49
2.2.3	Hypercarte multirésolution . . . . .	51
2.2.4	2-carte multirésolution . . . . .	52
2.3	Opérateurs de manipulation . . . . .	53
2.3.1	Opérateurs de base . . . . .	53
2.3.2	Opérateurs de haut niveau . . . . .	55
2.4	Mise-en-œuvre pratique . . . . .	62
2.4.1	Implantation des 2-cartes . . . . .	63
2.4.2	Implantation des 2-cartes multirésolution . . . . .	66
<b>II Applications géométriques</b>		<b>71</b>
<b>3</b>	<b>Surfaces de subdivision multirésolution</b>	<b>73</b>
3.1	Introduction . . . . .	74
3.1.1	Surfaces de subdivision . . . . .	74
3.1.2	L'édition multirésolution . . . . .	78
3.1.3	Obtention d'une surface de subdivision multirésolution . . . . .	82
3.1.4	Support du maillage multirésolution . . . . .	84
3.2	Génération des niveaux . . . . .	85
3.2.1	Génération régulière . . . . .	85
3.2.2	Génération adaptative . . . . .	99
3.2.3	Simplification . . . . .	106
3.2.4	Critère de subdivision . . . . .	108
3.2.5	Mise à jour du maillage . . . . .	111
3.2.6	Niveaux de subdivision des cellules . . . . .	114
3.3	Accès aux niveaux . . . . .	120
3.3.1	Accès régulier . . . . .	120
3.3.2	Accès adaptatif . . . . .	121
3.3.3	Affichage d'une 2-carte adaptative . . . . .	125
3.4	Schémas de subdivision originaux . . . . .	129
3.4.1	Le schéma Quad/Triangle . . . . .	129
3.4.2	Le schéma $\sqrt{3}$ . . . . .	132
3.5	Comparaison . . . . .	140
3.5.1	Structure . . . . .	142
3.5.2	Complexité en temps . . . . .	144
3.5.3	Complexité en espace . . . . .	147
3.6	Conclusion . . . . .	150
<b>4</b>	<b>Maillages progressifs</b>	<b>153</b>
4.1	Introduction . . . . .	153

---

4.2	Applications des maillages progressifs multirésolution . . . . .	155
4.2.1	Filtrage . . . . .	155
4.2.2	Compression . . . . .	158
4.3	Supports topologiques classiques des maillages progressifs . . . . .	161
4.4	Utilisation des cartes multirésolution . . . . .	162
4.4.1	Gestion des ensembles de brins . . . . .	162
4.4.2	Gestion des coutures . . . . .	165
4.4.3	Vérification des contraintes topologiques . . . . .	167
4.4.4	Contractions groupées . . . . .	171
4.4.5	Economies de mémoire . . . . .	173
4.4.6	Application aux algorithmes de traitement de la géométrie	175
4.4.7	Accès au maillage . . . . .	180
4.4.8	Comparaison du coût mémoire . . . . .	182
4.5	Conclusion . . . . .	186
<b>Conclusion</b>		<b>189</b>
1	Conclusion . . . . .	189
2	Perspectives . . . . .	193
<b>Liste des figures</b>		<b>195</b>
<b>Bibliographie</b>		<b>200</b>



# INTRODUCTION

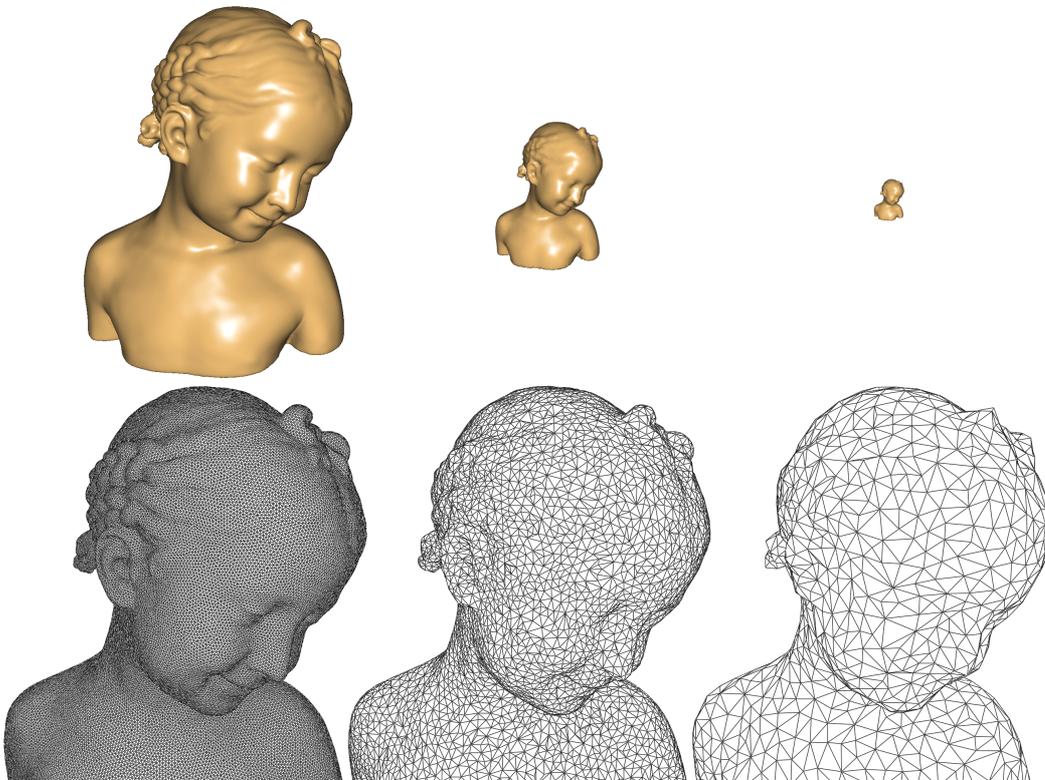
---

## 1 Contexte et objectif

En modélisation géométrique, l'amélioration de la représentation et de la manipulation des objets géométriques est un enjeu dont l'objectif est de mettre en avant différentes propriétés de ces objets, et de rendre possibles différentes méthodes d'interaction et d'édition de ces objets. L'un des modes de représentation usuels consiste à décomposer les objets en cellules de différentes dimensions (sommets, arêtes, faces, ...) partageant entre elles des relations d'adjacence et d'incidence.

Dans le cadre d'applications de modélisation ou de traitement de la géométrie sur de tels objets, où la souplesse et l'efficacité de la représentation sous-jacente sont primordiales, le modèle des cartes combinatoires [Edm60, Vin83, Lie89, Lie91] constitue une référence. Le choix de ce modèle, dit à base topologique, est d'autant plus pertinent face aux structures ad hoc qui existent dans la littérature que son socle mathématique permet aussi bien la garantie de certaines propriétés sur les objets représentés qu'une définition consistante en dimension arbitraire.

Dans le cadre de la modélisation *multirésolution* [FIQ02, DFS04], la décomposition des objets en cellules est effectuée à des niveaux de détail ou de raffinement



**Figure 1** - Exemple d'application d'une représentation multirésolution surfacique : plus l'objet est distant, plus le maillage utilisé pour l'affichage est grossier

différents. Cette représentation multi-échelles a de nombreuses applications en informatique graphique. Suivant les cas, elle sert à accélérer des traitements existants tels que l'affichage, comme illustré dans la figure 1 où plus l'objet à afficher est éloigné du point de vue de l'utilisateur, plus le maillage utilisé pour l'affichage est simplifié, accélérant d'autant l'exécution de la procédure d'affichage.

Dans d'autres applications, cette représentation constitue le support d'outils puissants de traitement et de manipulation de la géométrie, tels que l'édition multirésolution de surfaces de subdivision [Zor05] qui permet de déformer une surface à des échelles différentes et ce de manière naturelle et cohérente relativement aux détails de l'objet. La compression de maillages ou encore la suppression de bruit [GSS99] sont d'autres applications qui s'appuient sur une telle représentation. Afin de pouvoir s'exécuter de manière optimale et pour un large éventail de types d'objets différents, tous ces algorithmes de traitement de la géométrie ont besoin d'un support topologique efficace et général pour la représentation des maillages sous-jacents.

Nous nous intéressons dans cette thèse aux modèles et structures de données utilisées pour la représentation de maillages multirésolution. Comme nous le verrons, les modèles existants, généralement basés sur la représentation conjointe d'un maillage courant encodé dans une structure standard et d'un ensemble de nœuds de modification, présentent de nombreux défauts. Ceux-ci sont principalement dus à l'absence de représentation de l'ensemble des cellules de la subdivision de l'objet modélisé au sein d'une structure topologique complète.

## 2 Contributions

Compte tenu des avantages que présente le modèle des cartes combinatoires, ce dernier nous est apparu comme un point de départ de choix pour l'élaboration d'un nouveau modèle de représentation multirésolution capable de répondre aux contraintes imposées par les applications prenant appui sur de telles représentations et de proposer des solutions aux défauts que présentent les structures existantes.

Nous proposons dans cette thèse une extension multirésolution des cartes combinatoires permettant la représentation multi-échelles d'objets géométriques. Une carte multirésolution peut être vue comme une hiérarchie de cartes combinatoires imbriquées les unes dans les autres. Chaque niveau de résolution de l'objet représenté est ainsi défini par une carte combinatoire. On bénéficie donc ici à chaque niveau de toutes les propriétés intéressantes des cartes, à savoir la généralité de la représentation, l'efficacité des algorithmes de parcours des maillages représentés et la légèreté des structures de données dérivées du modèle. En effet, le modèle est défini en dimension arbitraire et peut représenter des partitions cellulaires quelconque, quel que soit le niveau de résolution considéré. A tout niveau, les requêtes de voisinage sont exécutées de manière optimale. Enfin, le coût mémoire de la structure de données dérivée du modèle est comparable à celui des structures multirésolution classiquement utilisées.

Cette structure de données a été implantée au cœur d'un modèleur géométrique fondé sur une implantation générique des cartes combinatoires développée dans notre équipe. En dimension 2, c'est-à-dire dans le cadre de la représentation de maillages surfaciques, nous avons choisi deux cadres applicatifs différents afin d'illustrer les avantages de notre modèle par rapport à l'existant : les surfaces de subdivision multirésolution, et les maillages progressifs.

## 2.1 Surfaces de subdivision

Les surfaces de subdivision multirésolution ont été notre sujet d'étude principal. Une surface de subdivision est définie comme la limite d'une séquence de maillages raffinés successivement. Un schéma de subdivision définit les règles selon lesquelles ce raffinement est appliqué à partir d'un maillage de départ donné. On peut distinguer ici deux étapes : d'abord, la topologie du maillage est raffinée, puis de nouvelles informations géométriques sont associées au maillage obtenu. Celles-ci sont calculées en appliquant des masques locaux sur le maillage d'origine. Il existe un certain nombre de schémas qui, opérant sur des maillages constitués de types de cellules différents, génèrent des surfaces aux propriétés géométriques différentes.

Dans le cadre de l'édition multirésolution de surfaces de subdivision, des coefficients de détail sont introduits entre les différents maillages correspondant aux étapes de la subdivision, que l'on appelle alors niveaux de résolution. En éditant le maillage à un niveau donné, on applique une déformation plus ou moins globale à la surface, en conservant les détails de l'objet de manière cohérente.

Par rapport aux structures basées sur des arbres de faces habituellement utilisées dans ce cadre, les cartes multirésolution apportent ici de nombreux avantages. Là où les structures classiques doivent être déclinées en plusieurs versions spécifiques, la généralité de notre modèle autorise l'utilisation d'un même noyau avec des algorithmes de subdivision travaillant sur des cellules de types différents. Ainsi, nous avons implanté efficacement les algorithmes de Loop [Loo87] (quadrisection de triangles), Catmull-Clark [CC78] (quadrisection de quadrilatères), Doo-Sabin [DS78] (subdivision duale) et le schéma  $\sqrt{3}$  [Kob00] qui opère une trisection de triangles suivie de basculements d'arêtes. Plusieurs types de subdivision peuvent même être utilisés simultanément au sein d'un même objet, tel qu'illustré avec l'implantation du schéma quad-triangle [SL03] qui conserve à la fois des triangles et des quadrilatères dans le maillage.

Dans une carte multirésolution, la subdivision adaptative, c'est-à-dire le raffinement à une profondeur variable suivant les zones d'intérêt de l'objet, ne compromet pas la consistance topologique du maillage. Dans les structures à base d'arbres, une telle subdivision engendre des trous topologiques au niveau des frontières entre les zones ayant des profondeurs différentes. Cela résulte du fait que notre modèle ne manipule pas directement une seule entité topologique (en l'occurrence la face en ce qui concerne les modèles arborescents), mais opère la subdivision de manière cohérente avec toutes les cellules de la subdivision (sommets, arêtes, faces). Enfin, l'efficacité des algorithmes de calcul de l'information géométrique est améliorée, les informations d'adjacence et d'incidence

entre les cellules étant disponibles de manière optimale à tous les niveaux de résolution du maillage.

## 2.2 Maillages progressifs

Le second axe d'expérimentation du modèle des cartes multirésolution a été développé dans le cadre de la représentation de maillages progressifs. Un maillage progressif consiste en une hiérarchie de maillages construite par simplifications successives d'un maillage triangulaire initial. Chaque niveau plus grossier est obtenu en appliquant une ou plusieurs opérations de contraction d'arête, dans laquelle deux sommets adjacents sont fusionnés en un seul.

Dans ce cadre applicatif, notre modèle se révèle également compétitif par rapport aux structures classiquement utilisées. Ces dernières proposent en général de stocker le maillage courant dans une structure standard, conjointement à un ensemble de nœuds contenant l'information nécessaire pour inverser une contraction d'arête (c'est-à-dire appliquer un éclatement de sommet). Ces derniers peuvent être organisés dans un arbre de dépendances afin d'accéder au maillage à des profondeurs variables.

Certains algorithmes de traitement de la géométrie ont besoin de faire des calculs locaux à un niveau de résolution donné, puis de faire circuler de l'information au travers des différents niveaux du maillage. Dans une carte multirésolution, contrairement aux structures à base de nœuds de reconstruction, les informations d'adjacence entre cellules sont immédiatement disponibles à tous les niveaux de résolution du maillage sans avoir besoin de les reconstruire au fur et à mesure. Cela permet une implantation optimale des algorithmes de traitement de la géométrie qui tirent fortement parti de cette information. Cette amélioration de l'efficacité n'est pas obtenue au détriment du coût mémoire, ce dernier restant comparable à celui des structures existantes dans la plupart des cas.

## 3 Plan du mémoire

La suite du manuscrit est organisée comme suit.

Le premier chapitre est consacré à la présentation des modèles et structures de données standards utilisés pour la représentation de maillages "monorésolution". Nous nous intéressons plus particulièrement aux modèles à base

topologique issus du cadre des cartes combinatoires. Nous faisons ensuite un tour d'horizon des structures de données existantes pour la représentation et la manipulation de maillages multirésolution.

Le deuxième chapitre présente notre extension multirésolution des cartes combinatoires. En partant d'une définition très générale, nous nous focalisons ensuite sur leur spécialisation en dimension 2 topologique, c'est-à-dire au cadre de la représentation de maillages surfaciques. Nous proposons alors un ensemble d'opérateurs topologiques adaptés à la manipulation des surfaces multirésolution.

Dans le troisième chapitre, nous présentons et étudions les apports de l'application de notre modèle au cadre de la représentation et de la manipulation de surfaces de subdivision multirésolution. Nous montrons comment on peut l'utiliser dans le cadre de différents schémas de subdivision et analysons les performances et les coûts en mémoire par rapport à l'existant.

Dans le quatrième chapitre, c'est l'application des cartes multirésolution au cadre des maillages progressifs multirésolution qui est présentée. Ici aussi, nous décrivons la manière dont on construit une telle hiérarchie au sein de notre modèle, puis étudions les performances et comparons les coûts en mémoire par rapport aux structures existantes.

Enfin, nous résumons dans un chapitre de conclusion les apports et limitations de notre modèle. Nous proposons également des opportunités de travaux futurs concernant aussi bien des cadres applicatifs au sein desquels les cartes multirésolution pourraient présenter des avantages, que des évolutions du modèle en lui-même.

PREMIÈRE PARTIE

---

MODÈLE TOPOLOGIQUE  
MULTIRÉSOLUTION



---

# MODÈLES DE REPRÉSENTATION MULTIRÉSOLUTION

---

## Sommaire

---

<b>1.1</b>	<b>Introduction</b>	<b>9</b>
<b>1.2</b>	<b>Modèles topologiques standards</b>	<b>10</b>
1.2.1	Généralités	11
1.2.2	Cartes combinatoires	17
<b>1.3</b>	<b>Modèles multirésolution</b>	<b>32</b>
1.3.1	Généralités	32
1.3.2	Maillages progressifs	33
1.3.3	Multi-Triangulation	37
1.3.4	Quadtrees	39
1.3.5	Pyramides de cartes	42
<b>1.4</b>	<b>Conclusion</b>	<b>44</b>

---

Ce chapitre débute par une rapide introduction à la modélisation géométrique en général. Nous présentons ensuite un certain nombre de modèles et structures de données classiquement utilisés en modélisation géométrique, dont le modèle des cartes combinatoires, point de départ de notre travail. Nous nous intéressons ensuite aux modèles et structures de données existants pour la représentation de maillages multirésolution.

## 1.1 Introduction

En modélisation géométrique, l'objectif est de définir le moyen de *représenter* et de *manipuler* des objets géométriques grâce à l'outil informatique. Ce

dernier étant un outil éminemment discret et à capacité finie, il n'est pas envisageable de stocker l'ensemble infini des points qui constituent un objet dans un espace donné. Cet état de fait a donné naissance à deux principales familles de méthodes qui répondent à cette problématique.

La première propose de représenter les objets de façon *continue*. Les objets sont alors définis par des équations ou des fonctions dont les coefficients peuvent être stockés. Par exemple, un cercle en 2D peut être représenté par la position de son centre et la valeur de son rayon. Ces deux informations à elles seules permettent de capter l'infinité de points qui composent cet objet. Afin d'être visualisé il faut alors expliciter cet objet en évaluant l'équation qui le représente en un certain nombre de points.

La seconde famille de méthodes s'est développée en se fondant sur une structuration de l'ensemble de points à représenter. Cette structuration consiste en une partition de l'ensemble de points en *cellules* de différentes dimensions liées par des relations d'incidence et d'adjacence. Par exemple un objet en 2D sera discrétisé en un ensemble de sommets, d'arêtes et de faces. Etant en nombre fini, ces cellules et leurs relations peuvent alors être stockées explicitement. Cette représentation est une modélisation combinatoire de la topologie de l'ensemble de points définissant l'objet considéré. On parle alors de *modélisation à base topologique de subdivisions d'espaces géométriques*.

C'est cette dernière approche qui va nous intéresser tout au long de ce mémoire de thèse. De nombreux modèles aux propriétés et aux domaines de représentation différents ont déjà été proposés dans ce cadre. Avant de nous intéresser aux problématiques liées à la multirésolution, nous allons faire un tour d'horizon rapide des modèles à base topologique et plus précisément des modèles combinatoires.

## 1.2 Modèles topologiques standards

Nous débutons cette section par un rappel de quelques généralités sur les objets que l'on cherche à représenter. Un rapide historique des modèles existants est ensuite donné. Une large part de cette section est ensuite dévolue à la présentation plus détaillée des cartes combinatoires, ce modèle étant le fondement de notre travail que nous présentons dans le chapitre 2.

### 1.2.1 Généralités

On cherche à représenter des objets discrétisés en cellules partageant des relations d'incidence et d'adjacence. Un objet de dimension  $n$  est composé d'un ensemble de  $k$ -cellules où  $k \in [0, n]$  représente la dimension de la cellule. Ainsi, une 0-cellule est un sommet, une 1-cellule est une arête, une 2-cellule est une face, une 3-cellule est un volume, etc. . . . Le bord d'une  $k$ -cellule est composé d'un ensemble de  $(k - 1)$ -cellules, avec  $k \in [1, n]$  (une 0-cellule n'ayant pas de bord et une  $n$ -cellule ne faisant partie du bord d'aucune cellule). Une cellule  $c_1$  est dite incidente à une cellule  $c_2$  si  $c_2$  appartient au bord de  $c_1$ . De manière symétrique on dit alors aussi que  $c_2$  est incidente à  $c_1$ .

Si l'on considère ces informations d'incidence entre cellules en faisant abstraction de toute information géométrique liée à l'objet, on dispose alors d'une représentation combinatoire de la topologie de l'objet considéré. Afin de retrouver une représentation complète de l'objet, on lui redonne une existence géométrique dans un espace donné. On complète alors sa représentation topologique par un modèle de *plongement* qui associe un objet géométrique à chaque cellule de la subdivision. La dimension de l'espace de plongement n'est pas liée à la dimension topologique de l'objet. Ainsi, très classiquement, un objet surfacique de dimension 2 topologique, peut être plongé dans un espace à 3 dimensions. On associe alors par exemple un point à un sommet, un segment de droite ou une courbe à une arête et un polygone ou un élément de surface à une face. Les informations de plongement peuvent également consister en tout autre type d'information que l'on souhaite attacher aux cellules d'une subdivision. Dans le cadre de certaines simulations de phénomènes physiques, on peut associer des propriétés physiques aux cellules comme par exemple une masse à chaque sommet et une rigidité de ressort à chaque arête.

Cette séparation franche entre l'information topologique et le plongement qui lui est associé est une caractéristique fondamentale des modèles à base topologique.

### Graphes d'incidence

Les relations d'incidence entre les cellules de la partition cellulaire d'un objet géométrique peuvent être représentées par un graphe orienté, appelé *graphe d'incidence*. Les nœuds de ce graphe sont les cellules de la subdivision et les arcs orientés relient une  $k$ -cellule à une  $(k - 1)$ -cellule qui lui est incidente.

La figure 1.1-A illustre le graphe d'incidence associé à la partition cellulaire

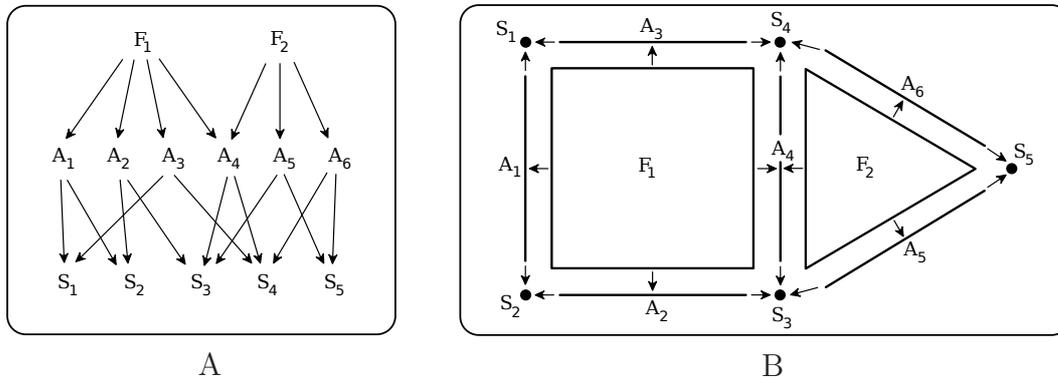


Figure 1.1 - Graphe d'incidence

d'un objet de dimension 2 composée de deux faces, six arêtes et cinq sommets. La figure 1.1-B donne une représentation graphique à cette partition cellulaire et reproduit également les arcs du graphe d'incidence. L'information topologique est entièrement contenue dans le graphe de la figure 1.1-A et cette représentation graphique n'est qu'une des réalisations géométriques possibles, un plongement particulier et arbitraire de cette subdivision. Néanmoins, cette représentation graphique est plus commode car elle donne une intuition de ce à quoi pourrait ressembler un objet géométrique ayant cette topologie. Tout au long de cette thèse, on donnera souvent une telle représentation graphique à l'information topologique, tout en gardant à l'esprit la séparation fondamentale entre topologie et plongement.

Certaines représentations proposent d'encoder directement ce graphe d'incidence. Cette approche qui semble de prime abord plutôt intuitive a néanmoins de nombreux inconvénients. Etant donnée une  $k$ -cellule, le parcours d'un voisinage de dimension supérieure à  $k$  (par exemple le parcours de l'ensemble des arêtes ou encore de l'ensemble des faces incidentes à un sommet) nécessite une recherche impliquant la totalité de la structure. Ce défaut d'efficacité peut être comblé en rajoutant des liens partiels liant des  $k$ -cellules à des  $k + 1$ -cellules. Mais cela ne règle pas la complexité de la gestion des  $n + 1$  structures de données différentes nécessaires à la représentation de tous les types de cellules composant un objet de dimension  $n$ . Des opérateurs différents doivent alors être écrits pour manipuler chacune de ces structures et les opérations de mise-à-jour de la subdivision deviennent d'une complexité rédhibitoire afin de maintenir la cohérence du modèle.

Le domaine de représentation d'un graphe d'incidence est vaste. On se place ici dans le cadre des objets ne présentant pas de multi-incidence (par exemple, on exclut les cas où une arête est incidente à un même sommet à ses deux extrémités). Tout objet de dimension  $n$  pouvant être partitionné en cellules peut

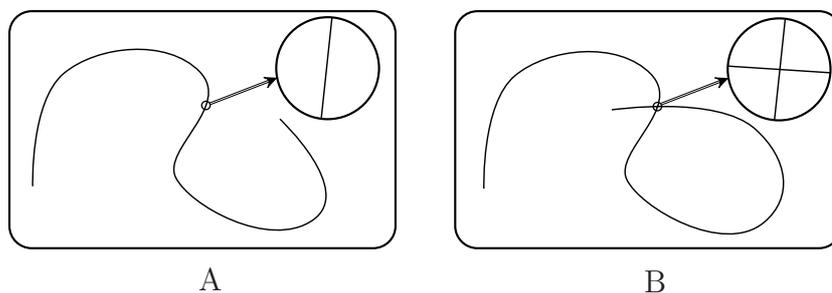


Figure 1.2 - Exemples d'objets variété et non-variété de dimension 1

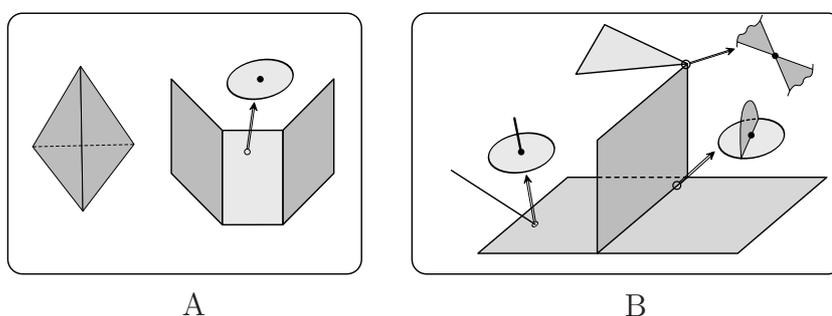
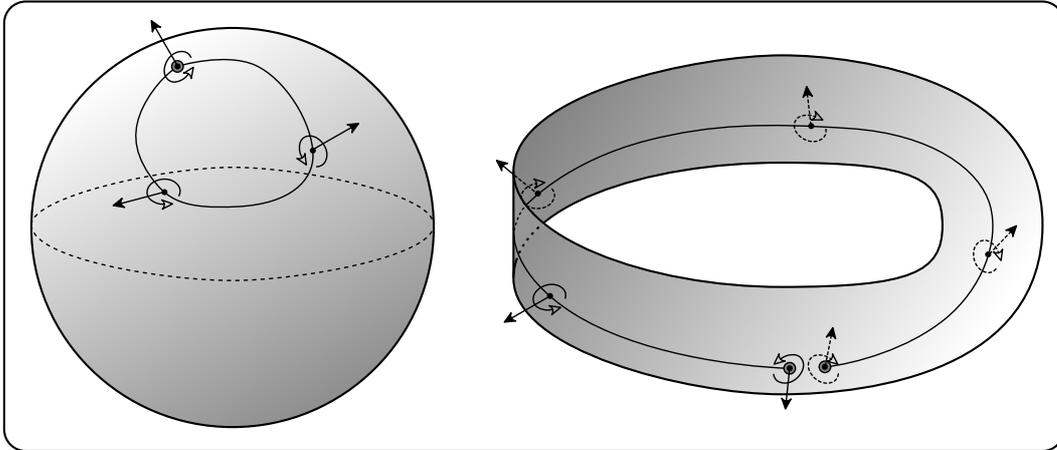


Figure 1.3 - Exemples d'objets variété et non-variété de dimension 2

être représenté par un tel graphe. Comme on vient de le voir, cette généralité se paie par une complexité élevée des opérations manipulant cette structure. Si l'on restreint le domaine de représentation, des modèles plus spécialisés et plus efficaces peuvent alors être définis. Quelques définitions sont ici nécessaires afin de définir les domaines de représentation auxquels nous nous intéressons. Nous donnons ici une approche intuitive des notions de variété et d'orientabilité.

On nomme *variété* de dimension  $n$ , un objet dont tout point admet un voisinage homéomorphe à l'espace euclidien  $\mathbb{R}^n$ . En dimension 1, tel qu'illustré dans la figure 1.2-A, un objet est une 1-variété si il admet en tout point un voisinage homéomorphe à un segment de droite. La figure 1.2-B illustre une courbe dont la condition de variété n'est pas vérifiée au niveau du croisement. En dimension 2, un objet est une 2-variété si il admet en tout point un voisinage homéomorphe à un morceau de plan. Dans le cadre de la discrétisation d'un objet en cellules, un certain nombre de cas de non-variété peuvent être énoncés qui se caractérisent par des points présentant un voisinage trop petit (sous-ensemble d'un disque) ou trop grand (sur-ensemble d'un disque). Ceux-ci sont illustrés dans la figure 1.3-B où l'on voit des faces adjacentes uniquement par un sommet, une arête incidente à plus de deux faces ou encore une arête pendante à un point. On appelle *surface* une variété de dimension 2. La figure 1.3-A montre une surface fermée (sans bord) à gauche et une surface ouverte (avec bord) à droite. Pour les dimensions supérieures à deux, le principe est



**Figure 1.4** - Exemples de surfaces orientable (une sphère) et non-orientable (une bande de Moebius)

similaire, même s'il est moins aisé d'en donner une approche intuitive.

Déterminer l'*orientation* d'une 2-variété en un point revient à choisir un sens de rotation autour de ce point sur la surface. Afin de déterminer l'orientation en un autre point de la surface, on peut ensuite transporter cette orientation par continuité le long d'un chemin. Une surface est dite *orientable* si pour tout point  $p$  de la surface et pour tout chemin fermé  $c$  sur la surface passant par  $p$ , une orientation de la surface en  $p$  est transportée par continuité en elle-même le long du chemin  $c$ . La figure 1.4 illustre à gauche un exemple de surface orientable (une sphère). Sur la droite on voit un exemple bien connu de surface non-orientable, la bande de Moebius : une fois une orientation choisie, si on parcourt le chemin dessiné ici, on peut se retrouver au même point mais avec l'orientation inverse.

## Modèles structurés

Un domaine de représentation très courant en modélisation 3D est celui des surfaces orientables, avec ou sans bord. De nombreuses structures de données ad-hoc ont donc été proposées au fil du temps pour représenter de tels objets. Le premier modèle introduit est celui dit *fil de fer* dans lequel les objets sont représentés par un ensemble de sommets connectés par des côtés. L'ambiguïté de cette représentation en a précipité l'abandon. En effet, comme on le voit à la figure 1.5, à un même modèle représenté en fil de fer peuvent correspondre plusieurs objets bien distincts.

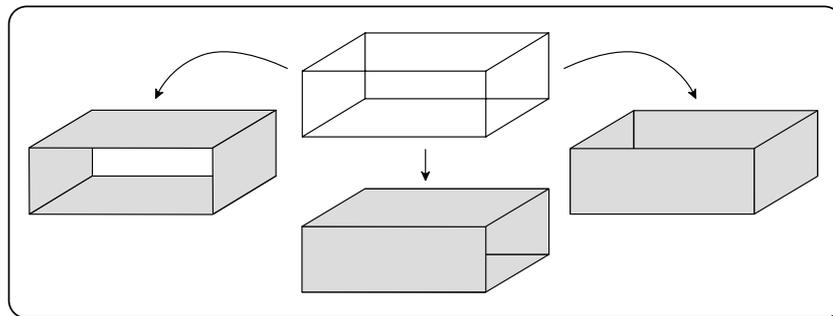


Figure 1.5 - Ambiguïté de la représentation en fil de fer

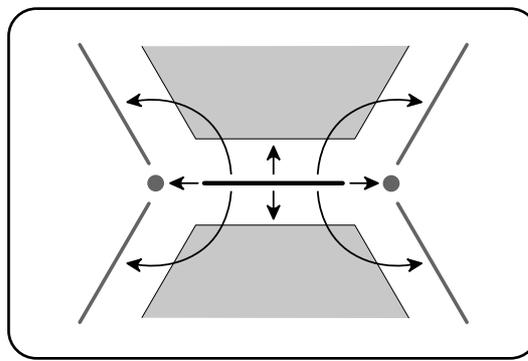


Figure 1.6 - Représentation par arêtes ailées

La structure d'*arêtes ailées*, proposée dans [Bau75] et dont une illustration est donnée à la figure 1.6 permet la représentation de surfaces orientables sans bord. Toute l'information est ici structurée autour des arêtes, qui stockent chacune des liens vers leurs deux sommets et leurs deux faces incidentes, ainsi que quatre liens permettant de rejoindre les arêtes connectées en tournant autour desdits sommets.

Le choix des arêtes comme entité centrale dans cette représentation est dû au fait que contrairement aux autres, au sein de la subdivision d'une 2-variété fermée, les relations d'incidence et d'adjacence liant les arêtes aux autres cellules impliquent un nombre constant d'éléments. En effet, dans le cas d'un maillage représentant la subdivision d'une surface orientable sans bord, une arête a toujours deux sommets terminaux et deux faces incidentes. Toutes les autres relations (nombre de faces autour d'un sommet, nombre d'arêtes autour d'une face, ...) impliquent un nombre variable de cellules. L'objectif est de trouver le moyen de coder l'ordonnancement de ces arêtes autour des sommets ou autour des faces de la subdivision.

La structure d'arêtes ailées a été améliorée par la suite par Weiler dans [Wei85] en divisant chaque arête en deux demi-arêtes orientées liées ensemble (figure

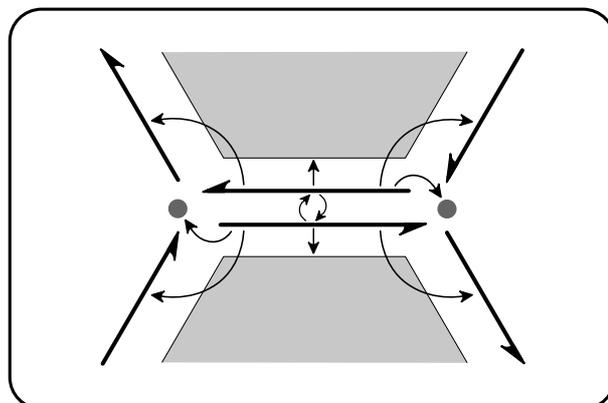


Figure 1.7 - Représentation par demi-arêtes

1.7). Combiné à l'ajout de nouveaux pointeurs, cela a permis d'améliorer l'efficacité d'un certain nombre d'algorithmes de parcours du maillage représenté. Cette structure de données a été déclinée sous d'autres noms incluant les *DCEL* ou liste doublement connectée d'arêtes, et est utilisée dans de nombreux projets ou bibliothèques comme CGAL <sup>1</sup> ou OpenMesh <sup>2</sup> [BSBK02]. Du point de vue de l'information combinatoire contenue dans ces structures, celles-ci sont équivalentes à la notion de *carte combinatoire* qui les précède pourtant largement chronologiquement [Edm60, Vin83].

Les cartes combinatoires proposent un cadre mathématique pour représenter des subdivisions de surfaces orientables sans bord. Contrairement aux structures de données complexes mettant en jeu de nombreuses entités différentes, un formalisme algébrique simple basé sur un unique type d'élément est ici fourni. Outre une unification des opérateurs topologiques (qui n'ont plus à agir que sur un seul type d'entité) ce formalisme permet une maintenance plus aisée de la cohérence du modèle à travers le respect de contraintes d'intégrité qui peuvent être exprimées simplement. Bon nombre de sources d'erreurs rencontrées avec les structures de données complexes sont ainsi évitées.

Ce cadre mathématique a également permis l'extension de ce modèle afin de permettre la représentation de la topologie de la subdivision de variétés de dimension quelconque (on parle alors de *n*-cartes) [Lie88, Lie89]. Ce modèle peut être spécifié algébriquement ce qui permet par exemple la création de modélisateurs 3D robustes conformes à ces spécifications [BDFL93, BD94, Duf97]. Enfin l'utilisation de méthodes formelles pour la preuve de théorèmes et d'algorithmes géométriques est rendue possible à travers une formalisation du modèle dans des langages de preuves au sein d'outils d'assistance à la preuve tels que

<sup>1</sup><http://www.cgal.org>

<sup>2</sup><http://www.openmesh.org>

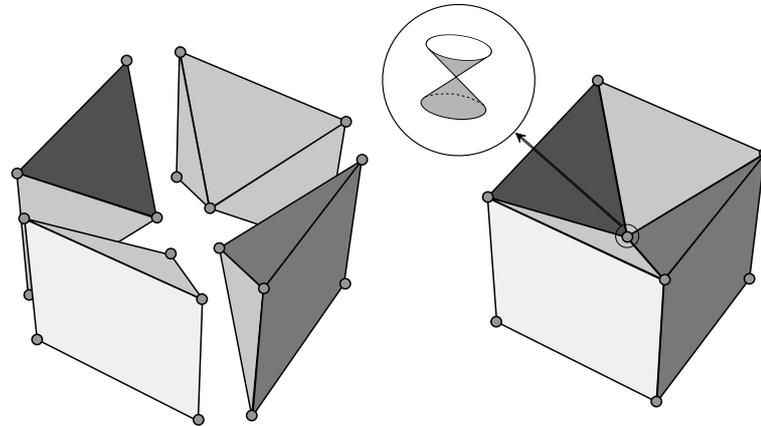


Figure 1.8 - La boule pincée : exemple de quasi-variété de dimension 3

Coq [CD99, DD04a, DD04b, Duf07].

Le domaine de représentation des cartes combinatoires a été étendu à celui des quasi-variétés de dimension  $n$  orientables ou non par la définition des cartes généralisées ou G-cartes [Lie91, Lie94]. On appelle quasi-variété de dimension  $n$  un objet pouvant être décrit de façon générale comme un assemblage de  $n$ -cellules deux à deux le long de  $n - 1$  cellules de leur bord. Pour un objet de dimension  $n > 2$  cet assemblage ne constitue plus nécessairement une variété, mais admet des configurations où le voisinage d'un point est un sous-ensemble de la boule ouverte de dimension  $n$ . Un exemple classique est celui de la boule pincée obtenue par assemblage de volumes le long de leurs faces, et dont le point de pincement n'admet pas un voisinage homéomorphe à  $\mathbb{R}^3$ . Un exemple de subdivision de cette boule pincée en quatre pyramides à base carrées est illustré à la figure 1.8.

## 1.2.2 Cartes combinatoires

Avant d'en donner une présentation formelle, nous pensons qu'il est important de passer en premier lieu par une approche plus intuitive. Nous avons choisi de présenter tout d'abord les G-cartes en s'inspirant largement de la présentation faite par Lévy dans [LM99] et [Lev00]. Comme nous l'avons déjà dit, des modèles plus spécifiques et plus efficaces peuvent être définis si l'on restreint le domaine de représentation. De la définition des  $n$ -G-cartes, nous déduisons ensuite la définition des  $n$ -cartes, modèle plus spécifique et optimisé permettant de représenter la topologie de la subdivision cellulaire de variétés orientables.

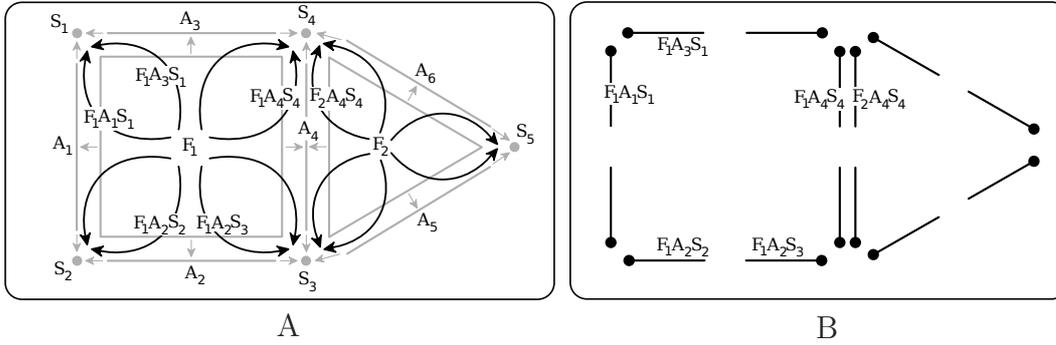


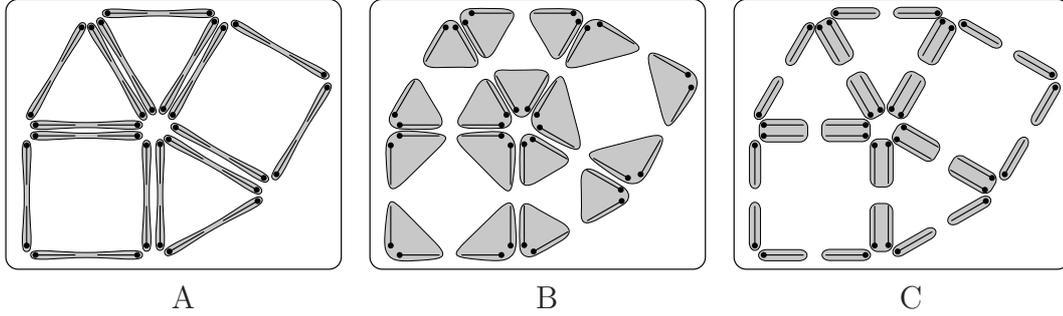
Figure 1.9 - Tuples de cellules d'un graphe d'incidence

## Cartes généralisées

La présentation qui suit permet de retrouver la définition des  $n$ -G-cartes en prenant pour point de départ la notion de graphe d'incidence que nous avons introduite au début de cette section. Au lieu de s'intéresser aux relations liant les cellules entre elles, on considère ici la notion de *tuple de cellules* telle qu'elle a été introduite par Brisson [Bri89, Bri93]. Dans un objet de dimension  $n$ , un tuple de cellules est défini comme une séquence ordonnée de cellules  $(c_n, c_{n-1}, \dots, c_0)$  de dimensions décroissantes, telles que  $\forall i \in [1, n]$ , la cellule  $c_i$  est incidente à la cellule  $c_{i-1}$ . Chaque tuple de cellules représente un chemin dans le graphe d'incidence. En dimension 2, un tuple de cellules est constitué d'une face, d'une arête et d'un sommet. La figure 1.9-A illustre l'ensemble des tuples de cellules que l'on peut construire sur le graphe d'incidence de la figure 1.1, dont certains sont étiquetés par l'ensemble des cellules qui les composent. La figure 1.9-B donne une représentation graphique plus légère de ces tuples de cellules.

Deux tuples de cellules sont dits  $i$ -adjacents ou  $i$ -voisins si ils partagent les mêmes cellules à l'exception de celle de dimension  $i$ . Par exemple, les tuples  $F_1A_2S_2$  et  $F_1A_2S_3$  de la figure 1.9 sont 0-adjacents,  $F_1A_1S_1$  et  $F_1A_3S_1$  sont 1-adjacents et  $F_1A_4S_4$  et  $F_2A_4S_4$  sont 2-adjacents.

Un résultat important montre que dans le cadre de la représentation de partition cellulaire d'objets quasi-variétés sans bord, ces relations d'adjacence lient toujours les tuples de cellule deux à deux. Si l'on représente des quasi-variétés à bord, les tuples de cellules composés de cellules du bord n'ont pas de  $n$ -voisin. La figure 1.10 illustre, sur un ensemble un peu plus complexe que précédemment, les regroupements de tuples de cellules que l'on peut former relativement à la 0-adjacence (figure 1.10-A), la 1-adjacence (figure 1.10-B) et la 2-adjacence (figure 1.10-C).



**Figure 1.10** - Adjacences entre les tuples de cellules de la subdivision d'une 2-variété

Une  $G$ -carte [Lie89, Lie94] est une structure combinatoire composée d'un type unique d'éléments appelé *brin*. Chacun des brins d'une  $n$ - $G$ -carte est mis en correspondance avec un tuple de cellules du graphe d'incidence de la partition cellulaire de la quasi-variété de dimension  $n$  représentée. Les relations d'adjacence entre les brins sont définies de la même manière que les adjacences entre les tuples de cellules. On appelle ici  $\alpha_i$  la fonction qui à un brin de la  $G$ -carte associe le brin qui lui est  $i$ -adjacent. Tout comme les tuples de cellules, les brins sont toujours  $i$ -adjacents deux à deux, exceptés les brins du bord qui n'ont pas de  $n$ -voisin. Les fonctions  $\alpha_i$  sont donc des involutions, i.e. des fonctions telles que pour tout brin  $b$ ,  $\alpha_i(\alpha_i(b)) = b$ . Le brin  $b' = \alpha_i(b)$  correspond donc au tuple de cellules partageant les mêmes cellules que  $b$  à l'exception de celle de dimension  $i$ . On dit que  $b$  et  $b'$  sont *cousus* par l'involution  $\alpha_i$  ou  $i$ -cousus, ce qui en fait deux  $i$ -voisins. Ces involutions respectent les conditions suivantes :

$$\forall i, j \mid 0 \leq i < i + 2 \leq j \leq n, \alpha_i \circ \alpha_j \text{ est une involution} \quad (1.1)$$

Si l'on construit une  $n$ - $G$ -carte à partir du graphe d'incidence de la partition cellulaire d'une quasi-variété et de ses tuples de cellules, alors cette contrainte est automatiquement respectée par construction. Elle doit en revanche être vérifiée si l'on veut obtenir une  $n$ - $G$ -carte par assemblage de cellules, comme nous le verrons par la suite.

Formellement, une  $n$ - $G$ -carte est un  $(n + 2)$ -uplet  $G = (B, \alpha_0, \dots, \alpha_n)$  où  $B$  est un ensemble fini de brins, et les  $\alpha_i$  sont des involutions sur  $B$  respectant les conditions 1.1. La figure 1.11 montre la 2- $G$ -carte construite à partir des tuples de cellules de la figure 1.10. La relation  $\alpha_0$  est illustrée par un trait en pointillés,  $\alpha_1$  par un trait plein et  $\alpha_2$  par un double trait.

En tant que tuple de cellules, chaque brin est donc le représentant d'un ensemble de cellules de chaque dimension  $i \in [0, n]$ , et chaque cellule est représentée par un ensemble de brins. Dans la mesure où l'on manipule ici explicitement

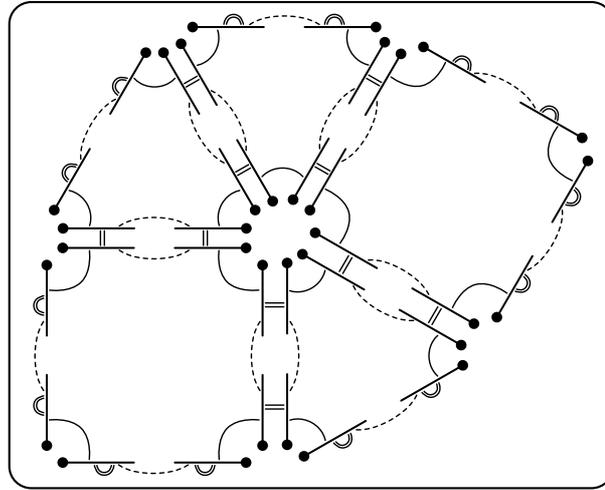


Figure 1.11 - Exemple de 2-G-carte

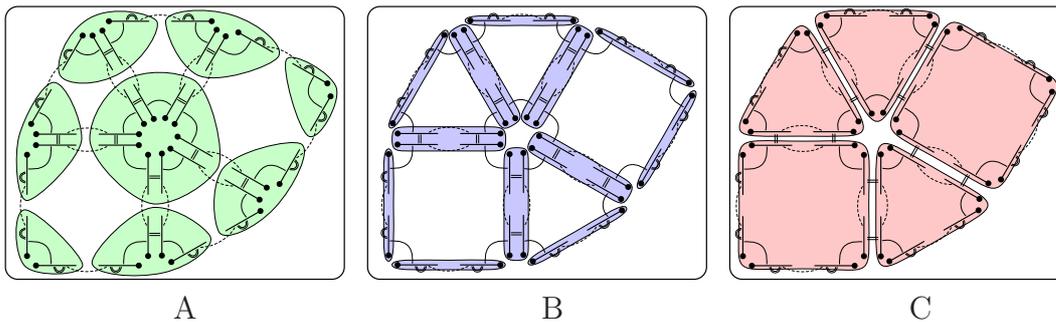
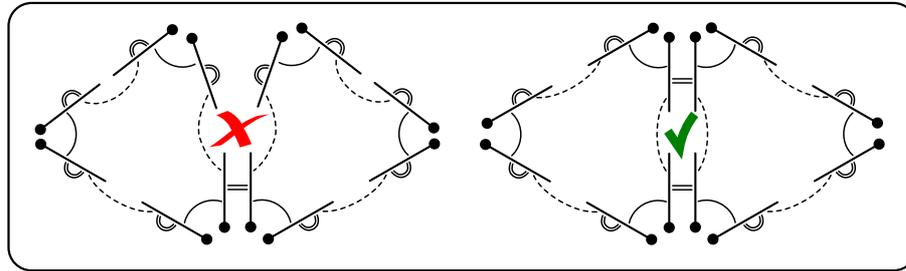


Figure 1.12 - Orbites d'une 2-G-carte

des ensembles de brins, une cellule donnée est identifiée par l'ensemble des brins qui la représentent, c'est-à-dire par l'ensemble des tuples de cellules qui la contiennent. Cette représentation implicite suffit à définir complètement les cellules et leurs relations d'incidence et d'adjacence. En partant d'un brin  $b$  donné, l'ensemble des brins correspondant à une même  $i$ -cellule est obtenu en appliquant récursivement à partir du brin  $b$  toutes les involutions  $\alpha_j$ ,  $j \neq i$ , c'est-à-dire toutes les relations d'adjacence qui ne nous font pas changer de  $i$ -cellule. La notion d'*orbite* exprime cela de la façon suivante : l'orbite  $\langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_n \rangle (b)$  définit l'ensemble des brins de la  $i$ -cellule du brin  $b$ .

La figure 1.12 met en évidence les cellules de la 2-G-carte de la figure 1.11. Les ensembles de brins correspondant aux orbites sommets ( $\langle \alpha_1, \alpha_2 \rangle$ ), arêtes ( $\langle \alpha_0, \alpha_2 \rangle$ ) et faces ( $\langle \alpha_0, \alpha_1 \rangle$ ) sont illustrés respectivement dans les figures 1.12-A, 1.12-B et 1.12-C. On retrouve ici les huit sommets, douze arêtes et cinq faces de la partition cellulaire à l'origine de cette G-carte.



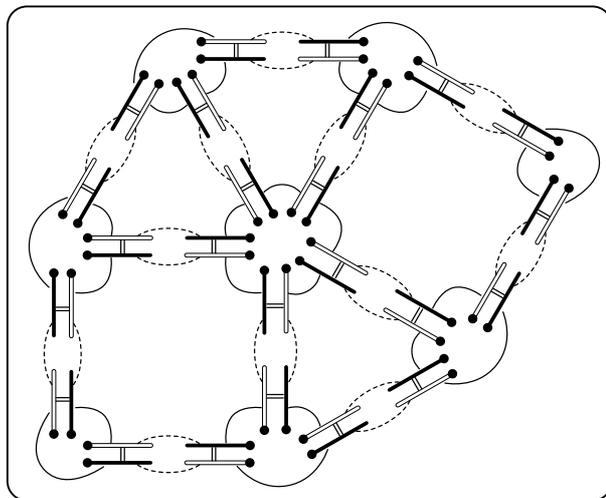
**Figure 1.13** - Couture de deux faces le long d'une arête dans une 2-G-carte

L'approche constructive consiste à assembler des cellules de dimension croissante le long de leur bord. Cette approche, qui peut paraître plus abordable que celle reposant sur les graphes d'incidence et leurs tuples de cellules, ne donne néanmoins pas la même intuition de la sémantique liée au brin. On débute par la représentation d'une partition cellulaire d'une variété de dimension 1. Celle-ci est composée d'une chaîne d'arêtes ouverte ou fermée. Chaque arête est composée de deux brins liés par l'involution  $\alpha_0$ . Les arêtes sont cousues entre elles deux à deux par l'involution  $\alpha_1$ .

Un ensemble fermé d'arêtes cousues définit le bord d'une face. La subdivision d'une variété de dimension 2 est composée d'un ensemble de faces ouvert ou fermé. Deux faces sont cousues le long d'une arête par l'involution  $\alpha_2$ . Afin de modéliser une partition cellulaire de dimension 2 valide, cette couture doit lier les deux arêtes *complètement*. Cela implique la dépendance entre  $\alpha_0$  et  $\alpha_2$  exprimée par les conditions (1.1) qui dit que  $\alpha_0 \circ \alpha_2$  est une involution. La figure 1.13 illustre la couture partielle de deux faces le long d'une arête à gauche, et la couture complète respectant la contrainte d'intégrité de la G-carte à droite. Les objets représentables sont des partitions cellulaires de variétés de dimension 2.

Un ensemble fermé de faces cousues entre elles définit le bord d'un volume. La subdivision d'une variété de dimension 3 est composée d'un ensemble de volumes ouvert ou fermé. Deux volumes sont cousus le long d'une face par l'involution  $\alpha_3$ . Afin de modéliser une partition cellulaire de dimension 3 valide, cette couture doit lier deux faces *complètement*, d'où les dépendances entre  $\alpha_0$  et  $\alpha_3$  et entre  $\alpha_1$  et  $\alpha_3$  exprimées par la contrainte de l'équation 1.1. Ce sont ici les relations  $\alpha_0 \circ \alpha_2$  et  $\alpha_1 \circ \alpha_3$  qui doivent être des involutions. Les objets représentables sont des partitions cellulaires de quasi-variétés de dimension 3.

Et ainsi de suite pour les dimensions supérieures ... On peut montrer par récursion que n'importe quelle partition cellulaire de quasi-variété de dimension  $n$  peut être représenté par une  $n$ -G-carte [Lie91, Lie94].



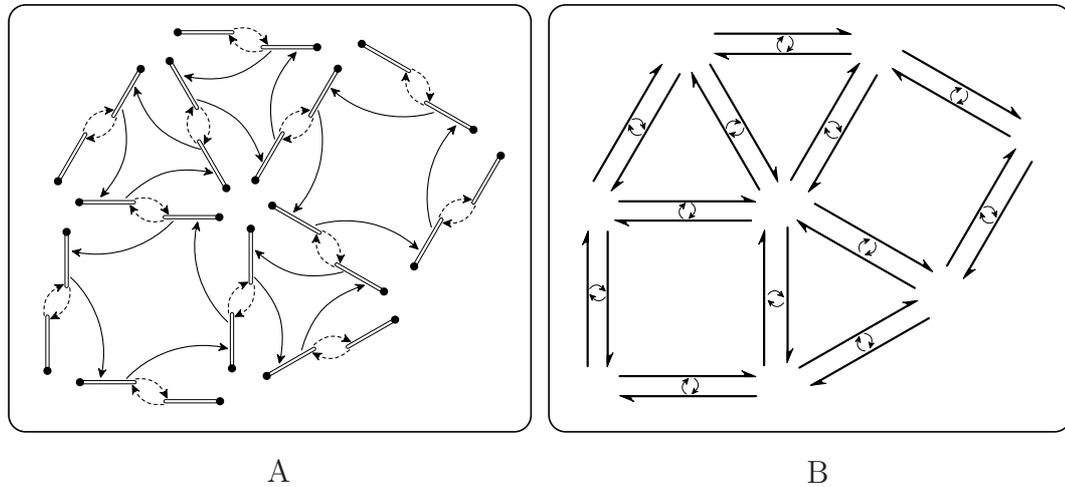
**Figure 1.14** - Les deux orientations d'une 2-G-carte orientable (brins noirs ou brins blancs)

## Cartes combinatoires

Les  $n$ -cartes sont une spécialisation des  $n$ -G-cartes aux objets orientables sans bord. Intuitivement, une  $n$ -carte est une demi  $n$ -G-carte orientable sans bord dans laquelle une seule orientation a été conservée. Les objets à bord étant tout de même relativement courants, nous verrons comment il est possible de gérer de tels objets avec une carte.

Revenons tout d'abord sur la notion d'orientation. Une méthode intuitive pour définir l'orientabilité d'un objet représentée par une  $n$ -G-carte consiste à opérer une *coloration* des brins en noir et blanc. La couleur d'un brin de la carte est positionnée en suivant la règle exprimant qu'un brin ne peut être cousu qu'à des brins de couleur opposée. Le processus de coloration est alors initialisé en fixant arbitrairement une couleur pour un brin donné de la G-carte. On colore ensuite les autres brins en respectant la règle précédemment exprimée. Si ce processus termine, c'est-à-dire si tous les brins peuvent être colorés en suivant cet algorithme, alors l'objet représenté est orientable, sinon il est non-orientable.

La figure 1.14 montre le résultat de l'orientation de la 2-G-carte de la figure 1.11. Cette dernière a été préalablement fermée en ajoutant une face à sept côtés. L'involution  $\alpha_2$  n'admet ici plus aucun point fixe. En pratique en dimension 2, on pourra initier l'algorithme de coloration sur une face en choisissant un sens de rotation en colorant un brin sur deux de la face dans une couleur donnée. Cette orientation est ensuite propagée aux faces voisines en traversant les arêtes communes, et ainsi de suite sur toutes les faces du maillage.



**Figure 1.15** - 2-carte correspondant à une des deux orientations d'une 2-G-carte orientable

Dans le cas où le processus de coloration termine, l'ensemble de brins  $B$  se trouve alors partitionné en deux sous-ensembles  $B^{noir}$  et  $B^{blanc}$  contenant exactement le même nombre de brins. En partant d'un brin  $b \in B^{blanc}$ , l'orbite  $\langle \phi_1, \phi_2, \dots, \phi_n \rangle$  avec  $\phi_i = \alpha_i \circ \alpha_0$  permet de retrouver tous les brins de  $B^{blanc}$ . De même en partant d'un brin de  $B^{noir}$ , la même orbite permet de retrouver tous les brins de  $B^{noir}$ .

Une  $n$ -carte est alors définie comme le  $(n + 1)$ -uplet  $C = (B^{blanc}, \phi_1, \dots, \phi_n)$  où  $B^{blanc}$  est un ensemble fini de brins,  $\phi_1$  est une permutation sur  $B$  et les  $\phi_i$  avec  $2 \leq i \leq n$  sont des involutions sur  $B$ . La permutation  $\phi_1$  exprime le cycle de sommets ordonnés autour de chaque face orientée. La contrainte imposant aux autres relations d'être des involutions provient de la contrainte d'intégrité exprimée sur les G-cartes par les conditions 1.1. La même carte contenant exactement la même information topologique mais d'orientation opposée est définie sur l'ensemble de brins  $B^{noir}$ .

La figure 1.15-A montre l'une des deux 2-cartes que l'on peut obtenir à partir de la 2-G-carte orientable de la figure 1.14. La permutation  $\phi_1$  est illustrée par une flèche pleine et l'involution  $\phi_2$  par une flèche en pointillés. Une version allégée est illustrée à la figure 1.15-B, où chaque brin est directement représenté par une flèche, permettant ainsi l'économie de la représentation graphique de la permutation  $\phi_1$ .

La définition des orbites permettant de retrouver les cellules dans une  $n$ -carte peut être déduite de celle des  $n$ -G-cartes. Une différence néanmoins va concerner les sommets. En effet, la relation d' $i$ -adjacence dans une carte est définie comme la composition de la 0-adjacence et de la  $i$ -adjacence dans une G-carte.

Traverser l'involution  $\phi_i$  nous fait donc changer de  $i$ -cellule mais également de 0-cellule (sommet). Toutes les relations d'une  $n$ -carte faisant donc changer de sommet, on ne peut donc pas définir l'orbite du sommet d'un brin  $b$  comme l'ensemble des brins atteints à partir de  $b$  en appliquant récursivement toutes les relations ne faisant pas changer de sommet.

Dans une  $n$ -G-carte, l'orbite sommet est égale à  $\langle \alpha_1, \dots, \alpha_n \rangle$ . Pour tout  $i$  tel que  $2 \leq i \leq n$  l'application de la composition des relations  $\alpha_i$  et  $\alpha_1$  sur un brin nous fait changer à la fois de  $i$ -cellule et d'arête tout en restant sur place relativement à toutes les autres cellules. L'orbite  $\langle \alpha_1 \circ \alpha_2, \dots, \alpha_1 \circ \alpha_n \rangle$  va donc nous permettre, tout en restant sur un sommet, de passer par l'ensemble des cellules incidentes à ce sommet.

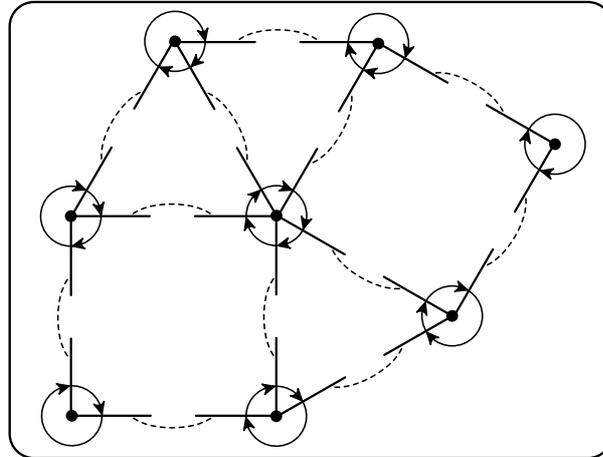
Cette orbite peut être exprimée au sein des  $n$ -cartes en écrivant tout d'abord  $\alpha_1 \circ \alpha_i = \alpha_1 \circ \alpha_0 \circ \alpha_0 \circ \alpha_i$ . Pour tout  $i \geq 2$ , d'après les conditions 1.1,  $\alpha_0 \circ \alpha_i$  est une involution et on a donc  $\alpha_0 \circ \alpha_i = \alpha_i \circ \alpha_0$ . On peut donc écrire  $\alpha_1 \circ \alpha_i = \alpha_1 \circ \alpha_0 \circ \alpha_i \circ \alpha_0$ , et finalement  $\alpha_1 \circ \alpha_i = \phi_1 \circ \phi_i$ . L'orbite sommet dans une  $n$ -carte est finalement la suivante :  $\langle \phi_1 \circ \phi_2, \dots, \phi_1 \circ \phi_n \rangle$ .

Pour les autres cellules, l'orbite permettant de parcourir les brins d'une  $i$ -cellule, avec  $i \geq 1$  est la suivante :  $\langle \phi_1, \dots, \phi_{i-1}, \phi_{i+1}, \dots, \phi_n \rangle$ . Tout comme pour les G-cartes, on traverse ici toutes les relations d'adjacence ne nous faisant pas changer de  $i$ -cellule.

Ici aussi une approche constructive est possible. Une chaîne d'arêtes fermée est représentée par un ensemble de brins liés par la permutation  $\phi_1$ . Un tel ensemble d'arêtes définit le bord d'une face. Deux faces sont cousues le long d'une arête par l'involution  $\phi_2$ . Un ensemble fermé de faces cousues entre elles définit le bord d'un volume. Deux volumes peuvent alors être cousus le long d'une face par l'involution  $\phi_3$ . Et ainsi de suite pour les dimensions supérieures.

Bien qu'étant l'approche la plus naturelle des cartes, cette représentation est communément appelée *représentation duale*. Il existe une autre expression de la même carte contenant exactement la même information appelée *représentation primale*. Une  $n$ -carte primale est un  $(n + 1)$ -uplet  $C = (B, \beta_0, \dots, \beta_{n-1})$  où  $B$  est un ensemble de brins,  $\beta_{n-1}$  est une permutation sur  $B$  et les  $\beta_i$  avec  $0 \leq i \leq n - 2$  sont des involutions sur  $B$ .

Ces deux formulations représentent exactement la même information et il est aisé d'exprimer l'une dans le formalisme de l'autre. Les relations  $\beta_i$  peuvent être exprimées en fonction des  $\phi_i$  de la version duale de la façon suivante :  $\beta_0 = \phi_n$  et, pour  $1 \leq i \leq n - 1$ ,  $\beta_i = \phi_i \circ \phi_n$ . Inversement, on peut aussi exprimer les  $\phi_i$  en fonction des  $\beta_i$  :  $\phi_n = \beta_0$  et, pour  $1 \leq i \leq n - 1$ ,  $\phi_i = \beta_i \circ \beta_0$ .



**Figure 1.16** - Exemple de 2-carte primale

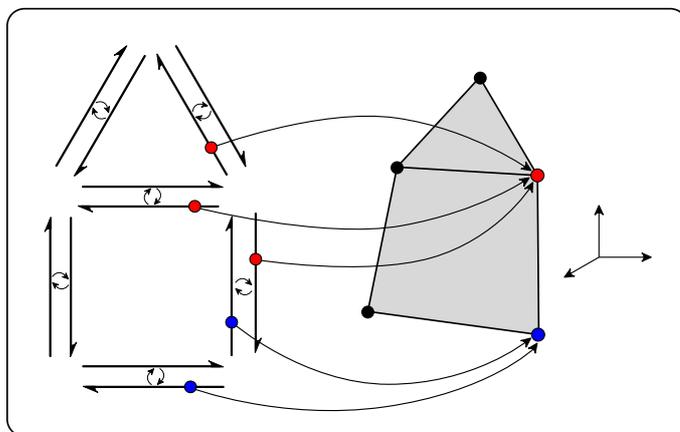
Dans la représentation primale, les orbites permettant de parcourir les brins des cellules de la subdivision représentée s'expriment de la façon suivante :  $\langle \beta_1 \circ \beta_0, \dots, \beta_{n-1} \circ \beta_0 \rangle$  pour le parcours des brins d'une  $n$ -cellule, et  $\langle \beta_0, \dots, \beta_{i-1}, \beta_{i+1}, \dots, \beta_{n-1} \rangle$  pour les  $i$ -cellules avec  $0 \leq i \leq n - 1$ .

Enfin, si l'on réexprime les  $n$ -cartes primales dans le formalisme des  $n$ -G-cartes, on obtient  $\beta_0 = \phi_n = \alpha_n \circ \alpha_0$  et, pour  $1 \leq i \leq n - 1$ ,  $\beta_i = \phi_i \circ \phi_n$ . Comme on peut également écrire  $\phi_i \circ \alpha_0 = \alpha_i \circ \alpha_0 \circ \alpha_0 = \alpha_i$ , l'expression précédente s'écrit alors  $\beta_i = \phi_i \circ \phi_n \circ \alpha_0 \circ \alpha_0 = \phi_i \circ \alpha_n \circ \alpha_0$ . Et comme pour  $n \geq 2$  on a  $\alpha_n \circ \alpha_0 = \alpha_0 \circ \alpha_n$ , finalement on peut écrire  $\beta_i = \phi_i \circ \alpha_0 \circ \alpha_n = \alpha_i \circ \alpha_n$ . La figure 1.16 illustre la représentation primale de la 2-carte de la figure 1.15.

## Plongement

L'objectif initial était de représenter des objets géométriques. Comme nous l'avons vu auparavant, afin de représenter complètement un tel objet, la description purement topologique que représente une carte ou une G-carte doit être complétée par un modèle de plongement associant des informations géométriques à chacune des cellules de la subdivision. Nous rappelons ici que toutes les représentations graphiques de l'information topologique qui ont été données jusqu'ici ne sont que des exemples possibles de son interprétation géométrique. En effet, il existe une infinité de plongements différents pouvant être associés à une même description topologique.

Chaque cellule étant identifiée par un ensemble de brins, tous les brins associés à une même  $i$ -cellule sont mis en correspondance avec la même information de



**Figure 1.17** - Exemple de plongement des sommets d'une 2-carte duale

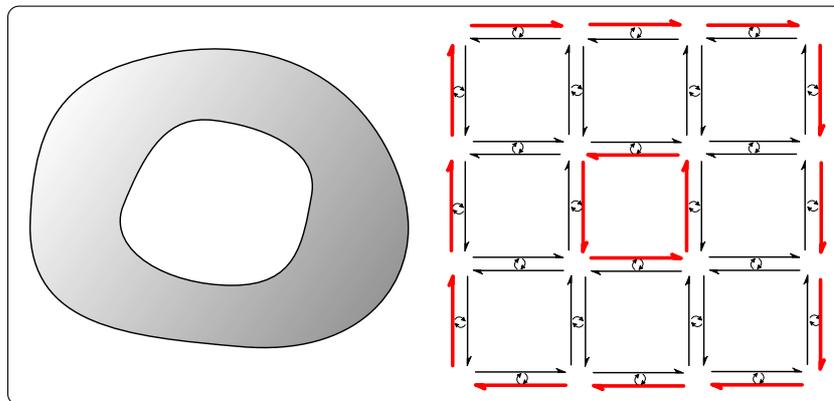
plongement de dimension  $i$ . Chaque brin, qu'il appartienne à une  $n$ -G-carte ou à une  $n$ -carte, en tant que représentant de  $n + 1$  cellules différentes (les  $n + 1$  cellules de dimension différente du tuple de cellules représenté par le brin), dispose donc de  $n + 1$  plongements différents. Formellement, si  $pl_i(b)$  est la fonction qui associe au brin  $b$  un plongement de dimension  $i$ , également appelé  $i$ -plongement, alors on a :

$$\forall b' \in orb_i(b), pl_i(b) = pl_i(b')$$

avec  $orb_i(b)$  l'orbite de la  $i$ -cellule du brin  $b$ .

La figure 1.17 illustre le 0-plongement d'une 2-carte duale. Chaque sommet du maillage est associé à un point de  $\mathbb{R}^3$ , ce qui est ici illustré pour deux des sommets du maillage. Il n'y a pas ici de 1-plongement explicite associé aux arêtes de la subdivision : leur géométrie est calculée par interpolation linéaire du plongement des sommets. Il en va de même pour le 2-plongement associé aux faces. Il y a évidemment de nombreuses autres manières de plonger une 2-carte dans  $\mathbb{R}^3$ , en associant par exemple des courbes de Bézier aux arêtes, ou encore des carreaux de Bézier aux faces. L'association de plongements complexes à une carte a été étudiée dans la thèse de Thery [The00].

Comme nous l'avons dit précédemment, les objets à bord sont suffisamment courants pour vouloir trouver un moyen de les représenter avec des  $n$ -cartes. Afin de réaliser cela sans pour autant briser les contraintes d'intégrité des cartes, une solution consiste à plonger les brins appartenant aux  $n$ -cellules du bord de l'objet avec un marqueur spécifique. On représente ainsi l'objet avec une carte fermée respectant les définitions données plus haut, et c'est aux algorithmes de traitement de cette carte qu'est déléguée la prise en compte de l'information supplémentaire apportée par ce marqueur.



**Figure 1.18** - Exemple de représentation d'une surface à bord avec une 2-carte duale

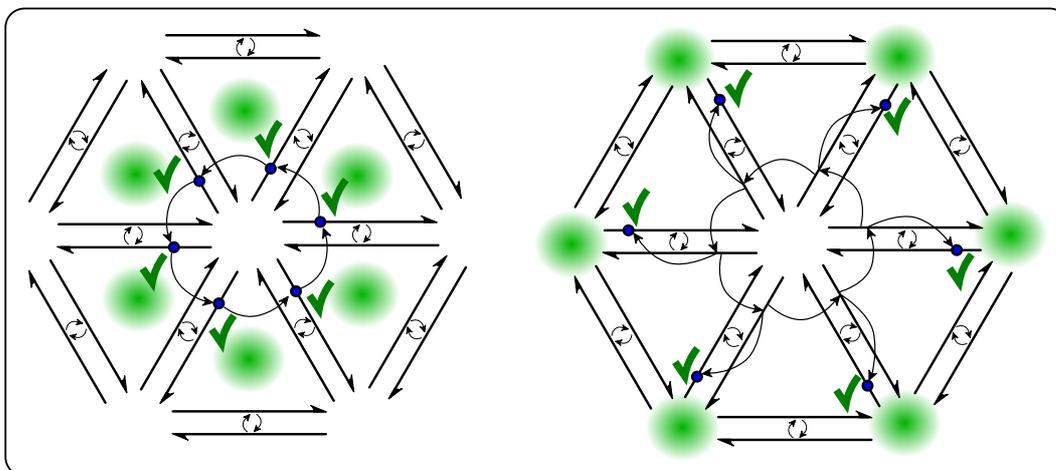
La figure 1.18 montre un exemple de surface à bord que l'on aimerait représenter à l'aide d'une 2-carte et un exemple d'une telle 2-carte. Les brins en rouges sont ceux qui constituent les faces marquées comme faisant partie du bord. Lorsqu'on souhaite afficher cette surface, la procédure d'affichage doit alors ignorer ces faces.

## Opérateurs d'adjacence

De nombreux algorithmes en traitement de la géométrie ou en simulation ont besoin d'effectuer des calculs mettant en jeu des informations de plongement contenues dans des cellules voisines. Plus l'information topologique contenue dans le modèle sous-jacent est complète, plus la récupération des informations nécessaires aux calculs est efficace.

Les cartes fournissent un support topologique optimal permettant de parcourir les maillages représentés de manière efficace, quelles que soient les cellules considérées. En effet, comme on l'a vu dans le parallèle avec les tuples de cellules (Section 1.2.2), chaque brin est représentatif d'un ensemble de cellules de chaque dimension. Etant donné un brin, on peut donc déjà atteindre l'ensemble de ces cellules sans même avoir à changer de brin. Les coutures qui lient un brin aux autres brins de la carte et la notion d'orbite permettent de parcourir l'ensemble des cellules incidentes à une cellule donnée dans un temps linéaire en la valence ou le degré de cette cellule. La figure 1.19 illustre deux exemples de parcours sur des cartes duales de dimension 2.

Le premier algorithme est illustré à gauche. Il consiste en le parcours suivant : étant donné un sommet du maillage, c'est-à-dire à partir de l'un des brins



**Figure 1.19** - Illustration des algorithmes permettant de parcourir l'ensemble des faces incidentes à un sommet, et l'ensemble des sommets adjacents à un sommet sur une 2-carte duale

représentant un sommet, on souhaite parcourir l'ensemble des faces incidentes à ce sommet. Il suffit pour cela de parcourir chacun des brins de l'orbite du sommet considéré. En effet, les tuples de cellules associés à ces brins ont tous en commun le sommet considéré, et sont chacun composés de l'une des faces appartenant au voisinage de ce sommet. Le temps nécessaire à l'exécution de ce parcours d'orbite est donc linéaire en le nombre de faces constituant ce voisinage, chaque passage d'une face à sa suivante étant lui effectué en temps constant. Si l'on souhaite faire un calcul mettant en jeu les plongements associés à toutes ces faces, il suffit, pour chaque brin  $b$  parcouru, de récupérer son 2-plongement associé  $pl_2(b)$ .

Le deuxième algorithme illustré à droite dans la figure 1.19 consiste à parcourir l'ensemble des sommets adjacents à un sommet donné. Le principe est très similaire à celui de l'algorithme précédent. Il consiste à parcourir tous les brins de l'orbite du sommet, et pour chacun, à traverser l'arête incidente au sommet afin d'atteindre le sommet voisin lié par cette arête. Ce parcours s'exécute également en un temps linéaire en le nombre de sommets du voisinage. En effet, passer d'un brin au suivant dans l'orbite du sommet et atteindre le sommet voisin lié par l'arête incidente sont toutes deux des opérations effectuées en temps constant. De même, si l'on veut calculer un résultat à partir des plongements de tous ces sommets voisins, on peut récupérer le 0-plongement associé à chacun des brins parcourus.

Des algorithmes similaires peuvent être écrits pour le parcours du voisinage de n'importe quelle cellule de la subdivision représentée.

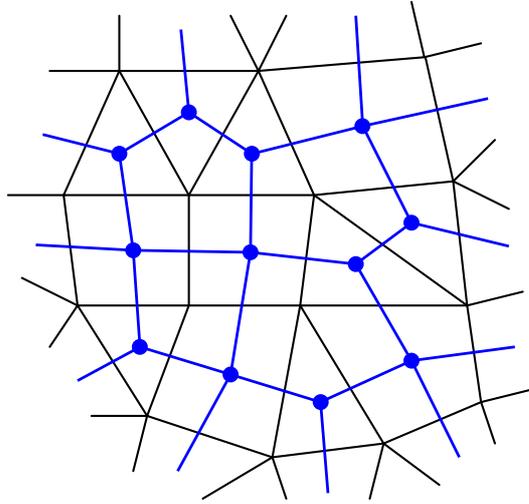


Figure 1.20 - Exemple d'un maillage polygonal et de son maillage dual

## Maillage dual

En dimension 2 topologique, le dual d'un maillage est le maillage dont chaque sommet est en correspondance avec une face du maillage original, chaque arête avec une arête, et chaque face avec un sommet. Les figures 1.20 et 1.22 montrent des exemples de maillages avec leur maillage dual associé.

Une 2-carte contient l'information combinatoire à la fois des maillages primal et dual. En effet, étant donnée une 2-carte primale  $C = (B, \beta_0, \beta_1)$ , si l'on interprète la permutation  $\beta_1$  comme la relation permettant de passer d'un brin à l'autre autour d'une face plutôt qu'autour d'un sommet, on parcourt en fait le maillage dual. On est en train de définir en réalité la 2-carte duale  $C = (B, \phi_1, \phi_2)$  telle que  $\phi_1 = \beta_1$  et  $\phi_2 = \beta_0$ . Cette 2-carte duale représente le dual du maillage de départ.

La figure 1.21 illustre la transition entre une 2-carte primale et la 2-carte duale représentant le maillage dual de l'objet de départ. Les brins de l'orbite d'un sommet de la 2-carte primale deviennent les brins de l'orbite d'une face de la 2-carte duale et inversement.

La transformation inverse est tout à fait possible. Si l'on part d'un maillage représenté par une 2-carte duale  $C = (B, \phi_1, \phi_2)$ , la 2-carte primale  $C = (B, \beta_0, \beta_1)$  telle que  $\beta_1 = \phi_1$  et  $\beta_0 = \phi_2$  représente le dual de ce maillage.

Supposons que l'on dispose d'une 2-carte primale dont les sommets sont plongés en un point 3D. La procédure standard d'affichage du maillage consiste à af-

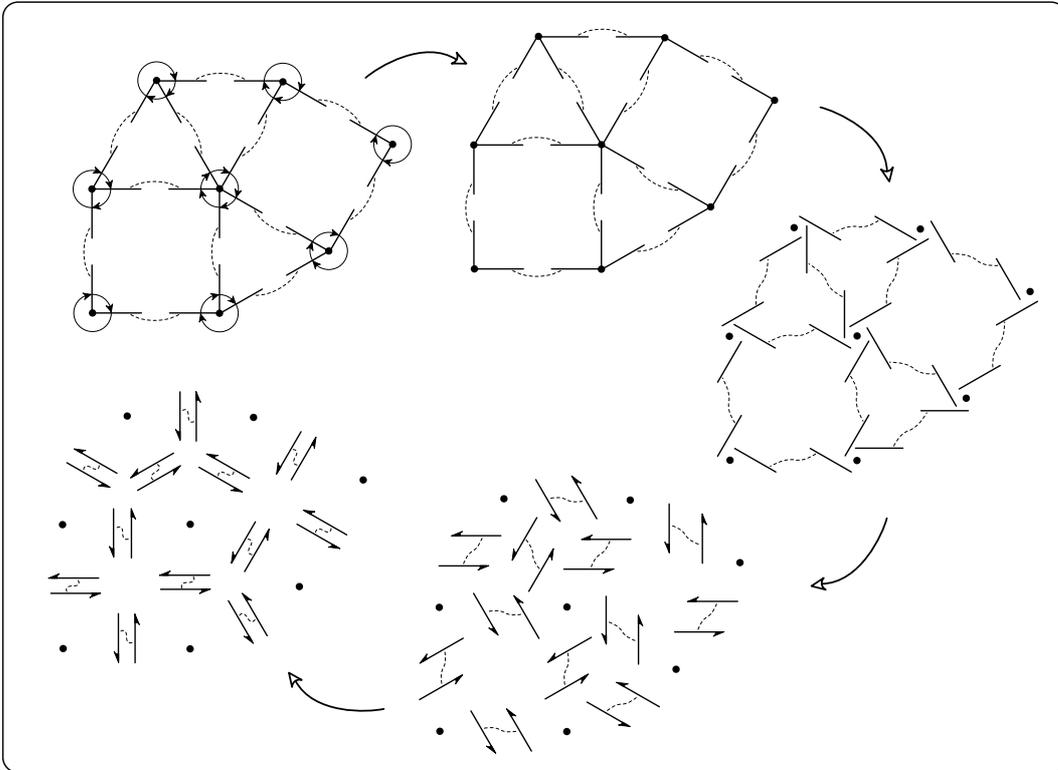


Figure 1.21 - Passage d'une 2-carte primale à la 2-carte duale représentant le dual du maillage de départ

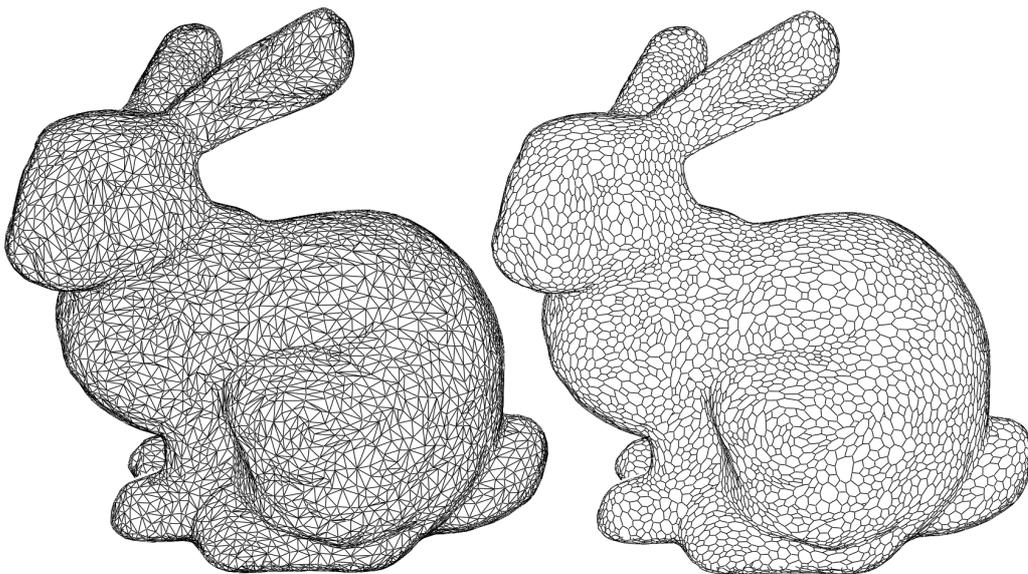


Figure 1.22 - Exemple d'un maillage triangulaire et de son maillage dual

ficher un polygone par face de la 2-carte. Pour chaque face, les positions des sommets de ce polygone sont récupérés en prenant, pour chaque brin de l'orbite de la face, son plongement de sommet associé. Si l'on souhaite maintenant afficher le maillage dual, la procédure consiste à afficher un polygone par sommet de la 2-carte, c'est-à-dire par face de la 2-carte duale représentant le maillage dual. Pour chaque sommet de la 2-carte primale, les sommets de ce polygone sont récupérés en prenant par exemple, pour chaque brin de l'orbite sommet, le barycentre de la face à laquelle il correspond.

## Hypercartes

Les modèles topologiques que nous avons présentés jusqu'ici, cartes ou G-cartes, ont tous en commun d'être définis par un ensemble de brins et un ensemble de permutations définies sur ces brins. D'un modèle à l'autre, les contraintes imposées sur ces permutations diffèrent.

Si l'on ne fixe pas de contraintes d'involution ou d'absence de point fixe sur ces permutations, et si l'on s'abstrait de la sémantique de tuple de cellule que nous avons associée au brin, on se retrouve en présence d'un cadre général commun à tous ces modèles, que l'on appelle *hypercarte*. Formellement, une hypercarte de dimension  $n$  est un  $(n + 1)$ -uplet  $H = (B, \alpha_0, \dots, \alpha_{n-1})$  où  $B$  est un ensemble fini de brins et les  $\alpha_i$ , avec  $i \in [0, n - 1]$  sont des permutations sur  $B$ .

Ce modèle constitue le cadre général abstrait dans lequel peuvent être exprimés tous les modèles issus des cartes combinatoires. Il nous servira de point de départ par la suite pour la définition de notre extension multirésolution des cartes combinatoires.

## 1.3 Modèles multirésolution

### 1.3.1 Généralités

Les modèles que nous avons présentés dans la section précédente permettent la représentation d'un maillage unique, figé à un moment donné. Certes, des opérateurs peuvent agir sur le maillage et permettent de le modifier, de le simplifier ou de le raffiner, mais le maillage précédant la modification est à chaque fois perdu et seule la version courante du maillage est accessible à un moment donné.

Un modèle de représentation multirésolution encode un maillage de manière à permettre d'accéder à différentes versions de ce maillage. Dans de nombreuses applications, ces différentes versions correspondent à des *résolutions* ou des *échelles* différentes d'un même objet. Un objet existe alors de sa version la plus grossière à sa version la plus fine, et toutes sont accessibles.

Une telle représentation est utilisée dans de nombreux cadres applicatifs et son utilisation ne se fait pas toujours dans le même objectif : soit la représentation multirésolution permet d'accélérer certains traitements ou algorithmes pré-existants, soit elle sert de support à la définition de nouveaux algorithmes. On peut citer dans la première catégorie l'affichage d'un objet en fonction du point de vue : plus un objet ou partie d'un objet est éloigné, plus on en affichera une version grossière, accélérant ainsi la tâche de rendu de l'objet sans sacrifier la qualité visuelle. La détection de collision peut également être accélérée en utilisant une version plus grossière d'un objet en tant que modèle de collision plutôt que l'objet dans toute sa complexité. Dans la deuxième catégorie, on peut citer tous les outils de traitement de la géométrie qui exploitent une hiérarchie de maillages, tels que l'édition multirésolution de surfaces de subdivision [Zor05], que l'on détaillera dans le chapitre 3, ou l'application d'outils de traitement du signal sur des maillages [GSS99], dont on parlera dans le chapitre 4.

Usuellement, la construction de la représentation multirésolution d'un maillage est classifiée en deux approches. L'approche "grossier vers fin" qui, partant d'un objet donné, va en créer des versions plus fines en raffinant successivement le maillage, et l'approche "fin vers grossier" qui, étant donné un objet fin de départ, va en créer des versions plus grossières par l'application de simplifications successives du maillage.

Comme on l'a dit, un modèle de représentation multirésolution doit permettre

d'encoder et d'accéder à ces différentes versions d'un maillage. Une première solution consiste simplement à stocker séparément les différentes résolutions de l'objet, chacun de ces maillages étant encodé grâce à un modèle standard que nous avons vu dans la section précédente. Si les maillages ont besoin d'être connectés entre eux, les structures peuvent alors être augmentées de liens entre les cellules de niveaux différents. Cette solution, bien que simple à mettre en place, est néanmoins très coûteuse en espace mémoire.

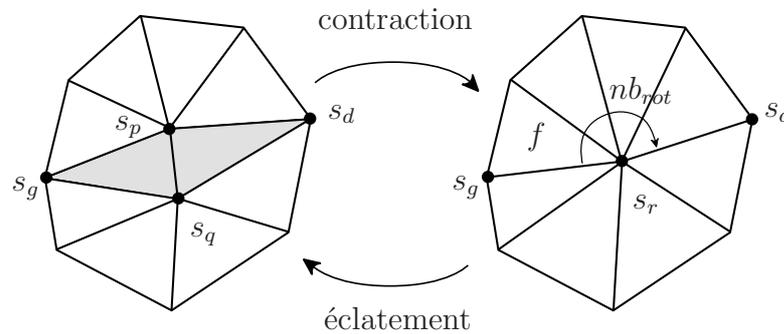
Quelle que soit l'approche de construction retenue, si ce que l'on cherche à encoder est bien les différentes résolutions d'un même objet, alors il existe des similarités entre ces différents maillages. Les encoder tous séparément introduit nécessairement une forte redondance dans le stockage de l'information. L'un des objectifs des modèles de représentation multirésolution est d'exploiter ces similarités pour supprimer autant que possible cette redondance.

Comme on va le voir, une des solutions les plus couramment utilisées [DM02, DKP04] consiste à stocker d'une part le maillage dans une structure standard et de maintenir d'autre part une structure arborescente dont les nœuds décrivent chacune des modifications que l'on applique successivement au maillage de départ afin de constituer l'ensemble des versions intermédiaires de l'objet. A partir du maillage d'origine, suivant l'ensemble de nœuds que l'on prend en compte et dont on applique les modifications correspondantes au maillage, on retrouve chacune des versions intermédiaires de l'objet.

### 1.3.2 Maillages progressifs

Les maillages progressifs ont été introduits dans [Hop96]. Ils se placent dans l'approche "fin vers grossier". Dans cette représentation, un maillage triangulaire fin de départ  $M$  est simplifié pas à pas par une série de  $n$  contractions d'arêtes successives, jusqu'à obtenir une version grossière  $M^0$  du maillage de départ. On obtient donc  $n + 1$  versions différentes du maillage, chacune étant composée d'un sommet de moins que la précédente. On a  $M \xrightarrow{\text{contract}_{n-1}} \dots \xrightarrow{\text{contract}_1} M^1 \xrightarrow{\text{contract}_0} M^0$ .

L'opérateur inverse de la contraction d'arête est l'éclatement de sommet. On peut donc partir du maillage grossier  $M^0$  et retrouver le maillage d'origine  $M$  en appliquant en sens inverse les  $n$  opérations d'éclatement de sommet correspondant aux opérations de contraction d'arête. On a  $M^0 \xrightarrow{\text{eclat}_0} M^1 \xrightarrow{\text{eclat}_1} \dots \xrightarrow{\text{eclat}_{n-1}} M$ . La figure 1.23 illustre les opérateurs de contraction d'arête et d'éclatement de sommet dans un maillage triangulaire.



**Figure 1.23** - L'opérateur de contraction d'arête et son inverse, l'éclatement de sommet

C'est le couple  $(M^0, \{eclat_0, \dots, eclat_{n-1}\})$  formé du maillage le plus grossier obtenu et de l'ensemble des opérations d'éclatement de sommet qui forme ce que l'on appelle une représentation en maillage progressif du maillage  $M$ . N'importe laquelle des  $n + 1$  versions intermédiaires du maillage peut être obtenue en n'appliquant à  $M^0$  qu'une partie de la séquence d'opérations d'éclatement de sommet.

La question de l'implantation des maillages progressifs est abordée dans [Hop98]. Le couple  $(M^0, \{eclat_0, \dots, eclat_{n-1}\})$  est directement encodé. Le maillage est stocké dans une structure contenant un tableau de sommets et un tableau de faces, chacune des faces contenant un tableau permettant de retrouver ses trois sommets ainsi qu'un tableau indiquant ses trois triangles adjacents. Conjointement à ce maillage est encodé l'ensemble des opérations d'éclatement de sommet.

Pour définir entièrement un éclatement de sommet il faut pouvoir identifier le sommet  $s_r$  à éclater, ainsi que les deux sommets  $s_g$  et  $s_d$  afin de savoir où insérer les deux nouvelles faces introduites lors de l'opération. L'information topologique n'étant disponible ici que sur les faces, c'est un index de face permettant d'identifier la face  $f$  qui est stocké ici. Un autre index compris entre 0 et 2 permet de retrouver le sommet  $s_r$  dans cette face. Le sommet  $s_g$  est défini comme le sommet suivant  $s_r$  dans le sens horaire dans cette même face. Le sommet  $s_d$  est quant à lui identifié par un autre index  $nb_{rot}$  indiquant la rotation à effectuer dans le sens horaire autour du sommet  $s_r$  à partir de l'arête composée des sommets  $s_r$  et  $s_g$ .

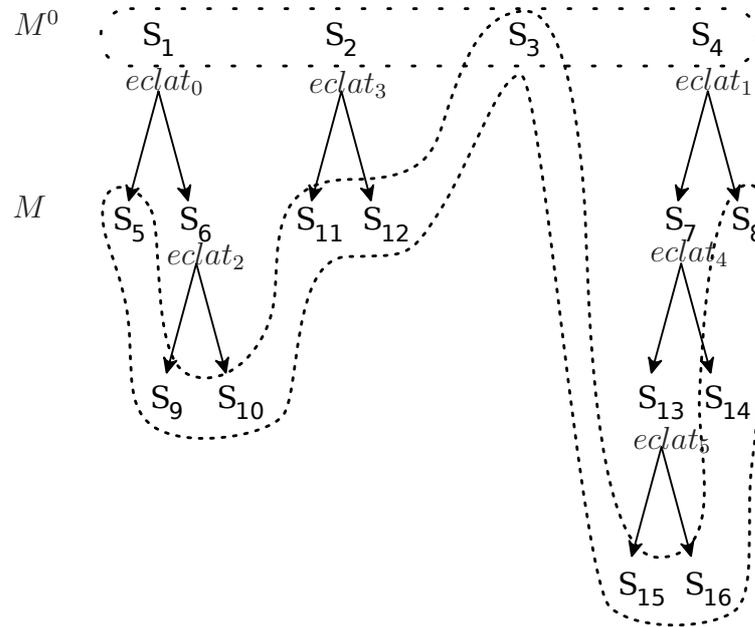
De plus, il faut pouvoir positionner géométriquement les deux nouveaux sommets  $s_p$  et  $s_q$ . Sachant que lors de la contraction d'arête, le sommet  $s_r$  a été positionné soit sur  $s_p$ , soit sur  $s_q$ , soit à mi-chemin entre ces sommets, un vecteur est stocké afin de pouvoir retrouver les positions de ces sommets à partir de celle du sommet  $s_r$ .

Pour obtenir un maillage  $M^i$  intermédiaire, on doit appliquer sur la structure encodant le maillage, les  $i$  premières opérations d'éclatement de sommet. L'application d'une de ces opérations sur le maillage implique l'ajout d'un sommet et de deux faces dans la structure, suivie de la mise-à-jour des informations d'adjacence entre les faces et enfin la mise-à-jour de la position des sommets. On peut également revenir en arrière et appliquer une contraction d'arête (l'information nécessaire à cette opération étant contenue dans le codage d'un éclatement de sommet). Cette application implique la suppression d'un sommet et de deux faces, la mise-à-jour des adjacences entre les faces et enfin l'attribution d'une position au sommet  $s_r$ .

Ces  $n + 1$  maillages différents proposent certes des versions d'un même objet à des résolutions différentes, mais le fait d'être restreint à la séquence d'éclatements de sommet définis lors de la construction du maillage progressif n'est pas satisfaisant dans toutes les applications. On peut en effet souhaiter que le maillage soit plus fin dans une certaine zone et plus grossier dans une autre, sans que cela corresponde à la séquence de simplifications effectuée lors de la construction de la représentation progressive.

Une première solution donnée dans [Hop96] est de n'effectuer que les éclatements de sommets souhaités lors de la traversée de la séquence d'éclatements en ignorant les autres. Malheureusement, les choses ne sont pas aussi simples. Un éclatement de sommet ne peut en effet être exécuté que si les sommets concernés  $s_r$ ,  $s_g$  et  $s_d$  existent effectivement dans le maillage actuel. Si l'on omet un éclatement de sommet, on élimine du même coup la possibilité d'éclater plus tard les deux sommets  $s_p$  et  $s_q$  qui auraient été générés et ainsi de suite pour toute leur descendance. On obtient avec cette technique des maillages dont la résolution peut s'adapter à un certain critère indépendamment de la séquence dans lequel la représentation progressive a été construite, mais pouvant présenter un sous-raffinement dû à l'impossibilité prématurée d'effectuer certaines opérations d'éclatement de sommet.

Dans [Hop97], une structure étendant les maillages progressifs est proposée afin de permettre un accès adaptatif plus précis. A partir d'une représentation progressive classique d'un maillage, une forêt d'arbres binaires de sommets est construite. Les racines de ces arbres sont l'ensemble des sommets du maillage le plus grossier  $M^0$ . Chaque opération d'éclatement de sommet donne naissance à deux nouveaux sommets, insérés dans la forêt d'arbres binaires en tant que fils du sommet éclaté. Comme on le voit dans la figure 1.24 le maillage le plus grossier  $M^0$  est composé des sommets constituant l'ensemble des racines des arbres, tandis que le maillage fin  $M$  de départ sur lequel a été construite la représentation progressive est composé de l'ensemble des sommets feuilles des arbres. Chaque double flèche correspond à une opération d'éclatement de



**Figure 1.24** - Forêt d'arbres binaires de sommets permettant un accès adaptatif dans un maillage progressif

sommet. Le nombre total de sommets stockés dans la structure est égal environ à deux fois le nombre de sommets du maillage  $M$  de départ.

Un front de sommets actifs est maintenu au sein de cette forêt de même qu'un ensemble de faces actives au sein de la structure représentant le maillage. Cet ensemble de cellules actives forme le maillage adaptatif courant. Si le front de sommet est initialisé comme étant égal à l'ensemble des racines des arbres, le maillage adaptatif courant est égal au maillage grossier  $M^0$ . Le front de sommets actifs peut être mis-à-jour de deux façons : soit en "avançant" pour intégrer les deux sommets fils d'un sommet par l'application d'un éclatement de sommet, soit en "reculant" pour intégrer le sommet parent de deux sommets par l'application d'une contraction d'arête. L'application d'une de ces opérations n'est valide que si les sommets concernés ainsi que leur voisinage de faces est actuellement actif. Si une opération est légale, alors le front de sommets actifs est déplacé dans la forêt d'arbres binaires et l'ensemble des faces actives est mis-à-jour ainsi que leurs relations d'adjacence.

Dans [ESV99] grâce à une numérotation particulière des sommets au sein des arbres binaires, la validité d'une opération d'éclatement ou de contraction est vérifiée par une simple opération arithmétique sur les labels des sommets concernés par l'opération.

### 1.3.3 Multi-Triangulation

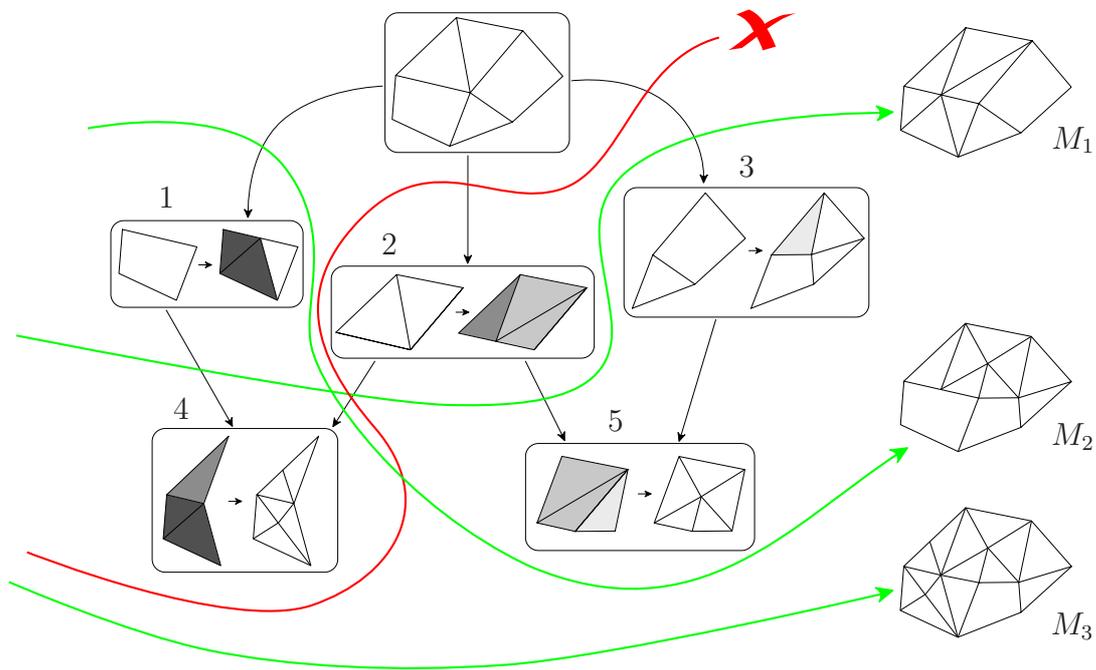
Le modèle dit Multi-Triangulation présenté dans [DP95, Pup98] permet la représentation de maillages triangulaires à des niveaux de résolution variables, indépendamment des opérations topologiques appliquées au maillage pour générer la hiérarchie. Ce modèle se positionne aussi bien dans l'approche "grossier vers fin" que dans l'approche "fin vers grossier". Les maillages progressifs que l'on vient de décrire peuvent être exprimés au sein de ce modèle. Une formulation plus générale du modèle Multi-Triangulation s'appliquant aux maillages polygonaux quelconques est décrite dans [DPM97, DM02].

Si l'on considère l'approche "grossier vers fin", l'idée est de partir d'un maillage grossier  $M^0$  que l'on va successivement raffiner par l'application d'une séquence d'opérations de mise-à-jour locales. Une mise-à-jour  $m$  est encodée par un couple  $(m^-, m^+)$  où  $m^-$  et  $m^+$  sont des ensembles de faces. L'application de cette mise-à-jour consiste à supprimer l'ensemble de faces  $m^-$  du maillage courant et à le remplacer par l'ensemble de faces  $m^+$  qui couvre la même surface et contient plus de faces. L'opération inverse consiste à simplifier localement le maillage en remplaçant les faces de  $m^+$  par celles de  $m^-$ .

Une mise-à-jour  $m_1$  est dépendante d'une autre mise-à-jour  $m_2$  si  $m_1$  supprime des faces introduits par  $m_2$ , ou autrement dit si  $m_2^+ \cap m_1^- \neq \emptyset$ . Cette dépendance entre les mise-à-jour implique un ordre partiel strict entre ces opérations qui peut être représenté par un graphe acyclique orienté [DMP98]. Si l'on part d'un maillage fin de départ, ce graphe peut également être construit par simplification successives, à l'image de ce qui est effectué dans le cadre des maillages progressifs.

La figure 1.25 illustre un exemple de maillage de départ ainsi qu'un graphe de dépendance de mises-à-jour de ce maillage. Pour chacun des nœuds sont illustrés l'ensemble des faces à supprimer ainsi que l'ensemble des faces par lesquelles les remplacer. Un nœud particulier ne supprimant aucune face et insérant l'ensemble des faces du maillage de départ  $M^0$  est ajouté en tant que racine de ce graphe.

L'accès aux maillages intermédiaires au sein de cette structure est effectué en effectuant des coupes dans le graphe. Si le front de coupe est initialisé juste au-dessous du nœud racine, le maillage courant auquel on accède est égal au maillage grossier  $M^0$ . Le front de coupe peut être mis-à-jour soit en "avançant" pour englober un nouveau nœud de modification induisant un raffinement du maillage courant, soit en "reculant" conduisant alors à une simplification du maillage courant.

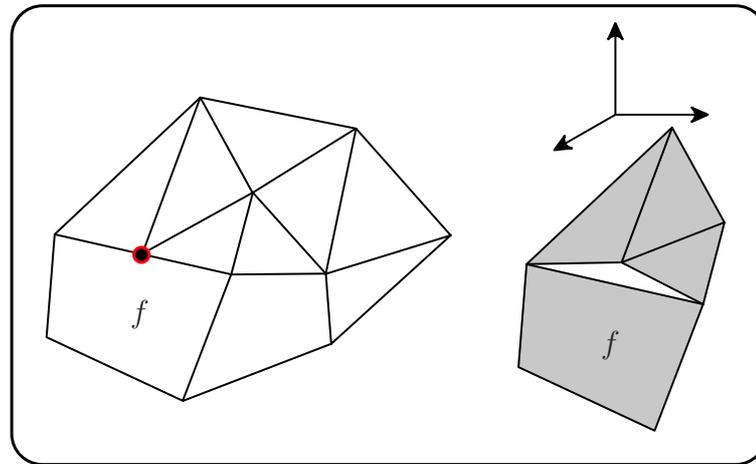


**Figure 1.25** - Graphe acyclique orienté encodant les dépendances entre des opérations de mise-à-jour d'un maillage polygonal

Un nœud de modification ne peut être ajouté que si l'ensemble des nœuds dont il dépend sont déjà actifs. Un nœud de modification ne peut être supprimé que si aucun des nœuds dépendant de lui n'est encore actif. Chaque déplacement du front de nœuds actifs entraîne l'application de l'opération de raffinement ou de simplification correspondante au sein de la structure de données représentant le maillage courant.

Plusieurs exemples de coupes sont illustrés pour la structure de la figure 1.25 : trois coupes légales en vert et une coupe illégale en rouge. L'illégalité de cette dernière provient du fait qu'elle ne respecte pas certaines dépendances entre les modifications. En effet, cette coupe englobe le nœud 4 alors que le nœud 2 duquel il est dépendant n'est pas inclus dans la coupe. Pour les trois autres coupes, les maillages extraits résultants  $M_1$ ,  $M_2$  et  $M_3$  sont illustrés sur la droite. Le maillage  $M_3$  correspond au maillage le plus fin représenté dans cette structure.

Dans le maillage  $M_2$ , un nouveau problème apparaît, illustré dans la figure 1.26. Le nœud de modification 2 remplace deux faces par trois faces en insérant un sommet sur l'arête du bas. La face  $f$  qui avait en commun cette arête avec l'un des triangles supprimés par le nœud de modification 2 reste elle inchangée lors de l'application de cette modification. Le résultat est un maillage dit *non-conforme* dont le plongement géométrique peut faire apparaître des trous dans



**Figure 1.26** - Maillage non-conforme généré par une coupe dans le graphe

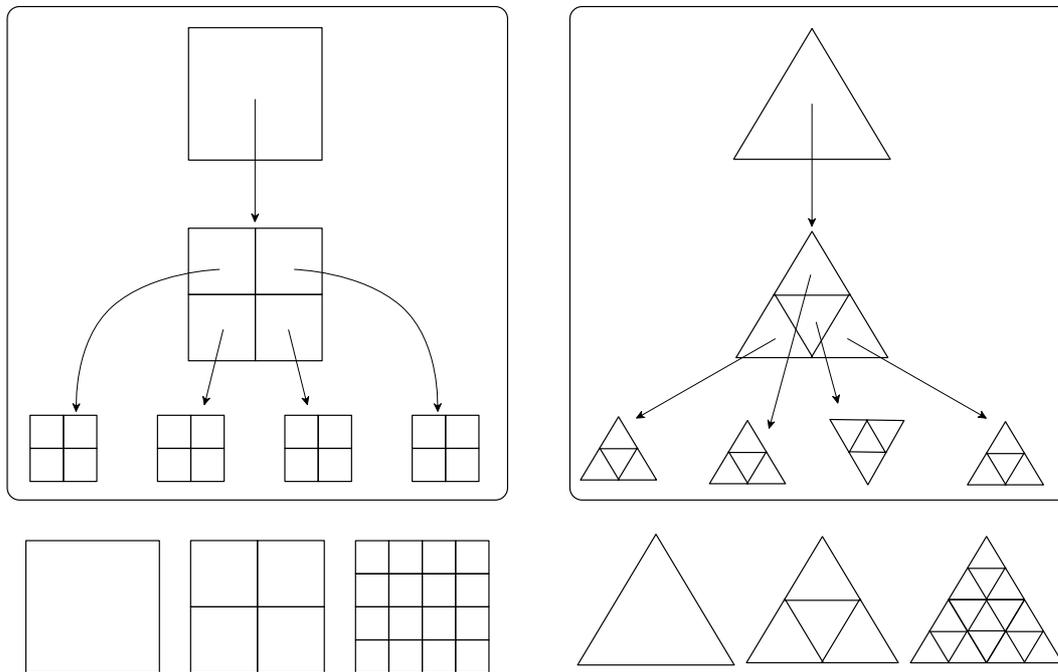
la surface. Ce problème est dû au fait que dans cette structure, toutes les entités topologiques du maillage ne sont pas représentées, et seules les faces sont maintenues. Si la représentation topologique était complète le fait de découper l'arête commune à la face  $f$  lors de l'insertion des nouvelles faces aurait un effet sur aussi sur  $f$ . En effet, celle-ci serait alors devenue une face à cinq côtés, et aucun trou topologique ne serait apparu dans le maillage.

La solution proposée dans [DM02] consiste à regrouper les nœuds de modification non-conformes afin d'obtenir un graphe composé uniquement de nœuds effectuant des modifications conformes. Dans notre exemple, les nœuds 1 et 2 seraient alors regroupés en un seul. Cette solution, bien que résolvant le problème posé, réduit cependant l'ensemble des maillages atteignables par coupe dans la structure.

### 1.3.4 Quadrees

La structure d'arbre quaternaire ou quadtree [Sam84, Sam90] peut être vue comme un cas particulier du modèle précédent dans lequel la seule opération de raffinement consiste à remplacer une seule face par un ensemble de quatre faces. Dans ce cadre, le graphe acyclique orienté décrivant l'ordre partiel entre les opérations de mise-à-jour du maillage est réduit à un arbre, ou plus exactement une forêt d'arbres, chaque face du maillage de départ  $M^0$  étant la racine d'un de ces arbres.

L'unicité de l'opération de mise-à-jour du maillage restreint l'utilisation de



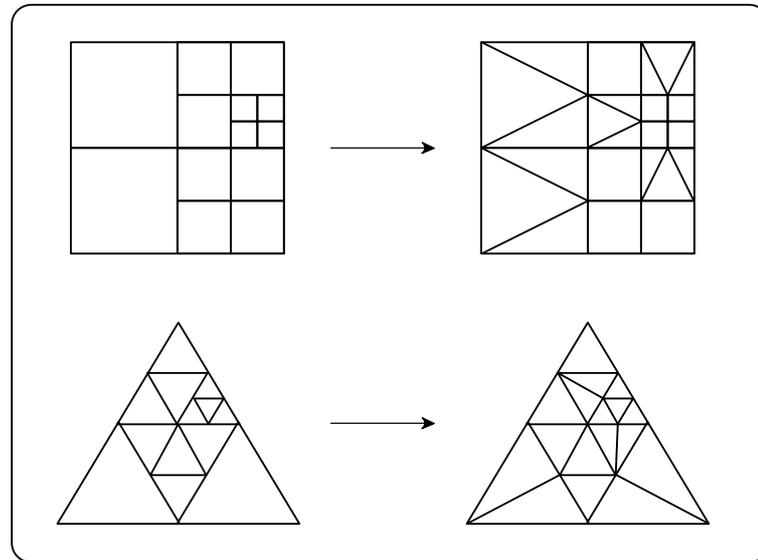
**Figure 1.27** - Exemples de quadtrees de faces à quatre et à trois côtés

ce modèle à l'approche "grossier vers fin". En effet, la condition pour pouvoir construire une hiérarchie de mises-à-jour aussi régulières sur un maillage fin de départ serait que ce maillage soit lui-même le résultat de la quadrisection successive des faces d'un maillage grossier. Nous reviendrons sur ce point dans le chapitre suivant concernant les surfaces de subdivision.

Il existe deux types d'arbres quaternaires distingués par le type de faces qu'ils représentent. La figure 1.27 illustre à gauche un quadtree construit par quadrisection de faces à quatre côtés, et à droite un quadtree construit par quadrisection de triangles.

Ici non plus, il n'existe pas de représentation topologique complète liant entre elles toutes les cellules du maillage. Les problèmes de conformité du maillage extrait sont donc également à résoudre. Si l'on considère la solution précédente consistant à regrouper les mises-à-jour donnant naissance à des maillages non-conformes, la diversité des maillages que l'on peut extraire se trouve alors fortement réduite. En effet, toutes les opérations de quadrisection d'un même niveau doivent alors être regroupées et seuls les maillages illustrés sous les arbres dans la figure 1.27, dont toutes les faces sont issues du même niveau, sont accessibles.

Afin d'accéder à des maillages dont la résolution est adaptative, une méthode



**Figure 1.28** - Exemples de maillages issus de restricted quadtrees et leurs maillages conformes correspondants

consiste à utiliser ce qui est alors appelé un quadtree restreint ou "restricted quadtree" [HB87, Paj98]. On s'autorise alors à extraire des maillages non-conformes sous la condition que toutes les feuilles adjacentes du sous-arbre correspondant à ce maillage ont au plus un niveau de différence. Les maillages de gauche de la figure 1.28 illustrent des exemples de maillages extraits d'un restricted quadtree de faces à quatre et trois côtés.

Comme précédemment, les arêtes n'étant pas représentées dans la structure, des trous topologiques apparaissent aux frontières entre les zones de niveaux différents. De nouvelles faces doivent alors être insérées dans le maillage afin de rendre ce dernier conforme. Les faces voisines ayant au maximum un seul niveau de différence, les faces sont ajoutées suivant un nombre réduit de motifs. Les figures de droite de la figure 1.28 montrent les maillages conformes obtenus après ajout des faces supplémentaires.

Les relations d'adjacence entre les faces ne sont pas directement représentées dans un quadtree, mais la régularité du raffinement effectué permet de calculer cette information de voisinage. Au sein d'un même groupe de quatre faces, le voisinage entre celles-ci est trivial. Dans le cas général, étant donné le sous-arbre correspondant à un maillage extrait d'un quadtree, les requêtes de voisinage entre les faces de ce maillage (les feuilles du sous-arbre) sont résolues en remontant dans l'arbre jusqu'au parent commun et en redescendant du côté de l'arête concernée par la requête d'adjacence. Ces requêtes d'adjacence sont résolues en un temps étant au pire logarithmique fonction de la taille de l'arbre.

Dans le cadre d'un quadtree complet dans lequel toutes les faces sont raffinées d'un niveau au suivant, des techniques ont été développées pour résoudre les adjacences entre faces d'un même niveau en temps constant. Les feuilles de l'arbre, dont le nombre est une puissance de 4, sont stockées dans un tableau, et les adjacences sont résolues en utilisant des opérations arithmétiques sur les indices. On parle de quadtree linéaire. Ces techniques ont été développées aussi bien pour les quadtrees de faces carrées [Sch92] que pour les quadtrees de triangles [LS00].

### 1.3.5 Pyramides de cartes

Les pyramides de cartes [BK03a, BK03b, BK06] se placent à l'origine dans l'approche "fin vers grossier", mais peuvent également être utilisées dans l'approche "grossier vers fin". Ce sont des structures topologiques multi-échelles définies à partir des cartes combinatoires que nous avons présentées dans la section précédente. Les applications de ce modèle sont généralement tournées vers l'analyse et la segmentation multi-échelles d'images discrètes. Dans ce cadre, une carte combinatoire initiale est construite à partir d'une image en associant une face de la carte à chaque pixel de l'image. Le processus de segmentation consiste à fusionner des faces voisines de la carte afin de regrouper en régions des pixels présentant une certaine similarité selon un critère donné. Une pyramide de cartes va permettre de représenter à chacun de ses niveaux une segmentation dont les régions sont de plus en plus larges, allant du premier niveau où chaque région ne contient qu'un pixel, au dernier niveau pouvant aller jusqu'à n'être composé plus que d'une face englobant toute l'image.

Plus précisément, une pyramide de cartes est définie comme une succession de cartes combinatoires de dimension 2 obtenues par l'application d'opérations de contractions et de suppressions d'arêtes. Ces opérations sont encodées dans ce qui est appelé des noyaux de contraction ou de suppression. Ces derniers sont définis de sorte qu'ils interdisent des modifications telles que la déconnexion d'une carte en deux composantes connexes séparées.

De manière pratique, étant donné la carte décrivant un niveau de segmentation, le niveau suivant (présentant moins de faces et donc des régions plus larges) est construit en deux étapes : un noyau de contractions d'arêtes est calculé dans le but de fusionner des régions voisines, puis un noyau de suppression est calculé afin d'éliminer les arêtes dégénérées pouvant être apparues lors de l'application des contractions.

De nombreuses informations peuvent ensuite être calculées à partir de cette

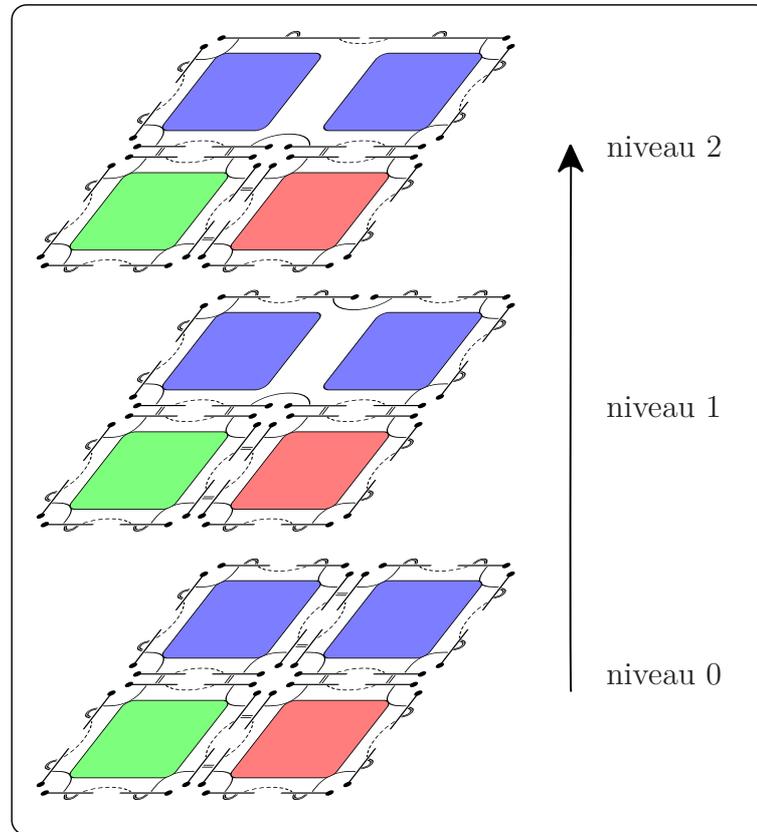


Figure 1.29 - Exemples de pyramide de 2-G-cartes

hiérarchie. Par exemple, l'ensemble des pixels qui composent une région d'un niveau donné de la segmentation peut être retrouvé en calculant l'ensemble des faces du niveau initial qui ont été fusionnées afin de former la face délimitant la région concernée.

La définition des opérations de contraction et de suppression de cellules au sein des cartes généralisées de dimension quelconque [DL03] a permis la définition des pyramides de cartes généralisées en dimension quelconque [GSDL06]. Ce modèle a été appliqué à la segmentation multi-échelles d'images discrètes 3D. Chaque niveau de segmentation est ici représenté par une 3-G-carte, obtenue par l'application d'opérations de contraction et de suppression de cellules sur la 3-G-carte de niveau précédent.

Formellement, une pyramide  $P$  de  $n$ -G-cartes composée de  $m + 1$  niveaux est l'ensemble  $P = \{G^a\}_{0 \leq a \leq m}$  où pour tout  $a$  tel que  $0 \leq a \leq m$ ,  $G^a = (B^a, \alpha_0^a, \dots, \alpha_n^a)$  est une  $n$ -G-carte. A chaque niveau  $a$  sont définis  $C_i^a$  et  $S_i^a$ , les ensembles de  $i$ -cellules de niveau  $a$  devant être contractées (respectivement supprimées) afin de d'obtenir la  $n$ -G-carte décrivant le niveau  $a + 1$ . Ces ensem-

bles de cellules sont construits en respectant un certain nombre de contraintes notamment d'exclusion mutuelle.

La figure 1.29 illustre un exemple de pyramide de 2-G-cartes. Le niveau 0 est construit à partir d'une image discrète de taille 2x2 et est composé de quatre faces. Les deux faces dont le pixel correspondant est de même couleur sont fusionnées au niveau 1 par la suppression de leur arête commune. Le sommet de valence 2 créé lors de cette opération est supprimé dans la 2-G-carte de niveau 2.

Plusieurs approches ont été proposées pour la représentation d'un tel modèle, appelées respectivement explicite, hiérarchique et implicite. Ces trois représentations sont équivalentes relativement à l'information qu'elles représentent, mais diffèrent par leurs complexités à la fois spatiale et temporelle. Aux deux extrêmes on trouve d'une part une représentation qui stocke séparément chacune des n-G-cartes correspondant à chacune des échelles de la segmentation, et d'autre part une représentation qui ne stocke que le niveau de départ conjointement à toute l'information nécessaire à la reconstruction de tous les niveaux de segmentation.

## 1.4 Conclusion

Comme on l'a vu dans la première section de ce chapitre, le cadre des cartes combinatoires présente de nombreux avantages en modélisation géométrique. Ce formalisme simple faisant usage d'une seule entité munie de relations permet de représenter la topologie de la subdivision cellulaire de quasi-variétés en dimension quelconque. En plus de la généralité du domaine de représentation, le fait de n'avoir qu'une seule entité à maintenir permet une mise-en-œuvre pratique simple au sein de structures de données peu gourmandes en mémoire relativement à l'information contenue, et dont les opérateurs de manipulation de base n'ont qu'un nombre restreint de contraintes à respecter afin de garantir des propriétés sur les objets modélisés. Les requêtes de voisinage, quelles que soit les cellules qu'elles mettent en jeu, sont résolues de manière optimale. Tous les algorithmes mettant en jeu dans leurs calculs des informations contenues dans un ensemble de cellules voisines peuvent donc tirer partie de cette efficacité.

Dans la deuxième section de ce chapitre, nous nous sommes intéressés aux modèles existants pour la représentation de maillages multirésolution. La plupart d'entre eux sont basés sur la représentation conjointe d'un maillage à l'aide d'une structure "monorésolution" et d'un ensemble d'opérations de mise-à-jour

dont les dépendances mutuelles sont représentées par un graphe.

Une première caractéristique de ces modèles est que l'accès à un niveau de résolution de la hiérarchie de maillages représentée se fait nécessairement par les applications successives des opérations de mise-à-jour correspondantes au sein de la structure représentant le maillage courant. Dans des applications mettant en œuvre des mises-à-jour dynamiques de la géométrie impliquant un parcours de l'ensemble des niveaux, il est souhaitable de pouvoir économiser l'application de ces opérations sur le support topologique. De plus, des accès simultanés à plusieurs niveaux impliquent la maintenance de plusieurs maillages courants, multipliant alors les coûts.

Si la régularité des opérations de mise-à-jour le permet, comme dans le cadre de la quadrisection de faces et des structures à base de quadrees, ces défauts sont pour la plupart évités. En effet, les relations entre les cellules sont représentées implicitement dans la structure et peuvent être résolues grâce à des parcours spécifiques de la hiérarchie. Il est alors possible d'accéder à une ou plusieurs versions du maillage sans avoir à reconstruire les adjacences entre les cellules.

Cependant, de manière plus générale, les structures utilisées pour la représentation du ou des maillages courants ne sont pas les plus adaptées à une utilisation générale dans le cadre de la multirésolution. Le plus souvent, ces structures sont soit limitées à la représentation de maillages composés uniquement d'un seul type de faces (maillages purement triangulaires ou quadrangulaires pour les quadrees), ou des structures ne représentant qu'un sous-ensemble des cellules de l'objet (typiquement les faces pour les modèles surfaciques). Or, lorsqu'on accède à un maillage intermédiaire au sein d'une hiérarchie multirésolution, nombreux sont les cas où pour maintenir l'intégrité du maillage, le nombre de côté des faces est amené à évoluer suivant les niveaux auxquels sont accédées ses cellules voisines. La rigidité due à l'absence de gestion de l'ensemble des cellules de la subdivision génère alors dans ces structures des maillages non-conformes, comme on l'a vu dans le cadre des Multi-Triangulation ou des quadrees. Les solutions adoptées dans ces structures ne résolvent pas fondamentalement le problème, mais proposent de rajouter des faces temporairement tout en réduisant l'éventail de maillages accessibles au sein de la structure.

Enfin, l'absence de représentation de l'ensemble des cellules de la subdivision et de leurs relations d'incidence et d'adjacence ne permet pas une résolution optimale des requêtes de voisinage au sein des maillages extraits de la structure. Or, dans de nombreux outils s'appuyant sur une représentation multirésolution sous-jacente, des algorithmes de traitement de la géométrie ont besoin de traverser des voisinages donnés afin de récupérer l'information nécessaire au calcul.

Au regard de ces défauts et des avantages des cartes combinatoires que l'on vient d'énumérer, il apparaît clairement qu'un modèle multirésolution issu de ces dernières doit avoir la capacité de proposer nombre de solutions afin de pallier les limitations rencontrées dans les structures multirésolution existantes.

Les pyramides de G-cartes proposent d'utiliser les cartes combinatoires pour décrire une hiérarchie de maillages. Cependant, bien qu'étant bien adaptée au cadre de la segmentation d'images dans une approche "fin vers grossier", leur définition trop rigide liée à l'utilisation d'opérateurs spécifiques de contraction et de suppression de cellules ne permettent pas de les utiliser dans un cadre plus général indépendant des opérations de raffinement appliquées au maillage entre les niveaux de résolution. De plus, l'utilisation des G-cartes, modèle très général dont le domaine de représentation inclut les quasi-variétés non-orientables, implique des besoins en mémoire deux fois plus importants par rapport aux cartes, plus spécialisées et limitées à la représentation de variétés orientables. Dans nombre d'applications classiques en informatique graphique, les objets modélisés sont des variétés orientables. Afin de pouvoir être compétitif en terme de coût mémoire avec les structures existantes, le choix des G-cartes peut alors se révéler pénalisant.

---

# EXTENSION MULTIRÉSOLUTION DES CARTES COMBINATOIRES

---

## Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>47</b>
<b>2.2</b>	<b>Extension multirésolution</b>	<b>48</b>
2.2.1	Ensembles de brins	48
2.2.2	Relations indexées	49
2.2.3	Hypercarte multirésolution	51
2.2.4	2-carte multirésolution	52
<b>2.3</b>	<b>Opérateurs de manipulation</b>	<b>53</b>
2.3.1	Opérateurs de base	53
2.3.2	Opérateurs de haut niveau	55
<b>2.4</b>	<b>Mise-en-œuvre pratique</b>	<b>62</b>
2.4.1	Implantation des 2-cartes	63
2.4.2	Implantation des 2-cartes multirésolution	66

---

## 2.1 Introduction

Notre objectif est de concevoir un modèle capable de représenter et de manipuler efficacement n'importe quel maillage multirésolution quels que soient sa dimension, le type de cellules dont il est composé ou la manière dont il est construit. Comme nous l'avons vu au chapitre précédent, les cartes combinatoires, définies en dimension  $n$ , permettent de représenter des maillages quelconques et fournissent un accès optimal à l'information topologique de la subdivision de l'espace représentée. Prenant ce modèle comme point de départ, nous définissons une extension multirésolution tout aussi générale et efficace, quel que soit

le niveau de résolution considéré. La cohérence et l'efficacité induite par l'utilisation des cartes va nous permettre de proposer des solutions aux défauts que l'on a énuméré concernant les structures multirésolution existantes.

Ce chapitre est consacré à la définition de notre extension multirésolution des cartes combinatoires. Un des objectifs étant de définir un modèle indépendant de la dimension ou du type d'objet représenté (orientable ou non, ouvert ou fermé, variété ou non), nous nous plaçons pour commencer dans le cadre des hypercartes de dimension  $n$ . Nous définissons alors tout d'abord notre extension multirésolution à ce niveau. Une fois cette première étape franchie, le principe général de l'extension peut être retrouvé dans tous les modèles de carte, en réinjectant les contraintes spécifiques à chaque modèle.

Nous nous sommes par la suite principalement intéressés aux cartes en dimension 2. Nous détaillerons donc pour ces modèles un certain nombre d'opérateurs nécessaires à leur manipulation, et nous donnerons un aperçu de ce que peut être une implantation pratique de ces modèles.

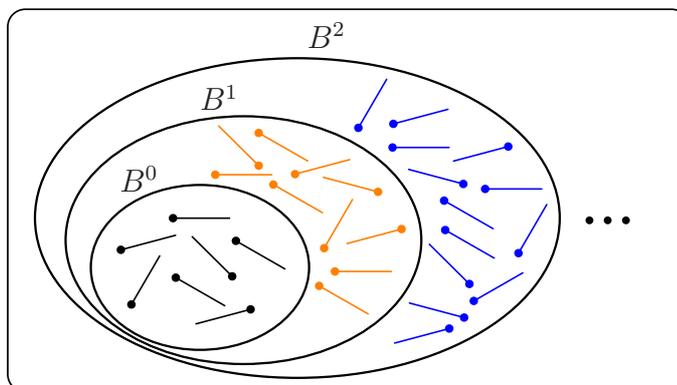
## 2.2 Extension multirésolution

### 2.2.1 Ensembles de brins

Un modèle multirésolution classique décrit une succession de maillages du plus grossier au plus fin. Entre chaque maillage, des cellules sont ajoutées par l'application de différents opérateurs (subdivision de face, éclatement de sommet, ...) entraînant des modifications locales des liaisons topologiques entre les cellules.

L'idée centrale de notre modèle est que ce principe peut être obtenu à partir d'une carte combinatoire à laquelle on ajoute progressivement des brins. Les brins sont conservés d'un niveau au suivant et leurs relations peuvent rester inchangées ou être modifiées. Cette propriété nous permettra par la suite de montrer que notre modèle reste peu gourmand en espace mémoire.

Dans une hypercarte classique, les brins sont contenus dans un unique ensemble et décrivent une seule subdivision en cellules d'un objet. Dans une hypercarte multirésolution, on veut décrire un ensemble de subdivision de plus en plus raffinées d'un même objet. Les brins sont alors organisés en une série d'ensembles imbriqués les uns dans les autres. La subdivision correspondant à chaque niveau de résolution  $i$  est décrite par un ensemble de brins appelé  $B^i$



**Figure 2.1** - Ensembles de brins imbriqués dans une carte multirésolution

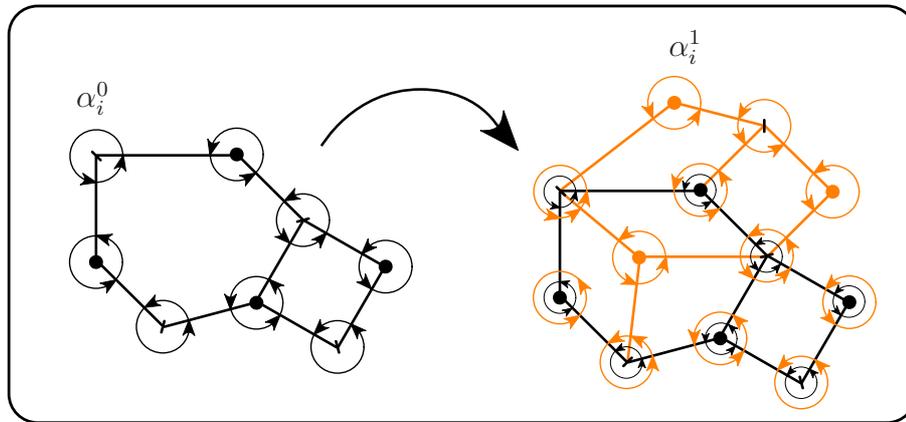
dont le cardinal croît de façon monotone avec les niveaux de résolutions. On a  $B^0 \subset B^1 \subset B^2 \dots$ . Autrement dit, une fois un brin inséré dans la carte à un niveau  $i$  donné, il fait partie intégrante des ensembles de brins décrivant toutes les cartes de niveaux supérieurs.

A chaque niveau de résolution, un nouvel ensemble de brins est inséré dans la carte. On définit  $N^i$  comme l'ensemble de brins insérés dans la carte au niveau de résolution  $i$ . On a, pour tout  $i > 0$ ,  $N^i = B^i \setminus B^{i-1}$ . Les brins décrivant l'objet au niveau 0 sont contenus dans l'ensemble de brins  $B^0 = N^0$ . Afin de décrire l'objet au niveau 1, on ajoute de nouveaux brins à l'ensemble  $B^0$ . Ces nouveaux brins forment l'ensemble  $N^1$ . A chaque niveau de résolution  $i$  un nouvel ensemble de brins  $N^i$  est ajouté aux brins déjà présents afin de pouvoir décrire l'objet de manière de plus en plus détaillée. On peut également définir les ensembles  $B^i$  de la façon suivante :  $B^i = \bigcup_{0 \leq j \leq i} N^j$ .

La figure 2.1 illustre ces ensembles de brins imbriqués. Les brins noirs, respectivement oranges et bleus, sont ceux de  $N^0$ , respectivement  $N^1$  et  $N^2$ .

### 2.2.2 Relations indexées

Au niveau 0, les brins de l'ensemble  $B^0$  (ceux de l'ensemble  $N^0$ ) sont liés entre eux par des permutations  $\alpha_i$  comme dans une hypercarte classique. Plaçons nous maintenant au niveau 1 et considérons les brins de l'ensemble  $B^1$  (composé des brins de  $N^0$  et de ceux de  $N^1$ ). On souhaite que les brins de  $N^1$  que l'on vient de rajouter puissent s'insérer dans des permutations qui existaient déjà au niveau 0 entre les brins de  $N^0$ . En effet, dans le cas contraire, cela signifierait que les brins que l'on ajoute à chaque niveau de résolution sont toujours totalement déconnectés des brins préexistants, ce qui doit rester fais-



**Figure 2.2** - Indexation des relations dans une carte multirésolution primale

able, mais ne doit pas être obligatoire. Dans le même temps, on ne veut pas perdre l'information concernant les permutations qui existaient au niveau 0 entre les brins de  $N^0$ .

Une solution consiste à définir l'ensemble des permutations  $\alpha_i$  à chaque niveau. On note alors  $\alpha_i^0$  la permutation  $\alpha_i$  liant les brins de  $B^0$ . Au niveau 1, c'est une autre permutation, notée  $\alpha_i^1$  qui lie les brins de  $B^1$ . D'une manière générale, on note  $\alpha_i^l$  la permutation  $\alpha_i$  agissant sur les brins de l'ensemble  $B^l$ . On voit apparaître ici le fait qu'un brin dispose d'une relation différente par la permutation  $\alpha_i$  pour chaque niveau où il existe, c'est-à-dire chaque niveau supérieur ou égal au niveau auquel il a été inséré dans la carte.

La figure 2.2 montre un exemple de carte multirésolution à deux niveaux consécutifs. Au niveau 0, les brins sont liés par des permutations appelées  $\alpha_i^0$ . Au niveau 1, les brins de  $N^1$  ont été rajoutés et les brins de l'ensemble  $B^1$  sont maintenant liés par les permutations  $\alpha_i^1$ . Les permutations  $\alpha_i^0$  sont ici reportées en plus petit pour signifier qu'elles n'ont pas disparu dans l'opération.

Différents cas sont visibles dans cet exemple. Certaines permutations de niveau 1 sont égales à ce qu'elles étaient au niveau précédent, c'est-à-dire qu'aucun nouveau brin n'y a été inséré. D'autres permutations lient uniquement des brins nouvellement insérés. Enfin, des nouveaux brins de  $N^1$  ont été insérés dans des permutations qui existaient entre des brins de  $N^0$  au niveau 0.

### 2.2.3 Hypercarte multirésolution

Formellement, on définit une hypercarte multirésolution de dimension  $n$  comme un  $(n + 1)$ -uplet

$$H_{MR} = \left( \{B^i\}_{i \geq 0}, \{\alpha_0^i\}_{i \geq 0}, \dots, \{\alpha_{n-1}^i\}_{i \geq 0} \right)$$

tel que  $B^0 \subset B^1 \subset B^2 \dots$  et pour tout  $i \geq 0$ , le  $(n + 1)$ -uplet

$$H^i = (B^i, \alpha_0^i, \dots, \alpha_{n-1}^i)$$

est l'hypercarte qui décrit l'objet au niveau de résolution  $i$ .

Jusqu'à présent, aucune contrainte n'a été définie sur les permutations *entre les niveaux*, et donc aucune cohérence ni lien logique n'existe entre les hypercartes décrivant les différents niveaux de résolution. En effet, les liaisons existant entre les brins de l'ensemble  $N^0$  par la permutation  $\alpha_i^0$  ne laissent présager en rien de leurs liaisons au niveau suivant par la permutation  $\alpha_i^1$ . Rien n'empêche les relations entre les brins d'être totalement différentes d'un niveau au suivant.

Les seules contraintes présentes dans le modèle à ce niveau afin d'autoriser la généralité souhaitée tout en garantissant la cohérence du modèle et une certaine efficacité sont au nombre de deux. La première impose à l'hypercarte de niveau  $i$  de contenir un nombre de brins supérieur ou égal au nombre de brins présents dans l'hypercarte de niveau  $i - 1$ . Rien n'interdit à un niveau de représenter un objet, et au niveau suivant un objet tout à fait différent, pour peu qu'il soit plus fin et nécessite plus de brins pour sa représentation. Simplement, plus les relations sont égales d'un niveau au suivant, plus grande sera la cohérence et la similarité des objets entre les niveaux.

La deuxième contrainte est que le  $(n + 1)$ -uplet  $H^i$  défini à chaque niveau de résolution est bien une hypercarte conforme à la définition que l'on en a donné à la section précédente. La liberté laissée aux opérateurs manipulant la topologie dans un niveau et entre les niveaux n'est donc pas totale. Ils doivent au minimum garantir l'intégrité de la carte correspondant à chaque niveau.

Nous faisons le choix de laisser ici la définition de notre modèle dans cet état totalement libre de liens logiques entre les niveaux. La cohérence sera introduite par la suite par les opérateurs topologiques qui vont servir à la construction de la hiérarchie des objets multirésolution. Nous pensons en effet que ce n'est pas au modèle d'imposer des contraintes fortes *a priori* sur les objets représentables et la manière de les construire. Ainsi ce modèle se destine à des domaines d'application divers qui utilisent des modes de construction variés, introduisant chacun leur propre cohérence entre les niveaux à travers les opérateurs de construction utilisés.

### 2.2.4 2-carte multirésolution

Comme nous l'avons dit plus haut, une fois le principe de l'extension multirésolution défini dans le cadre des hypercartes, on peut le retrouver pour chacun des modèles exprimables dans ce cadre en réinjectant les contraintes spécifiques à chaque modèle. Ainsi, on retrouve la définition des 2-cartes en partant des hypercartes de dimension 2 et en contraignant l'une des permutations en une involution sans point fixe.

On déduit alors qu'une 2-carte multirésolution primale est un triplet

$$C_{MR} = \left( \{B^i\}_{i \geq 0}, \{\beta_0^i\}_{i \geq 0}, \{\beta_1^i\}_{i \geq 0} \right)$$

tel que  $B^0 \subset B^1 \subset B^2 \dots$  et pour tout  $i \geq 0$ , le triplet

$$C^i = (B^i, \beta_0^i, \beta_1^i)$$

est la 2-carte primale qui décrit l'objet au niveau de résolution  $i$ .

Tout comme pour les cartes standards, on peut exprimer la version duale d'une 2-carte multirésolution de la façon suivante :

$$C_{MR} = \left( B, \{\phi_1^i\}_{i \geq 0}, \{\phi_2^i\}_{i \geq 0} \right)$$

tel que pour tout  $i \geq 0$ , le triplet

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

est la 2-carte duale qui décrit l'objet au niveau de résolution  $i$ .

La contrainte forçant chacun des triplets  $C^i$  à être une 2-carte valide garantit que chacun des niveaux de résolution représentés par la 2-carte multirésolution est bien la subdivision en cellules d'une 2-variété orientable fermée (ou ouverte si on considère la gestion des marqueurs telle que décrite au chapitre 1 pour les cartes classiques). Les opérateurs agissant sur une 2-carte multirésolution doivent donc être définis de manière à toujours respecter cette contrainte d'intégrité.

Les cellules des maillages des différents niveaux sont, comme dans le cadre des 2-cartes classiques, identifiées au sein de chaque niveau de résolution par des ensembles de brins. Ces ensembles sont également construits ici grâce à la notion d'orbite. Si l'on considère le maillage de niveau  $i$ , les cellules de la carte  $C^i$  sont composées d'ensembles de brins de  $B^i$  qui sont parcourus, dans la version primale de la définition, grâce aux orbites suivantes :  $\langle \beta_1^i \rangle$  pour

les sommets,  $\langle \beta_0^i \rangle$  pour les arêtes et  $\langle \beta_1^i \circ \beta_0^i \rangle$  pour les faces. Dans la version duale de la définition, les orbites permettant de parcourir les brins des cellules de la subdivision de niveau  $i$  sont les suivantes :  $\langle \phi_1^i \circ \phi_2^i \rangle$  pour les sommets,  $\langle \phi_2^i \rangle$  pour les arêtes et  $\langle \phi_1^i \rangle$  pour les faces. Toutes ces orbites sont exactement les mêmes que celles définies au chapitre 1 sur les 2-cartes, le paramètre de niveau  $i$  en plus. Cette similitude découle tout naturellement du fait que chacune des cartes  $C^i$  est définie comme une 2-carte classique.

Nous allons illustrer tout cela dans des exemples plus détaillés une fois que nous aurons introduit les opérateurs de base nécessaires à la manipulation d'une 2-carte multirésolution.

## 2.3 Opérateurs de manipulation

Dans cette section, nous introduisons un certain nombre d'opérateurs topologiques pour la manipulation d'une 2-carte multirésolution. Nous commençons par détailler les opérateurs de base pour manipuler les brins et leur relations. Ces derniers vérifient toujours les contraintes d'intégrité des cartes et constituent les briques élémentaires qui vont nous servir ensuite à construire des opérateurs de plus haut niveau. Étant construits à partir des opérateurs de base, ces opérateurs de plus haut niveau garantissent que pour tout  $i$ , les cartes  $C^i$  sont bien des 2-cartes après leur application.

La complexité de ces opérateurs est minimale, compte tenu de la quantité d'information topologique maintenue au sein des modèles de cartes. Il n'y a en effet qu'un seul type d'éléments à manipuler, ce qui en plus de faciliter l'écriture des opérateurs, permet de ne pas avoir de nombreuses structures à mettre à jour afin de garantir l'intégrité de l'objet. Manipuler les coutures des brins en respectant les simples contraintes du modèle suffit à modifier les cellules de la subdivision de manière implicite.

### 2.3.1 Opérateurs de base

Les opérations que nous allons détailler ici concernent la façon dont sont gérées les coutures des brins entre eux ainsi que leur insertion et suppression de la carte. Nous commençons par décrire ces opérateurs dans le cadre des 2-cartes classiques, et montrons ensuite comment on les adapte au sein des 2-cartes multirésolution.

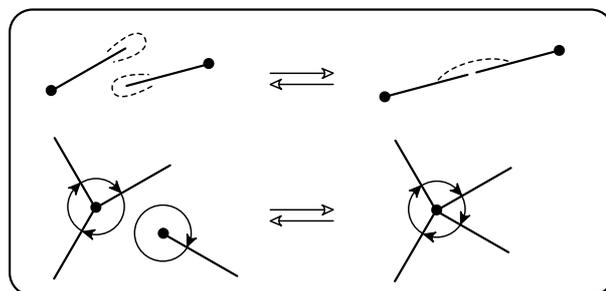


Figure 2.3 - Opérations de couture dans une 2-carte primale

## 2-cartes classiques

Les brins d'une 2-carte peuvent être cousus par deux relations différentes : une involution  $\beta_0$ , et une permutation  $\beta_1$ . Quand un brin n'est pas cousu à un autre brin par une relation, il est alors cousu à lui-même et est un point fixe de cette relation.

L'opérateur de couture de deux brins par  $\beta_0$  prend simplement deux brins  $b_1$  et  $b_2$  et les lie ensemble. L'opérateur inverse effectue la découteure de deux brins. Pour la permutation  $\beta_1$ , on veut s'assurer que le cycle formé par les brins de la permutation est toujours bien un cycle fermé. Ainsi, les opérateurs que l'on va définir sont l'insertion d'un brin dans une permutation et la sortie d'un brin d'une permutation. Ces opérateurs de couture sont illustrés à la figure 2.3.

Afin de garantir le respect des contraintes d'intégrité des 2-cartes, une attention particulière doit être portée à la gestion de la relation  $\beta_0$ . En effet, celle-ci est définie comme une involution sans point fixe. Cela veut dire qu'il ne peut exister dans la carte de brin cousu à lui-même par cette involution comme on le voit dans la figure 2.3. L'insertion des brins dans la carte est donc une opération atomique qui crée un couple de brins et les coud par  $\beta_0$ . La suppression des brins de la carte doit alors également se faire par couple de brins cousus par  $\beta_0$ . L'opérateur de découteure de brins par cette involution devrait donc être inexistant. Cependant, certains opérateurs de haut niveau peuvent avoir besoin de découdre temporairement des brins par  $\beta_0$  avant de les recoudre. Tout comme pour l'insertion de brins, ces opérateurs devront s'assurer d'être atomiques et donc impossibles à interrompre. De la sorte, toute carte accessible respecte bien les contraintes imposées par sa définition.

## 2-cartes multirésolution

Dans une 2-carte multirésolution, les relations entre les brins sont définies à chaque niveau de résolution. Chaque 2-carte  $C^i$  dispose d'une involution  $\beta_0^i$  et d'une permutation  $\beta_1^i$  qui lient les brins de l'ensemble  $B^i$ . Les opérateurs de base de couture et découture sont les mêmes que ceux que l'on vient de décrire pour les 2-cartes.

L'unique différence vient du fait que les brins de  $B^i$  ont été insérés dans la carte à des niveaux différents, inférieurs ou égal à  $i$ . Une opération de couture se fait donc aussi bien entre des brins qui ont été insérés au même niveau qu'entre des brins issus de niveaux différents. Comme nous l'avons déjà illustré à propos des hypercartes multirésolution, si les coutures ne se faisaient qu'entre brins insérés au sein d'un même niveau, toutes les cartes  $C^i$  seraient totalement déconnectées les unes des autres.

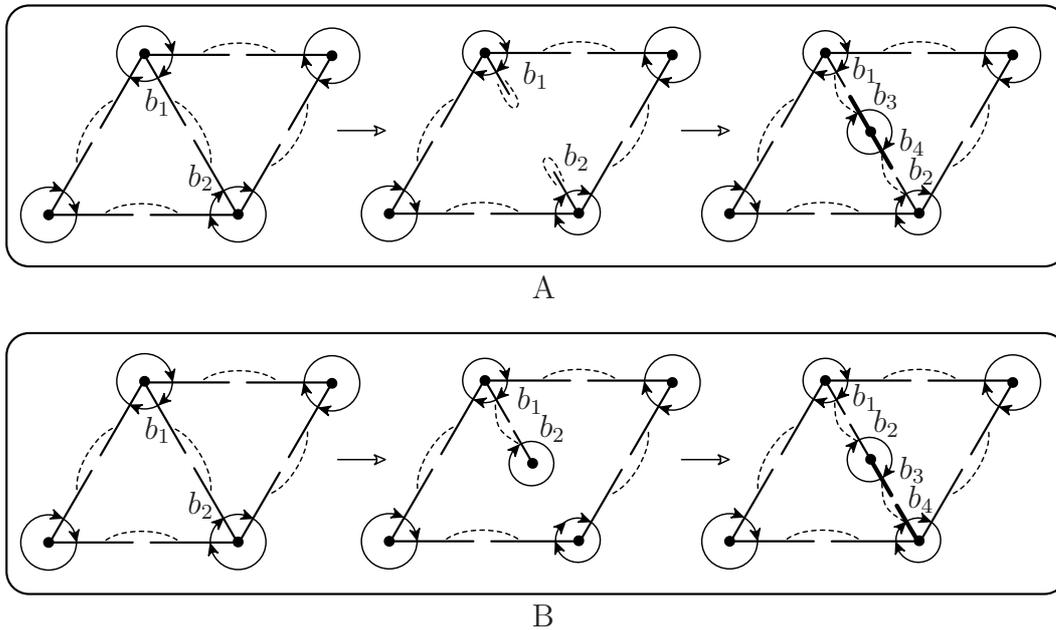
Nous allons illustrer cela avec des exemples dans la suite avec la description d'opérateurs de plus haut niveau, que l'on construit en combinant ces opérateurs d'insertion et de couture de brins.

### 2.3.2 Opérateurs de haut niveau

Dans un souci de simplicité de lecture, nous allons tout d'abord illustrer ces opérateurs comme s'ils étaient simplement appliqués dans une 2-carte classique. Nous montrerons ensuite dans des exemples plus complets la façon dont ces opérateurs peuvent être exécutés à cheval sur des niveaux de résolution différents afin de lier les niveaux ensemble.

## Caractéristique d'Euler

Tous les opérateurs classiquement utilisés en modélisation géométrique peuvent être formulés dans le formalisme des cartes, et *a fortiori* les opérateurs d'Euler et leurs combinaisons, ainsi nommés car leur application sur la décomposition cellulaire d'une variété conserve la caractéristique d'Euler-Poincaré de la variété. Cette caractéristique est un invariant topologique indépendant de la décomposition cellulaire, qui, pour une variété fermée, vaut  $\chi = c_0 - c_1 + c_2 + \dots + (-1)^n c_n$  où  $c_i$  est le nombre de  $i$ -cellules présents dans la décomposition. Par exemple, quelle que soit la décomposition en cellules que l'on fera d'une sphère, sa caractéristique d'Euler-Poincaré sera toujours égale à 2.



**Figure 2.4** - Insertion / Suppression d'un sommet dans une arête dans une 2-carte primale

### Insertion de sommet

L'insertion d'un sommet dans une arête consiste à couper une arête existante en deux en insérant un nouveau sommet au milieu. Cet opérateur est illustré pour le cas primal dans la figure 2.4. Avant l'application de l'opérateur, les deux brins  $b_1$  et  $b_2$  sont cousus par l'involution  $\beta_0$ . Une manière d'effectuer l'opération, illustrée à la figure 2.4-A découd ces deux brins et insère deux nouveaux brins  $b_3$  et  $b_4$ . Ces derniers sont cousus ensemble par la permutation  $\beta_1$  et aux brins  $b_1$  et  $b_2$  par l'involution  $\beta_0$ .

Cette manière d'exécuter cette opération d'insertion de sommet passe par une phase où des brins sont cousus à eux-mêmes par  $\beta_0$ . Il existe une autre manière d'effectuer l'opération, illustrée à la figure 2.4-B, qui évite ce passage par le point fixe. Elle consiste à laisser les brins  $b_1$  et  $b_2$  cousus par  $\beta_0$ , à sortir  $b_2$  de sa permutation  $\beta_1$ . On introduit ensuite dans la carte les brins  $b_3$  et  $b_4$  cousus par  $\beta_0$ , puis on insère  $b_3$  dans la permutation  $\beta_1$  de  $b_2$  et  $b_4$  dans celle du sommet où était initialement le brin  $b_2$  (en-bas à droite).

L'opération inverse consiste à supprimer un sommet de valence deux du maillage. Il suffit pour cela, selon la première méthode, de supprimer de la carte les deux brins qui composent ce sommet et de recoudre par  $\beta_0$  les deux brins  $b_1$  et  $b_2$  laissés orphelins pour cette involution lors de la suppression. Selon la

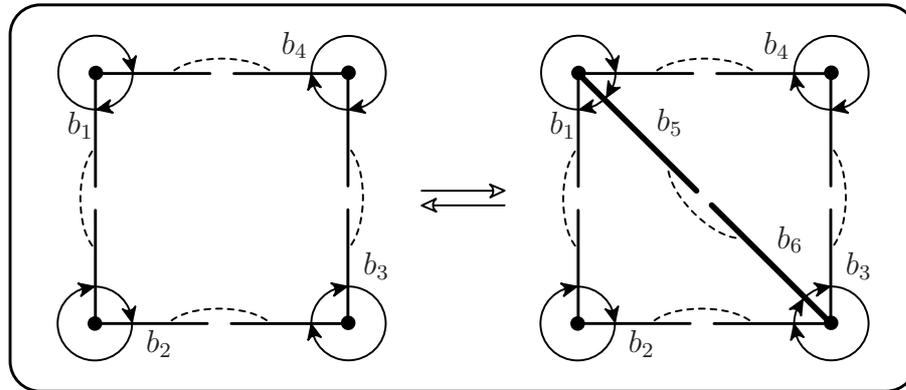


Figure 2.5 - Coupe / Fusion de faces dans une 2-carte primale

seconde méthode, on commence par sortir  $b_3$  et  $b_4$  de leur permutation  $\beta_1$ , on supprime ces deux brins de la carte, et on réinsère  $b_2$  dans la permutation du sommet où était initialement  $b_4$ .

Dans une 2-carte classique, ces deux méthodes sont pratiquement équivalentes et les cartes résultantes sont identiques à un isomorphisme près. Dans une 2-carte multirésolution, suivant les niveaux de résolution où les brins sont introduits, les effets peuvent être différents. Nous détaillerons cela plus tard dans des exemples plus complets en multirésolution.

### Insertion d'arête

Couper une face en deux revient à insérer une arête entre deux sommets dans cette face. On s'intéresse ici uniquement à l'aspect topologique de l'opération et on ne tient pas compte des problèmes géométriques qui peuvent apparaître si la face considérée n'est pas convexe ou si elle est constituée de moins de quatre côtés. La figure 2.5 illustre l'application de cet opérateur sur une face à quatre côtés composée des brins  $b_1, b_2, b_3$  et  $b_4$ . Deux nouveaux brins  $b_5$  et  $b_6$ , cousus par l'involution  $\beta_0$ , sont introduits dans la carte. Ils sont ensuite insérés dans les permutations  $\beta_1$  des deux sommets concernés par l'opération, c'est-à-dire ici les sommets des brins  $b_1$  et  $b_3$ .

L'opération inverse consiste ici à fusionner deux faces adjacentes du maillage en supprimant l'arête commune à ces deux faces. Il suffit ici de sortir les deux brins de l'arête à supprimer des permutations  $\beta_1$  auxquelles elles appartiennent, puis de supprimer ces deux brins de la carte.

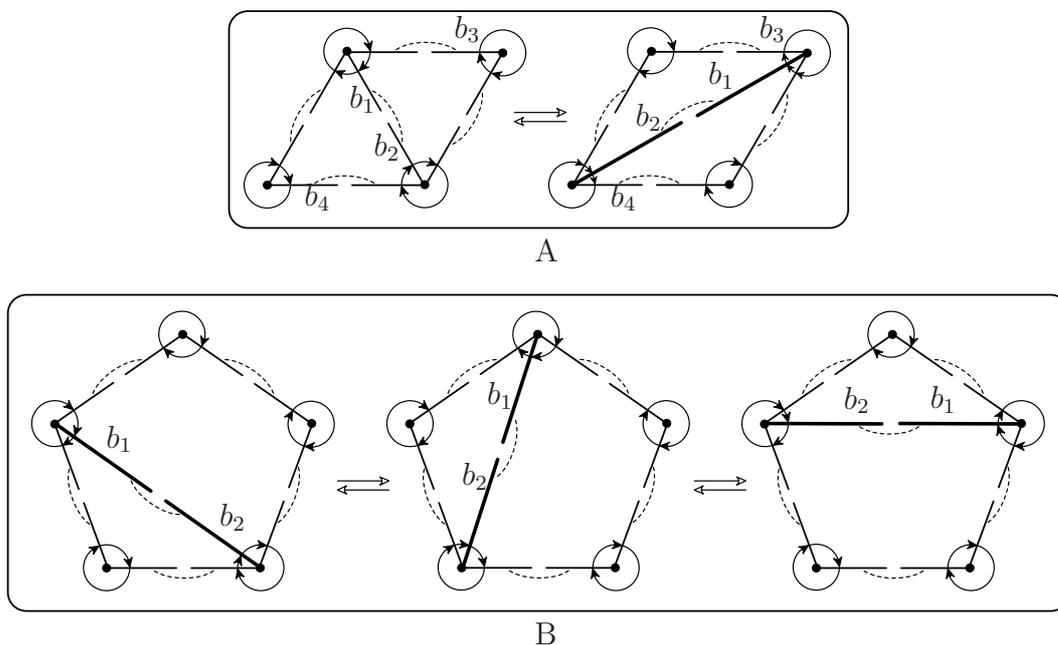


Figure 2.6 - Basculement d'arête dans une 2-carte primale

### Basculement d'arête

L'opération de basculement d'arête est généralement utilisée dans des maillages triangulaires. La figure 2.6-A illustre le basculement de l'arête composée des brins  $b_1$  et  $b_2$  au sein d'un maillage triangulaire. Ces deux brins sont sortis de leur permutation  $\beta_1$  puis insérés dans celle du sommet précédent dans la face. En partant d'un brin donné, le brin dans sa face appartenant au sommet précédent est atteint en parcourant la face en sens inverse, c'est-à-dire en suivant l'orbite  $\langle \beta_0 \circ \beta_{-1} \rangle$ , où  $\beta_{-1}$  est l'inverse de la permutation  $\beta_1$ . A partir des brins  $b_1$  et  $b_2$  on arrive donc ici sur les brins  $b_3$  et  $b_4$ . C'est donc dans la permutation  $\beta_1$  de ces brins que  $b_1$  et  $b_2$  vont être insérés pour opérer le basculement.

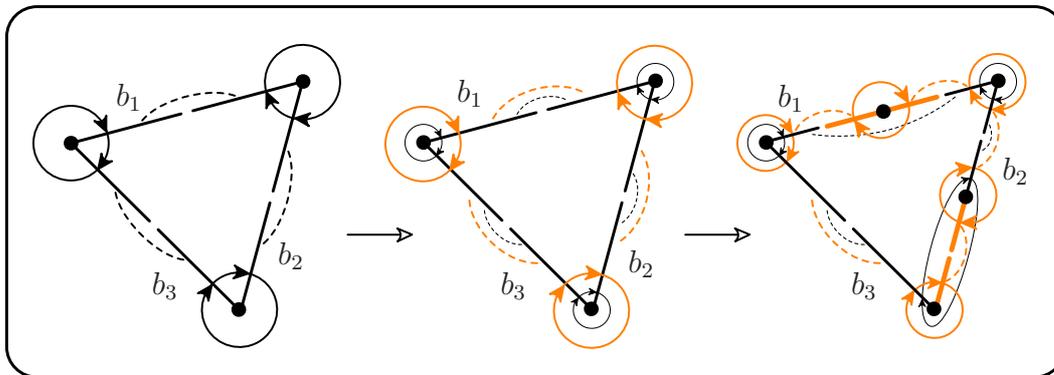
Dans un maillage triangulaire, cet opérateur et son inverse ont un effet équivalent, de même que faire basculer l'arête vers les sommets précédents est équivalent à la faire basculer vers les sommets suivants. Dans un maillage quelconque cette distinction a cependant du sens, tel qu'on peut le voir à la figure 2.6-B. De gauche à droite, l'opérateur est appliqué normalement, et l'opérateur inverse est appliqué de droite à gauche.

## Cadre multirésolution

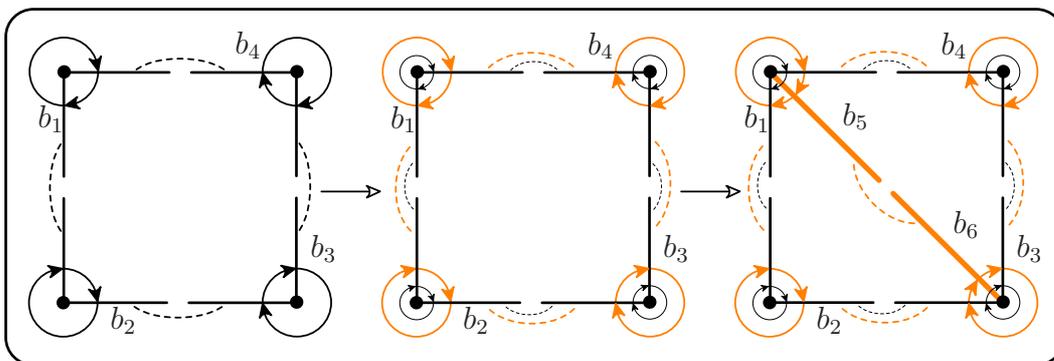
On va commencer ici à intégrer de la cohérence entre les niveaux par l'utilisation d'un nombre restreint d'opérateurs dont l'effet est connu. Si l'on veut garantir que l'objet que l'on décrit conserve sa caractéristique d'Euler-Poincaré de niveau en niveau, la création d'un nouveau niveau va se passer en deux temps. Tout d'abord, on commence par définir la carte correspondant au nouveau niveau de résolution  $i + 1$  comme étant égale à celle de niveau précédent  $i$ . C'est-à-dire que l'on a  $N^{i+1} = \emptyset$  (aucun nouveau brin n'a encore été inséré à ce niveau) et pour tout brin  $b \in B^{i+1}$ ,  $\beta_0^{i+1}(b) = \beta_0^i(b)$  et  $\beta_1^{i+1}(b) = \beta_1^i(b)$ . Ensuite, sur cette carte de niveau  $i + 1$ , on va appliquer des opérateurs d'Euler. Quand ceux-ci impliquent l'insertion de nouveaux brins, ces derniers sont insérés dans l'ensemble  $N^{i+1}$ , et toutes les modifications de couture se font sur les relations de niveau  $i + 1$ . On s'assure ainsi que de niveau en niveau, quelles que soient les opérations que l'on effectue sur l'objet, celui-ci conserve sa caractéristique d'Euler-Poincaré. Bien sûr, cette garantie n'est pas une limitation imposée par le modèle et on peut tout à fait imaginer des opérateurs qui modifient cette caractéristique entre les niveaux (création de tunnels, d'anses, ...). Nous nous focalisons néanmoins ici sur les opérateurs qui conservent cet invariant topologique sur les objets manipulés.

Le premier exemple, illustré à la figure 2.7, montre la différence, dans le cadre des cartes multirésolution, entre les deux méthodes d'insertion de sommet présentées précédemment. On voit ici deux niveaux consécutifs d'une 2-carte multirésolution, les brins noirs étant ceux de  $B^i$ , les oranges ceux de  $N^{i+1}$  (insérés au niveau  $i + 1$ ), et les relations noires et oranges étant celles de niveau  $i$  et  $i + 1$ . L'étape du milieu montre le niveau  $i + 1$  après la phase de copie du niveau précédent. Les relations de niveau  $i$  sont répétées en noir sur cette carte pour illustrer le fait que ces dernières ne sont pas perdues par les opérations effectuées entre les deux niveaux.

Sur la figure de droite, l'arête du brin  $b_1$  a été coupée suivant la première méthode, l'arête du brin  $b_2$  selon la seconde méthode et celle du brin  $b_3$  n'a pas été touchée. Les quatre brins créés lors de ces opérations d'insertion de sommet ont été insérés au niveau  $i + 1$ , de même que les coutures et découpures nécessaires ont été effectuées sur les relations de niveau  $i + 1$ . Dans un cas comme dans l'autre, l'arête a bien été coupée et un sommet inséré. La décomposition cellulaire que représente la carte de niveau  $i + 1$  est la même quelle que soit la méthode utilisée. Un des avantages de l'opération en multirésolution est que dans la première méthode de coupure, il n'y a pas à découdre temporairement les brins  $b_1$  et  $\beta_0^i(b_1)$ , leur couture par  $\beta_0$  au niveau  $i$  restant intouchée.



**Figure 2.7** - Opération d'insertion de sommets entre deux niveaux consécutifs d'une 2-carte multirésolution primale

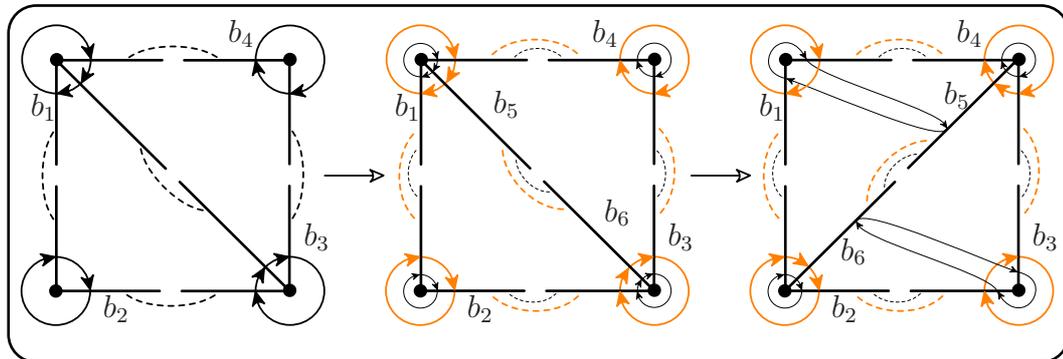


**Figure 2.8** - Opération d'insertion d'arête entre deux niveaux consécutifs d'une 2-carte multirésolution primale

C'est la cohérence entre les cartes des différents niveaux qui est ici différente. Pour le sommet du brin  $b_1$ , les brins de l'orbite sommet aux niveaux  $i$  et  $i + 1$  sont les mêmes. Alors que pour le sommet du brin  $b_3$ , l'orbite sommet au niveau  $i + 1$  est composée d'un ancien et d'un nouveau brin. Nous verrons dans le chapitre 3 l'importance que revêt cette différence lorsque l'on va construire des opérateurs d'encore plus haut niveau à partir de ces opérateurs.

La figure 2.8 montre l'application de l'opérateur d'insertion d'arête entre deux niveaux consécutifs d'une 2-carte multirésolution. L'étape du milieu consiste comme précédemment en la copie au niveau  $i + 1$  de la carte de niveau  $i$ . L'insertion d'arête est ensuite effectuée au niveau  $i + 1$ . Deux nouveaux brins  $b_5$  et  $b_6$  sont introduits dans la carte et insérés dans les permutations  $\beta_1^{i+1}$  des brins  $b_1$  et  $b_3$ . Les permutations  $\beta_1^{i+1}$  des brins  $b_2$  et  $b_4$  sont égales à ce qu'elles étaient au niveau  $i$ .

La figure 2.9 montre l'application de l'opérateur de basculement d'arête en-



**Figure 2.9** - Opération de basculement d'arête entre deux niveaux consécutifs d'une 2-carte multirésolution primale

tre deux niveaux consécutifs d'une 2-carte multirésolution. L'étape du milieu consiste à nouveau en la copie au niveau  $i + 1$  de la carte de niveau  $i$ . Le basculement est ensuite effectué au niveau  $i + 1$ . Aucun nouveau brin n'est inséré, mais les brins  $b_5$  et  $b_6$  sont extraits des permutation  $\beta_1^{i+1}$  de  $b_1$  et  $b_3$  pour être insérés dans celles de  $b_2$  et  $b_4$ . Bien sûr, rien ne change pour les permutations  $\beta_1^i$  qui sont toujours illustrées ici sur la carte de niveau  $i + 1$  pour information.

La figure 2.10 montre un exemple de deux niveaux  $i$  et  $i + 1$  d'une 2-carte multirésolution. L'étape intermédiaire de copie du niveau  $i$  n'est ici pas représentée. Dans cet exemple de nombreux opérateurs ont été appliqués pour construire la carte de niveau  $i + 1$ . L'arête du brin  $b_1$  a été basculée, un sommet a été inséré dans les arêtes des brins  $b_2$  et  $b_3$  en suivant la première des deux méthodes décrites précédemment (les brins  $b_2$  et  $b_3$  ne sont plus indiqués au niveau  $i + 1$  dans un souci de lisibilité). Une nouvelle arête a été insérée entre ces deux nouveaux sommets, découpant l'ancien triangle en un triangle et une face à quatre côtés. La face à quatre côtés de gauche a été découpée en trois triangles par l'insertion de deux arêtes entre l'un des nouveaux sommets et deux anciens sommets. L'insertion d'une nouvelle arête entre deux anciens sommets découpe la face à quatre côtés de droite en deux triangles.

Dans cet exemple, on utilise un certain nombre d'opérateurs de façon arbitraire pour créer un maillage plus raffiné au nouveau niveau de résolution. Nous verrons dans le chapitre 3 comment la combinaison d'opérateurs de façon maîtrisée et ordonnée dans le cadre de l'application de schémas de subdivision permet d'apporter une cohérence forte entre les niveaux de résolution, et ainsi de déduire de nombreuses informations sur les maillages constituant ces différents niveaux.

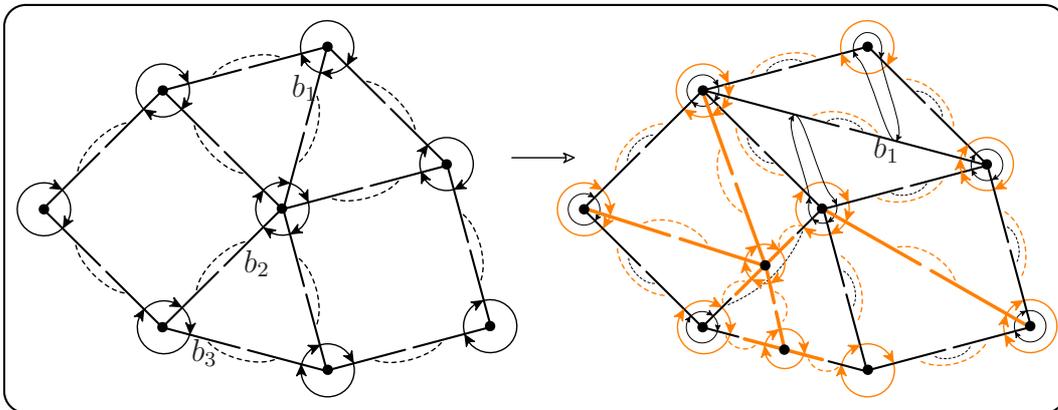


Figure 2.10 - Différentes opérations de couture entre deux niveaux consécutifs d'une 2-carte multirésolution primale

## 2.4 Mise-en-œuvre pratique

Les cartes combinatoires ont déjà été mises en pratique, notamment au sein du modèleur Topofil [Ber98]. Nous avons pour notre part réalisé un prototype de modèleur géométrique multirésolution dont le noyau est l'implantation du modèle des cartes multirésolution, codé en C++ comme une extension du noyau de modèleur générique développé au sein de l'équipe IGG du LSIIT [The07]. Ce dernier fait une utilisation intensive de concepts de programmation disponibles en C++ tels que l'héritage, le polymorphisme et la généricité ainsi que de la *Standard Template Library* (STL). Ces techniques de programmation ont déjà été éprouvées dans le cadre du développement de la structure de demi-arêtes au sein de la bibliothèque CGAL [Ket99].

Dans le modèleur générique de l'équipe IGG, la généricité porte aussi bien sur le modèle topologique utilisé que sur le modèle de plongement qui lui est associé. En effet, ce modèleur permet d'instancier un noyau de base pour n'importe lequel des modèles de cartes que nous avons présentés au chapitre 1 (cartes, G-cartes) et ce en dimension quelconque. De plus les cellules de chaque dimension peuvent ou non être plongées, et ce plongement peut être du type souhaité par le programmeur (géométrique ou autre).

Une telle implantation permet d'éviter autant que possible la réécriture de code, en écrivant par exemple les parcours d'orbites de manière générique, ou en permettant la réutilisation au sein d'un modèle de dimension  $n$ , d'algorithmes déjà écrits pour une dimension inférieure. Le noyau reste pour autant performant, une bonne partie des choix génériques étant réglés lors de la compilation.

Ces problématiques pratiques liées à la programmation n'étant pas centrales ici et à des fins de facilité de lecture, nous allons présenter ici une version très simplifiée dans un pseudo-C++ du noyau de modeleur, spécialisé dans la représentation de 2-cartes primales. Dans un premier temps nous décrivons cette implantation en monorésolution, puis nous montrons comment on peut effectuer son extension afin de gérer la multirésolution.

### 2.4.1 Implantation des 2-cartes

L'objet de base constituant une carte est le brin. Chaque brin doit disposer d'une part des relations topologiques vers les brins auxquels il est cousu, et d'autre part des plongements des différentes cellules auxquelles il appartient. Les relations topologiques entre les brins vont être matérialisées par des pointeurs vers d'autres brins, et les plongements par des pointeurs vers des structures contenant une information de plongement pour chaque dimension à laquelle la carte est plongée. Tous les brins d'une même  $i$ -cellule vont référencer la même structure contenant le plongement de cette  $i$ -cellule.

Une 2-carte est un ensemble de brins munis de leurs relations topologiques et associés à des plongements. Cet ensemble de brins peut être stocké dans une liste. Ce conteneur est plus adapté qu'un tableau dans une application telle qu'un modeleur géométrique où, contrairement à une application de visualisation d'objets, les changements dans la topologie du maillage entraînent de nombreuses opérations d'insertion et de suppression de brins. Contrairement à un tableau, l'insertion et la suppression d'éléments sont exécutées dans une liste en un temps optimal. La contrepartie à cette efficacité est un plus grand partitionnement de la mémoire. L'accès aléatoire à un élément donné est certes lui optimal dans un tableau, mais les accès aux brins de la carte sont le plus souvent effectués au travers de parcours topologiques qui exploitent directement les relations stockées au sein des brins eux-mêmes. Le parcours de la liste de brins elle-même n'est en général effectué que dans les algorithmes où justement l'ensemble des brins doivent être parcourus.

Un exemple basique d'implantation d'une 2-carte primale dont les sommets sont plongés en un point 3D est donné dans le listing 2.4.1. La carte contient une liste d'objets brins. On définit un `Brin` comme étant un itérateur – ou un pointeur – sur un élément de cette liste. C'est ce type d'objet que l'on manipulera dans toutes les fonctions agissant sur les brins. L'objet brin lui-même contient deux de ces pointeurs – un pour chaque relation topologique – et un pointeur vers un plongement de sommet.

---

**Listing 2.4.1** Exemple d'implantation des classes Carte et Brin pour une 2-carte primale dont les sommets sont plongés en un point 3D

---

```

class Carte
{
    list<BrinObj> brins;
    typedef list<BrinObj>::iterator Brin;

    Brin insererBrin();
    void supprimerBrin(Brin b);

    void coutureBeta0(Brin b1, Brin b2);
    void decoutureBeta0(Brin b);
    void insereDansBeta1(Brin b1, Brin b2);
    void sorsDeBeta1(Brin b);

    Brin beta0(Brin b);
    Brin beta1(Brin b);

    void plongeSommet(Brin b, Point3D * p);
}

class BrinObj
{
    Brin beta0;
    Brin beta1;

    Point3D * p;
}

```

---

La carte dispose d'opérateurs permettant d'insérer et de supprimer des brins de la carte, d'opérateurs de couture des brins entre eux ainsi que de fonctions `beta0` et `beta1` permettant de faire des requêtes sur les relations entre les brins. On suppose que quand un brin est inséré, toutes ses liaisons topologiques portent sur lui-même. Cela, ainsi que le fait d'avoir la fonction `decoutureBeta0` à disposition, permet de construire des cartes qui ne vérifient pas les contraintes d'intégrité imposées par la définition. Comme nous l'avons dit précédemment, ces fonctions sont présentes à très bas niveau et ne servent qu'à l'élaboration d'opérateurs de plus haut niveau qui font eux en sorte de respecter les contraintes d'intégrité du modèle. Elles se doivent d'être cachées et interdites à toute autre utilisation.

L'opérateur de plongement de sommet doit parcourir l'orbite sommet du brin passé en paramètre, et pour chaque brin de cette orbite, positionner son plonge-

---

**Listing 2.4.2** Exemple de code pour le parcours et le plongement d'une orbite sommet dans une 2-carte

---

```

void Carte::plongeSommet(Brin b, Point3D * p)
{
    Brin tmp = b;
    do
    {
        tmp->p = p;
        tmp = beta1(tmp);
    } while (tmp != b);
}

```

---

**Listing 2.4.3** Exemple de code pour l'insertion d'une nouvelle arête entre deux sommets dans une 2-carte

---

```

Brin Carte::insererArete(Brin b1, Brin b2)
{
    Brin bb1 = insererBrin();
    Brin bb2 = insererBrin();
    coutureBeta0(bb1, bb2);
    insereDansBeta1(bb1, b1);
    insereDansBeta1(bb2, b2);
    bb1->p = b1->p;
    bb2->p = b2->p;
    return bb1;
}

```

---

ment au point également passé en paramètre. Ce parcours d'orbite peut s'implanter de la manière présentée au listing 2.4.2 : en partant du brin  $b$  des applications successives de la fonction `beta1` qui traverse la permutation  $\beta_1$  permettent de tourner autour du sommet en passant par tous ses brins. Pour chacun des brins, son pointeur de plongement est positionné.

Le listing 2.4.3 montre une façon d'implanter l'insertion d'arête vue précédemment. Cette fonction prend en paramètres les deux brins permettant d'identifier les sommets entre lesquels la nouvelle arête doit être insérée. On suppose que  $b_1$  et  $b_2$  sont bien les brins après lesquels les nouveaux brins doivent être insérés dans les permutations  $\beta_1$ . La fonction commence par insérer deux nouveaux brins dans la carte, puis les lie par l'involution  $\beta_0$ . On insère ensuite ces brins dans les permutations  $\beta_1$  des deux sommets et on positionne le plongement des nouveaux brins avec l'information de plongement déjà contenue dans les sommets existants. Enfin, l'un des deux brins de la nouvelle arête est retourné.

### 2.4.2 Implantation des 2-cartes multirésolution

Chaque brin d'une 2-carte multirésolution est muni de relations topologiques  $\beta_0^i$  et  $\beta_1^i$  pour chaque niveau de résolution  $i$  entre son niveau d'introduction dans la carte, et le niveau maximum courant de la carte. Un brin multirésolution va donc stocker son niveau d'introduction et, pour chaque relation topologique, un tableau de pointeurs vers d'autres brins. Les cases d'indice 0 de ces tableaux contiennent les coutures correspondant au niveau d'introduction du brin. En effet, un brin n'a pas de relations topologiques à des niveaux inférieurs à son niveau d'introduction dans la carte. Soient  $insNiv$  le niveau d'introduction du brin  $b$ , et  $maxNiv$  le niveau maximum courant de la carte multirésolution. Les tableaux de relation du brin  $b$  contiennent chacun  $1 + maxNiv - insNiv$  éléments, et la liaison topologique de niveau  $i$  se trouve dans la case d'indice  $i - insNiv$ .

Pour chaque niveau de résolution où il existe dans la carte (c'est-à-dire supérieur ou égal à son niveau d'introduction), un brin dispose d'une information de plongement. Cela est implanté par un tableau de pointeurs de plongement dont la taille évolue de la même façon que celle des tableaux de liens topologiques.

Une 2-carte multirésolution est composée d'une hiérarchie d'ensembles imbriqués  $B^i$ , que l'on peut exprimer ainsi :  $B^i = \bigcup_{j \leq i} N^j$ , où  $N^j$  est l'ensemble de brins introduits dans la carte au niveau de résolution  $j$ . Au lieu de n'avoir qu'une seule liste contenant tous les brins de la carte, comme dans le cadre monorésolution, nous allons maintenir ici une liste de brin pour chaque ensemble  $N^i$ . Ces listes vont être stockées dans un tableau de listes de brins, dont la case d'indice  $i$  renferme l'ensemble de brins  $N^i$ . Ce tableau a donc autant d'éléments qu'il y a de niveaux de résolution dans la carte.

Le listing 2.4.4 montre comment cette extension peut être réalisée en se basant sur l'implantation monorésolution précédente. Toutes les fonctions d'insertion de brin, de manipulation et d'accès aux coutures ainsi que de plongement de cellules prennent toutes un paramètre supplémentaire précisant le niveau de résolution auquel l'opération concernée doit être effectuée.

Le listing 2.4.5 montre comment est implanté l'accès à la relation  $\beta_0$  d'un brin à un niveau donné. Une précondition à cette fonction est évidemment que le niveau demandé doit être supérieur ou égal au niveau d'insertion du brin. Dans ce listing, on illustre également une manière d'effectuer un traitement sur chacune des faces du maillage au niveau `niv`. Ce maillage étant composé de l'ensemble des brins insérés aux niveaux `niv` et inférieurs, on parcourt l'ensemble des listes de brins contenues dans les cases du tableau dont l'indice est

---

**Listing 2.4.4** Exemple d'implantation des classes Carte et Brin pour une 2-carte multirésolution primale dont les sommets sont plongés en un point 3D

---

```

class CarteMR
{
    vector< list<BrinObjMR> > brins;
    typedef list<BrinObjMR>::iterator Brin;

    BrinMR insererBrin(int niv);
    void supprimerBrin(Brin b);

    void coutureBeta0(Brin b1, Brin b2, int niv);
    void decoutureBeta0(Brin b, int niv);
    void insereDansBeta1(Brin b1, Brin b2, int niv);
    void sorsDeBeta1(Brin b, int niv);

    Brin beta0(Brin b, int niv);
    Brin beta1(Brin b, int niv);

    void plongeSommet(Brin b, Point3D * p, int niv);
}

class BrinObjMR
{
    int insNiv;

    vector<Brin> beta0;
    vector<Brin> beta1;

    vector<Point3D *> p;
}

```

---

inférieur ou égal à `niv`. Pour chacun de ces brins, on considère ses coutures de niveau `niv`.

Afin de ne pas traiter plusieurs fois une même face, on va utiliser un mécanisme de marqueurs. Plusieurs marqueurs peuvent être associés aux brins. Un marqueur étant une information binaire, un bit suffit pour le stocker. Attacher un entier de 32 bits à chaque brin permet la gestion de 32 marqueurs. Dans le cadre de notre parcours de faces, pour chacun des brins considérés, s'il n'est pas encore marqué, on effectue un traitement sur la face de ce brin au niveau `niv`, puis on marque l'ensemble des brins de cette face afin de ne plus les traiter par la suite.

---

**Listing 2.4.5** Exemple d'implantation de l'accès aux liaisons topologiques et au parcours des faces du maillage d'un niveau donné

---

```

Brin Carte::beta0(Brin b, int niv)
{
    assert(niv >= b->insNiv);
    return b->beta0[niv - b->insNiv];
}

// parcours des faces du maillage de niveau niv
for(int i = 0; i <= niv; ++i)
{
    for(Brin b = brins[i].begin(); b != brins[i].end(); ++b)
    {
        if(! estMarque(b))
        {
            faire quelque chose avec la face du brin b au niveau niv;
            marquer les brins de la face du brin b au niveau niv;
        }
    }
}

```

---

Les marqueurs sont utilisés dans de nombreux algorithmes tels que des parcours comme on vient de l'illustrer, ou dans la cadre d'une interface graphique, pour identifier par exemple les brins sélectionnés.

Comme on l'a vu dans la section précédente avec des exemples de construction de 2-cartes multirésolution, une méthode permettant de conserver une certaine cohérence d'un niveau à l'autre consiste à copier la carte de niveau  $i$  au niveau  $i+1$ , puis à appliquer des opérateurs à ce niveau. Cette phase de copie consiste ici à créer une liste vide à la fin du tableau de listes de brins de la carte, puis à dupliquer la dernière case des tableaux de relations de chaque brin, avant d'appliquer des opérateurs topologiques au niveau  $i+1$ .

Or, suivant les opérateurs topologiques utilisés pour la construction des niveaux, certaines coutures ne sont pas modifiées d'un niveau de résolution au suivant. Ainsi, dans la carte de niveau  $maxNiv$ , un brin dont la couture par une relation donnée n'a pas été modifiée depuis le niveau  $lastNiv$  contient  $1 + maxNiv - lastNiv$  fois la même valeur dans les dernières cases de son tableau de coutures. Si l'on effectue les mêmes opérations de raffinement pour de nombreuses cellules d'un objet, alors cela est répété pour un grand nombre de brins.

---

**Listing 2.4.6** Exemple d'implantation de l'accès aux liaisons topologiques d'un niveau donné

---

```

Brin Carte::beta0(Brin b, int niv)
{
    assert(niv >= b->insNiv);
    if( niv >= b->insNiv + b->beta0.size() )
        return b->beta0.back();
    else
        return b->beta0[niv - b->insNiv];
}

```

---

Pour éviter cette redondance inutile de l'information due à la duplication systématique des liens topologiques entre les niveaux et synonyme de perte d'espace mémoire, nous mettons en place un mécanisme simple qui consiste à tronquer le tableau au niveau *lastNiv*. Le tableau de coutures d'un brin est agrandi uniquement dans le cas où un opérateur exécuté au niveau  $i + 1$  modifie la couture existant au niveau précédent.

La procédure de création du niveau  $i + 1$  est modifiée pour ne comporter plus que la phase de création d'une liste vide à la fin du tableau de listes de brins, et les liaisons topologiques des brins ne sont plus dupliquées par défaut.

L'accès aux coutures d'un brin doit être modifié en conséquence. Si la taille du tableau de liens n'augmente plus régulièrement à chaque niveau, la case d'indice  $i - insNiv$  censée contenir le lien de niveau  $i$  n'existera plus nécessairement. On utilise alors l'algorithme suivant : si la case d'indice  $i - insNiv$  existe, prendre le lien présent dans cette case, sinon prendre celui présent dans la case d'indice le plus élevé. Le listing 2.4.6 illustre l'implantation d'un tel accès pour l'involution `beta0`.

De ce mode d'accès aux coutures, on déduit aussi que l'on peut maintenant également accéder aux coutures, et donc à une carte, à n'importe quel niveau supérieur à 0, même s'il n'est pas encore défini. Si une 2-carte multirésolution a été construite jusqu'au niveau 3, l'accès à la 2-carte de niveau 3 permettra effectivement de parcourir la carte de niveau 3, mais l'accès à la 2-carte de niveau 15 parcourra en réalité la même carte. Nous verrons dans la suite quelques cas de figure où le fait de pouvoir interroger la carte à des niveaux supérieurs à son niveau maximum actuel nous sera utile.

Un dernier problème se pose si, pour un brin, une couture est modifiée après avoir été laissée inchangée sur plusieurs niveaux. Par exemple si la couture par  $\beta_0$  du brin  $b$  inséré au niveau 0 n'a pas changé aux niveaux 1 et 2, puis qu'un

opérateur la modifie au niveau 3. Afin de rester cohérent avec le mode d'accès aux coutures que l'on vient de décrire, il faut alors combler les "trous" dans le tableau en dupliquant la couture de niveau 0 dans les cases d'indices 1 et 2 du tableau, afin que l'opérateur modifiant cette couture au niveau 3 puisse positionner celle-ci dans la case d'indice 3. Une autre solution consisterait à non plus utiliser un simple tableau de coutures dans les brins, mais à maintenir une structure plus complexe qui stockerait une couture par intervalle de niveaux donné. Cette solution serait en contrepartie bien plus coûteuse dans les cas d'un maillage multirésolution où les coutures changent à chaque niveau.

Dans le chapitre 3, nous allons appliquer notre modèle à la représentation de maillages multirésolution construits à l'aide d'algorithmes de subdivision. Les opérations topologiques appliquées au maillage par ces algorithmes d'un niveau au suivant présentent une forte régularité. Le problème décrit au paragraphe précédent n'apparaît pas dans ce cadre et on peut alors gérer l'accès aux coutures simplement de la manière décrite au listing 2.4.6 en évitant ainsi une surconsommation de mémoire.

DEUXIÈME PARTIE

---

APPLICATIONS GÉOMÉTRIQUES



---

# SURFACES DE SUBDIVISION MULTIRÉSOLUTION

---

## Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>74</b>
3.1.1	Surfaces de subdivision	74
3.1.2	L'édition multirésolution	78
3.1.3	Obtention d'une surface de subdivision multirésolution	82
3.1.4	Support du maillage multirésolution	84
<b>3.2</b>	<b>Génération des niveaux</b>	<b>85</b>
3.2.1	Génération régulière	85
3.2.2	Génération adaptative	99
3.2.3	Simplification	106
3.2.4	Critère de subdivision	108
3.2.5	Mise à jour du maillage	111
3.2.6	Niveaux de subdivision des cellules	114
<b>3.3</b>	<b>Accès aux niveaux</b>	<b>120</b>
3.3.1	Accès régulier	120
3.3.2	Accès adaptatif	121
3.3.3	Affichage d'une 2-carte adaptative	125
<b>3.4</b>	<b>Schémas de subdivision originaux</b>	<b>129</b>
3.4.1	Le schéma Quad/Triangle	129
3.4.2	Le schéma $\sqrt{3}$	132
<b>3.5</b>	<b>Comparaison</b>	<b>140</b>
3.5.1	Structure	142
3.5.2	Complexité en temps	144
3.5.3	Complexité en espace	147
<b>3.6</b>	<b>Conclusion</b>	<b>150</b>

---

Ce chapitre traite de l'application principale que nous avons étudiée afin d'illustrer les capacités de notre modèle de cartes multirésolution, à savoir les *surfaces de subdivision multirésolution*. Nous commençons par introduire le principe des surfaces de subdivision ainsi que de l'analyse multirésolution qui permet, entre autres applications, l'édition multirésolution de telles surfaces. Nous détaillerons ensuite la manière dont les cartes multirésolution peuvent servir de support efficace pour la représentation et la manipulation de tels objets.

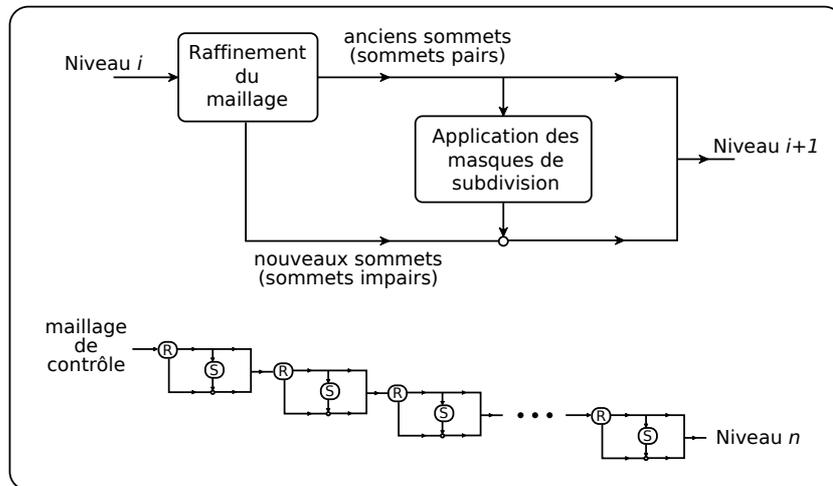
## 3.1 Introduction

### 3.1.1 Surfaces de subdivision

Nous ne prétendons pas donner ici une définition stricte des surfaces de subdivision et des propriétés mathématiques qui sous-tendent la création de ces objets, mais uniquement une introduction intuitive aux mécanismes mis en œuvre dans ces algorithmes.

Au début du premier chapitre, nous avons classifié en deux catégories les modèles de représentation des objets géométriques : ceux qui les représentent de façon continue par des fonctions, et ceux qui en représentent explicitement une discrétisation en cellules. Les surfaces de subdivision jettent un pont entre ces deux catégories en proposant de représenter l'infinité des points d'une surface par la limite d'un processus de raffinements successifs d'une partition cellulaire surfacique de départ, aussi appelé *maillage de contrôle*.

L'un des avantages de cette technique, comparée par exemple aux assemblages de carreaux B-spline surfaciques, est de pouvoir générer et contrôler des surfaces lisses dont la topologie est arbitraire, et ce sans avoir à gérer des problèmes de continuité aux frontières entre les différents patches. Par "topologie arbitraire" on entend ici que le nombre d'anses que présente la surface est indifférent et que la connectivité du maillage de contrôle est indifférente. La surface limite obtenue aura exactement la même topologie que le maillage de contrôle.

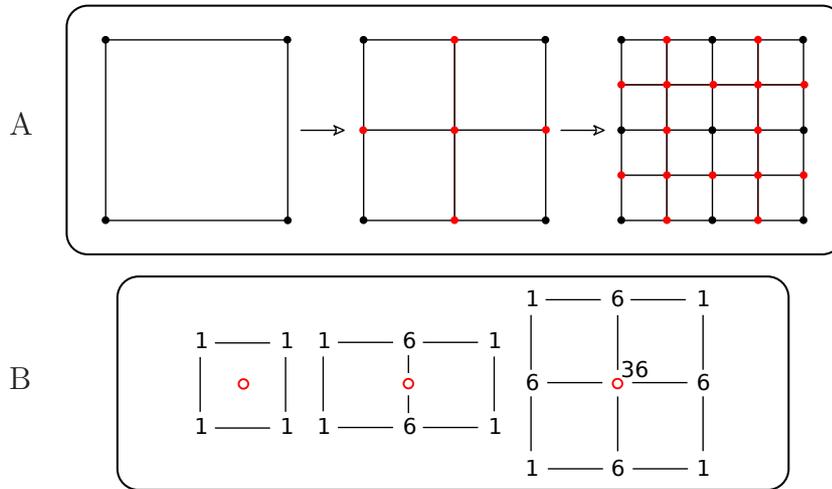


**Figure 3.1** - Diagramme d'exécution d'un schéma de subdivision interpolant

## Obtention

A chaque pas de subdivision, la topologie du maillage est raffinée et la géométrie qui lui est associée est calculée de manière à être plus proche de la surface limite. Dans la pratique, on ne va évidemment pas subdiviser le maillage une infinité de fois, et on arrêtera le processus une fois que la version courante du maillage, c'est-à-dire l'approximation courante de la surface limite, aura atteint la précision souhaitée. On appellera *niveau de résolution* chacun des maillages intermédiaires obtenus durant le processus de subdivision. On peut choisir de numéroter les niveaux de manière croissante à partir du maillage de départ – ou maillage de contrôle – qui porte le numéro 0. Des techniques récentes basées sur les évolutions du matériel graphique proposent de n'envoyer à ce dernier que le maillage de contrôle. La surface sous-jacente est directement calculée et générée sur le matériel graphique [BS05, BS08].

Un *schéma de subdivision* définit les règles selon lesquelles le raffinement est appliqué au maillage à chaque pas de ce processus itératif. On peut distinguer ici deux étapes : d'abord, la topologie du maillage est raffinée, puis de nouvelles informations géométriques sont associées aux sommets du maillage obtenu. Les coordonnées de chaque nouveau sommet sont calculées comme une combinaison affine des coordonnées des sommets appartenant à un voisinage donné dans le maillage d'origine (on parle de *masque de subdivision*). Selon que la surface limite interpole les sommets de départ (schéma *interpolant*) ou non (schéma *approximant*), les anciens sommets conservent ou non leur position initiale. Les schémas approximatifs sont souvent préférés car leurs surfaces limites ont des propriétés de continuité plus intéressantes et que leurs masques

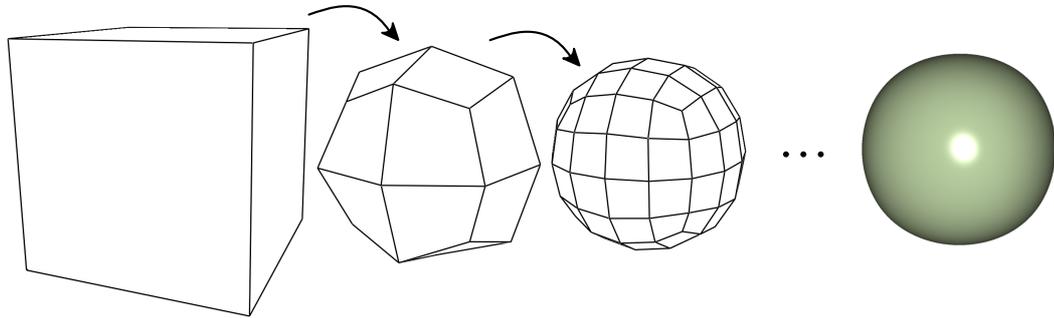


**Figure 3.2** - Subdivision de la topologie du maillage et masques de subdivision dans le schéma de subdivision de Catmull-Clark

de subdivision sont plus compacts et donc plus efficaces à calculer. Les schémas interpolants ont cependant l'avantage propre à leur nature qui est que les sommets du maillage de contrôle sont situés sur la surface limite, ce qui permet de mieux percevoir ce vers quoi va tendre la surface limite et peut éventuellement permettre une manipulation plus intuitive de la surface.

La figure 3.1 donne une vue schématique de l'exécution d'un schéma de subdivision. Pour la simplicité de la présentation, nous choisissons d'illustrer ici le processus avec un schéma interpolant. Le maillage de niveau  $i$  est pris en entrée à gauche, puis, après la phase de raffinement de ce maillage, on considère d'une part les anciens sommets (dits *pairs*) et d'autre part les sommets que l'on vient d'introduire (dits *impairs*) qui n'ont pas encore de position attribuée. La position des sommets impairs est calculée par l'application des masques de subdivision sur les anciens sommets. Dans le cas d'un schéma approximant, la position des anciens sommets devrait également être mise à jour. Finalement, on fusionne les anciens et les nouveaux sommets pour constituer le maillage de niveau  $i + 1$ . Ce processus de raffinement (R) et subdivision (S) est appliqué en cascade jusqu'au niveau  $n$  que l'on souhaite atteindre.

Il existe de nombreux schémas de subdivision qui opèrent sur des maillages constitués de types de cellules différents (triangles, quadrangles, ...), effectuent des opérations de raffinement du maillage différentes et utilisent des voisinages ainsi que des coefficients différents pour le calcul de la géométrie. Ces différents schémas génèrent des surfaces aux propriétés géométriques différentes, notamment en ce qui concerne la continuité de la surface limite.



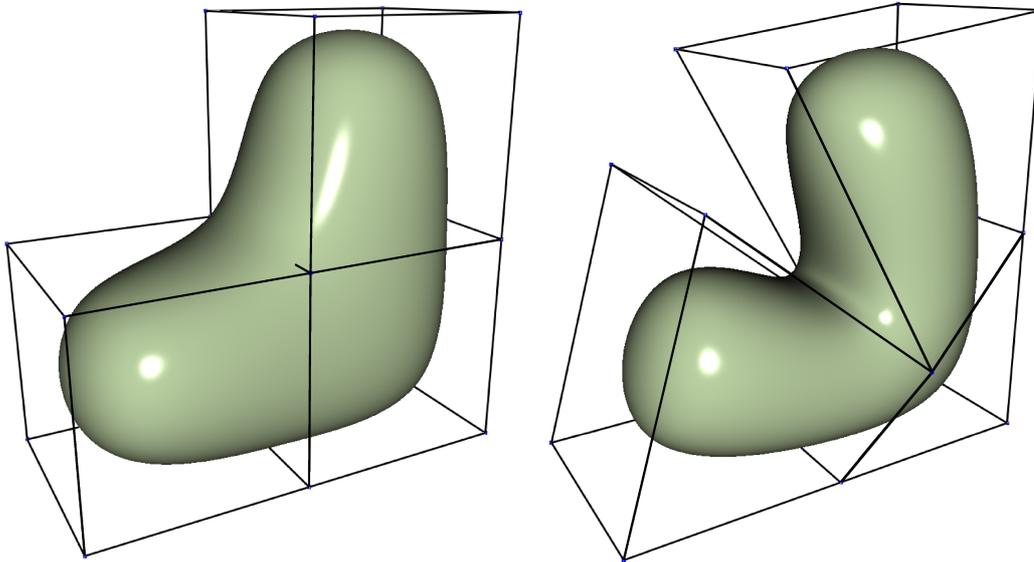
**Figure 3.3** - Premiers pas de subdivision d'un cube et surface limite avec le schéma de Catmull-Clark

Le schéma de Catmull-Clark [CC78] opère une quadrisection de faces sur des maillages quadrangulaires, comme illustré dans la figure 3.2-A. Les sommets impairs sont dessinés en rouge à chaque niveau. A chaque pas de subdivision, les positions des sommets sont calculées suivant les masques de la figure 3.2-B. Ces derniers ont été calculés de manière à ce que le maillage approxime de plus en plus finement et finalement tende vers une surface B-spline bi-cubique de continuité  $C^2$ . La figure 3.3 illustre les deux premiers pas de subdivision d'un cube ainsi que la surface limite obtenue avec cet algorithme.

Des masques particuliers doivent être précisés pour les sommets dits *extraordinaires* qui sont, dans le cas de ce schéma, les sommets ayant une valence différente de 4. Le nombre de ces sommets extraordinaires n'augmente pas avec les pas de subdivision comme nous le verrons plus tard, tous les sommets introduits lors de la quadrisection étant des sommets de valence 4. Nous ne rentrons pas ici dans les détails de ces masques, mais précisons simplement que ces derniers sont calculés de manière à garantir la continuité maximale de la surface limite au niveau de ces sommets. Des masques spécifiques peuvent également être construits de manière à ce que, plutôt que d'opérer un lissage régulier, le processus de subdivision suive des contraintes particulières dans certaines zones du maillage [BLZ00].

## Edition

L'édition d'une surface de subdivision s'effectue par le déplacement des sommets de son maillage de contrôle. Une fois ce maillage modifié, les positions des sommets des niveaux de résolution supérieurs doivent être recalculés afin d'être conformes à la nouvelle géométrie du maillage de contrôle. Les déformations induites à la surface limite par le déplacement d'un sommet du maillage de contrôle dépendent des propriétés du schéma utilisé. La zone d'effet de cette



**Figure 3.4** - Edition d'une surface de subdivision de Catmull-Clark

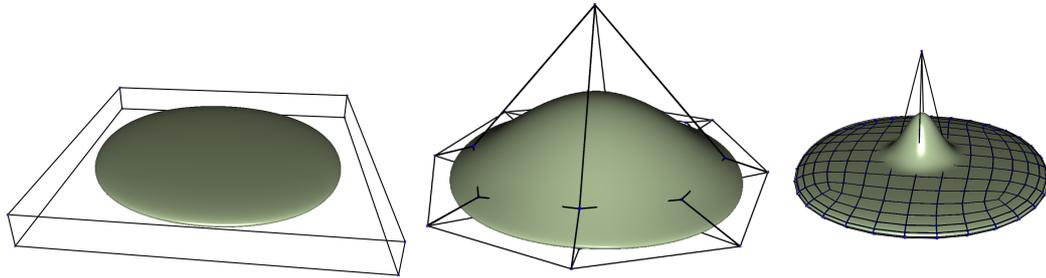
déformation est localisée autour du sommet déplacé, les sommets mis-en-jeu étant dépendants de la taille du voisinage considéré dans la phase de calcul de la géométrie, lors de l'application du masque du schéma utilisé.

La figure 3.4 illustre un exemple d'édition d'une surface de subdivision obtenue avec le schéma de Catmull-Clark. La figure de gauche montre le maillage de contrôle initial ainsi que la surface limite associée. Sur la figure de droite, certains sommets du maillage de contrôle ont été déplacés et la surface limite a été recalculée à partir de ces nouvelles positions.

### 3.1.2 L'édition multirésolution

De manière simplifiée, l'édition multirésolution consiste à considérer tous les niveaux de résolution de la surface de subdivision comme autant de maillages de contrôle potentiels. On veut pouvoir éditer la surface de manière cohérente en déplaçant des sommets au sein de n'importe quel maillage intermédiaire. De prime abord, on va considérer que chaque déplacement de sommets est suivi de la mise à jour des positions des sommets des maillages de niveaux supérieurs au niveau d'édition. Plus le niveau considéré pour l'édition est élevé, plus la déformation de la surface sous-jacente va être localisée.

La figure 3.5 montre des éditions à des niveaux différents sur un objet subdivisé à l'aide du schéma de Catmull-Clark [CC78]. La surface limite est ici à



**Figure 3.5** - Edition multirésolution d'une surface de subdivision de Catmull-Clark

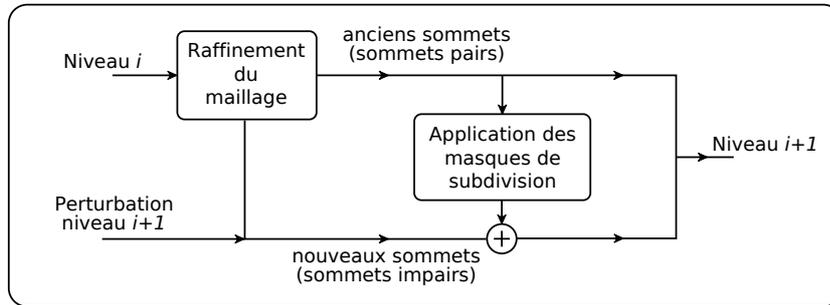
chaque fois dessinée conjointement au maillage de contrôle actuel : à gauche, le maillage de niveau 0, au milieu le maillage de niveau 1 après une édition, et à droite le maillage de niveau 3 dont un sommet a également été déplacé. On voit bien que la zone d'effet de l'édition décroît avec l'augmentation du niveau auquel celle-ci est appliquée.

Considérons le cas où suite à ces éditions aux niveaux 1 et 3, un sommet est déplacé dans le maillage de niveau 0. Cette dernière édition entraîne la mise à jour des niveaux de résolution supérieurs, ce qui inclut les niveaux 1 et 3 précédemment édités. Le recalcul de la géométrie alors effectué va écraser les déformations plus fines précédemment appliquées au niveau 1 et 3 qui seront définitivement perdues. Le résultat sera exactement le même que si on avait directement édité le maillage au niveau 0. Cela est dû au fait qu'il n'existe dans ce que nous avons décrit pour l'instant qu'un lien *ascendant* entre les niveaux permettant la *synthèse* de la géométrie du maillage d'un niveau en fonction de la géométrie du maillage de niveau inférieur.

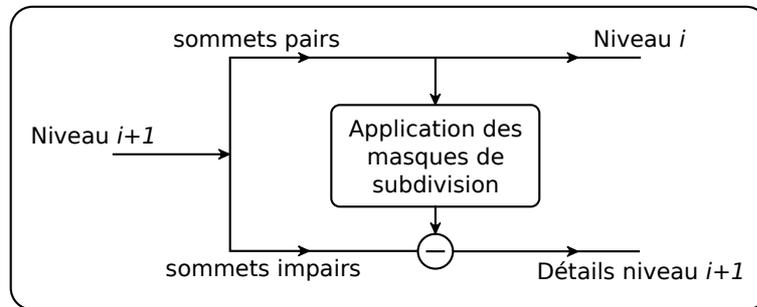
Afin de ne pas perdre les détails ajoutés à la surface aux niveaux de résolutions supérieurs à 0, il faut ajouter un lien *descendant* entre les niveaux. Cela se fait par la création d'une phase dite d'*analyse*, permettant de déduire un maillage de niveau  $i$  à partir du maillage de niveau  $i+1$ , tout en conservant l'information nécessaire à la reconstruction du maillage de niveau  $i+1$  modifié.

En reprenant le diagramme de subdivision de la figure 3.1, et en y ajoutant une légère modification, on peut obtenir le diagramme de la figure 3.6. A chaque pas de subdivision, on se propose d'ajouter une certaine perturbation aux coordonnées calculées pour les sommets impairs par l'application des masques de subdivision sur le maillage de niveau  $i$ . Le maillage de niveau  $i+1$  résultant n'est donc plus égal au résultat direct de la subdivision du maillage de niveau  $i$ , mais à l'addition de cette subdivision et des perturbations de niveau  $i+1$ .

Considérons maintenant le problème en sens inverse. Partant du maillage de



**Figure 3.6** - Diagramme d'exécution d'un schéma de subdivision interpolant modifié pour intégrer des perturbations à chaque niveau

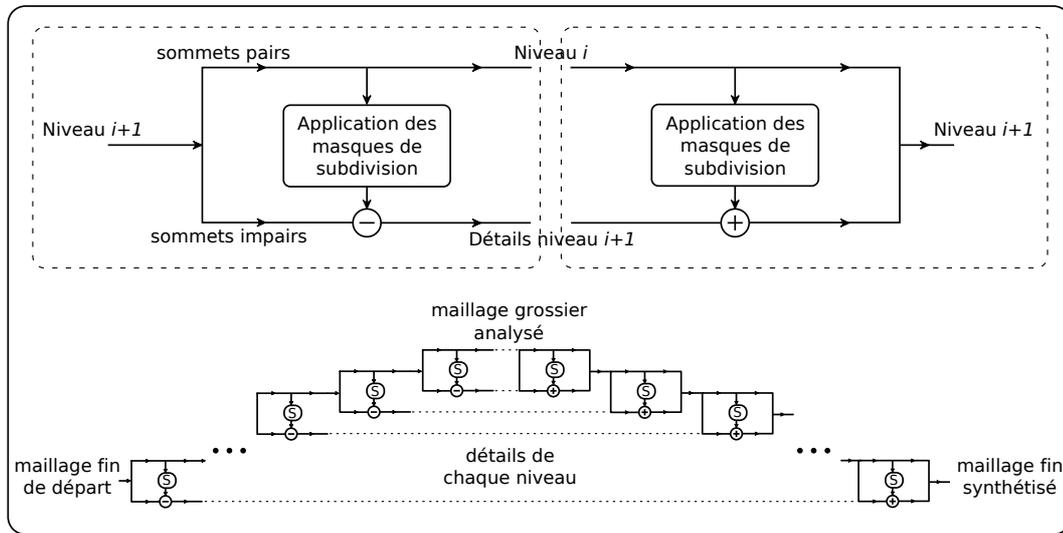


**Figure 3.7** - Diagramme d'exécution de l'analyse associée à un schéma de subdivision interpolant

niveau  $i + 1$ , dont la géométrie n'est pas nécessairement égale au résultat de la subdivision du maillage de niveau  $i$ , on souhaite obtenir ce maillage de niveau  $i$  conjointement avec l'information nécessaire à la reconstruction du niveau  $i + 1$  de départ. Il nous suffit alors d'inverser le diagramme de la figure 3.6 afin d'obtenir celui illustré à la figure 3.7.

Partant du maillage de niveau  $i + 1$  on considère d'une part les sommets pairs et d'autre part les sommets impairs. La connectivité du maillage constitué des sommets pairs va directement former le maillage de niveau  $i$ . L'application des masques de subdivision sur ce maillage de niveau  $i$  va générer des positions pour les sommets impairs du maillage de niveau  $i + 1$  qui ne correspondent pas nécessairement à leurs positions courantes. La différence pour chaque sommet impair entre la position résultant du calcul de la subdivision du maillage de niveau  $i$  et la position réelle de ce sommet constitue ce que l'on appelle alors un *vecteur de détail* qui va être stocké.

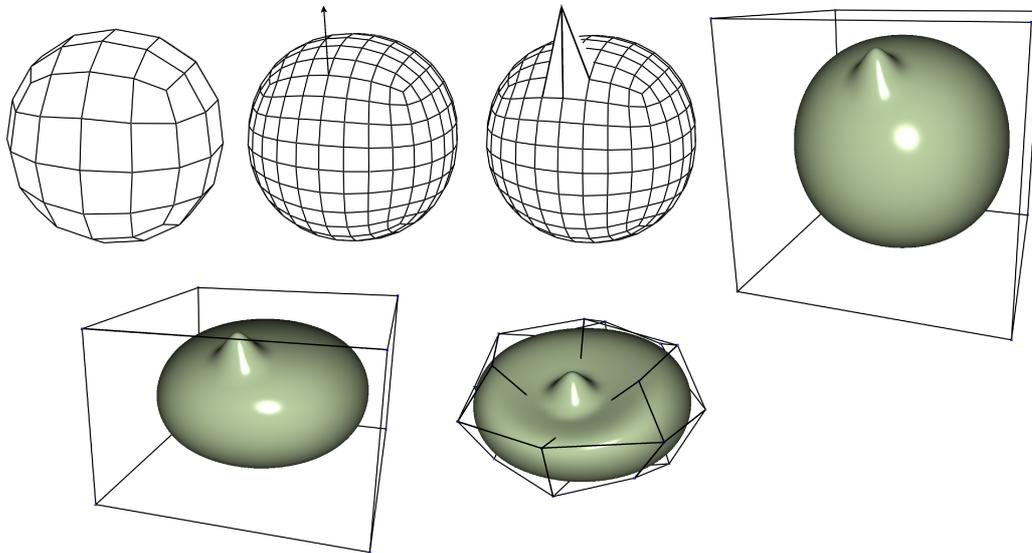
Ce sont ces vecteurs de détail qui vont être ajoutés durant la phase de synthèse en tant que perturbation du résultat de la subdivision. En combinant les deux processus d'analyse et de synthèse, on obtient le diagramme présenté à la figure 3.8. Celui-ci est appliqué en cascade suivant le nombre d'étapes d'analyse et



**Figure 3.8** - Diagramme d'exécution de l'analyse (à gauche) et de la synthèse (à droite) associées à un schéma de subdivision interpolant

de synthèse que l'on souhaite appliquer au maillage. Clairement, si on part d'un maillage de contrôle donné, que l'on applique un certain nombre d'étapes de synthèse, puis que l'on effectue les étapes d'analyse correspondantes, tous les vecteurs de détail générés seront nuls. Ceux-ci n'auront de valeur que si la géométrie associée au maillage a été modifiée à des niveaux de résolution supérieurs.

La figure 3.9 illustre un exemple simple d'édition multirésolution sur une surface de subdivision multirésolution obtenue avec le schéma de Catmull-Clark. Un vecteur de détail est ici ajouté au maillage de niveau 3 qui n'est donc plus égal au résultat de la subdivision du maillage de niveau 2. La surface limite intègre cette déformation de niveau 3. Grâce à ce vecteur de détail, la modification de la géométrie d'un niveau plus grossier entraîne maintenant une déformation qui conserve les déformations plus fines qui ont déjà été ajoutées à la surface. Afin de rendre les déformations de la surface plus intuitives, les vecteurs de détail devront être exprimés dans un repère local au sommet auquel il s'applique plutôt que global relativement à l'espace dans lequel est plongée la surface. Comme on le voit dans les images du bas de la figure 3.9, des éditions aux niveaux inférieurs, ici 0 et 1, conservent cette déformation fine.



**Figure 3.9** - Insertion d'un vecteur de détail dans la hiérarchie multirésolution. Les déformations appliquées par après aux niveaux plus grossiers conservent la déformation fine.

### 3.1.3 Obtention d'une surface de subdivision multirésolution

Cette décomposition d'un maillage en niveaux de résolutions liés par ces processus d'analyse et de synthèse est similaire à la décomposition en ondelettes bien connue en traitement du signal. L'analyse multirésolution de surfaces de subdivision s'assimile à cette méthode dans le cadre des surfaces. Une fois ce mécanisme mis en place, de nombreuses applications deviennent possibles et ont été développées.

L'édition multirésolution [ZSS97, ZSD<sup>+</sup>00, Zor05] que l'on a déjà décrite permet de déformer des objets à différents niveaux de résolution tout en conservant leurs caractéristiques fines. L'édition multirésolution sous contrainte [HSB05, SHB07] propose en plus de conserver certaines propriétés des objets déformés telles que leurs surface ou volume. Ces techniques permettent également de compresser les maillages [KSS00, AG05] sans perte en opérant une quantification et des codages particuliers sur les vecteurs de détail, ou avec perte en appliquant en plus un seuillage sur les vecteurs conservés. Le filtrage de maillages bruités est une application qui s'apparente à cette dernière méthode en supprimant les vecteurs de détails correspondant aux hautes fréquences de l'objet. Enfin, des outils de modélisation géométrique telle que l'approximation d'opération booléennes [BKZ01], le copier-coller de morceaux de surfaces [BMBZ02] ou l'insertion d'arêtes franches [BMZB02] prennent également appui

sur cette représentation.

L'approche que nous avons décrite jusqu'à présent consiste à partir d'un maillage initial grossier, puis à le raffiner et lui ajouter des détails, tout en conservant la possibilité de le déformer à un niveau global. Cette approche est souvent nommée "grossier vers fin". On peut également souhaiter partir d'un maillage fin donné, contenant déjà de nombreux détails, et lui appliquer des déformations multirésolution telles que nous venons de les décrire. Cette approche est nommée "fin vers grossier". On peut alors distinguer principalement deux cas de figure.

Soit le maillage fin donné est *semi-régulier* et dispose d'une connectivité de subdivision, c'est-à-dire telle qu'elle a pu être générée par l'application d'un schéma de subdivision sur un maillage plus grossier de départ. Dans ce cas, il est relativement aisé de remonter topologiquement le fil de la subdivision jusqu'au maillage le plus grossier [Tau02]. Pour ce qui est de la géométrie, on utilise le mécanisme d'analyse que l'on vient de décrire. Dans le cas peu probable où l'objet de départ disposait d'une géométrie proche de celle générée par le schéma de subdivision utilisé, les vecteurs de détail générés seront petits, voire inexistants. Sinon, la phase d'analyse constitue un ensemble de vecteurs de détail rendant possible l'édition multirésolution de l'objet.

La phase de mise à jour des sommets pairs de l'analyse dans le cas approximant doit opérer une inversion des masques de subdivision. Celle-ci n'est pas aisée à obtenir directement. Dans [ZSS97], le filtre approximant de Taubin [Tau95] est utilisé pour calculer la géométrie des sommets du niveau  $i$  à partir du niveau  $i + 1$ . Des techniques plus récentes [LDW97, Ber04] telles que le Lifting Scheme [Swe97], issu de la théorie des ondelettes, permettent d'obtenir une bien meilleure approximation des maillages aux niveaux plus grossiers, minimisant ainsi les vecteurs de détail, et apportant plus de stabilité numérique au système.

La situation est totalement différente dans le cas plus courant où le maillage fin donné dispose d'une connectivité quelconque. En effet, il n'est alors pas possible de remonter directement le cours d'un raffinement topologique correspondant à un schéma de subdivision. La méthode que l'on vient de décrire ne fonctionne plus. Deux cas de figures sont alors à envisager.

Soit on veut absolument conserver le maillage tel qu'il est et on abandonne les techniques d'édition multirésolution basées sur les surfaces de subdivision au profit d'autres méthodes d'analyse multirésolution telles que celle présentées dans [KCVS98] ou [GSS99].

Soit on accepte de perdre le maillage d'origine et on fait subir à l'objet un

prétraitement consistant à le remailler en lui donnant la connectivité de subdivision souhaitée. Un algorithme tel que MAPS [LSS<sup>+</sup>98] rend ce remaillage possible : il consiste à simplifier le maillage fin de départ à l'aide d'opérateurs de contraction d'arête, tout en maintenant une paramétrisation des sommets supprimés au sein des faces grossières. A partir du maillage grossier obtenu, on applique ensuite un schéma de subdivision en reprojétant au mieux les sommets ainsi créés sur la géométrie de l'objet d'origine. Une fois ce remaillage effectué, on peut à nouveau appliquer les techniques basées sur les surfaces de subdivision que l'on vient de décrire.

### 3.1.4 Support du maillage multirésolution

Quelle que soit la méthode par laquelle la surface de subdivision multirésolution est obtenue, les algorithmes de manipulation de surfaces de subdivision multirésolution nécessitent un support efficace pour la représentation des maillages sous-jacents. Pour répondre au mieux aux exigences de telles applications, plusieurs critères doivent être satisfaits.

Premièrement, les différents niveaux de résolution du maillage doivent être accessibles directement et ce de la manière la plus simple qui soit. En effet, les éditions de la géométrie peuvent être effectuées à n'importe quel niveau intermédiaire de la surface multirésolution, ce qui sous-entend de pouvoir déplacer des sommets dans chacun des maillages correspondant à ces niveaux.

Les requêtes d'adjacence doivent être très performantes. L'accès à l'information de voisinage est extrêmement sollicité pendant l'exécution des processus aussi bien de synthèse que d'analyse. En effet comme nous l'avons vu, la position d'un sommet est le résultat de la combinaison affine des positions des sommets appartenant à un voisinage donné. La mise à jour des positions des sommets des maillages des niveaux supérieurs étant effectuée à chaque édition, cette opération se doit d'être la plus efficace possible.

La subdivision adaptative doit être supportée. Celle-ci consiste à raffiner la surface à une profondeur variable selon des zones d'intérêt de l'objet. Cela permet notamment d'économiser de la mémoire en ne générant pas toutes les faces ni tous les sommets qui auraient été calculés si la subdivision avait été appliquée de manière uniforme. Un critère de subdivision est défini et évalué pour chaque cellule afin de déterminer si celle-ci doit ou non être subdivisée vers le niveau suivant. Ce critère peut être la distance à la surface limite si elle est calculable, ou bien encore l'appartenance à une fenêtre de vue ou une zone d'intérêt sur l'objet.

De nombreux schémas de subdivision doivent pouvoir être appliqués. Afin de simplifier le développement d'applications tirant parti d'algorithmes de subdivision différents, un modèle général pouvant représenter des maillages quelconques et permettant de leur appliquer des raffinements quelconques est préférable. En plus de n'avoir à développer qu'un seul noyau de modéleur, cela permet d'utiliser plusieurs schémas au sein d'une même application sans avoir à maintenir plusieurs structures de données. En allant plus loin, une structure présentant une telle généralité permettrait aussi d'utiliser plusieurs schémas au sein d'un même objet.

Les 2-cartes multirésolution que nous avons présentées au chapitre 2 présentent toutes les caractéristiques nécessaires à la satisfaction de ces contraintes. Nous allons montrer dans les prochaines sections comment on peut les utiliser efficacement, et mettre en évidence leurs apports dans le cadre de la représentation de tels maillages multirésolution.

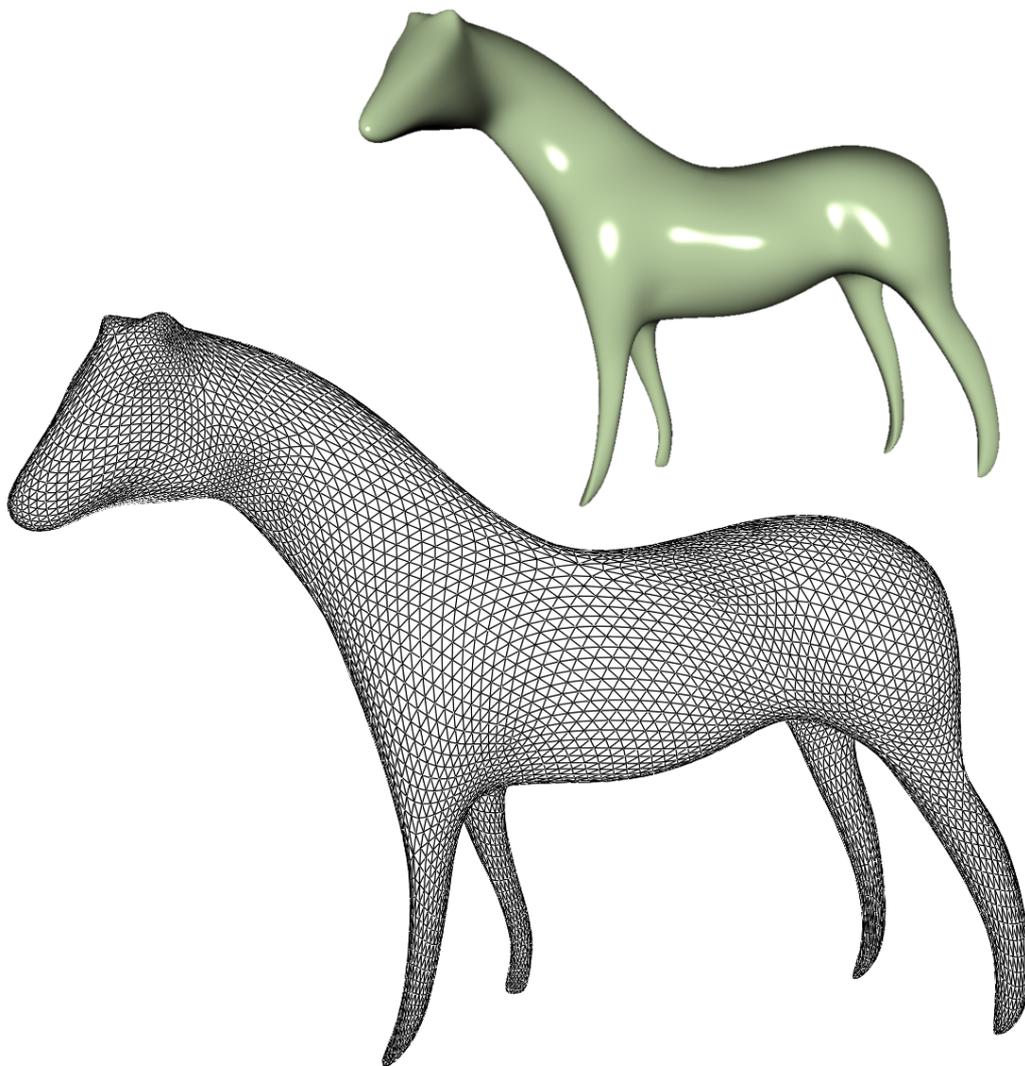
## 3.2 Génération des niveaux

Dans cette section, nous allons détailler comment les niveaux de résolution sont générés au sein d'une 2-carte multirésolution. Il existe, suivant les schémas de subdivision, différentes stratégies pour définir le raffinement topologique à effectuer à chaque pas de subdivision. Elles sont usuellement classifiées en deux catégories appelées raffinement *primal* et *dual*. Les premières opèrent une subdivision des faces du maillage, tandis que les secondes en éclatent les sommets s'assimilant aux méthodes dites de corner-cutting pour les courbes. D'autres méthodes telles que le schéma  $\sqrt{3}$  [Kob00] ou le schéma quad/triangle [SL03, PS04] utilisent d'autres algorithmes de raffinement ou combinent des méthodes existantes. Nous traiterons de ces méthodes dans une prochaine section.

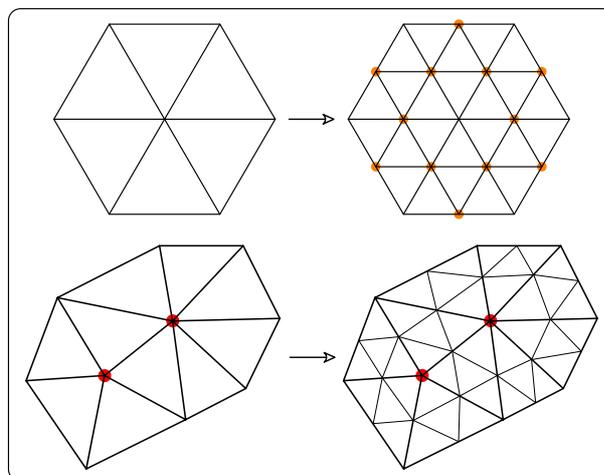
La création des niveaux de résolution peut se faire soit de manière régulière, soit de manière adaptative en fonction d'un critère donné. Nous commençons par détailler l'approche régulière avant de présenter les problèmes propres à la subdivision adaptative.

### 3.2.1 Génération régulière

Dans le cadre d'une génération régulière des niveaux de résolution, toutes les cellules du maillage sont subdivisées d'un niveau au suivant. La figure



**Figure 3.10** - Exemple de maillage obtenu par une subdivision régulières



**Figure 3.11** - Quadrisection de triangles

3.10 illustre un exemple de maillage obtenu en subdivisant régulièrement un maillage triangulaire de départ en utilisant le schéma de subdivision de Loop. Nous allons ici traiter tout d'abord le cas des schémas de subdivision primaux, puis de ceux qui opèrent un raffinement dual.

### Schémas primaux

Au sein des algorithmes de subdivision primaux, il existe deux principaux algorithmes de raffinement. Celui utilisé dans les schémas de Loop [Loo87] et Butterfly [DLG90, ZSS96] et celui utilisé dans le schéma de Catmull-Clark [CC78].

Le premier, illustré à la figure 3.11, opère sur les maillages triangulaires uniquement et effectue une quadrisection de triangle. L'image du haut montre un détail d'un maillage triangulaire régulier où tous les sommets sont de valence 6. Chaque arête du maillage est subdivisée, et les nouveaux sommets ainsi introduits – en orange sur la figure – sont reliés par de nouvelles arêtes. Le cas plus général de maillages où des sommets irréguliers ou extraordinaires existent dans le maillage est illustré par l'image du bas – en rouge sur la figure. La valence de ces sommets est conservée d'un niveau au suivant. En revanche, aucun nouveau sommet extraordinaire n'est introduit durant la subdivision, tous les nouveaux sommets étant de valence 6.

Le second algorithme, illustré à la figure 3.12 opère sur des maillages quadrangulaires et effectue également une quadrisection de faces. Comme on le voit sur l'image du haut, dans le cas régulier, tous les sommets sont de valence 4.

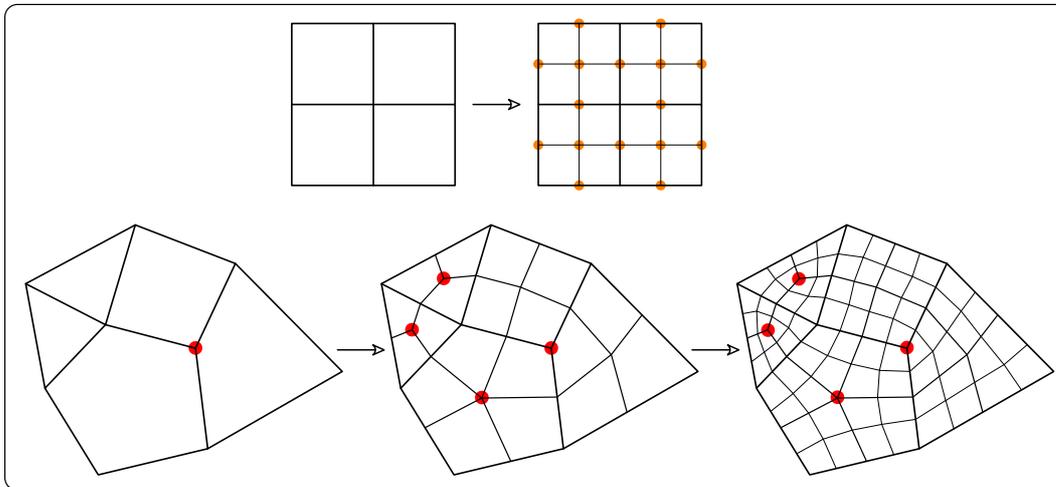
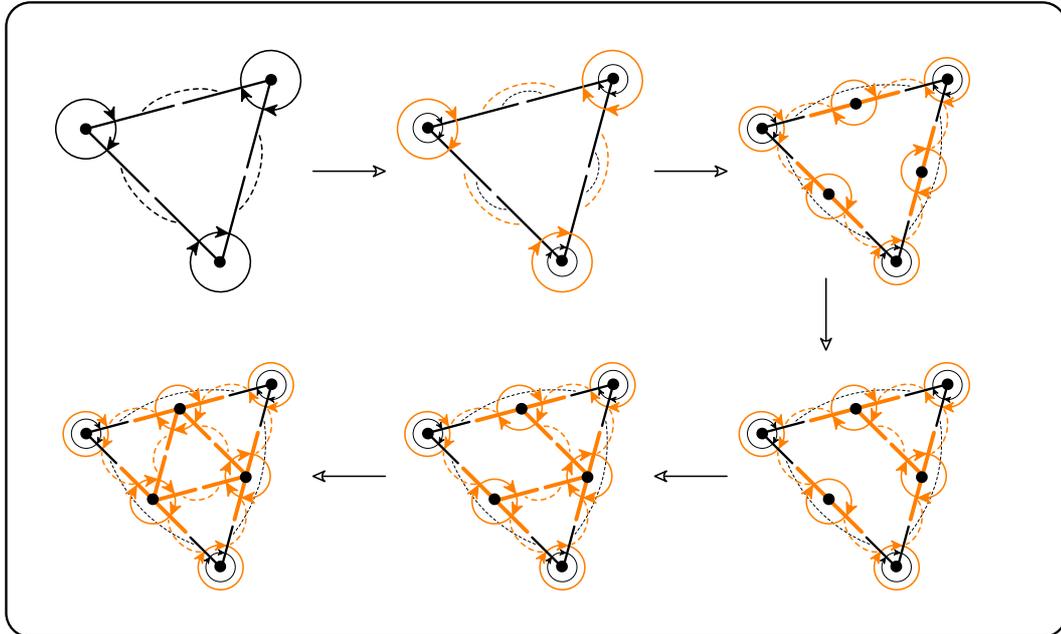


Figure 3.12 - Quadrisection de polygones

Un nouveau sommet est introduit au milieu de la face, toutes les arêtes du maillage sont subdivisées, et de nouvelles arêtes relient les nouveaux sommets au sommet central. Ce principe de subdivision fonctionne également sur des maillages polygonaux quelconques comme on peut le voir sur la figure du bas. Après le premier pas de subdivision, toutes les faces du maillage sont quadrangulaires. Les sommets extraordinaires déjà présents dans le maillage sont conservés. De nouveaux sommets extraordinaires sont introduits lors de ce premier pas de subdivision au centre de toutes les faces dont le nombre de côtés était irrégulier. Tous les pas de subdivision suivants conservent la valence de ces sommets, mais n'en introduisent plus de nouveaux.

Les 2-cartes multirésolution permettant de représenter des maillages quelconques, ces deux processus de raffinement peuvent aisément être implantés au cœur du modèle. La figure 3.13 illustre une quadrisection de triangle entre deux niveaux de résolution consécutifs  $i$  et  $i + 1$  d'une 2-carte multirésolution. La première étape consiste à définir le niveau  $i + 1$  comme étant égal au niveau  $i$ , puis, toutes les arêtes du maillage sont découpées par l'insertion d'un nouveau sommet en leur centre. Cette phase consiste à appliquer sur chaque arête l'opérateur d'insertion de sommet décrit dans le chapitre 2. Enfin, de nouvelles arêtes sont introduites dans les permutations  $\beta_1$  de ces sommets afin de les relier, en utilisant l'opérateur d'insertion d'arête également décrit au chapitre 2.

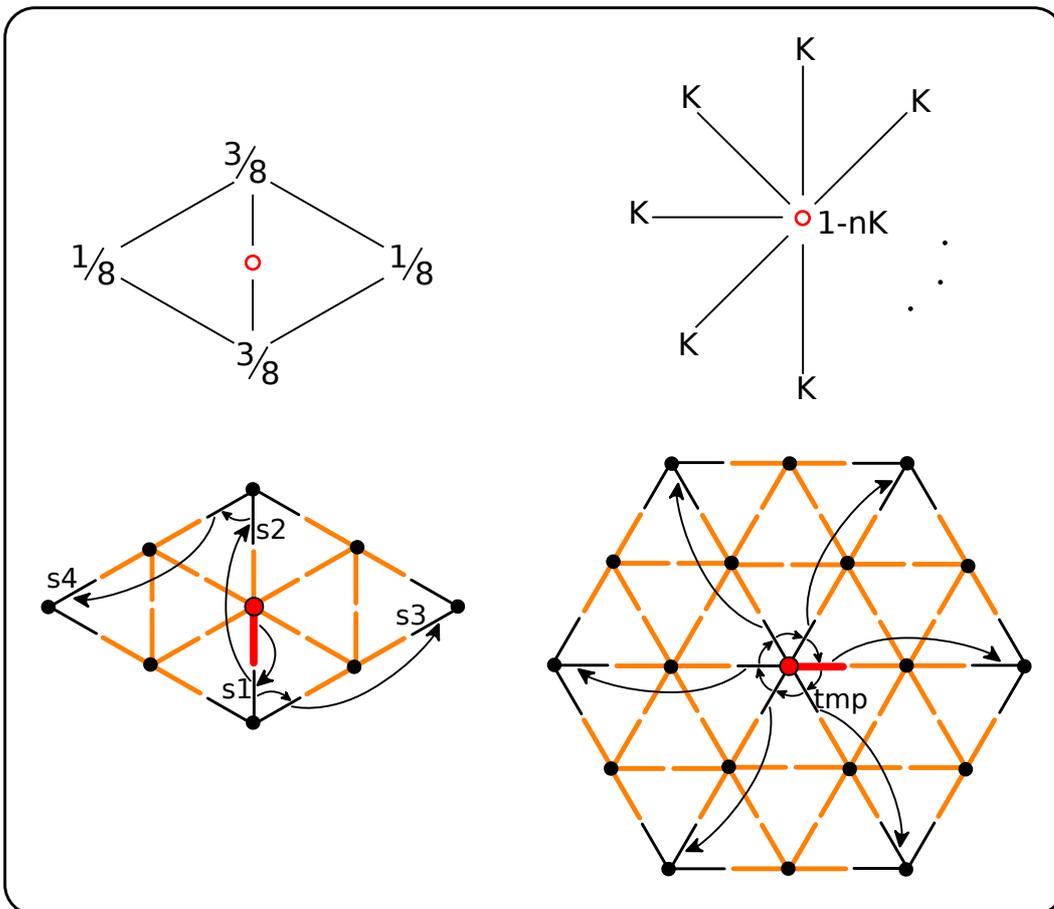
Dans le listing 3.2.1, on montre comment les algorithmes de calcul de la position d'un sommet pour le schéma de Loop peuvent être implantés dans la structure de données issue des 2-cartes multirésolution. Les masques utilisés dans le schéma de Loop sont illustrés dans les dessins du haut de la figure 3.14. Le



**Figure 3.13** - Quadrisection de triangles au sein d'une 2-carte multirésolution

masque de gauche sert au calcul de la position des sommets impairs, et celui de droite au calcul de la nouvelle position des sommets pairs (le schéma de Loop est un schéma approximant ce qui implique la mise à jour de la position des sommets pairs). Le poids  $K$  appliqué aux sommets voisins a été initialement proposé par Loop [Loo87] comme étant égal à  $\frac{1}{n} \left( \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{n} \right)^2 \right)$ , où  $n$  est la valence du sommet.

La première fonction `positionImpairLoop` retourne la position d'un sommet impair. Ce sommet est identifié ici par un brin `b` et un niveau de résolution `niv`. Une précondition, dont la vérification n'est pas illustrée dans cet exemple de code, consiste à s'assurer que le sommet identifié par le brin et le niveau de résolution est bien un sommet impair, c'est-à-dire qui vient d'être introduit à ce niveau de résolution. Cette vérification est aisée : il suffit de vérifier que tous les brins de l'orbite sommet de `b` ont bien été introduits dans la carte au niveau de résolution `niv`. Une deuxième précondition impose que le brin `b` soit l'un des deux brins issus de la subdivision de l'arête du niveau précédent (les autres brins du sommet étant ceux appartenant aux arêtes introduites lors de la phase de liaison des nouveaux sommets). Là encore, une procédure assez simple consiste à vérifier si le brin  $\beta_0^{\text{niv}}(b)$  dispose d'un niveau d'insertion dans la carte inférieur à `niv` (s'il est égal à `niv`, c'est que l'on est sur l'une des nouvelles arêtes). Dans la pratique, la régularité de la subdivision nous permet d'appeler cette fonction directement avec des paramètres qui vérifient



**Figure 3.14** - Masques utilisés dans le schéma de subdivision de Loop et illustration des algorithmes de calcul au sein d'une 2-carte multirésolution

---

**Listing 3.2.1** Exemple d'implantation du calcul des positions des sommets dans le schéma de subdivision de Loop

---

```

Point3D Carte::positionImpairLoop(Brin b, int niv)
{
    Point3D resultat;
    Brin s1 = beta0(b, niv);
    Brin s2 = beta0(s1, niv - 1);
    Brin s3 = beta0(beta1(s1, niv), niv - 1);
    Brin s4 = beta0(beta1(s2, niv), niv - 1);
    resultat = 3*pos(s1, niv) + 3*pos(s2, niv)
              + pos(s3, niv) + pos(s4, niv);
    resultat /= 8.0;
    return resultat;
}

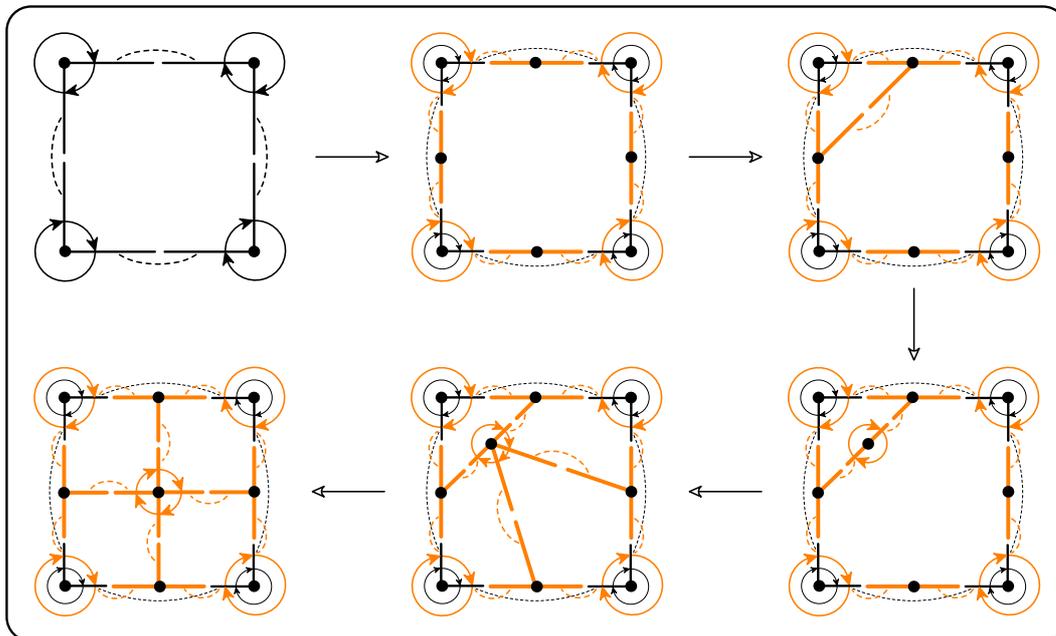
Point3D Carte::positionPairLoop(Brin b, int niv)
{
    Point3D resultat;
    int n = 0;
    Brin tmp = b;
    do
    {
        ++n;
        resultat += K * pos(beta0(tmp, niv - 1), niv - 1);
        tmp = beta1(tmp, niv);
    } while (tmp != b);
    resultat += (1 - n*K) * pos(b, niv - 1);
    return resultat;
}

```

---

ces préconditions. Dans ce calcul, on commence par récupérer un brin par sommet concerné dans l'application du masque. Le premier brin  $s_1$  est atteint en traversant l'involution  $\beta_0^{\text{niv}}$ . Les autres brins sont atteints très rapidement en naviguant sur le maillage de niveau inférieur en exploitant les relations topologiques de niveau  $\text{niv}-1$ . Il suffit ensuite de récupérer les plongements associés aux sommets de ces brins, de calculer le résultat et de le retourner.

La fonction suivante, `positionPairLoop`, effectue le calcul de la position d'un sommet pair à partir d'un brin  $b$  et un niveau de résolution  $\text{niv}$ . L'algorithme consiste ici simplement à faire le tour de l'orbite sommet de  $b$ , et pour chaque brin, à traverser sa couture  $\beta_0^{\text{niv}-1}$  pour atteindre le sommet voisin au niveau  $\text{niv}-1$  et récupérer son plongement. L'entier  $N$  stocke en même temps la valence du sommet. Le calcul de la nouvelle position est ensuite effectué et le résultat

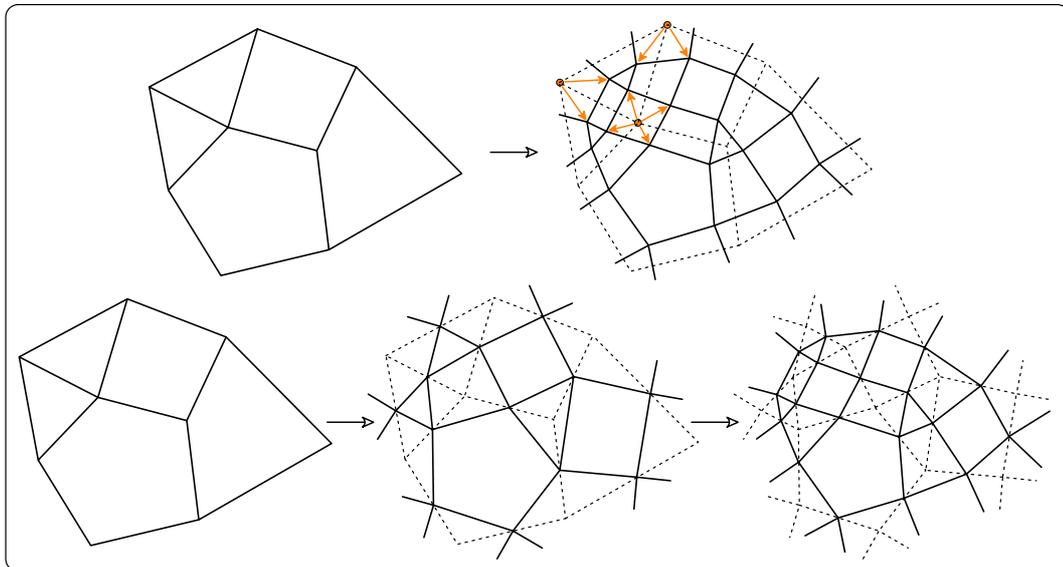


**Figure 3.15** - Quadrisection de quadrangles au sein d'une 2-carte multirésolution

retourné.

Les dessins du bas de la figure 3.14 illustrent l'application de cette fonction sur des extraits de maillage. Le brin  $b$  de départ est à chaque fois illustré en rouge, tout comme le sommet dont on est en train de calculer la position.

Le deuxième processus de subdivision primaire d'un maillage est la quadrisection de quadrangles (qui comme on l'a vu précédemment peut commencer sur un maillage composé de faces polygonales quelconques). La figure 3.15 illustre le processus de quadrisection d'une face quadrangulaire dans une 2-carte multirésolution. De la même manière que pour l'algorithme précédent, le niveau  $i+1$  est d'abord créé comme copie du niveau  $i$  (cette étape n'est pas illustrée ici), puis toutes les arêtes du maillage sont coupées en insérant de nouveaux sommets. On utilise ensuite l'opérateur d'insertion d'arête entre deux nouveaux sommets issus de deux anciennes arêtes contiguës. L'opérateur d'insertion de sommet est ensuite utilisé au niveau  $i+1$  sur cette nouvelle arête afin d'insérer un sommet qui sera le sommet central de la subdivision de la face. Enfin, de nouvelles arêtes sont insérées entre ce sommet central et les autres sommets issus de la subdivision des arêtes de la face d'origine. La dernière étape (en-bas à gauche) est topologiquement équivalente à l'avant-dernière, la représentation graphique est simplement modifiée afin de paraître plus conforme à ce que l'on s'attend à obtenir en tant que résultat de la quadrisection d'une face.



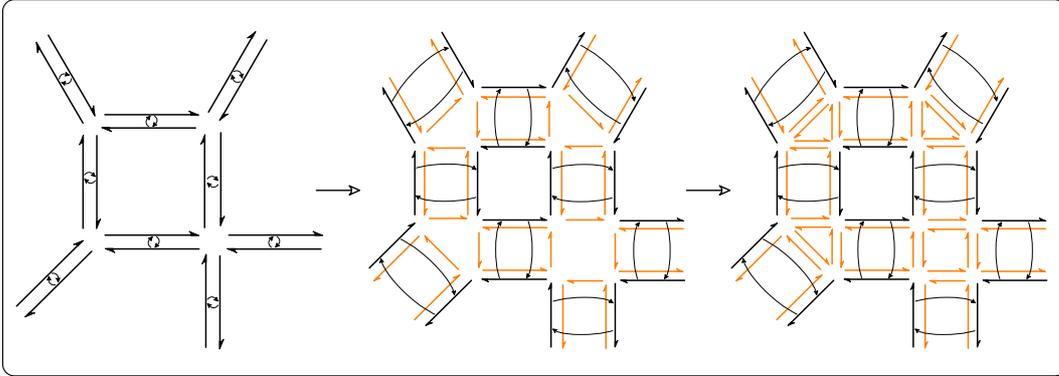
**Figure 3.16** - Subdivision duale de maillage polygonal

De la même manière, des algorithmes peuvent être écrits afin de calculer les positions des sommets au nouveau niveau en appliquant les masques illustrés à la figure 3.2. Ceux-ci vont également tirer fortement parti de l'information topologique du niveau précédent afin d'atteindre plus rapidement les sommets concernés par le calcul.

### Schémas duaux

Le raffinement de maillage dual utilisé dans le schéma de Doo-Sabin [Doo78, DS78] consiste à subdiviser les sommets du maillage plutôt que les faces. Dans le dessin du haut de la figure 3.16, chaque ancien sommet est subdivisé en autant de nouveaux sommets que de faces incidentes au sommet de départ. Ces nouveaux sommets sont ensuite maillés en trois types de faces : celles qui couvrent les anciennes faces, celles qui couvrent les anciennes arêtes et celles qui couvrent les anciens sommets. On peut également voir ce processus de subdivision comme un rétrécissement des anciennes faces suivi par un remplissage par de nouvelles faces des trous ainsi créés.

Deux pas de subdivision du schéma décrit dans [PR97] résultent en le même raffinement du maillage. Chaque pas de subdivision consiste ici simplement à insérer un nouveau sommet sur chaque arête, puis à lier ces nouvelles arêtes tel qu'illustré dans le dessin du bas de la figure 3.16.



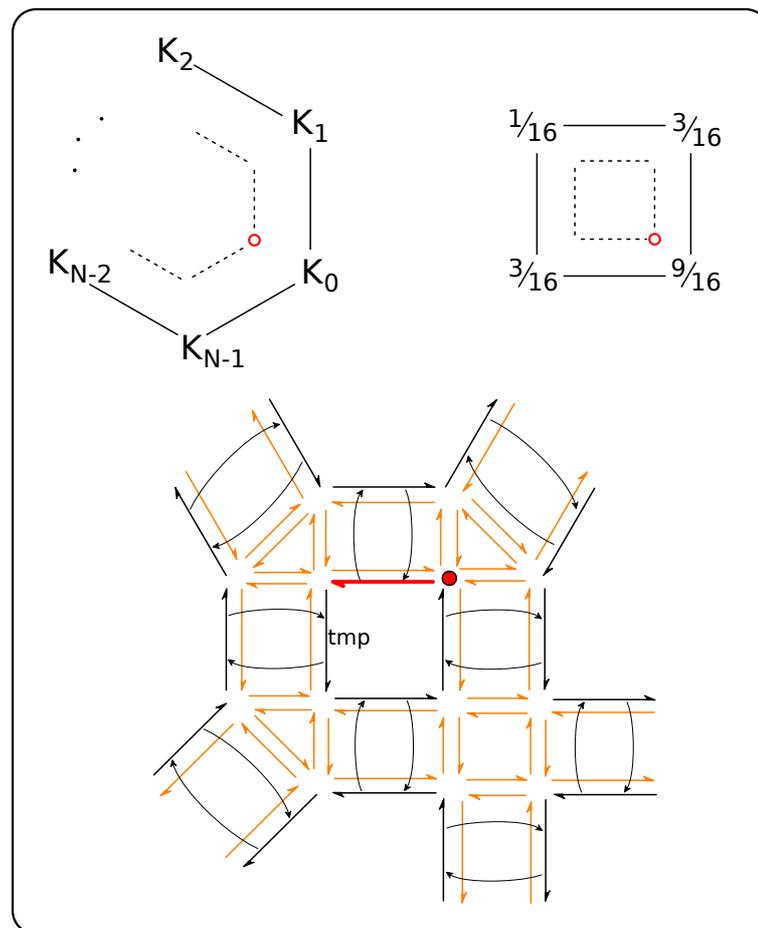
**Figure 3.17** - Subdivision duale de maillage polygonal au sein d'une 2-carte multirésolution duale

Bien qu'il pourrait également être implémenté au sein des 2-carte multirésolution primales, il se trouve que cet algorithme de subdivision dual est bien plus simple à décrire dans la formulation duale du modèle. Ainsi, la figure 3.17 illustre comment ce raffinement de maillage est effectué dans une 2-carte multirésolution duale entre les niveaux  $i$  et  $i + 1$ . Comme précédemment la carte de niveau  $i$  est d'abord copiée au niveau  $i + 1$ , puis de nouvelles faces sont introduites dans la carte. Les faces couvrant les anciennes faces conservent le nombre de côtés de ces faces. Il est donc naturel de conserver les cycles de brins liés par la permutation  $\phi_1$  pour représenter ces faces. Les faces couvrant les anciennes arêtes ont toutes quatre côtés et sont insérées entre les anciennes faces en modifiant les coutures par l'involution  $\phi_2^{i+1}$ . Les faces couvrant les anciens sommets ont un nombre côtés égal à la valence du sommet qu'elles couvrent. Ces faces sont cousues aux nouvelles faces que l'on vient juste de créer.

Dans les schémas duaux, un seul masque de subdivision est nécessaire. Celui utilisé dans le schéma de Doo-Sabin est illustré dans les dessins du haut de la figure 3.18. La position de chaque nouveau sommet est calculée à partir de l'ensemble des positions des sommets de la face dont il est issu. La figure de droite montre le cas régulier d'un maillage composé de faces à quatre côtés, et la figure de gauche montre le cas général. Les valeurs proposées initialement pour les poids appliqués aux sommets sont les suivantes :  $K_0 = \frac{1}{4} + \frac{5}{4}N$ ,

$$K_j = \frac{3 + 2 \cos \frac{2j\pi}{N}}{4N} \text{ avec } j \in [1, N - 1] \text{ et } N \text{ le nombre de sommets de la face.}$$

Implanter ce calcul dans les 2-cartes multirésolution est aisé et efficace. On peut par exemple réaliser le code illustré dans le listing 3.2.2 pour calculer la position d'un sommet identifié par un brin  $\mathbf{b}$  et un niveau  $\mathbf{niv}$ . On peut remarquer qu'il existe un sommet de niveau  $\mathbf{niv}$  pour chaque brin de niveau  $\mathbf{niv}-1$ . Une solution simple pour calculer les positions de tous les sommets



**Figure 3.18** - Masque utilisé dans le schéma de subdivision de Doo-Sabin et illustration des algorithmes de calcul au sein d'une 2-carte multirésolution

---

**Listing 3.2.2** Exemple d'implantation du calcul des positions des sommets dans le schéma de subdivision de Doo-Sabin

---

```

Point3D Carte::positionDooSabin(Brin b, int niv)
{
    Point3D resultat;
    int N = nbCotesFace(b, niv);
    K0 = 1/4 + 5/4 * N;
    resultat += K0 * pos(b, niv - 1);
    Brin tmp = phil(b, niv);
    int j = 1;
    do
    {
        float Kj = (3 + 2*cos(2*j*PI/N)) / 4*N;
        resultat += Kj * pos(tmp, niv - 1);
        tmp = phil(tmp, niv);
        ++j;
    } while (tmp != b);
    return resultat;
}

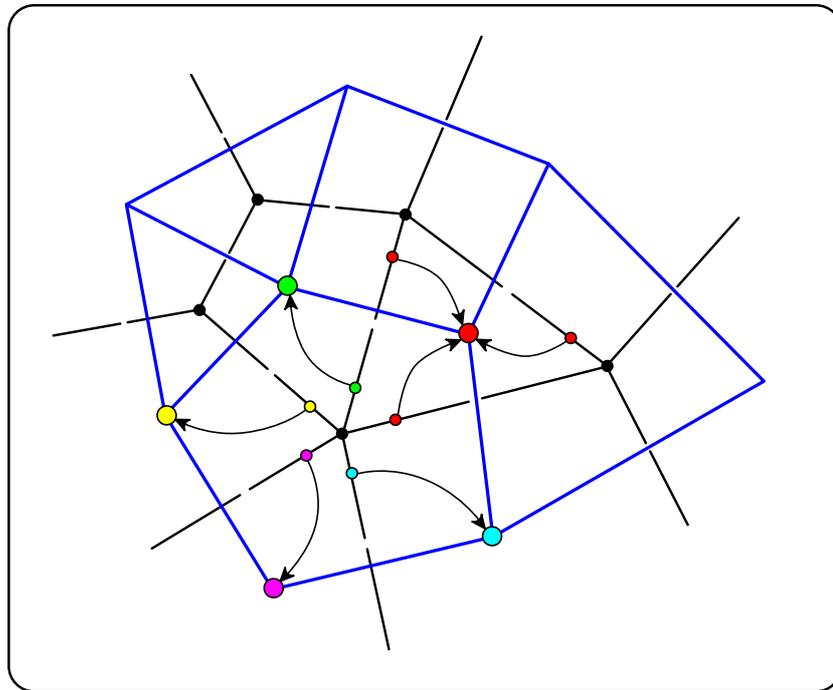
```

---

de niveau `niv` consiste alors à parcourir l'ensemble des brins de  $B^{\text{niv}-1}$  et à appliquer cette fonction de calcul au niveau `niv` pour chacun d'entre eux. L'algorithme consiste à faire le tour de la face de niveau `niv-1` dont le sommet est issu en suivant les relations  $\phi_1$ , et à récupérer la position au niveau `niv-1` de chacun des sommets traversés en appliquant le masque de subdivision. Le dessins du bas de la figure 3.18 illustre l'application de cette fonction sur un extrait de maillage. Le brin `b` de départ est à nouveau illustré en rouge, tout comme le sommet dont on calcule la position.

Une tout autre stratégie pour effectuer une subdivision duale avec des 2-cartes multirésolution consiste à exploiter le fait que le raffinement topologique du schéma de Doo-Sabin est le dual du raffinement topologique du schéma de Catmull-Clark, et que l'information combinatoire du maillage dual est, comme on l'a vu dans le chapitre 1, entièrement contenue dans une 2-carte.

Etant donné un maillage polygonal d'entrée que l'on souhaite subdiviser avec le schéma de Doo-Sabin, on va stocker en mémoire, non pas la 2-carte représentant le maillage lui-même, mais son dual. On représente cette 2-carte sous forme d'une 2-carte primale. Chaque face de cette 2-carte primale correspond à un sommet du maillage de départ, chaque sommet correspond à une face et chaque arête à une arête orthogonale (géométriquement). La figure 3.19 illustre un maillage en bleu, et la 2-carte primale encodant son maillage dual en

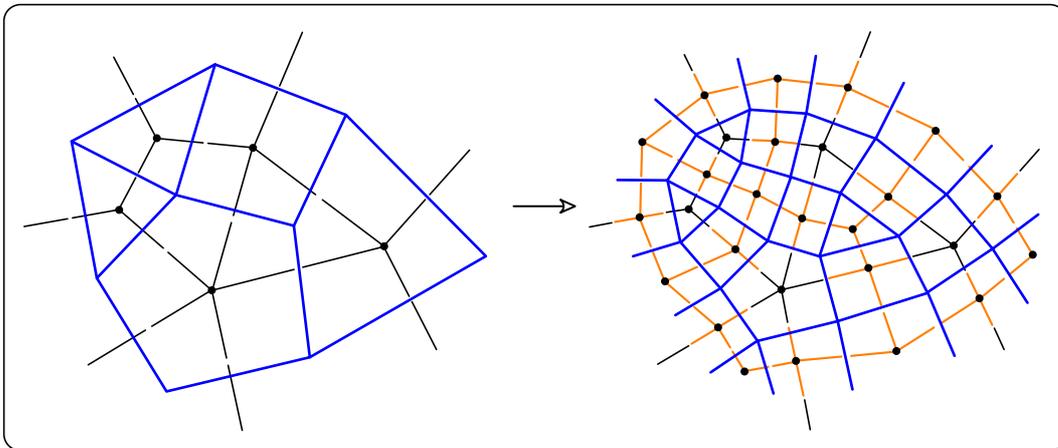


**Figure 3.19** - Représentation d'un maillage polygonal donné par la 2-carte primale encodant son maillage dual

noir.

On ne souhaite pas afficher classiquement le maillage correspondant à cette 2-carte primale, mais son maillage dual, qui correspond au maillage de départ (le dual du dual d'un maillage étant le maillage lui-même). Au lieu de plonger les sommets, ce sont les faces de cette 2-carte primale qui sont plongées en un point 3D. Comme on le voit sur le schéma de la figure 3.19, tous les brins d'une orbite face sont associés au même point 3D du maillage de départ. En conséquence, le cycle de brins d'une orbite sommet est associé au cycle de points 3D qui forment une face du maillage original. Afin d'afficher le maillage on va donc parcourir l'ensemble des sommets de la 2-carte, et pour chacun, afficher le polygone formé par les points 3D associés à chacun des brins de ce sommet.

Pour effectuer une subdivision de Doo-Sabin sur le maillage de départ, il suffit maintenant d'appliquer une subdivision de Catmull-Clark telle qu'on l'a décrite précédemment sur la 2-carte primale, et d'afficher toujours le maillage dual correspondant. La figure 3.20 illustre deux niveaux consécutifs entre lesquels ce principe est appliqué. Le maillage dual de la 2-carte subdivisée avec le schéma de Catmull-Clark est bien le résultat de la subdivision du maillage de départ avec le schéma de Doo-Sabin.



**Figure 3.20** - Subdivision de Doo-Sabin sur un maillage polygonal par l'application du schéma de Catmull-Clark sur la 2-carte primale encodant son maillage dual

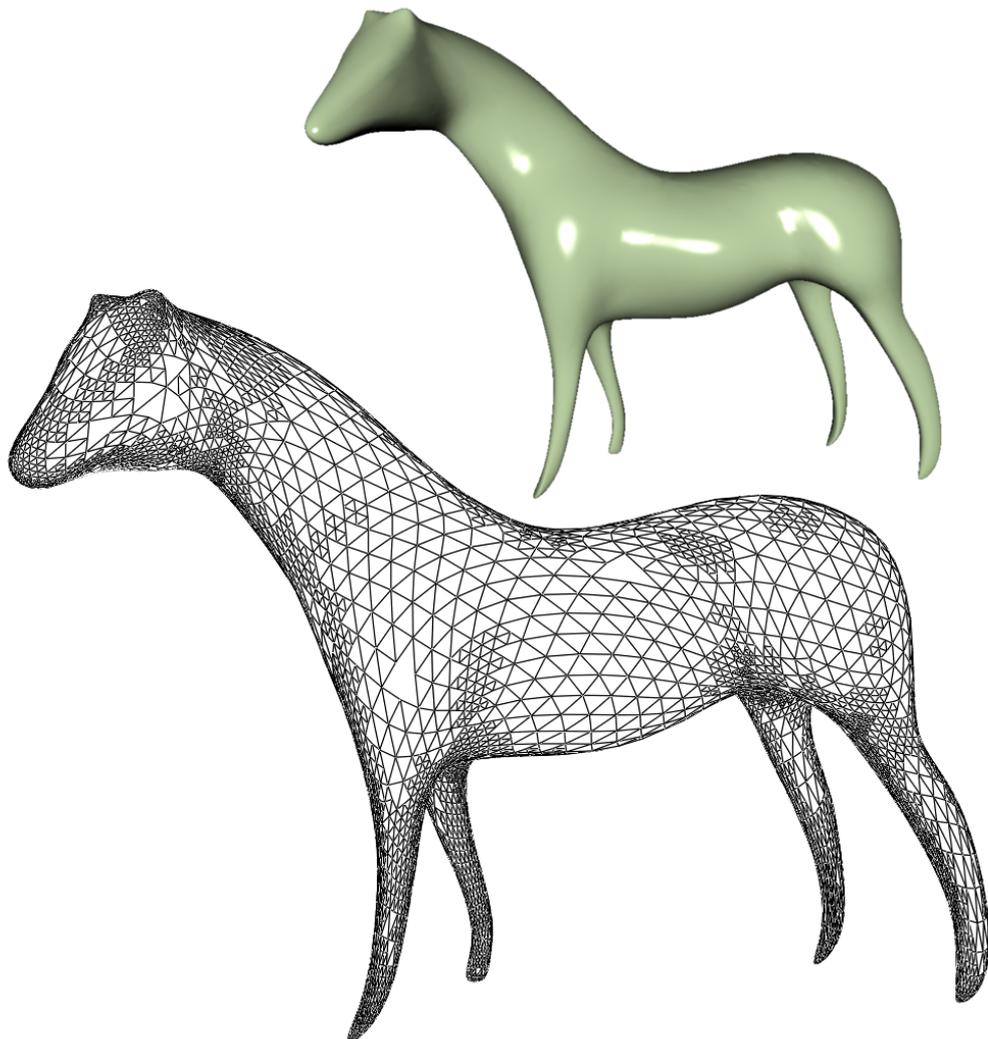
Au niveau du calcul de la géométrie, on continuera à appliquer le masque de Doo-Sabin illustré dans la figure 3.18. L'algorithme sera similaire à celui décrit dans le listing 3.2.2 avec pour seule différence qu'il faudra maintenant tourner autour des sommets de la 2-carte plutôt que des faces afin de récupérer l'ensemble des sommets participant au calcul de la position d'un nouveau sommet.

Dans [ZS01] un cadre est défini avec comme objectif l'unification au sein d'une même structure de schémas de subdivision primaux et duaux. Cette structure est basée sur des arbres quaternaires et est alors limitée aux maillages quadrilatéraux réguliers pour lesquels les versions primales et duales présentent toutes deux des faces à quatre côtés et des sommets de valence quatre. Les 2-cartes multirésolution permettent de dépasser cette limitation grâce à leur capacité à manipuler des maillages quelconques.

### Remarques générales sur la subdivision régulière

Que ce soit dans les algorithmes de raffinement primaux ou duaux, les permutations  $\beta_1$  ou  $\phi_1$  ne sont pas modifiées d'un niveau au suivant. Dans un cas, c'est la valence des anciens sommets qui n'est pas modifiée et de nouveaux sommets qui sont insérés, dans l'autre c'est le nombre de côtés des faces qui n'est pas modifié et de nouvelles faces qui sont insérées. C'est donc toujours sur l'involution  $\beta_0$  ou  $\phi_2$  que se joue la multirésolution.

Du point de vue de l'occupation mémoire, cela signifie que les tableaux de relations topologiques contenues dans les brins ont une taille fixe de 1 pour ce

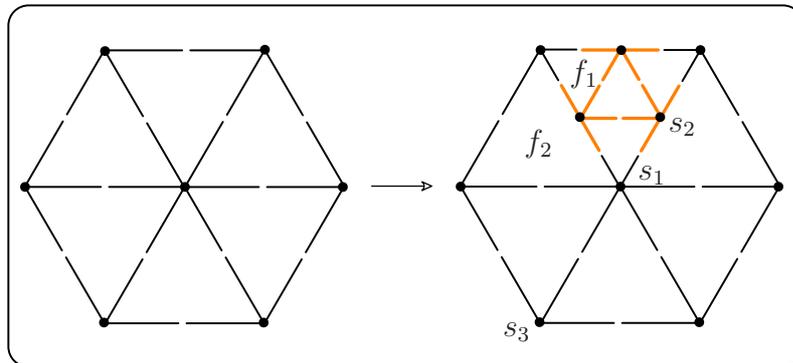


**Figure 3.21** - Exemple de maillage obtenu par une subdivision adaptative

qui concerne le stockage des permutations, quel que soit le nombre de niveaux de résolution générés. Les tableaux stockant les involutions et les liens vers le plongement auront eux tous une taille égale à la différence entre le niveau d'insertion du brin dans la carte multirésolution, et le niveau maximal représenté.

### 3.2.2 Génération adaptative

Dans le cadre d'une génération adaptative des niveaux de résolution, seules les cellules du maillage sélectionnées en fonction d'un certain critère de subdivision



**Figure 3.22** - Subdivision adaptative primale de maillage triangulaire :  
une seule face a été subdivisée entre les deux niveaux

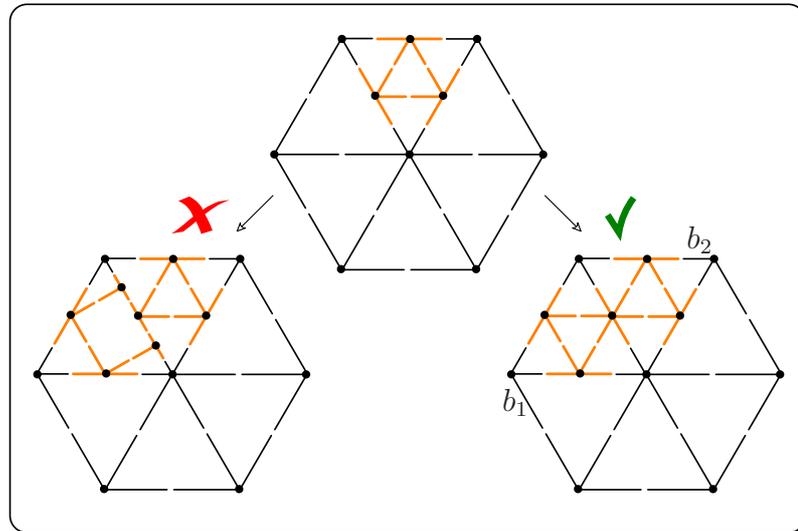
sont subdivisées d'un niveau au suivant. Ce critère peut être par exemple lié à la distance du maillage courant à la surface limite ou à une estimation de la courbure locale de la surface. La figure 3.21 montre un exemple de maillage obtenu en subdivisant un maillage triangulaire de départ de façon adaptative en utilisant le schéma de subdivision de Loop et une estimation de la courbure locale en tant que critère de subdivision. Le maillage obtenu est le maillage le moins raffiné respectant en tout lieu le critère de subdivision.

Nous allons ici nous intéresser essentiellement aux problèmes topologiques posés par un tel raffinement, et proposer des solutions en exposant la manière dont les 2-cartes multirésolution supportent ce type de raffinement.

### Schémas primaux

Si l'on étudie le déroulement d'un raffinement primal de la topologie d'un maillage on observe que durant une quadrisection de faces, ce sont les arêtes qui jouent le rôle central. Les opérations effectuées sont en effet soit des subdivisions d'arête par l'insertion d'un sommet, soit des insertions d'arêtes entre des sommets existants. C'est donc par une gestion fine des arêtes au cours de la subdivision que l'on va pouvoir gérer un raffinement adaptatif du maillage tout en conservant l'intégrité de la connectivité du maillage. Une nouvelle problématique concernant le niveau de subdivision d'une cellule au sein d'un maillage adaptatif d'un niveau donné va également devoir être prise en compte.

Pour commencer simplement, considérons les maillages de la figure 3.22. Pour la légèreté des illustrations, nous ne représentons plus ici les relations topologiques entre les brins. A gauche se trouve un maillage triangulaire de niveau 0, et à droite, le maillage de niveau 1 après un pas de subdivision adaptative où un



**Figure 3.23** - Mise-à-jour d'une subdivision adaptative primale de maillage triangulaire en tenant compte du voisinage déjà subdivisé

seul triangle a été sélectionné au niveau 0 pour être subdivisé vers le niveau 1. Contrairement au cas régulier, toutes les cellules n'ont pas été subdivisées et le maillage de niveau 1 est donc composé d'un ensemble de cellules aux niveaux hétérogènes. Ce maillage de niveau 1 reste cependant une 2-variété fermée. Celle-ci n'est simplement plus composée uniquement de faces triangulaires. En effet la face  $f_1$  est maintenant une face à quatre côtés, ce qui ne pose pas de problème dans une 2-carte, cette dernière étant un modèle cellulaire permettant la représentation de faces quelconques.

Supposons que la face  $f_2$  soit maintenant elle aussi sélectionnée au niveau 0 pour être subdivisée vers le niveau 1. L'algorithme de subdivision ne peut plus être appliqué simplement sans prendre en compte le niveau de subdivision actuel des cellules du voisinage. Le risque est en effet de se retrouver avec un résultat similaire à celui du dessin de gauche de la figure 3.23, alors que le résultat correct du raffinement correspond au dessin de droite.

Afin de gérer correctement ce raffinement, il nous faut définir la notion de *niveau de subdivision* d'une cellule au sein du maillage d'un niveau de résolution donné. De manière intuitive, on voit dans la figure 3.22, qu'au sein du maillage de niveau 1, la face  $f_1$  a un niveau de subdivision égal à 1 et la face  $f_2$  un niveau de subdivision égal à 0. De même, l'arête formée par les sommets  $s_1$  et  $s_2$  a un niveau de subdivision égal 1 tandis que celle formée par les sommets  $s_1$  et  $s_3$  a un niveau de subdivision égal à 0. La définition du niveau de subdivision des sommets n'est pas pertinente dans un schéma de subdivision primal, ces derniers n'intervenant pas dans le processus de raffinement topologique. Nous

```

pour chaque arête  $a$  de la face  $f$  de niveau  $i$  faire
  | si  $a$  n'est pas encore subdivisée vers le niveau  $i + 1$  alors
  | |   Subdiviser  $a$ ;
  | fin
fin
Insérer les nouvelles arêtes;

```

**Algorithme 1** : Algorithme de subdivision d'une face

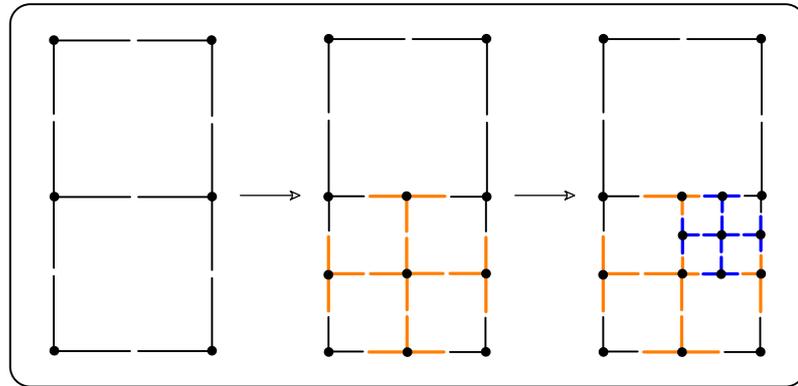
donnerons par la suite une définition plus formelle de cette notion de niveau de subdivision et montrerons comment on peut le calculer dans une 2-carte multirésolution, et ce sans avoir à stocker la moindre donnée supplémentaire. Nous nous contenterons cependant de cette approche intuitive pour le moment.

Au moment de subdiviser la face  $f_2$  vers le niveau 1, on va donc commencer par vérifier au niveau 0 pour chacune de ses arêtes, si elle a déjà été subdivisée vers le niveau 1. Si une de ces arêtes n'est pas encore subdivisée, alors on la subdivise vers le niveau 1. Afin de vérifier si une arête a déjà été subdivisée, il suffit de prendre un brin de l'arête au niveau 0, puis de regarder quel est le niveau de subdivision de son arête au sein de la carte de niveau 1. Si celui-ci est 0, c'est que l'arête n'a pas encore été subdivisée; si son niveau est 1, c'est qu'elle a déjà été subdivisée. Par exemple, pour le brin  $b_1$  de la figure 3.23, son arête au sein de la 2-carte de niveau 1 a un niveau de subdivision égal à 0. On en déduit qu'elle n'a pas encore été subdivisée vers le niveau 1. L'arête du brin  $b_2$  a quant à elle un niveau de subdivision égal à 1 au sein de la 2-carte de niveau 1, et on en déduit qu'elle a déjà été subdivisée vers ce niveau.

Une fois toutes les arêtes correctement subdivisées, on peut alors insérer les nouvelles arêtes qui vont lier les nouveaux sommets et obtenir un raffinement correct. Cet algorithme fonctionne de manière générale en remplaçant respectivement 0 et 1 par  $i$  et  $i + 1$ .

L'algorithme 1 illustre la manière de procéder pour subdiviser une face en tenant compte du niveau de subdivision de ses arêtes. L'algorithme est le même selon que l'on effectue une quadrisection de triangle ou une quadrisection de quadrangle, de même que pour la subdivision d'une face polygonale lors du premier pas d'une subdivision de Catmull-Clark. Cet algorithme de raffinement est d'ailleurs illustré dans la figure 3.24 où l'on voit deux niveaux consécutifs de subdivision adaptative d'un maillage de départ à faces quadrilatérales.

De la même manière que l'on a vérifié qu'une arête était déjà subdivisée, on aura besoin lors de la mise à jour de la subdivision, de vérifier si une face de la carte de niveau 0 a déjà été subdivisée vers le niveau 1 ou non. Le principe est



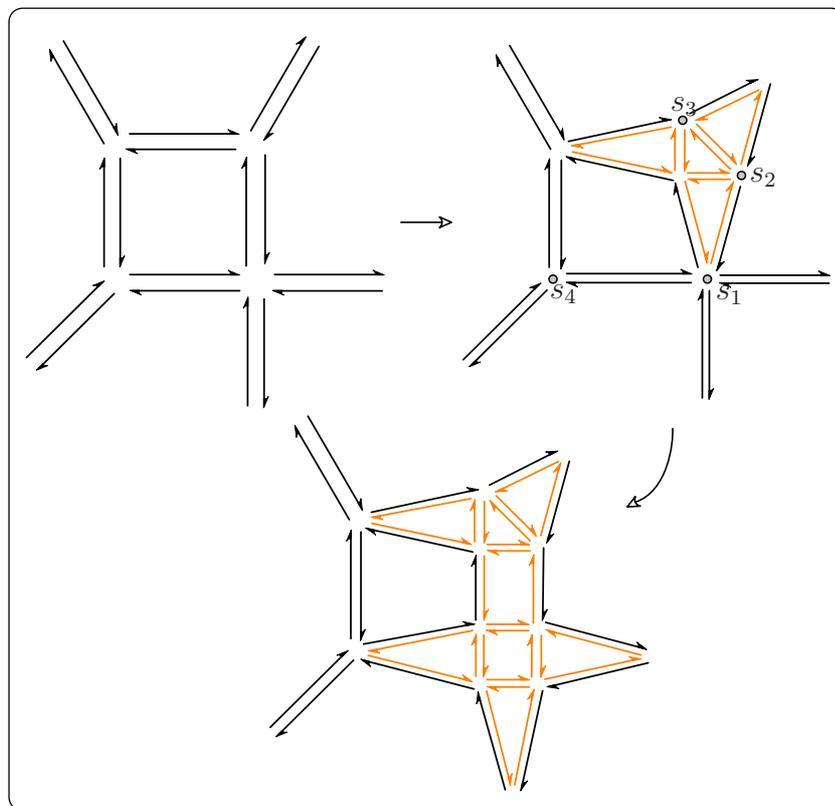
**Figure 3.24** - Exemple de subdivision adaptative primale de maillage quadrangulaire

similaire à celui utilisé pour les arêtes. Il consiste à prendre un brin de la face à tester, puis à vérifier quel est le niveau de sa face au sein de la carte de niveau 1. Si celui-ci est 0, c'est que la face n'a pas encore été subdivisée ; si son niveau est 1, c'est que l'on est dans l'une de ses sous-faces de niveau 1 et qu'elle a donc déjà été subdivisée. Si on considère le brin  $b_1$  de la figure 3.23, sa face au sein de la 2-carte de niveau 1 a un niveau de subdivision égal à 0 ce qui permet de déduire qu'elle n'a pas encore été subdivisée vers le niveau 1. La face du brin  $b_2$  a quant à elle un niveau de subdivision égal à 1 au sein de la 2-carte de niveau 1, et on en déduit qu'elle a déjà été subdivisée vers ce niveau. Ici également, cet algorithme fonctionne de manière générale en remplaçant respectivement 0 et 1 par  $i$  et  $i + 1$ .

On exploite ici la capacité à interroger la 2-carte multirésolution à un niveau supérieur au niveau maximum courant, soulevée à la fin du chapitre 2. En effet, si le niveau maximum courant est égal à 1 et que l'on veut appliquer l'algorithme de subdivision de face sur une face de niveau 1, on va alors tester si cette face ainsi que ses arêtes sont déjà subdivisés en interrogeant la 2-carte multirésolution au niveau 2 alors même que ce niveau n'existe pas encore.

## Schémas duaux

Dans un raffinement dual, le critère de subdivision est évalué sur les sommets du maillage. Une fois un sommet sélectionné, il se subdivise en autant de nouveaux sommets qu'il avait de faces incidentes. Tout comme une face subdivisait l'ensemble de ses arêtes lors d'un processus de raffinement primal, la création de la face couvrante d'un sommet lors de sa subdivision "écarter" en quelque sorte l'ensemble de ses arêtes incidentes. Topologiquement, l'arête étant son



**Figure 3.25** - Exemple de subdivision adaptative obtenue avec un raffinement dual sur un maillage polygonal

propre dual dans un maillage surfacique, c'est tout naturellement que dans le cadre de la subdivision duale, l'arête va également jouer un rôle central dans la gestion de la subdivision adaptative.

La figure 3.25 illustre sur les deux figure du haut un exemple de maillage de niveau 0 dans lequel un sommet a été sélectionné pour la subdivision vers le niveau 1. La face couvrant le sommet a été créée, et les faces couvrant les arêtes incidentes, qui dans le cadre d'un raffinement régulier sont toutes des faces à quatre côtés, sont ici triangulaires.

Ici aussi on obtient un maillage de niveau 1 composé d'un ensemble de cellules aux niveaux de subdivision hétérogènes. Comme dans le cas primal, on donne ici une approche intuitive et illustrée de ce qu'est le niveau de subdivision d'une cellule. Dans la figure 3.25, au sein du maillage de niveau 1, le sommet  $s_1$  a un niveau de subdivision égal à 1, tout comme les sommets  $s_2$  et  $s_3$ . Le sommet  $s_4$  a pour sa part un niveau de subdivision égal à 0. Les arêtes formées par les sommets  $s_1$  et  $s_2$  et  $s_2$  et  $s_3$  ont un niveau de subdivision égal 1 tandis que celle formée par les sommets  $s_1$  et  $s_4$  a un niveau de subdivision égal à 0. Dans

```

pour chaque arête  $a$  du sommet  $s$  de niveau  $i$  faire
  | si  $a$  n'est pas encore subdivisée vers le niveau  $i + 1$  alors
  | | Subdiviser  $a$ ;
  | fin
fin
Insérer les nouvelles arêtes;

```

**Algorithme 2** : Algorithme de subdivision d'un sommet

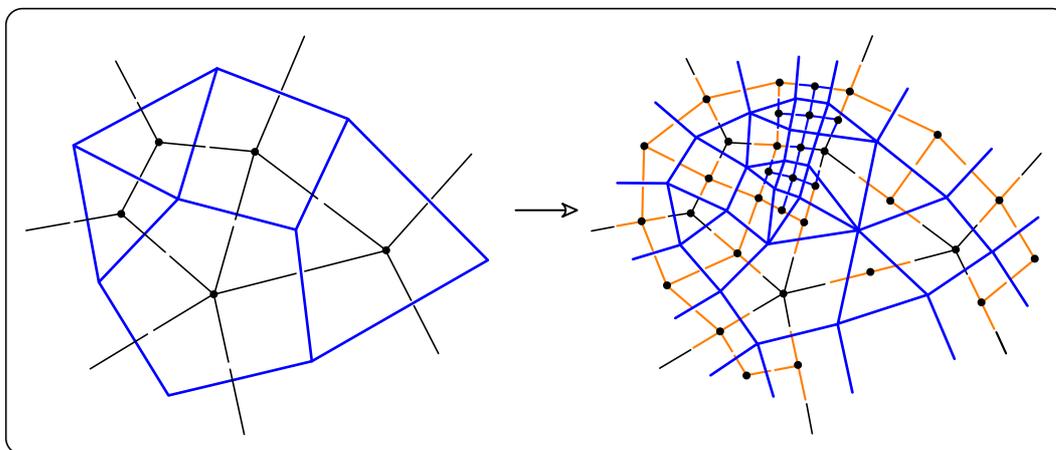
un schéma de subdivision dual, c'est la définition du niveau de subdivision des faces qui n'est pas pertinente, ces dernières n'intervenant pas dans le processus de raffinement topologique.

Lorsqu'un sommet voisin est aussi sélectionné pour être subdivisé, tel qu'illustré dans le dessin du bas de la figure 3.25, celui-ci doit commencer par vérifier l'état de ses arêtes incidentes. Pour chacune, si elle a déjà été partiellement subdivisée, c'est-à-dire si la face triangulaire de niveau 1 couvrant l'arête de niveau 0 est déjà présente, ce triangle est transformé en quadrangle par l'insertion d'un brin dans son orbite  $\phi_1^1$ . Sinon, cette face triangulaire est créée. Enfin la face couvrant le sommet est créée et liée par  $\phi_2^1$  aux faces couvrant les arêtes. De même que pour la subdivision primale, cet algorithme fonctionne de manière générale en remplaçant respectivement 0 et 1 par  $i$  et  $i + 1$ .

L'algorithme 2, sensiblement similaire à celui de la subdivision d'une face dans le cas primal, illustre la manière de procéder pour subdiviser un sommet en tenant compte du niveau de subdivision de ses arêtes.

Comme on l'avait vu dans le cadre de la subdivision régulière, ici aussi on va pouvoir exploiter le fait que ce raffinement topologique est le dual du raffinement utilisé dans le schéma de Catmull-Clark. De la même manière, on ne va plus encoder directement le maillage de départ mais son maillage dual au sein d'une 2-carte primale dont les faces sont plongées en un point 3D. On applique ensuite simplement le raffinement de Catmull-Clark de manière adaptative sur cette 2-carte primale. Ainsi en considérant le maillage dual de cette 2-carte, on va obtenir un maillage subdivisé de manière adaptative avec le schéma de Doo-Sabin.

La figure 3.26 illustre un exemple de subdivision de Doo-Sabin adaptative sur un maillage polygonal. Celui-ci est obtenu en subdivisant de manière adaptative avec le schéma de Catmull-Clark la 2-carte primale encodant le maillage dual du maillage de départ.



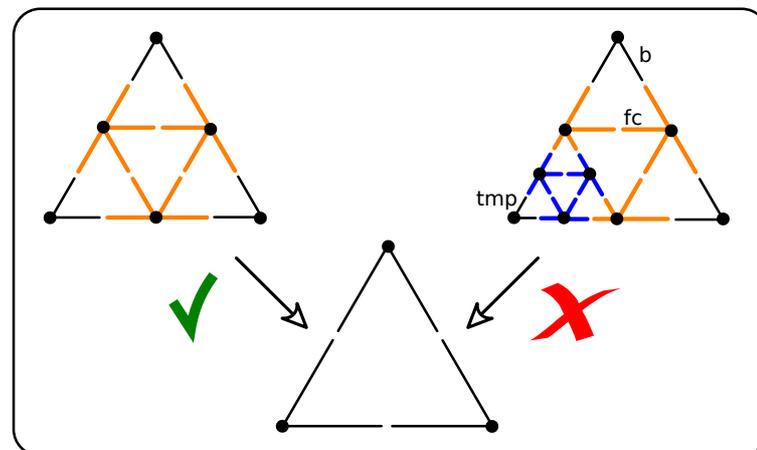
**Figure 3.26** - Subdivision adaptative de Doo-Sabin sur un maillage polygonal par l'application adaptative du schéma de Catmull-Clark sur la 2-carte primale encodant son maillage dual

### 3.2.3 Simplification

Lorsque la subdivision adaptative d'un maillage doit être mise à jour, on rencontre aussi bien le cas de cellules pour lesquelles la subdivision doit être appliquée plus avant, que le cas de cellules dont le niveau de subdivision courant n'a au contraire plus lieu d'être.

On a alors besoin de simplifier le maillage et pour cela, les opérateurs de subdivision que l'on a décrit dans cette section peuvent tout aussi bien être appliqués en sens inverse. Des contraintes doivent ici aussi être vérifiées pour que la simplification topologique puisse avoir lieu. Quand, au sein d'une 2-carte de niveau  $i$ , on subdivise une cellule de niveau  $i$  vers le niveau  $i + 1$ , on s'assure tout d'abord qu'elle n'a pas déjà été subdivisée vers ce niveau. De manière similaire, on ne va pouvoir simplifier une cellule du niveau  $i + 1$  vers le niveau  $i$  que si elle est effectivement déjà subdivisée vers le niveau  $i + 1$ , *et que ce pas de subdivision est le dernier pour cette cellule*. Cette dernière contrainte est le pendant du fait que l'on subdivise les cellules niveau après niveau sans appliquer plusieurs pas de subdivision d'un seul coup.

La figure 3.27 illustre en-haut à gauche une face de niveau 0 qui a été subdivisée une fois vers le niveau 1. Aucune de ses sous-faces n'a été subdivisée vers des niveaux supérieurs. On peut donc directement inverser le raffinement topologique et simplifier la face vers le niveau 0. En-haut à droite, on voit une face de niveau 0 qui a été subdivisée une fois vers le niveau 1, et dont une sous-face de niveau 1 a encore été subdivisée vers le niveau 2. Avant de pouvoir simplifier la face vers le niveau 0, il faudra donc d'abord simplifier vers



**Figure 3.27** - Une face subdivisée par quadrisection de triangles ne peut être simplifiée directement que si ses sous-faces ne sont pas subdivisées vers des niveaux supérieurs

le niveau 1 la sous-face subdivisée.

Le listing 3.2.3 montre un exemple de code illustrant le test permettant de savoir si une face peut être simplifiée ou non. Cette fonction reçoit en paramètres un brin `b` et un niveau `niv` qui identifient une face. Une précondition à cette fonction est que cette face de niveau `niv` a bien déjà été subdivisée vers le niveau `niv+1`. Chaque brin `tmp` de l'orbite de cette face au niveau `niv` est également un brin de chacune de ses sous-faces de niveau `niv+1`. Pour chacune, on teste si elle a déjà été subdivisée vers le niveau `niv+2`. Cela est effectué de la manière classique déjà décrite précédemment, en testant si la face du brin `tmp` dans la carte de niveau `niv+2` est ou non une face de niveau `niv+2`. Si c'est le cas, c'est que la sous-face de niveau `niv+1` a déjà été subdivisée, et que l'on ne peut pas directement simplifier la face de départ vers le niveau `niv`.

Dans le cadre d'une quadrisection de triangle, le parcours de l'orbite face du brin `b` au niveau `niv` ne permet pas d'atteindre la face centrale issue de la subdivision au niveau `niv+1` afin de tester son éventuelle subdivision vers le niveau `niv+2`. Le test correspondant à cette face est effectué en atteignant le brin `fc`, tel qu'illustré dans la figure 3.27.

La simplification d'une face vers le niveau 0 est illustrée dans la figure 3.28. Les arêtes intérieures sont d'abord tour à tour extraites de leurs permutations  $\beta_1^1$ . Une fois cela effectué, ces brins sont supprimés de la carte. Les brins des sommets subdivisant les arêtes en deux sont ensuite déconnectés des brins de niveau inférieur, puis également supprimés de la carte. Evidemment, ces simplifications d'arêtes n'ont lieu que dans le cas où la face partageant cette

---

**Listing 3.2.3** Exemple d'implantation du test de faisabilité de la simplification d'une face

---

```

bool Carte::facePeutEtreSimplifiee(Brin b, int niv)
{
    bool resultat = true;
    Brin fc = beta_1(beta0(b, niv), niv);
    if(niveauFace(fc, niv+2) == niv+2)
        resultat = false;
    Brin tmp = b;
    do
    {
        if(niveauFace(tmp, niv+2) == niv+2)
            resultat = false;
        tmp = beta1(beta0(tmp, niv), niv);
    } while(tmp != b && resultat);
    return resultat;
}

```

---

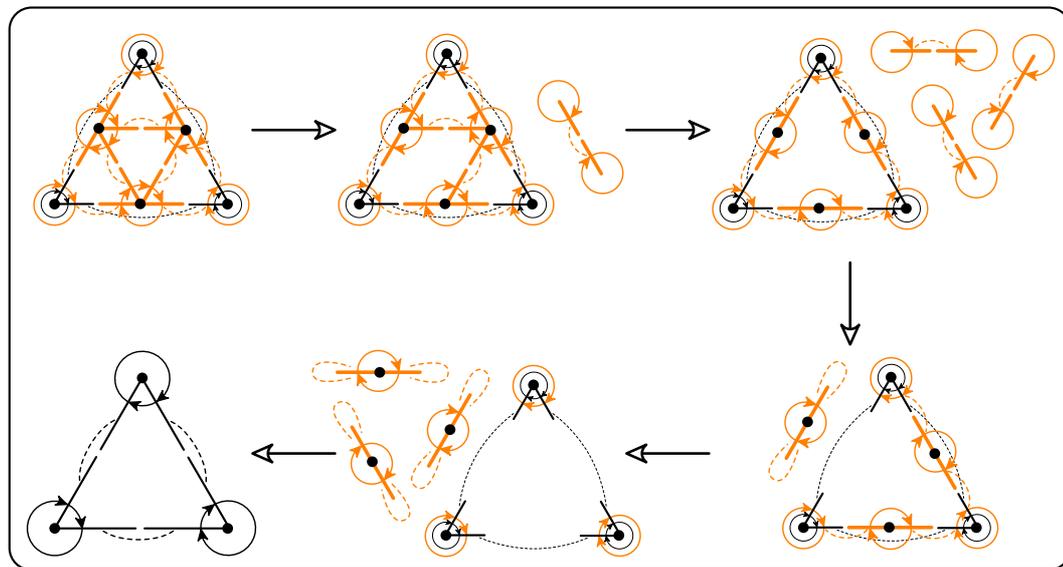
arête n'est elle-même pas subdivisée vers le niveau 1.

Des algorithmes de simplification similaires sont définis dans le cadre de l'utilisation d'un processus de raffinement de Catmull-Clark. Dans le cadre d'un schéma dual de Doo-Sabin, la simplification peut soit être effectuée de la même manière en inversant les opérateurs de raffinement au sein d'une 2-carte duale, soit profiter de la simplification de maillage adaptatif de Catmull-Clark par l'intermédiaire de l'encodage du maillage dual au sein d'une 2-carte primale.

### 3.2.4 Critère de subdivision

Dans le cadre d'une génération adaptative des niveaux de résolution, seules les cellules sélectionnées sont subdivisées d'un niveau au suivant. C'est l'évaluation d'un *critère de subdivision* qui détermine si une cellule est sélectionnée ou non pour être raffinée. Nous n'allons pas entrer ici dans les détails des différents critères qui peuvent être utilisés. Ceux-ci peuvent être par exemple liés aux propriétés géométriques locales de la surface, telles que la courbure ou la distance à la surface limite.

Nous allons en revanche distinguer deux cas de figure pour l'évaluation de ce critère. Soit il est évalué de manière uniforme, c'est-à-dire que pour tout l'objet, le même critère est utilisé pour décider de la subdivision ou non d'une cellule. Soit il est évalué de manière variable et selon les zones de l'objet, des



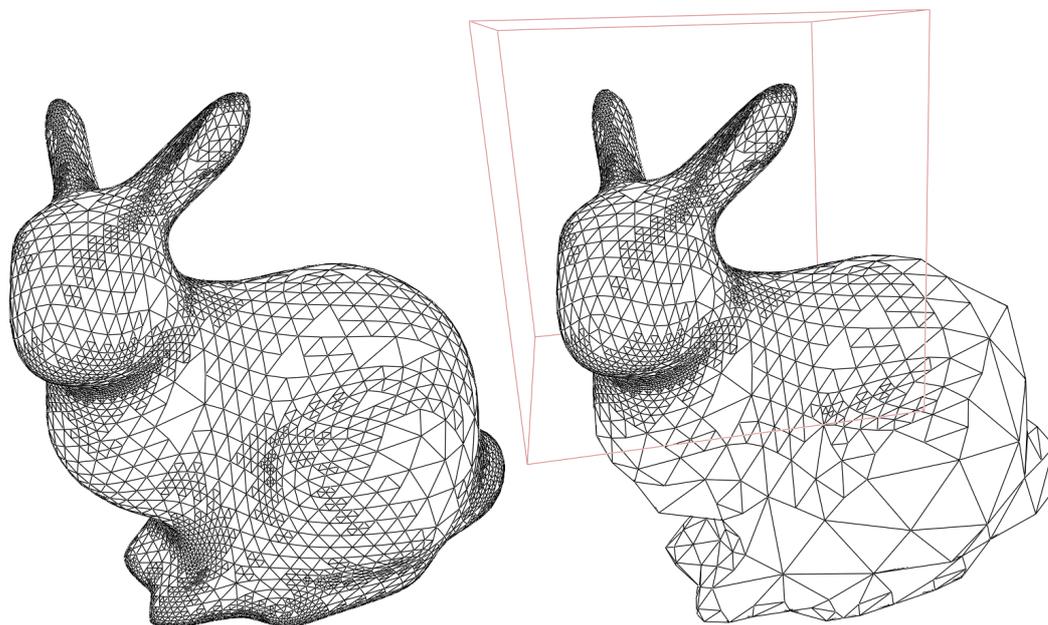
**Figure 3.28** - Simplification d'une face subdivisée par quadrisection de triangles

critères différents pourront être utilisés.

Si le critère de subdivision est uniforme, c'est tout l'objet qui est subdivisé en fonction d'un seul et même critère qui sera évalué de la même manière pour toutes les faces de l'objet. La figure 3.29 illustre à gauche un objet subdivisé en utilisant le même critère pour tout l'objet. Le maillage est mis à jour au fil des déformations en utilisant des algorithmes que nous développerons dans la suite.

Un critère de subdivision variable va pour sa part autoriser une évaluation différente suivant les zones de l'objet. Par exemple toujours dans le cadre d'un critère lié à la courbure, on va, à courbure équivalente, accepter une approximation de la surface plus grossière dans les zones invisibles d'un objet que dans les zones visibles. Les régions cachées vont donc être moins subdivisées que les autres, allégeant ainsi le modèle sans pour autant altérer sa visualisation. La figure 3.29 montre le même objet que précédemment mais cette fois-ci subdivisé en utilisant un critère variable. Une attention plus forte est portée aux parties de l'objet situées dans la boîte pour lesquelles une meilleure approximation de la surface est demandée. Les régions situées en dehors se contentent d'une approximation plus grossière et atteignent donc des profondeurs de subdivision moins élevées. Le maillage est ici mis à jour aussi bien au fil des éditions multirésolution que lors d'un déplacement de la boîte d'intérêt.

Une attention particulière doit être portée sur la simplification lorsqu'on utilise



**Figure 3.29** - Exemple de maillage subdivisé à l'aide d'un critère uniforme (à gauche) ou variable (à droite)

un critère de subdivision variable. Comme on l'a vu précédemment, lors de la phase de synthèse, la géométrie des sommets est calculée comme le résultat de la subdivision auquel on additionne un éventuel vecteur de détail. Quand le critère de subdivision permet plus de simplification dans une zone de l'objet, des sommets auxquels étaient attachés de tels vecteurs de détail peuvent être supprimés et perdus. Si l'on ne se donne pas le moyen de conserver ces vecteurs, au moment où le critère de subdivision demandera à nouveau plus de raffinement dans cette zone, la géométrie reconstruite sera alors simplement égale au résultat de la subdivision. Les vecteurs de détail perdus n'ayant pas été ajoutés, la géométrie de l'objet dans cette zone ne sera plus égale à ce qu'elle était avant la simplification, mais une version lissée de l'objet.

Une première méthode simple pour ne pas perdre d'information lors de la simplification est de ne pas autoriser la simplification d'une cellule si elle porte des vecteurs de détail significatifs à ce niveau. La conséquence est évidemment que le maillage va rester finement subdivisé, même dans des zones de l'objet où le critère de subdivision ne demandait pas une telle profondeur de subdivision.

Une autre méthode est de tout de même simplifier ces faces, mais en encodant les vecteurs de détails au fur et à mesure au sein des faces plus grossières en tant que plongement de face par exemple. La régularité de la subdivision permet de maintenir un codage pour ces vecteurs autorisant par la suite, au

moment où le critère demandera à nouveau une subdivision plus profonde dans cette zone du maillage, de propager correctement les vecteurs de détail et de les ajouter aux sommets adéquats dans les niveaux de résolution plus élevés, permettant alors une reconstruction de la géométrie équivalente à ce qu'elle était avant la simplification.

### 3.2.5 Mise à jour du maillage

Quelle que soit la solution choisie pour l'évaluation du critère de subdivision et suivant les applications, le maillage doit régulièrement être mis à jour. Cette mise à jour concerne aussi bien la géométrie associée aux sommets que la topologie des maillages des différents niveaux de résolution. En effet, lors d'une édition multirésolution de niveau  $i$  par exemple, la géométrie associée aux maillages de niveaux plus élevés doit être mise à jour afin de voir les effets de la déformation induite par l'édition.

Dans le même temps, si le critère de subdivision est lié à la courbure locale, cette dernière va être modifiée par la déformation. Certaines zones de l'objet seront plus fortement courbées, nécessitant alors une profondeur de subdivision plus élevée, quand d'autres le seront moins et se contenteront d'une profondeur de subdivision moindre. De même lors du déplacement d'une zone d'intérêt sur l'objet, certaines parties du maillage entrent dans cette zone tandis que d'autres en sortent, entraînant une modification de l'évaluation du critère de subdivision et donc la profondeur de subdivision à atteindre dans ces zones.

### Mise à jour de la géométrie

L'algorithme 3 réalise la synthèse de la géométrie. Pour cela, les maillages correspondant à chaque niveau de résolution  $i$  de 1 jusqu'au niveau maximum sont traversés. Au sein de chacun de ces maillages on va considérer l'ensemble des sommets de niveau  $i$ , et pour chacun, calculer sa position en tant que résultat de la subdivision de son voisinage au niveau précédent. Si un vecteur de détail était attaché à ce sommet pour ce niveau, on ajoute ce vecteur au résultat de la subdivision. Si on ne souhaite mettre à jour la géométrie que des niveaux de résolution supérieurs à un niveau  $j$  donné, on n'exécute alors simplement cet algorithme qu'entre le niveau  $j$  et le niveau maximum.

L'algorithme 4 réalise quant à lui l'analyse du maillage. Les maillages correspondant aux niveaux de résolution  $i$  à partir de celui précédant le plus fin

```

pour chaque carte de niveau  $i$  de 1 à  $max$  faire
  |
  | pour chaque sommet  $s$  de niveau  $i$  faire
  | | Calculer le résultat de la subdivision du niveau  $i - 1$ ;
  | | Ajouter un éventuel vecteur de détail;
  | | Assigner le résultat à  $s$ ;
  | fin
fin

```

**Algorithme 3** : Algorithme de synthèse

```

pour chaque carte de niveau  $i$  de  $max-1$  à 0 faire
  |
  | pour chaque sommet  $s$  de niveau  $i$  faire
  | | Calculer le résultat de l'analyse du niveau  $i + 1$ ;
  | | Assigner le résultat à  $s$ ;
  | fin
  | Dans la carte de niveau  $i + 1$  : pour chaque sommet  $s'$  de niveau
  |  $i + 1$  faire
  | | Calculer le résultat de la subdivision du niveau  $i$ ;
  | | Retrancher à la position actuelle de  $s'$ ;
  | | Assigner le résultat au vecteur de détail de  $s'$ ;
  | fin
fin

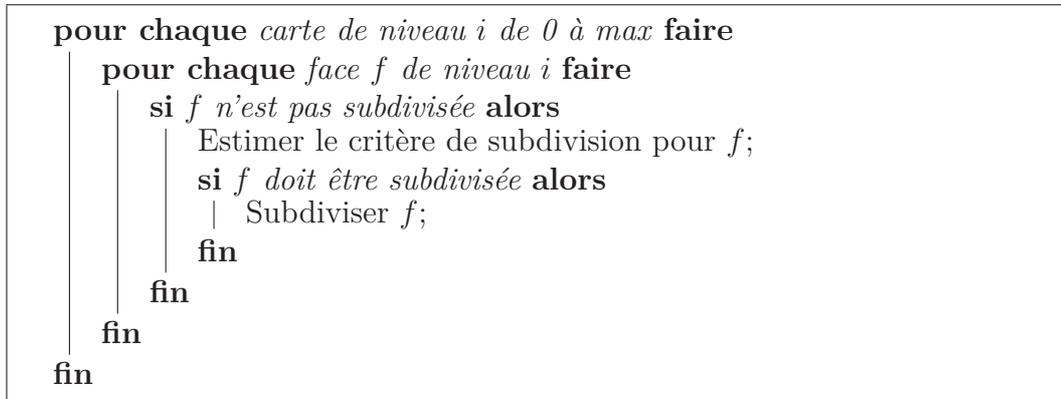
```

**Algorithme 4** : Algorithme d'analyse

jusqu'au plus grossier sont parcourus. Pour chacun, la position des sommets de niveau  $i$  est calculée à partir des sommets de niveau  $i + 1$ . Quand les sommets de niveau  $i$  ont été calculés, les vecteurs de détail de niveau  $i + 1$  sont calculés. Ceux-ci sont égaux à la différence entre le résultat de la subdivision de la nouvelle géométrie du maillage de niveau  $i$  et la position actuelle des sommets au niveau  $i + 1$ , et exprimés dans un repère local. Ici également, si on ne souhaite mettre à jour la géométrie que des niveaux de résolution inférieurs à un niveau  $j$  donné, on n'exécute alors simplement cet algorithme qu'entre le niveau  $j$  et le niveau 0.

### Mise à jour de la topologie

Les algorithmes 5 et 6 permettent de mettre à jour la profondeur de subdivision du maillage afin qu'il soit conforme au critère de subdivision. Le premier algorithme a pour objectif d'augmenter la profondeur de subdivision dans les zones où cela est nécessaire. Pour cela il traverse les maillages de chaque niveau de résolution  $i$  du plus grossier au plus fin et pour chacun, considère toutes les

**Algorithme 5** : Algorithme de subdivision

faces de niveau  $i$  au sein de ce maillage. Au niveau 0, toutes les faces seront considérées; au niveau 1, seules les faces issues de la subdivision au niveau 0 sont considérées; etc. Pour chacune de ces faces, si elle n'est pas encore subdivisée vers le niveau supérieur, le critère de subdivision est estimé, et suivant le résultat, la face est ou non subdivisée. Si elle est subdivisée, ses sous-faces seront à leur tour candidates à la subdivision lors du passage de l'algorithme au niveau de résolution suivant.

Le second algorithme a lui pour objectif de réduire la profondeur de subdivision là où elle n'est plus nécessaire. Pour cela il parcourt les maillages correspondant aux niveaux de résolution  $i$ , à partir de celui précédant le plus fin jusqu'au plus grossier et considère toutes les faces de niveau  $i$  au sein de chacun de ces maillages. Pour chaque face, si elle est déjà subdivisée vers le niveau supérieur et que ses sous-faces ne sont pas subdivisées, on estime le critère de subdivision. Si le résultat est négatif, c'est que cette face n'a plus besoin d'être subdivisée. Cette dernière est alors simplifiée vers le niveau  $i$ .

On peut ajouter à ces algorithmes une restriction supplémentaire imposant qu'au sein du maillage d'un niveau de résolution donné, deux faces voisines aient au maximum un niveau de subdivision de différence. Les maillages ainsi générés sont alors similaires à ceux obtenus dans le cadre des arbres quaternaires restreints que l'on a présenté dans le chapitre 1. Même si cette restriction n'est absolument pas imposée par le modèle des 2-cartes multirésolution, elle apporte un certain nombre d'avantages. Elle permet aux algorithmes de calcul des positions des sommets de toujours disposer du voisinage nécessaire à l'application des masques. Dans le cas où plus de niveaux de différence sont autorisés entre les faces, il manquerait des voisins et la subdivision devrait quand même être exécutée, ne serait-ce que temporairement. D'autre part, cette restriction simplifie l'affichage des maillages adaptatifs, comme on le verra par la suite.

```

pour chaque carte de niveau  $i$  de  $max-1$  à  $0$  faire
  |
  | pour chaque face  $f$  de niveau  $i$  faire
  | |
  | | si  $f$  est déjà subdivisée alors
  | | |
  | | | si les sous-faces de  $f$  ne sont pas subdivisées alors
  | | | |
  | | | | Estimer le critère de subdivision pour  $f$ ;
  | | | | si  $f$  ne doit pas être subdivisée alors
  | | | | |
  | | | | | Simplifier  $f$ ;
  | | | | fin
  | | | fin
  | | fin
  | fin
fin

```

**Algorithme 6** : Algorithme de simplification

Pour intégrer cette restriction, l'algorithme 5 est modifié de la façon suivante : lorsqu'une face  $f$  est sélectionnée pour la subdivision, pour chacune de ses faces voisines, si elle a un niveau de subdivision de moins que  $f$ , alors cette face voisine est d'abord subdivisée. L'algorithme 6 est quant à lui modifié de la façon suivante : lorsqu'une face  $f$  est sélectionnée pour la simplification, pour chacune de ses faces voisines, si elle est déjà subdivisée vers le niveau suivant et que ses sous-faces le sont également, alors on ne simplifie pas  $f$ . Ces modifications entraînent une subdivision plus profonde que celle imposée par l'évaluation du critère de subdivision, mais apportent en contrepartie les avantages que l'on vient de citer.

### 3.2.6 Niveaux de subdivision des cellules

Après avoir utilisé une définition intuitive de ce qu'est le niveau de subdivision d'une cellule au sein du maillage correspondant à un niveau donné d'une 2-carte multirésolution, nous revenons dans cette section sur cette notion et en donnons une définition plus formelle.

#### Cas général

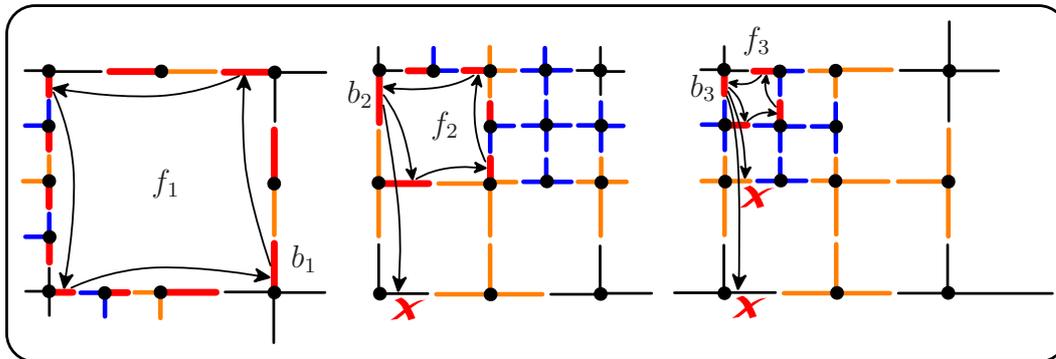
Au sein de la 2-carte courante de niveau  $i$ , on appelle  $cell_k^i(b)$  l'ensemble des brins de la  $k$ -cellule du brin  $b$ . On définit le *niveau de subdivision* de la  $k$ -cellule du brin  $b$  comme étant égal au plus petit niveau  $j$  tel que  $cell_k^j(b') \subset cell_k^i(b)$ , avec  $b' \in cell_k^i(b)$ .

Cette définition sous-entend que le fait d'ajouter des brins dans une cellule ne modifie pas le niveau de subdivision de cette cellule. En revanche, le fait de séparer l'ensemble des brins d'une cellule en plusieurs sous-ensemble distincts fait augmenter ce dernier. Par exemple, dans le cas d'une face au sein d'un maillage adaptatif de niveau  $i$  généré par un algorithme de subdivision primal, le fait de subdiviser ses arêtes aux niveaux supérieurs ne modifie pas le niveau de la face elle-même car cette dernière n'a pas été découpée et on peut toujours en faire le tour. Nous illustrons cela en donnant une procédure permettant de retrouver le niveau de subdivision d'une face dans un maillage adaptatif généré par un algorithme de subdivision primal.

Afin de déterminer le niveau d'une face  $f$  au sein de la carte de niveau  $i$ , on part de l'un de ses brins  $b$  ayant le plus petit niveau d'insertion dans la carte  $niv(b) \leq i$ . En effet, le niveau de cette face est nécessairement compris entre le minimum des niveaux d'insertion de ses brins et le niveau  $i$  de la 2-carte considérée. Afin de tester les niveaux du plus faible au plus élevé, on démarre donc sur l'un des brins présentant le niveau d'insertion le plus faible. On teste en premier lieu si les brins de l'orbite face du brin  $b$  au niveau  $niv(b)$  sont contenus dans l'orbite face de  $b$  au niveau  $i$ . Si oui, alors  $f$  est une face de niveau  $niv(b)$  au sein de la carte de niveau  $i$ . Sinon, on teste si les brins de l'orbite face de  $b$  au niveau  $niv(b) + 1$  sont contenus dans l'orbite de  $b$  au niveau  $i$ . On répète ce processus jusqu'à trouver le plus petit niveau pour lequel tous les brins de l'orbite face de  $b$  sont contenus dans l'orbite de  $f$  au niveau  $i$ . Au maximum, ce niveau est le niveau  $i$  lui-même.

La figure 3.30 illustre différents cas extraits du maillage adaptatif de niveau 2 de la figure 3.33. Pour chaque face considérée, les brins de son orbite sont dessinés en rouge. Pour la face  $f_1$ , on part du brin  $b_1$  inséré dans la carte au niveau 0, puis on commence à parcourir l'orbite face de  $b_1$  au niveau 0. Comme tous les brins de cette orbite sont contenus dans l'orbite face de  $b_1$  au niveau 2, on en conclut que  $f_1$  est une face de niveau 0. Pour la face  $f_2$ , on part du brin  $b_2$  inséré dans la carte au niveau 0. Le brin suivant dans l'orbite face de  $b_2$  au niveau 0 n'est pas contenu dans l'orbite face de  $b_2$  au niveau 2. On passe donc au parcours de l'orbite face de  $b_2$  au niveau 1, on voit que tous les brins de cette orbite sont contenus dans l'orbite face de  $b_2$  au niveau 2, et on en conclut que  $f_2$  est une face de niveau 1. Enfin, le même processus est appliqué pour  $f_3$  et détermine que cette face est de niveau 2.

Cet algorithme de calcul du niveau de subdivision d'une face au sein d'un maillage obtenu par subdivision adaptative avec le schéma de Catmull-Clark fonctionne également dans le cadre de la quadrisection de triangles, et donc en général pour les maillages obtenus grâce à des algorithmes de raffinement primaux.



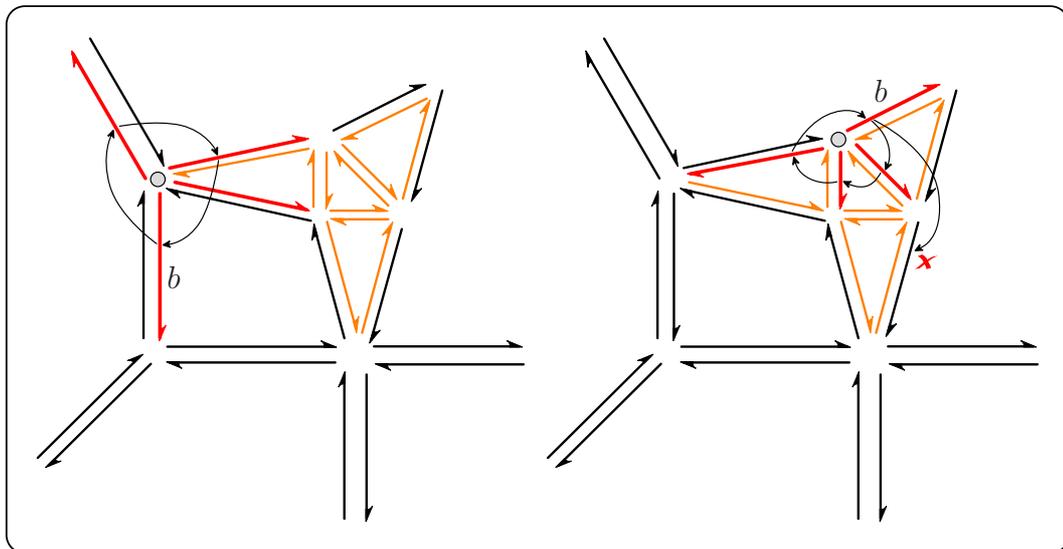
**Figure 3.30** - Illustration pour la formulation générale du calcul du niveau d'une face au sein d'un maillage obtenu par subdivision adaptative à l'aide d'un schéma primal

Le cas du niveau de subdivision d'une arête est très simple. En effet, une arête étant toujours composée d'exactly deux brins, le niveau de subdivision d'une arête est simplement égal au maximum des niveaux d'insertion de ses brins.

Suivant cette définition, le niveau de subdivision d'un sommet est égal, quel que soit le niveau de résolution considéré, au niveau auquel ce sommet a été inséré dans la carte. En effet, dans un processus de subdivision primal, aucun brin n'est jamais inséré dans l'orbite d'un sommet aux niveaux supérieurs à son niveau de création. En revanche, à partir de son niveau d'insertion dans la carte, et pour chaque niveau supérieur où une nouvelle position a été calculée pour ce sommet, les brins du sommet disposent de l'accès à un plongement différent par niveau. Un sommet n'a donc qu'un seul niveau de subdivision (il n'est jamais subdivisé), mais des niveaux de plongement différents.

Dans le cas d'un raffinement dual, la définition du niveau de subdivision des cellules fonctionne de la même façon. Si l'on exécute la subdivision duale au sein d'une 2-carte duale, tel qu'illustré dans la figure 3.31, le niveau de subdivision d'un sommet est retrouvé en utilisant le même algorithme. Ainsi, dans la carte de niveau 1 illustrée ici, le sommet du brin  $b$  a dans le dessin de gauche un niveau de subdivision égal à 0, et à droite un niveau de subdivision égal à 1. Le niveau de subdivision des arêtes est là aussi donné directement par le maximum des niveaux d'insertion des deux brins de l'arête. Les faces n'intervenant pas dans la subdivision, ce sont elles ici qui sont toujours composées du même ensemble de brins, et qui ont donc le même niveau de subdivision à partir du niveau de résolution où elles sont insérées dans la carte.

Si on exécute le raffinement dual en subdivisant avec le schéma de Catmull-Clark une 2-carte primale encodant le dual du maillage, tel qu'illustré dans la



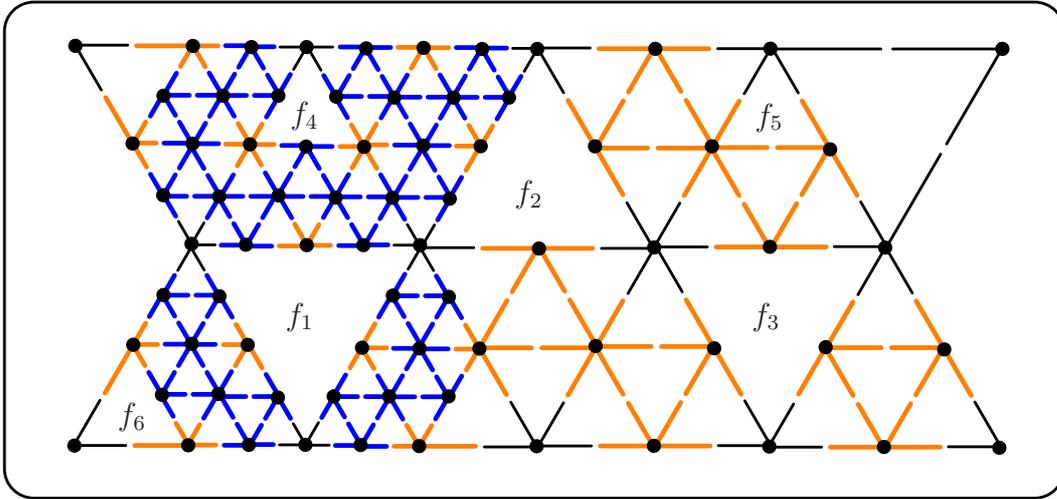
**Figure 3.31** - Illustration pour la formulation générale du calcul du niveau d'un sommet au sein d'un maillage obtenu par subdivision adaptative à l'aide d'un schéma dual

figure 3.26, alors le niveau de subdivision des cellules du maillage est simplement donné en calculant le niveau de subdivision de leur cellule duale correspondante dans le maillage dual. Ainsi, le niveau de subdivision d'un sommet est obtenu en récupérant le niveau de subdivision de sa face correspondante dans le maillage représenté par la 2-carte primale.

### Cas particuliers

Si l'on travaille sur des maillages qui présentent une certaine régularité dans le type de faces qui les composent, tels que des maillages triangulaires subdivisés à l'aide du schéma de Loop ou encore des maillages purement quadrangulaires subdivisés avec le schéma de Catmull-Clark, alors on peut définir d'autres méthodes spécifiques plus directes pour calculer le niveau de subdivision des cellules. Dans le cadre d'une subdivision de Catmull-Clark qui opère sur un maillage de départ polygonal quelconque, la formulation générale du niveau de subdivision d'une cellule est en revanche la seule qui fonctionne dans tous les cas.

Nous nous intéressons tout d'abord au cas d'un maillage triangulaire subdivisé de manière adaptative avec le schéma de Loop. Considérons les faces  $f_1$ ,  $f_2$ ,  $f_3$  et  $f_4$  du maillage adaptatif de niveau 2 illustré dans la figure 3.32. On va distinguer ici deux cas de figure : soit les faces sont régulières, c'est-à-dire



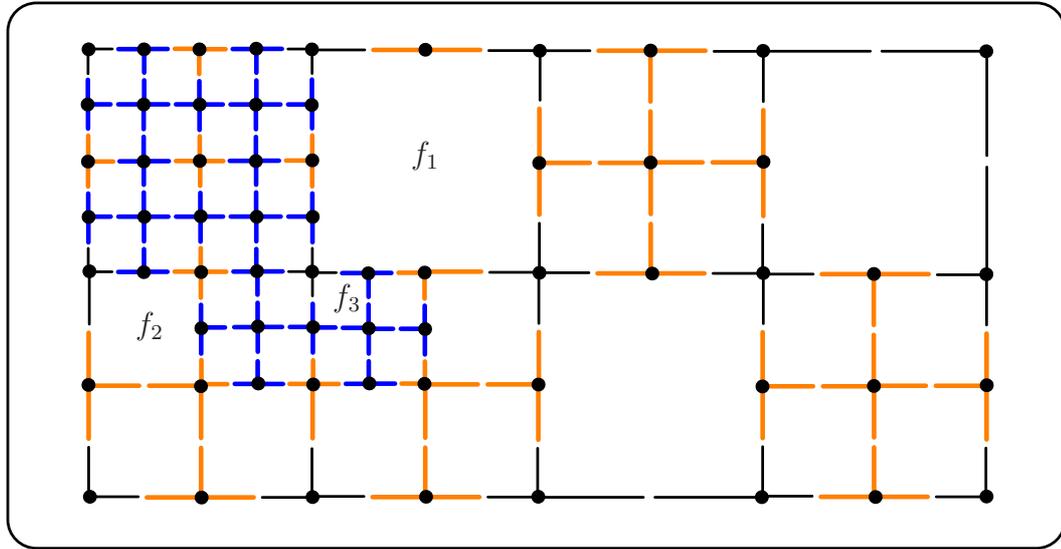
**Figure 3.32** - Cas particuliers du calcul du niveau d'une face au sein d'un maillage obtenu par subdivision adaptative de triangles

composées d'arêtes présentant toutes le même niveau de subdivision, soit elles sont irrégulières.

Les faces  $f_1$ ,  $f_3$  et  $f_4$  sont régulières.  $f_1$  a 12 arêtes de niveau 2,  $f_3$  a 6 arêtes de niveau 1 et  $f_4$  a 6 arêtes de niveau 2. On va exploiter ici le fait que l'on sait que le nombre d'arêtes des faces régulières vaut toujours  $3 * 2^i$  où  $i$  est le niveau des arêtes – ce qui est équivalent au nombre de fois où elles ont été subdivisées. On définit alors le niveau d'une face régulière comme étant égal à  $aNiv - \log_2(\frac{aNb}{3})$ , où  $aNiv$  est le niveau de ses arêtes et  $aNb$  le nombre de ses arêtes. Pour  $f_1$  comme pour  $f_3$ , on retrouve bien 0, et pour  $f_4$  on tombe bien sur 1. Cette formule fonctionne également pour les faces régulières "normales" à trois côtés telle que  $f_5$ .

Pour les faces irrégulières comme  $f_2$  ou  $f_6$  on commence par récupérer le niveau minimum parmi les niveaux de subdivision de ses arêtes, puis on se place dans la carte correspondant à ce niveau. Au sein de cette 2-carte, on se retrouve alors dans le cas d'une face régulière. Pour  $f_2$ , le niveau minimum parmi les niveaux de ses arêtes est égal à 1. Dans la carte de niveau 1, on se retrouve alors dans la même configuration que pour la face  $f_3$  et on trouve donc bien 0 pour le niveau de cette face. Pour  $f_6$ , le niveau minimum parmi les niveaux de ses arêtes est 1, et dans la 2-carte de niveau 1, la face devient régulière à trois côtés ce qui nous amène donc bien à trouver 1 pour le niveau de cette face.

Si on introduit la restriction dans la subdivision imposant à deux faces adjacentes d'avoir au maximum un niveau de différence, alors les faces  $f_1$  et  $f_2$  de la figure 3.32 auraient toutes deux été subdivisées une fois de plus. Le calcul



**Figure 3.33** - Exemple de subdivision adaptative obtenue avec un raffinement consistant en une quadrisection de quadrangles

du niveau des faces est par la même occasion légèrement simplifié, puisque l'unique cas de face régulière ayant plus de trois côtés est celui d'une face régulière ayant 6 côtés, comme la face  $f_3$  de la figure 3.32. Suivant la formule énoncée précédemment pour le calcul du niveau d'une face régulière, le niveau d'une face à 6 côtés est simplement égal à  $aNiv - 1$ , c'est-à-dire un niveau de moins que le niveau de subdivision de ses arêtes.

Dans le cadre d'un maillage composé uniquement de quadrangles et subdivisé à l'aide du schéma de Catmull-Clark, le niveau de subdivision des faces peut être obtenu de la même manière. La figure 3.33 montre un exemple de maillage adaptatif obtenu après l'application adaptative d'un tel raffinement sur un maillage composé de faces quadrilatérales.

La technique permettant de retrouver le niveau de subdivision des faces irrégulières est toujours valide dans ce contexte. On retrouve bien ici 0, 1 et 2 respectivement pour le niveau de subdivision des faces  $f_1$ ,  $f_2$  et  $f_3$ . La seule différence concerne les faces dites régulières. On avait donné une définition qui tenait compte de la connaissance *a priori* du nombre de côtés que peut présenter une face régulière. Si dans le cadre de la quadrisection de triangles, celui-ci est de la forme  $3 * 2^i$ , dans le cadre de maillages issus de quadrisection de quadrangles, celui-ci est de la forme  $4 * 2^i$ . Ainsi, le niveau de subdivision d'une face régulière est égal à  $aNiv - \log_2(\frac{aNb}{4})$ . Une fois ajoutée la même restriction que précédemment concernant la subdivision, l'unique cas de face régulière ayant plus de quatre côtés est celui d'une face présentant 8 côtés. Une nouvelle fois, le niveau d'une telle face régulière est alors simplement égal

à un niveau de moins que le niveau de subdivision de ses arêtes.

Une toute autre solution si l'on ne souhaite pas effectuer ces calculs et qu'en contrepartie on est prêt à utiliser un peu plus de mémoire consiste à stocker le niveau d'une face en tant que plongement de face. Néanmoins, ces calculs ne sont pas très coûteux car ils consistent au pire à faire deux tours de face (dans un cas comme celui de la face  $f_2$ , un tour pour trouver le niveau minimum des arêtes, et un tour pour compter le nombre d'arêtes de la face régulière) accompagné d'un petit calcul entier sur les niveaux des arêtes traversées.

### 3.3 Accès aux niveaux

On va établir dans cette section la distinction entre *construction* régulière ou adaptative, et *accès* régulier ou adaptatif.

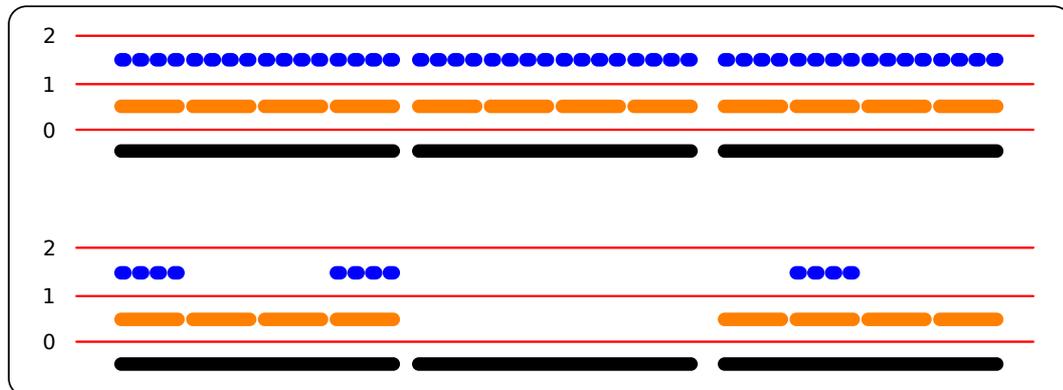
Jusqu'à présent, tous les accès aux maillages des différents niveaux  $i$  que nous avons effectués consistent à parcourir la carte  $C^i$  composée des brins de  $B^i$  liés par les relations de niveau  $i$ . On qualifie cette manière de parcourir les maillages d'*accès régulier*.

Comme on l'a vu en *construisant* ces niveaux de manière adaptative, les maillages représentés par ces cartes ne sont pas forcément composés de cellules partageant toutes la même profondeur de subdivision. Ces différentes profondeurs de subdivision peuvent même être modifiées en faisant varier le critère de subdivision ou les zones d'intérêt de l'objet. Néanmoins il existe des manières d'accéder aux maillages qui restent impossibles que nous allons expliciter maintenant.

#### 3.3.1 Accès régulier

Les figures qui suivent illustrent schématiquement un maillage multirésolution dont les faces de niveau 0 sont en noir, celles de niveau 1 en orange et celles de niveau 2 en bleu. Les traits en rouge illustrent les accès que l'on fait dans ce maillage multirésolution.

La figure 3.34 illustre en quoi consistent les *accès réguliers*. Dans le schéma du haut, les niveaux de résolution ont été construits de façon régulière. Les maillages parcourus par des accès réguliers sont donc composés de cellules présentant toutes la même profondeur de subdivision. Dans le schéma du bas, les



**Figure 3.34** - Schématisation d'un accès régulier dans une 2-carte multirésolution construite de façon régulière (figure du haut) ou adaptative (figure du bas)

niveaux de résolution ont été construits de façon adaptative. Des accès réguliers permettent donc de parcourir des maillages présentant des profondeurs de subdivision différentes.

La restriction imposée par l'accès régulier est que si l'on souhaite accéder au niveau 2 dans une certaine zone de l'objet, on accèdera alors aussi automatiquement au niveau 2 s'il existe dans tout le reste de l'objet. Si cela n'est pas souhaitable, la seule solution est alors de simplifier le maillage dans les zones où l'on veut accéder à une profondeur de subdivision moins importante. Cela peut être obtenu par une modification du critère de subdivision dans ces zones suivie d'une mise à jour de la subdivision. Même si la simplification effectuée permet en outre une économie de mémoire, les traitements qu'elle impose ne sont pas toujours souhaitables.

### 3.3.2 Accès adaptatif

L'accès adaptatif consiste à accéder aux profondeurs de subdivision voulues suivant les zones de l'objet sans avoir pour autant à simplifier ou subdiviser la structure sous-jacente. La figure 3.35 illustre des exemples d'accès adaptatif dans un maillage multirésolution. Dans le schéma du haut, les niveaux de résolution ont été construits de façon régulière. On accède cependant au même maillage que celui correspondant à l'accès régulier de niveau 2 sur l'objet construit de façon adaptative du schéma du bas de la figure 3.34. Les autres schémas montrent que l'on peut également effectuer des accès adaptatifs sur des maillages construits de façon adaptative.



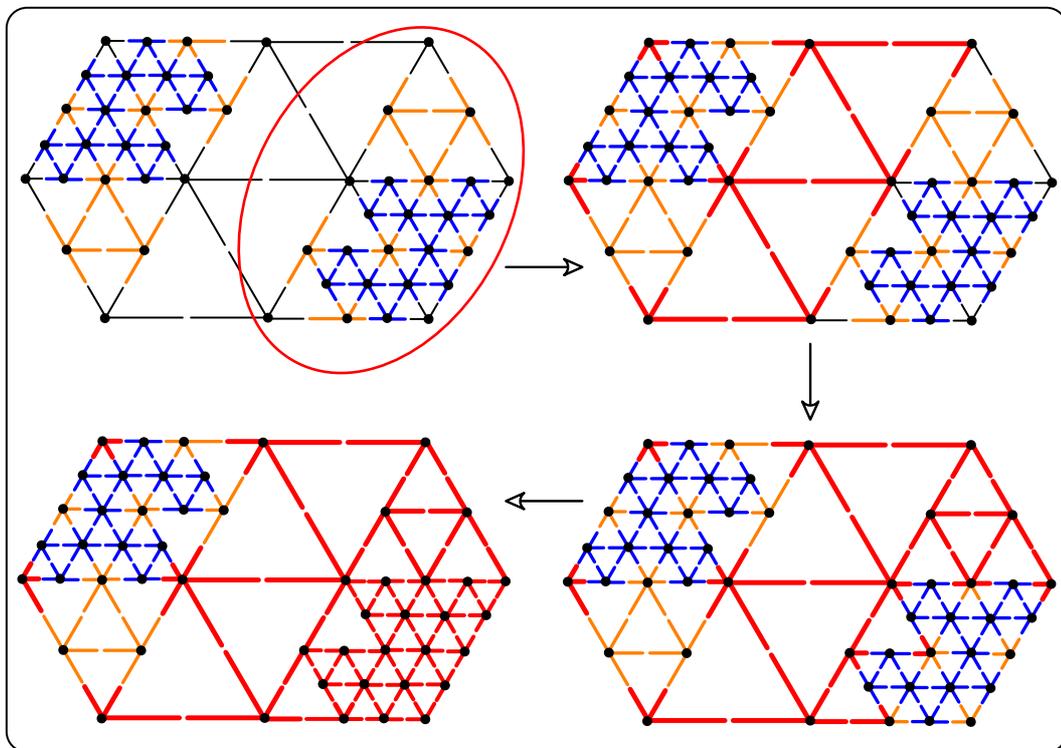
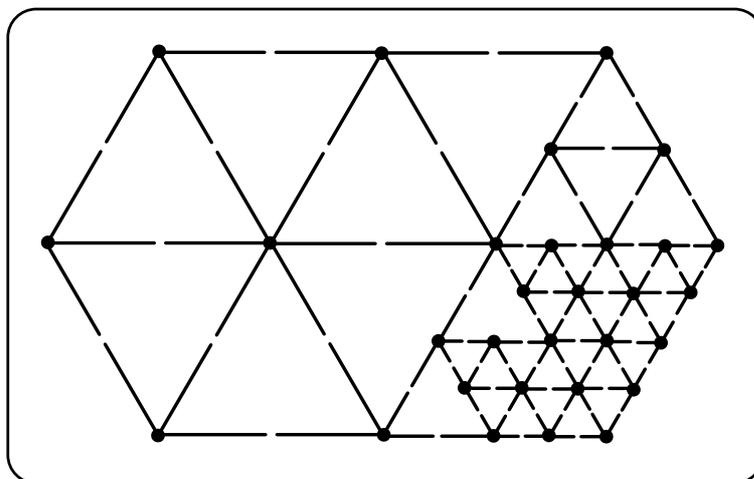


Figure 3.36 - Déroulement de l'algorithme d'accès adaptatif dans une 2-carte multirésolution



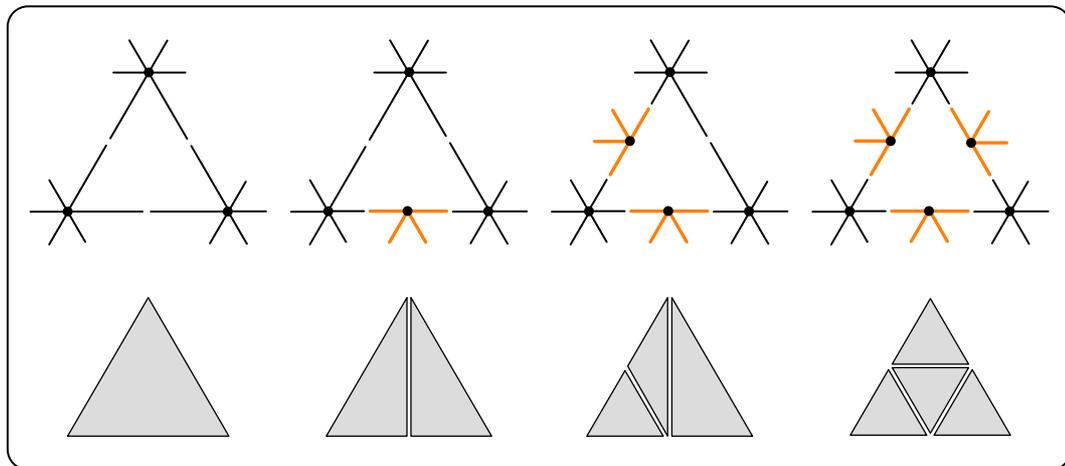
**Figure 3.37** - Résultat de l'accès adaptatif dans une 2-carte multirésolution

s'arrêter le plus tôt possible à l'extérieur. Dans les trois autres figures, on voit le déroulement de l'algorithme en parallèle pour chacune des faces de niveau 0, les arêtes marquées au fur et à mesure étant indiquées en rouge.

On voit dans la figure d'en-haut à droite que les arêtes ont été marquées au niveau 0 pour toutes les faces étant en-dehors de la zone d'intérêt. Pour les autres faces, le parcours en profondeur continue au niveau suivant. La figure d'en-bas à droite montre la suite de l'algorithme au niveau 1. Les faces situées en-haut à droite du maillage n'étant pas subdivisées plus profondément, leurs arêtes sont marquées au niveau 1. Les brins déjà marqués précédemment pour être affichés au niveau 0 sont maintenant marqués pour le niveau 1. Pour les autres faces, l'algorithme continue au niveau 2. La dernière figure en-bas à gauche illustre la fin de l'algorithme et l'ensemble des brins sélectionnés.

Une fois cette étape de marquage terminée, il faut dans un deuxième temps afficher le maillage représenté par l'ensemble des brins marqués et leurs relations aux niveaux sélectionnés. On peut pour cela parcourir l'ensemble des brins de la carte et ne considérer que le sous-ensemble de brins et leurs relations sélectionnés. L'affichage se fait alors exactement de la même façon qu'au sein d'une 2-carte représentant un accès régulier à un niveau de résolution.

La figure 3.37 montre la 2-carte résultant de l'accès adaptatif sur la 2-carte multirésolution de la figure 3.36. Ce maillage est le résultat d'un accès *adaptatif* sur une 2-carte multirésolution construite par subdivision *adaptive*. Si on utilise plusieurs marqueurs différents, on peut également maintenir plusieurs accès adaptatifs simultanément sur une même 2-carte multirésolution.



**Figure 3.38** - Gestion de la triangulation du maillage lors de l'affichage d'une 2-carte générée par l'application adaptative du schéma de Loop

### 3.3.3 Affichage d'une 2-carte adaptative

Le matériel graphique actuel est conçu et optimisé pour afficher des simplexes tels que des points, des segments ou des triangles. Dans le cadre de l'affichage d'une surface représentée par un maillage et quel que soit le type de faces dont est composé ce maillage, c'est finalement toujours un ensemble de triangles qui est rendu à l'écran. Le triangle est en effet le seul polygone dont la planarité dans un espace de plongement comme  $\mathbb{R}^3$  est toujours assurée.

Lorsque l'on travaille sur un maillage triangulaire subdivisé de manière régulière avec le schéma de Loop, et que l'on souhaite afficher le maillage correspondant à un niveau de résolution donné auquel on accède de manière régulière, alors le maillage à afficher ne sera composé que de triangles. En revanche, si la subdivision est appliquée de manière adaptative, ou si l'on travaille sur un maillage polygonal quelconque subdivisé avec le schéma de Catmull-Clark, alors le maillage à afficher ne sera pas composé uniquement de triangles.

En effet, les faces situées à la frontière entre les différentes profondeurs de subdivision du maillage auront, suivant les cas de figure, un nombre variable de côtés. Si l'on applique la restriction imposant qu'au sein du maillage d'un niveau de résolution donné, deux faces voisines aient au maximum un niveau de subdivision de différence, alors le nombre de cas à traiter chute considérablement. En effet, dans le cadre d'une subdivision de Loop appliquée sur des triangles, les seules faces non-triangulaires rencontrées présenteront 4, 5 ou 6 côtés. On pourrait envoyer directement ces faces polygonales au matériel graphique et le laisser choisir une triangulation, mais cette dernière ne serait

```

pour chaque face  $f$  de la 2-carte de niveau  $i$  faire
  si  $nbCôtés(f) == 3$  alors
    | parcourir les sommets et afficher un triangle;
  fin
  sinon
    | remplir le polygone avec des triangles;
  fin
fin

```

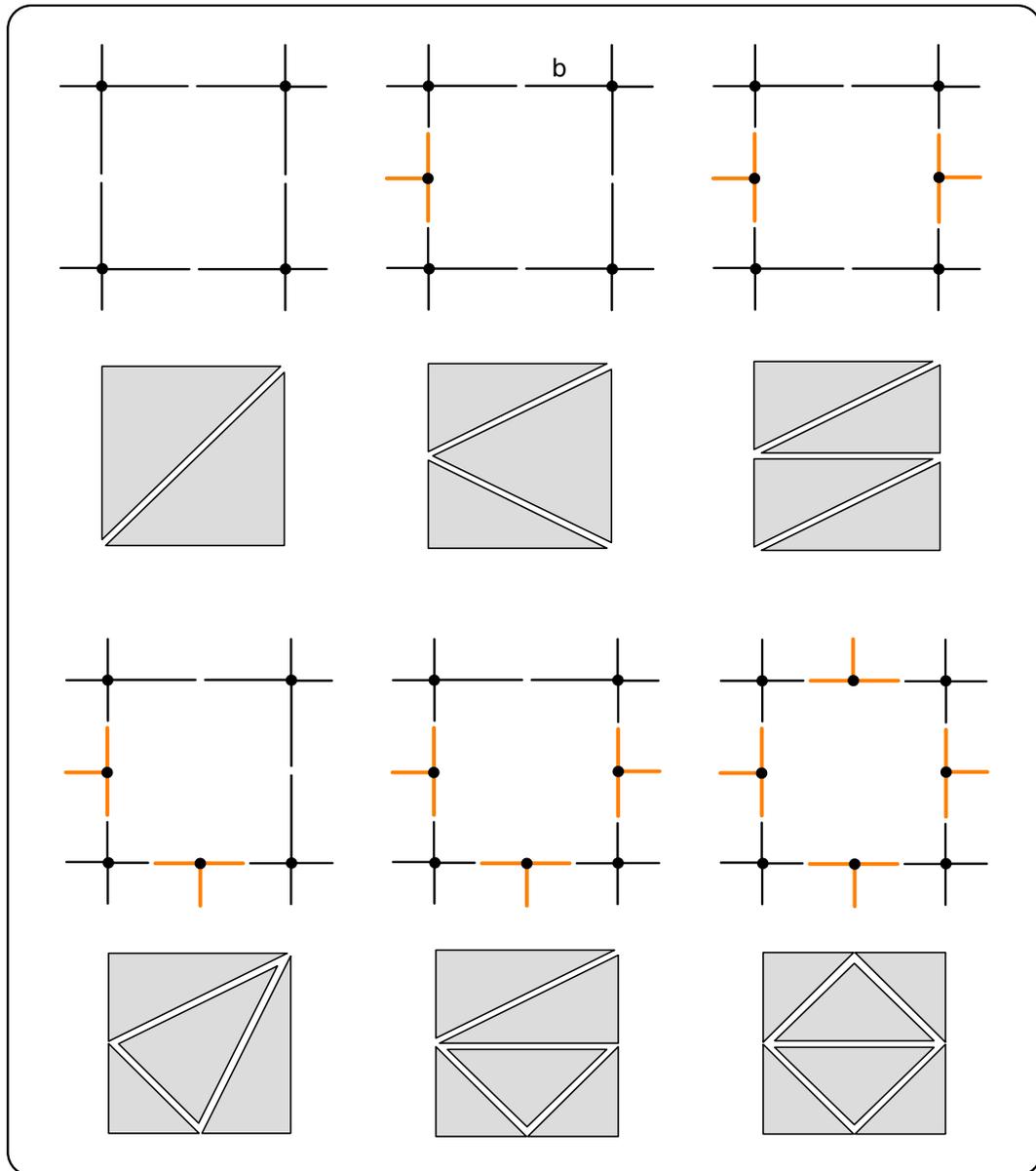
**Algorithme 7** : Affichage du maillage correspondant à un niveau de résolution donné

pas forcément la meilleure, surtout dans notre cas où ces polygones ont certains de leurs sommets quasiment alignés. Comme le nombre de cas rencontrés est faible, il est possible d'énumérer la meilleure triangulation à effectuer pour chacune de ces faces polygonales. La figure 3.38 illustre tous ces cas de figure, avec pour chacun l'ensemble des triangles à afficher pour le remplissage du polygone.

Dans le cadre d'une subdivision de Catmull-Clark appliquée à un maillage polygonal, la plupart des faces seront des quadrangles (l'application d'un pas de subdivision régulier génère un maillage entièrement quadrangulaire), tandis que les autres faces présenteront de 5 à 8 côtés. Ici aussi on peut énumérer la meilleure triangulation à effectuer pour chacune de ces faces polygonales, illustrées dans la figure 3.39.

L'affichage de la 2-carte correspondant à un niveau de résolution  $i$  donné se fait alors tel qu'illustré dans l'algorithme 7. Pour chacune des faces de la 2-carte de niveau  $i$  que l'on souhaite afficher, si c'est un triangle alors on fait simplement le tour de la face et on affiche le triangle, sinon on génère l'ensemble de triangles correspondant au cas rencontré. L'avantage ici par rapport aux structures basées sur des arbres est qu'il n'est pas nécessaire d'aller vérifier le niveau de subdivision des faces voisines pour savoir le cas dans lequel on se trouve. Il suffit ici de faire le tour des arêtes de la face à afficher, ce qui est de toute manière inévitable, ne serait-ce que pour récupérer les coordonnées des sommets à afficher.

Parmi les cas rencontrés dans le cadre d'une subdivision de Loop et ceux rencontrés dans le cadre d'une subdivision de Catmull-Clark, certaines faces polygonales présentent le même nombre de côtés tandis que leur triangulation n'est pas similaire. Ainsi, si on rencontre une face à 4 côtés, on peut être soit dans le deuxième cas de la figure 3.38, soit dans le premier cas de la figure 3.39. On peut distinguer ces différents cas simplement en effectuant un tour



**Figure 3.39** - Gestion de la triangulation du maillage lors de l'affichage d'une 2-carte générée par l'application adaptative du schéma de Catmull-Clark

---

**Listing 3.3.1** Exemple d'implantation de l'algorithme d'affichage d'une face à 5 côtés

---

```

bool Carte::afficheFace(Brin b, int niv)
{
    Brin tmp = b;
    std::vector<Point3D*> pos;
    do
    {
        p.push_back(pos(tmp, niv));
        tmp = beta1(beta0(tmp, niv), niv);
    } while(tmp != b);
    glBegin(GL_TRIANGLES);
    glVertex(p[0]->x, p[0]->y, p[0]->z);
    glVertex(p[1]->x, p[1]->y, p[1]->z);
    glVertex(p[2]->x, p[2]->y, p[2]->z);
    glVertex(p[0]->x, p[0]->y, p[0]->z);
    glVertex(p[2]->x, p[2]->y, p[2]->z);
    glVertex(p[4]->x, p[4]->y, p[4]->z);
    glVertex(p[2]->x, p[2]->y, p[2]->z);
    glVertex(p[3]->x, p[3]->y, p[3]->z);
    glVertex(p[4]->x, p[4]->y, p[4]->z);
    glEnd();
}

```

---

de face et en exploitant le niveau de subdivision des arêtes qui composent ces faces. En effet, on voit que dans le premier cas, deux arêtes sont de niveau  $i$  et les deux autres de niveau  $i + 1$ , tandis que dans le deuxième cas, toutes les arêtes sont de même niveau. De la même manière, en regardant le niveau de subdivision des arêtes on va pouvoir distinguer les deux cas correspondant à une face à 5 et à 6 côtés, et ainsi pouvoir générer l'ensemble de triangles adéquat.

Par exemple pour le cas d'une face à 5 côtés dont trois arêtes sont de niveau  $i$  et deux sont de niveau  $i + 1$ , un exemple de fonction permettant l'affichage des triangles correspondants est donné dans le listing 3.3.1. On fait l'hypothèse ici que le brin passé en paramètre est le brin  $b$  illustré dans la figure 3.39. Il est aisé de toujours satisfaire cette hypothèse en exploitant l'analyse précédente des niveaux de subdivision des arêtes. Le paramètre `niv` correspond au niveau de la 2-carte que l'on est en train d'afficher.

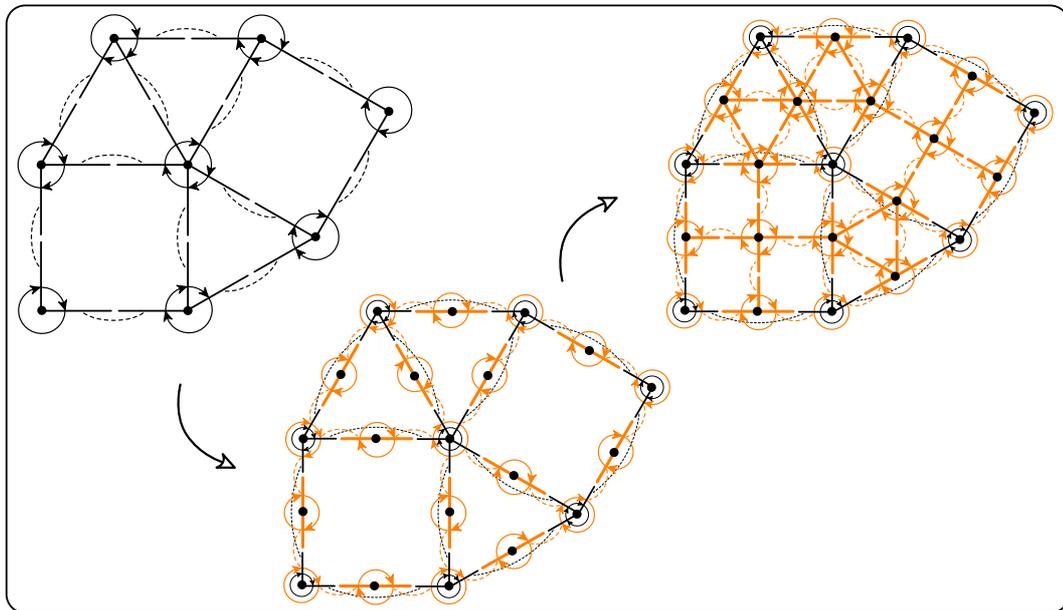


Figure 3.40 - Application régulière du schéma de subdivision quad/triangle

### 3.4 Schémas de subdivision originaux

Dans cette section nous allons illustrer comment la généralité du modèle des 2-cartes multirésolution permet de l'utiliser comme support topologique pour la représentation de maillages multirésolution dont les niveaux sont générés par l'application de schémas de subdivision qui ne pas supportés directement par les structures de données classiques.

Nous utilisons ici les 2-cartes multirésolution dans le cadre des schémas quad / triangle [SL03] et  $\sqrt{3}$  [Kob00] qui opèrent dans un cas une combinaison de raffinements que nous avons vus précédemment et dans l'autre un raffinement totalement différent.

#### 3.4.1 Le schéma Quad/Triangle

Le schéma quad/triangle [SL03] propose d'unifier les schémas de subdivision primaires triangulaire et quadrangulaire de Loop [Loo87] et de Catmull-Clark [CC78]. Les auteurs de ce schéma partent de deux observations. Tout d'abord, de nombreux maillages conçus manuellement sont composés de zones plus naturellement quadrangulaires ainsi que de zones plus naturellement triangu-

lares. Deuxièmement, l'utilisation d'un schéma triangulaire sur un maillage polygonal triangulé, tout comme celle d'un schéma quadrangulaire sur un maillage triangulaire, aboutit à des surfaces limites de moins bonne qualité pouvant laisser apparaître des oscillations non désirées.

Dans le schéma de subdivision quad/triangle, le raffinement de la topologie consiste en de la quadrissection de face, en conservant des triangles dans les zones triangulaires, et des quadrangles dans le reste du maillage. La gestion des frontières entre les zones triangulaires et quadrangulaires ne pose pas de problème particulier. En effet, indépendamment de la manière dont sont insérées les nouvelles arêtes à l'intérieur de la face subdivisée, dans un cas comme dans l'autre, les arêtes formant la frontière de la face sont subdivisées de la même manière. Pour le calcul de la géométrie, différentes stratégies peuvent être appliquées. En général, les masques de subdivision de Loop et de Catmull-Clark sont utilisés directement dans les zones triangulaires et quadrangulaires. En revanche, des masques particuliers sont développés pour les sommets situés à la frontière entre ces zones. Plus de détails sur les développements récents de ces masques peuvent être trouvés dans [SL03, PS04, SW05].

La figure 3.40 illustre l'application régulière du schéma de subdivision quad/triangle sur une 2-carte encodant un maillage composé de trois triangles et deux quadrangles. Lors de la première étape, aucune différence n'est faite entre les faces, et toutes les arêtes du maillage sont subdivisées vers le niveau 1. La deuxième étape est différente suivant les faces. Pour les faces présentant 6 côtés dans la 2-carte de niveau 1 (correspondant aux triangles du niveau 0), trois nouvelles arêtes sont insérées entre les nouveaux sommets résultant en une quadrissection du triangle original. Pour les faces présentant plus de 6 côtés (correspondant aux polygones de plus de 3 côtés du niveau 0), un sommet est inséré au centre de la face et de nouvelles arêtes sont insérées pour relier ce sommet aux autres nouveaux sommets.

Une subdivision adaptative peut ici aussi être appliquée au maillage. La figure 3.41 illustre un exemple de maillage de niveau 2 subdivisé de manière adaptative avec le schéma quad/triangle. Par rapport aux raffinements des schémas présentés dans la section précédente, deux types de raffinements différents sont appliqués ici suivant le type de face. Avant de subdiviser une face, il faut donc tout d'abord déterminer le type de raffinement que l'on va lui appliquer. Dans le cadre du schéma quad/triangle, il faut pour cela déterminer si la face concernée est issue d'un triangle ou d'un polygone à plus de 3 côtés. Si dans le cas régulier il suffit de compter le nombre de côtés de la face à subdiviser, les choses ne sont pas aussi simple dans le cas adaptatif où les faces peuvent avoir un nombre de côtés irrégulier.

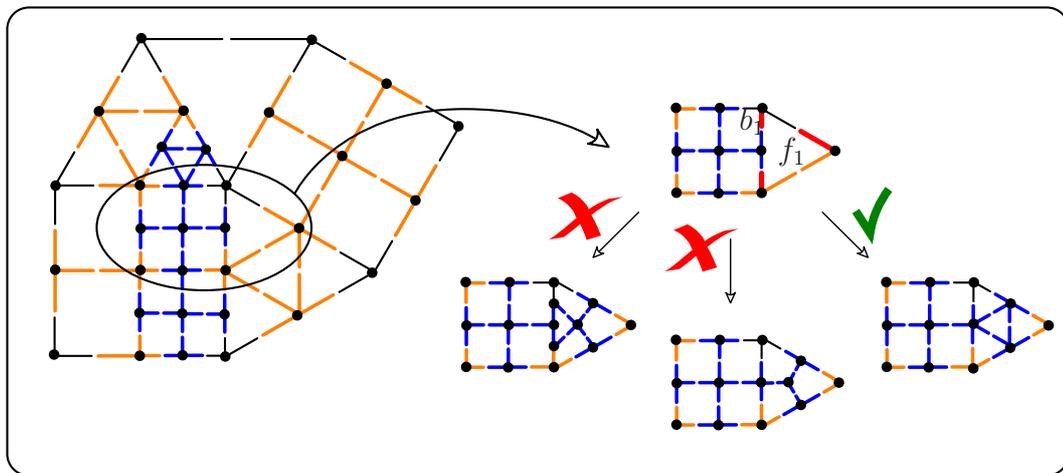


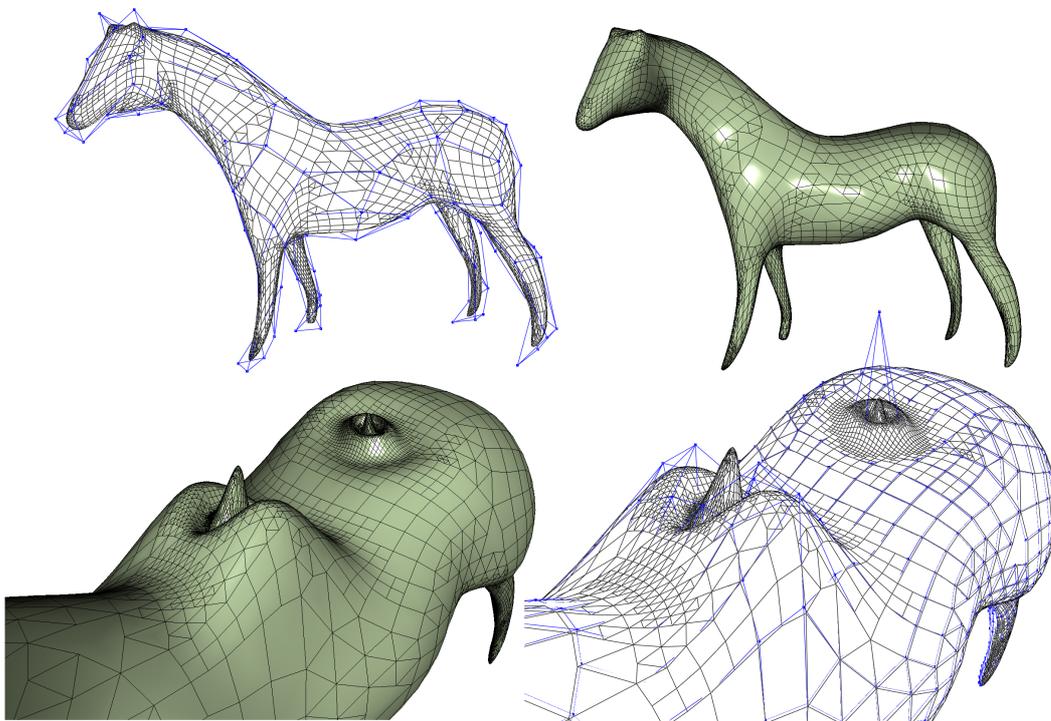
Figure 3.41 - Application adaptative du schéma de subdivision quad/triangle

Les définitions que nous avons données dans la section précédente pour déterminer le niveau de subdivision des arêtes et des faces au sein d'un maillage adaptatif montrent ici à nouveau tout leur intérêt. Celles-ci étant indépendantes du type de maillage et de l'algorithme de raffinement utilisé, la procédure illustrée dans la figure 3.30 est toujours valide pour n'importe quelle face d'un maillage obtenu par subdivision adaptative avec le schéma quad/triangle.

Le niveau de subdivision d'une face va nous permettre de déterminer le nombre de côtés de la face dont est issue une face  $f$ . Celui-ci est obtenu en prenant le brin  $b$  de  $f$  ayant le plus petit niveau d'insertion  $niv(b)$  dans la carte, puis en comptant le nombre de côtés de l'orbite face de  $b$  au sein de la 2-carte de niveau  $niv(b)$ .

La face  $f_1$  de la figure 3.41 est une face ayant un niveau de subdivision égal à 1 au sein du maillage de niveau 2. Bien qu'étant une face 4 côtés au sein de la 2-carte de niveau 2, en partant du brin  $b_1$  présentant le plus petit niveau d'insertion, on détermine que la face de  $b_1$  au niveau 1 est un triangle. Les brins de cette face dans la carte de niveau 1 sont ici dessinés en rouge. Ce test permet donc de savoir qu'il faut lui appliquer une quadrisection de triangle lors de sa subdivision vers le niveau 2.

La figure 3.42 montre un exemple de maillage obtenu par subdivision adaptative avec le schéma quad/triangle, avant et après des éditions effectuées à différents niveaux de résolution.



**Figure 3.42** - Exemple de maillage obtenu par une subdivision adaptative avec le schéma quad/triangle

### 3.4.2 Le schéma $\sqrt{3}$

Le schéma  $\sqrt{3}$  proposé dans [Kob00] effectue une subdivision sur des maillages triangulaires. L'opération de raffinement topologique consiste en la procédure suivante : une division en 3 de chaque triangle du maillage est effectuée en insérant un nouveau sommet en son centre ; cela introduit trois nouvelles arêtes connectant le nouveau sommet central aux anciens sommets du triangle ; les arêtes formant les triangles du maillage de départ sont basculées pour régulariser la valence des sommets.

La figure 3.43 illustre l'application régulière de ce schéma de subdivision dans un maillage triangulaire. On peut reprendre ici la terminologie de sommets pairs et impairs, les sommets impairs étant les nouveaux sommets insérés au centre des faces lors de la subdivision (en orange dans la figure), et les sommets pairs ceux du maillage précédent conservés d'un niveau à l'autre (en bleu dans la figure).

La géométrie associée aux sommets est quant à elle calculée en utilisant les masques illustrés dans la figure 3.44. Le masque de gauche sert au calcul de la position des sommets impairs, et celui de droite au calcul de la nouvelle position

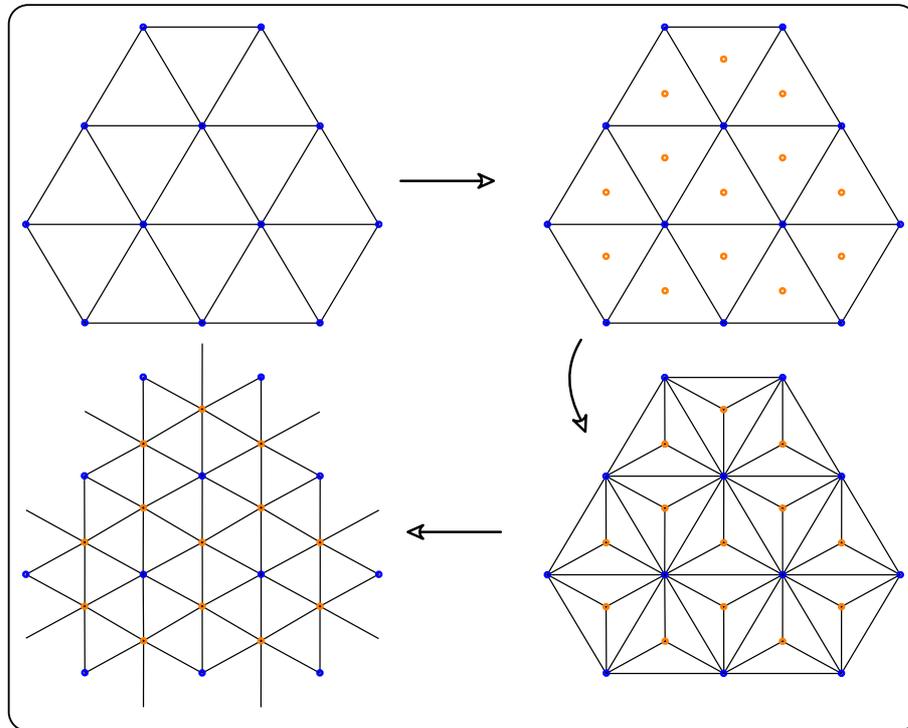


Figure 3.43 - Processus de raffinement de la topologie dans le schéma de subdivision  $\sqrt{3}$

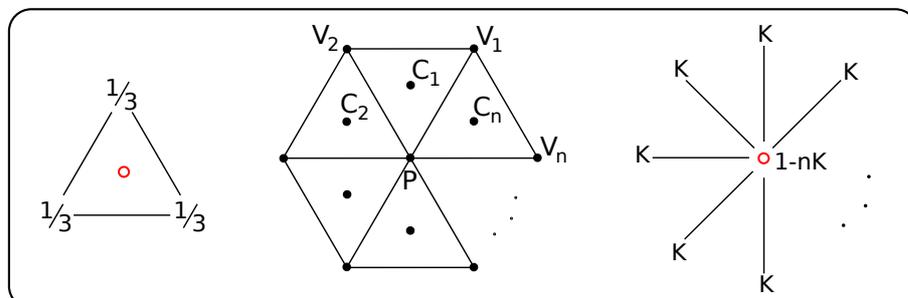
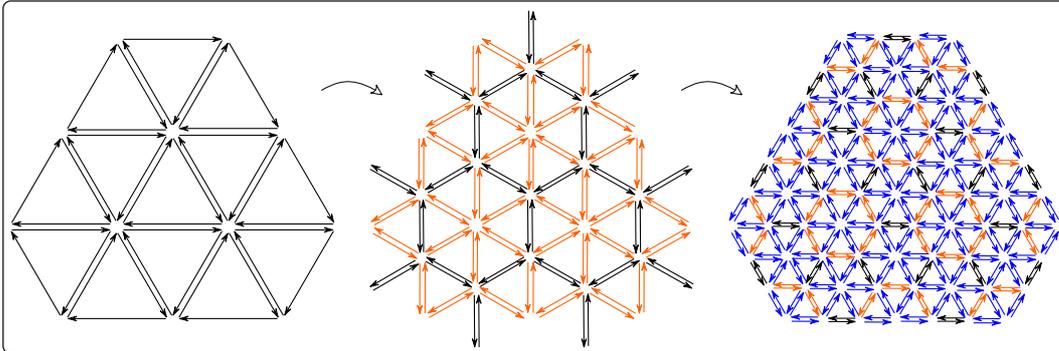


Figure 3.44 - Masques utilisés dans le schéma de subdivision  $\sqrt{3}$



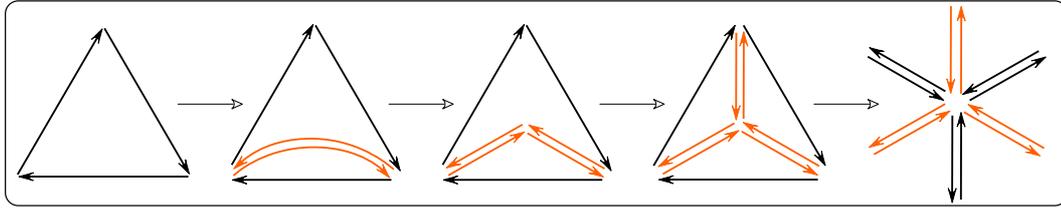
**Figure 3.45** - Deux pas de subdivision avec le schéma  $\sqrt{3}$  au sein d'une 2-carte multirésolution duale

des sommets pairs (le schéma  $\sqrt{3}$  est approximant). Le poids  $K$  appliqué aux sommets voisins est proposé dans [Kob00] comme étant égal à  $\frac{4 - 2 \cos \frac{2\pi}{n}}{9}$ , où  $n$  est la valence du sommet. On peut exprimer les positions des sommets de la façon suivante :  $C_i = \frac{V_i + V_{(i+1)\%n} + P}{3}$  et  $P' = K \frac{1}{n} \sum_{i=0}^n V_i + (1 - K)P$  où  $P'$  est la nouvelle position du sommet pair  $P$ .

La figure 3.45 illustre un exemple de 2-carte multirésolution duale dont les niveaux de résolution sont générés par l'application du schéma de subdivision  $\sqrt{3}$ . Les deux premiers niveaux de résolution sont ici illustrés. Pour chaque triangle, trois nouvelles arêtes sont insérées d'un niveau au suivant. La figure 3.46 illustre la manière dont ce raffinement est exécuté en utilisant les opérateurs définis dans le chapitre 2. On applique d'abord une opération d'insertion d'arête entre deux sommets du triangle. Cela a pour effet de découper temporairement le triangle en une face à deux côtés et une face à trois côtés. Un nouveau sommet est ensuite inséré au milieu de cette nouvelle arête, puis une nouvelle arête est insérée entre ce nouveau sommet et le troisième sommet du triangle de départ.

Dans la deuxième phase du raffinement, toutes les anciennes arêtes du maillage sont basculées. Cela est également exécuté en utilisant l'opérateur de basculement d'arête décrit dans le chapitre 2.

On peut remarquer plusieurs choses concernant les brins des cellules du maillage après l'application du raffinement. On note tout d'abord que la liaison  $\phi_2$  d'un brin n'est jamais modifiée lors de la subdivision, contrairement aux autres schémas vus précédemment où c'était l'involution qui était modifiée de niveau en niveau et la permutation qui demeurait inchangée. Du point de vue de l'implantation, c'est ici le tableau contenant les coutures pour l'involution



**Figure 3.46** - Insertion de nouvelles arêtes dans un triangle dans le processus de raffinement du schéma  $\sqrt{3}$

$\phi_2$  qui ne grandit jamais et conserve une taille constante égale à 1.

A chaque niveau de résolution, les sommets pairs sont toujours composés uniquement de brins nouvellement insérés et conservent leur valence. En effet, dans un maillage fermé, tout sommet a une valence égale au nombre de ses faces incidentes, et les sommets pairs sont constitués d'une arête pour chaque face incidente dans le maillage de départ. Les sommets impairs sont eux toujours de valence 6 et sont composés de trois nouveaux brins et de trois brins provenant des niveaux précédents. Au niveau des faces, on note que deux applications de ce schéma reviennent à subdiviser chaque face de départ en 9, subdivisant chaque arête d'origine en 3.

Il existe une autre façon d'appréhender ce raffinement qu'il est aisé de voir à l'aide des 2-cartes. Étant donné un maillage triangulaire, on considère son maillage dual. Comme on l'a vu dans le chapitre 1, celui-ci peut être obtenu très facilement, sa structure combinatoire étant incluse dans celle de la 2-carte du maillage de départ. Les sommets de ce maillage dual sont les sommets impairs du raffinement  $\sqrt{3}$ . D'après les masques de subdivision la position de ces sommets est simplement égale au barycentre de leur triangle correspondant dans le maillage primal. Les faces de ce maillage dual ont un nombre de côté égal à la valence de leur sommet correspondant dans le maillage primal. Pour obtenir le résultat du raffinement  $\sqrt{3}$  il suffit ensuite de trianguler ce maillage dual en ajoutant un sommet au milieu de chaque face. Ces sommets centraux sont les sommets pairs du raffinement  $\sqrt{3}$ . Leur position peut maintenant être exprimée en fonction de la position des sommets impairs déjà calculés. En reprenant la terminologie introduite dans la figure 3.44, on a  $\sum_{i=0}^n C_i = \frac{2 \cdot \sum_{i=0}^n V_i + n \cdot P}{3}$ , d'où on déduit  $\sum_{i=0}^n V_i = \frac{3 \cdot \sum_{i=0}^n C_i - n \cdot P}{2}$ .  
Finalement, on obtient  $P' = K \cdot \frac{1}{n} \cdot \frac{3 \sum_{i=0}^n C_i - n \cdot P}{2} + (1 - K) \cdot P$ .

Le listing 3.4.1 montre un exemple de code pour le calcul des positions des sommets. La première fonction calcule la position d'un sommet impair identifié par un brin `b` et un niveau de résolution `niv`. D'après le masque de subdivision,

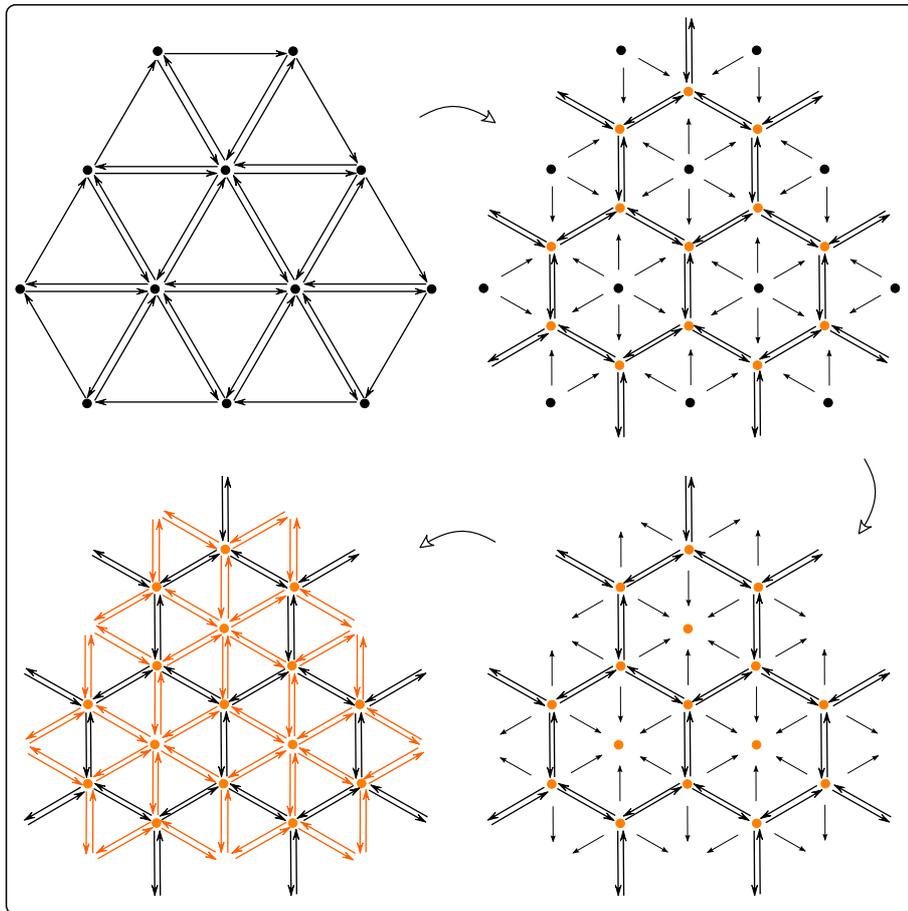


Figure 3.47 - Subdivision  $\sqrt{3}$  en deux temps en passant par le maillage dual

---

**Listing 3.4.1** Exemple d'implantation du calcul des positions des sommets dans le schéma de subdivision  $\sqrt{3}$

---

```

Point3D Carte::positionImpairSqrt3(Brin b, int niv)
{
    Point3D resultat;
    Brin old = ancienBrin(b, niv);
    Brin tmp = old;
    do
    {
        resultat += 1.0/3.0 * pos(tmp, niv - 1);
        tmp = phi1(tmp, niv - 1);
    } while (tmp != old);
    return resultat;
}

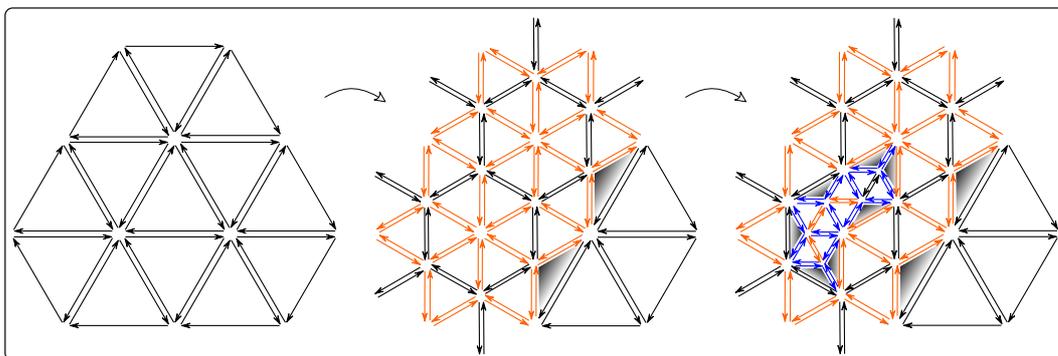
Point3D Carte::positionPairSqrt3(Brin b, int niv)
{
    Point3D resultat, Ci, P;
    int n = 0;
    Brin tmp = b;
    do
    {
        ++n;
        Ci += pos(phi2(tmp, niv), niv);
        tmp = phi1(phi2(tmp, niv), niv);
    } while (tmp != b);
    P = pos(phi1(b, niv), niv - 1);
    resultat = K * (3.0 * Ci - n * P) / 2 * n + (1.0 - K) * P;
    return resultat;
}

```

---

sa position est le barycentre du triangle dont le sommet impair est issu. On a vu que chaque sommet impair est composé de trois nouveaux brins et de trois brins provenant des niveaux précédents. Ce sont ces trois derniers brins qui formaient son triangle d'origine. Il suffit donc de récupérer le plongement de niveau  $niv-1$  pour chacun de ces trois brins et d'en faire la moyenne. Afin de passer par ces trois brins, on se place d'abord sur un brin du sommet ayant un niveau d'insertion inférieur à  $niv$  avec la fonction `ancienBrin`. Puis on parcourt le triangle d'origine du sommet grâce aux coutures  $\phi_1^{niv-1}$ .

La seconde fonction calcule la position d'un sommet pair identifié par un brin  $b$  et un niveau de résolution  $niv$ . Si, grâce à la formulation donnée précédemment, on calcule la position d'un sommet pair en fonction des sommets impairs



**Figure 3.48** - Deux pas de subdivision adaptative avec le schéma  $\sqrt{3}$  au sein d'une 2-carte multirésolution duale

déjà calculés, il suffit de parcourir l'ensemble des voisins du sommet de  $\mathbf{b}$  au niveau  $\text{niv}$  en traversant pour chaque brin de l'orbite du sommet leur couture par  $\phi_2$ , et de récupérer leur plongement de niveau  $\text{niv}$ . On a néanmoins toujours besoin aussi de la position de ce sommet au niveau  $\text{niv}-1$ . Tous les brins de cet ancien sommet forment la face polygonale du maillage dual de la deuxième étape de la figure 3.47. Afin de récupérer l'un de ces brins, il suffit de partir de l'un des brins du sommet de  $\mathbf{b}$  au niveau  $\text{niv}$  et de traverser sa couture par  $\phi_1^{\text{niv}}$ . On peut ensuite récupérer son plongement de niveau  $\text{niv}-1$ .

Ce schéma peut également être appliqué de manière adaptative. La figure 3.48 illustre un exemple de maillage triangulaire obtenu par l'application adaptative du schéma  $\sqrt{3}$ . Ce raffinement adaptatif suit une règle simple dictant que si un triangle est subdivisé alors que son voisin ne l'est pas, leur arête commune n'est simplement pas basculée.

Deux types de triangles différents, que l'on va appeler *pair* et *impair*, doivent alors être considérés. Les triangles dits impairs sont ceux obtenus par la division en 3 d'un triangle quand le basculement d'arête n'a pas eu lieu. Ces triangles, mis en évidence dans la figure 3.48 sont à la frontière entre les zones ayant des profondeurs de subdivision différentes. Les triangles pairs sont tous les autres triangles du maillage.

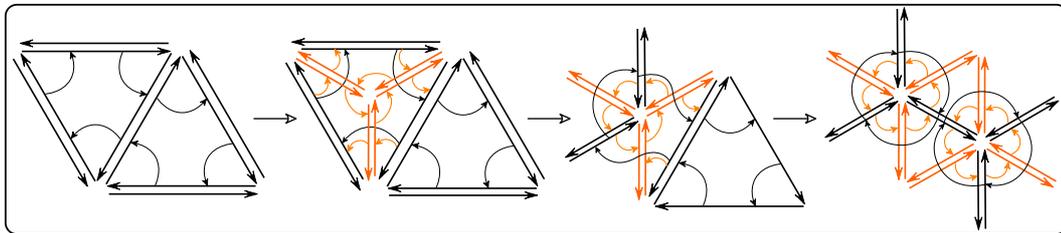
Pour définir le niveau de subdivision des faces, la définition générale donnée précédemment est toujours valide. Les choses sont néanmoins plus simples dans ce cadre où même dans le cas d'une subdivision adaptative, toutes les faces du maillage sont toujours des triangles. On définit alors le niveau d'une face au sein d'un maillage adaptatif comme le maximum des niveaux d'insertion de ses brins. Une face impaire est alors identifiée comme une face dont une face voisine est de niveau inférieur (les autres étant de niveau égal). Toutes les autres faces sont des faces paires, c'est-à-dire dont les faces voisines sont de

```

si  $t$  est pair alors
  | Insérer un sommet central et trois arêtes;
  | pour chaque triangle voisin  $t_i$  de  $t$  faire
  | | si  $t_i$  est impair et de niveau  $n + 1$  alors
  | | | Basculer l'arête commune;
  | | fin
  | fin
fin

```

**Algorithme 8** : Algorithme de subdivision d'un triangle dans le schéma de subdivision  $\sqrt{3}$



**Figure 3.49** - Détail d'un maillage lors d'une subdivision adaptative avec le schéma  $\sqrt{3}$

niveau supérieur ou égal.

La subdivision d'une face dans un maillage adaptatif se fait alors en suivant l'algorithme 8. La figure 3.49 illustre un détail d'un maillage lors d'une subdivision adaptative. Les coutures par la permutation  $\phi_1$  sont illustrées. La face de gauche est d'abord sélectionnée pour la subdivision. Trois nouvelles arêtes sont insérées en suivant la procédure illustrée dans la figure 3.46. On suppose ensuite que les deux triangles voisins invisibles ici étaient déjà subdivisés. Leurs arêtes communes sont donc basculées. On observe que les coutures par  $\phi_1$  au niveau précédent, toujours illustrées ici en noir, n'ont pas été modifiées. La face de droite est ensuite également sélectionnée pour la subdivision et l'arête centrale est alors aussi basculée après l'insertion des trois nouvelles arêtes.

Le calcul de la position des sommets impairs peut se faire en suivant le même algorithme que celui présenté dans le cadre d'une subdivision régulière. La position d'un sommet pair ne peut être calculée grâce à l'algorithme du listing 3.4.1 que si tous les sommets impairs nécessaires au calcul sont présents. Ce n'est pas le cas pour les sommets situés à la frontière entre plusieurs profondeurs de subdivision. Le masque d'origine mettant en jeu les sommets voisins au niveau précédent doit alors être utilisée. Ces sommets pairs, contrairement aux autres qui sont composés uniquement de nouveaux brins, ont également dans leur orbite, des brins provenant des niveaux précédents. On se sert de

---

**Listing 3.4.2** Implantation alternative du calcul des positions des sommets pairs dans le schéma de subdivision  $\sqrt{3}$

---

```

Point3D Carte::positionPairSqrt3(Brin b, int niv)
{
    Point3D resultat;
    int n = 0;
    Brin old = ancienBrin(b, niv);
    Brin tmp = old;
    do
    {
        ++n;
        resultat += K * pos(phi2(tmp, niv - 1), niv - 1);
        tmp = phi1(phi2(tmp, niv - 1), niv - 1);
    } while (tmp != old);
    resultat /= n;
    resultat += (1.0 - K) * pos(old, niv - 1);
    return resultat;
}

```

---

cette particularité pour les différencier et on utilise alors la fonction présentée dans le listing 3.4.2. On récupère d'abord un des brins ayant le niveau d'insertion minimal grâce à la fonction `ancienBrin`. Ce brin faisant partie du sommet de niveau `niv-1`, on peut s'en servir comme point de départ pour parcourir l'orbite de ce sommet et récupérer le plongement de ses voisins au niveau `niv-1`.

La figure 3.50 illustre un exemple de maillage triangulaire subdivisé de manière adaptative selon un critère lié à la courbure locale avec le schéma  $\sqrt{3}$ , puis le même objet après édition et mise à jour de la subdivision du maillage.

## 3.5 Comparaison

Dans cette section, nous établissons une comparaison entre les 2-cartes multirésolution et les structures de données généralement utilisées dans le cadre de la représentation de surfaces de subdivision multirésolution. Cette comparaison va tout d'abord porter sur des points d'ordre structurel, puis traiter de problèmes d'efficacité en présentant une étude de la complexité aussi bien en temps qu'en espace de ces structures.

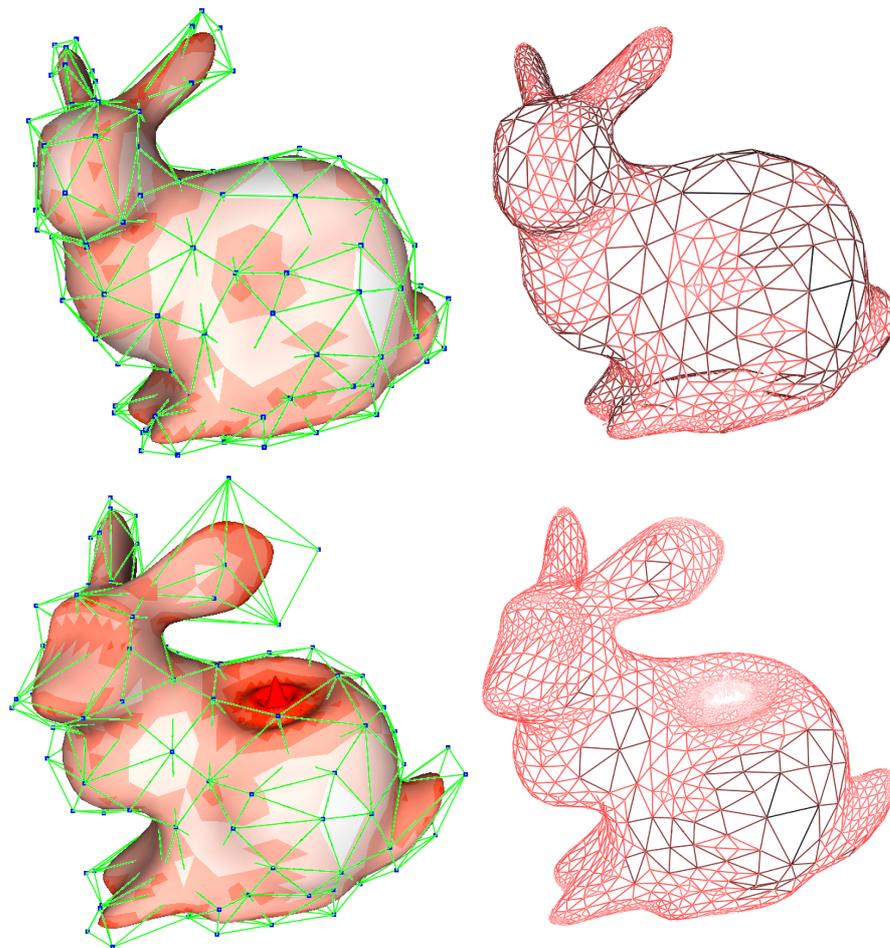


Figure 3.50 - Exemple de maillage subdivisé de manière adaptative et édité avec le schéma  $\sqrt{3}$

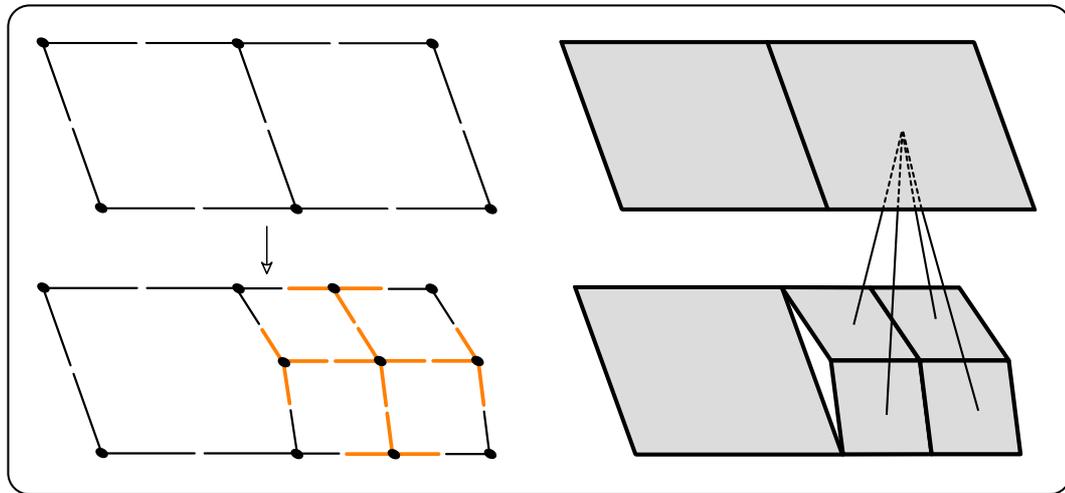
### 3.5.1 Structure

La structure de données la plus couramment utilisée pour représenter des maillages multirésolution issus d'algorithmes de subdivision est l'arbre quaternaire ou quadtree que l'on a présenté dans le chapitre 1. Cette structure peut être déduite naturellement quand on observe le fonctionnement des procédures de raffinement des schémas de Loop ou de Catmull-Clark. En effet, ces derniers opèrent une quadrisection des faces du maillage. Chaque face du maillage de départ est alors la racine d'un arbre dont chaque nœud est une face et possède quatre nœuds fils, un pour chacune des quatre sous-faces obtenues lors de la subdivision. On obtient au final une forêt de quadtrees dont l'ensemble des racines constitue le maillage de niveau 0.

Une construction adaptative est obtenue en stoppant la construction de l'arbre à des profondeurs différentes suivant les zones du maillage. Les feuilles au sein d'un arbre et entre les arbres ne partagent donc pas toutes la même profondeur. Un accès régulier au niveau  $i$  consiste à descendre dans les arbres jusqu'au niveau  $i$  s'il existe et à s'arrêter avant au niveau des feuilles sinon. Un accès adaptatif consiste à descendre dans les arbres jusqu'à la profondeur souhaitée suivant les zones de l'objet.

L'apparente simplicité de cette structure cache en réalité une rigidité qui, dans le cadre applicatif des surfaces de subdivision multirésolution, est la source de nombreuses limitations. Tout d'abord les maillages générés par quadrisection de triangles et ceux générés par quadrisection de quadrangles ne peuvent être tous les deux représentés dans une seule et même structure de données. Des versions spécifiques de cette structure ainsi que des opérateurs d'adjacence propres à chacune doivent être développés pour supporter de tels maillages. Les maillages générés par les schémas quad/triangle ou  $\sqrt{3}$  présentés précédemment ne sont eux simplement pas supportés naturellement pas de telles structures arborescentes, car ils ne présentent pas une uniformité et une régularité suffisantes.

Plus fondamentalement, le fait de ne représenter que les faces de la subdivision en omettant les autres cellules topologiques engendre de nombreux problèmes de consistance topologique, notamment lors de la construction ou de l'accès à des maillages adaptatifs. En effet, tel qu'illustré dans la figure 3.51, des trous topologiques se forment dans le maillage représenté par le quadtree à la frontière entre les zones de profondeur différente. Cela est dû au fait que le maillage n'est pas géré en tant que partition cellulaire d'une 2-variété. Les relations topologiques entre les cellules de la subdivision sont inexistantes dans cette structure où seules les faces sont représentées. Les relations d'adjacence



**Figure 3.51** - Dans un maillage adaptatif des trous topologiques apparaissent dans le quadtree aux frontières entre les niveaux, alors qu'il reste topologiquement fermé avec une 2-carte multirésolution

entre ces dernières ne sont par ailleurs pas accessibles directement mais sont données implicitement grâce à des algorithmes de parcours de l'arbre qui exploitent la régularité de la structure.

L'approche suivie dans le modèle des 2-cartes multirésolution permet de toujours maintenir la cohérence topologique du maillage, qu'il soit obtenu de manière adaptative ou non. Toutes les cellules de la partition sont représentées, et leurs relations d'incidence et d'adjacence sont toutes présentes et directement accessibles. De plus, des maillages polygonaux quelconques peuvent être représentés autorisant le support de nombreux algorithmes de raffinement. Comme on l'a vu lors de l'application des opérations de raffinement sur les 2-cartes multirésolution, une quadrissection de faces consiste en réalité à couper des arêtes et à en insérer de nouvelles. La gestion des arêtes dans notre structure permet d'effectuer correctement ce raffinement et donc d'éviter de créer des inconsistances topologiques tout en garantissant toujours que le maillage représenté est bien la partition cellulaire d'une 2-variété orientable. Dans la 2-carte multirésolution de la figure 3.51, la face de gauche est simplement devenue une face à cinq côtés évitant ainsi l'apparition d'un trou topologique entre les faces.

Au moment de l'affichage d'un tel maillage surfacique, des problèmes d'ordre géométrique peuvent survenir. Dans le cas des quadtrees, ces problèmes concernent la suppression des trous dans la surface. Ceux-ci sont résolus en découpant les triangles situés à la frontière entre différentes profondeurs de subdivision. Les quadtrees restreints [HB87] que l'on a présenté dans le chapitre 1, perme-

ttent de limiter le nombre de cas à traiter. Cependant, la configuration des triangles à ajouter dépend du niveau des faces voisines qu'il est alors nécessaire de visiter, ce qui est justement une opération coûteuse dans ce genre de structure dont on cherche à minimiser le nombre d'applications.

Dans le cas des 2-cartes multirésolution, les problèmes d'ordre géométriques sont liés à la non-planarité éventuelle des faces polygonales du maillage. Dans le cadre d'une subdivision de maillages triangulaires, celles-ci sont également situées au niveau de la frontière entre faces ayant des niveaux de subdivision différents. Comme on l'a vu dans la section sur l'affichage d'une 2-carte multirésolution, le matériel graphique n'affiche que des simplexes et ici aussi on résoud le problème en triangulant les polygones durant la routine d'affichage. En revanche, il n'est nullement nécessaire ici d'aller vérifier le niveau de subdivision des faces voisines, mais il suffit de faire simplement le tour de la face et de vérifier le niveau des arêtes en même temps que l'on récupère les positions des sommets.

### 3.5.2 Complexité en temps

Dans le cadre des surfaces de subdivision multirésolution, les requêtes d'adjacence entre sommets sont une des opérations les plus couramment exécutées. Celles-ci sont hautement sollicitées aussi bien lors de l'exécution de l'algorithme de synthèse de la géométrie que dans celui de l'analyse. En effet, les positions des sommets sont définies en fonction des positions de leurs voisins ce qui nécessite pour chaque sommet le parcours de l'ensemble de ces derniers.

Comme on l'avait vu dans le chapitre 1 au sujet des 2-cartes classiques, toutes les requêtes d'incidence et d'adjacence de base sont exécutées en temps constant. Par requête de base on entend, partant d'une cellule donnée (c'est-à-dire d'un des brins représentant une cellule), atteindre une cellule immédiatement incidente ou adjacente. Comme illustré dans la figure 3.52, étant donné un brin dans une 2-carte primale, en traversant sa liaison  $\beta_0$ , on change à la fois de sommet et de face tout en restant sur la même arête. En traversant sa liaison  $\beta_1$  on change de face et d'arête tout en restant sur le même sommet autour duquel on est en train de tourner. En traversant les liaisons  $\beta_0$  puis  $\beta_1$ , on change de sommet et d'arête en tournant autour d'une même face. Ces parcours au sein de la subdivision sont tous exécutés en temps constant, temps nécessaire pour suivre un ou deux pointeurs dans la structure de données.

Chaque niveau dans une 2-carte multirésolution étant une 2-carte, on retrouve naturellement cette efficacité au sein des maillages de chaque niveau de ré-

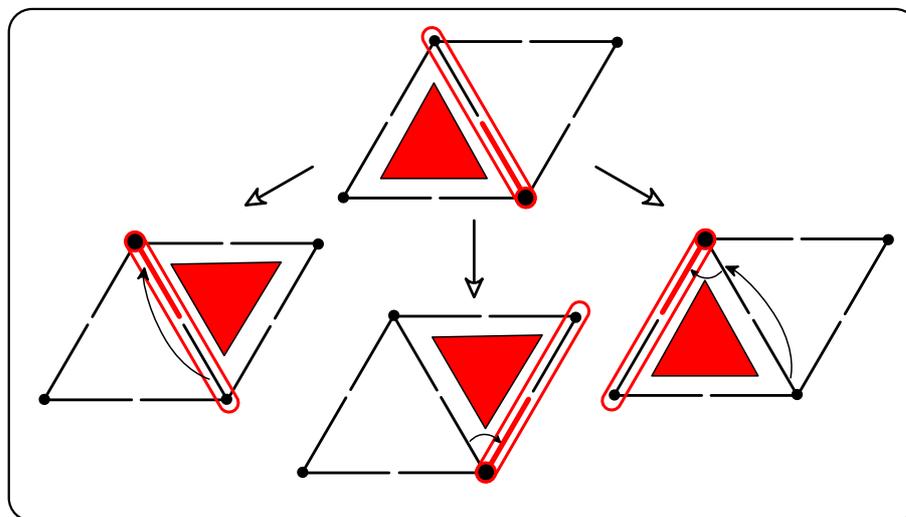


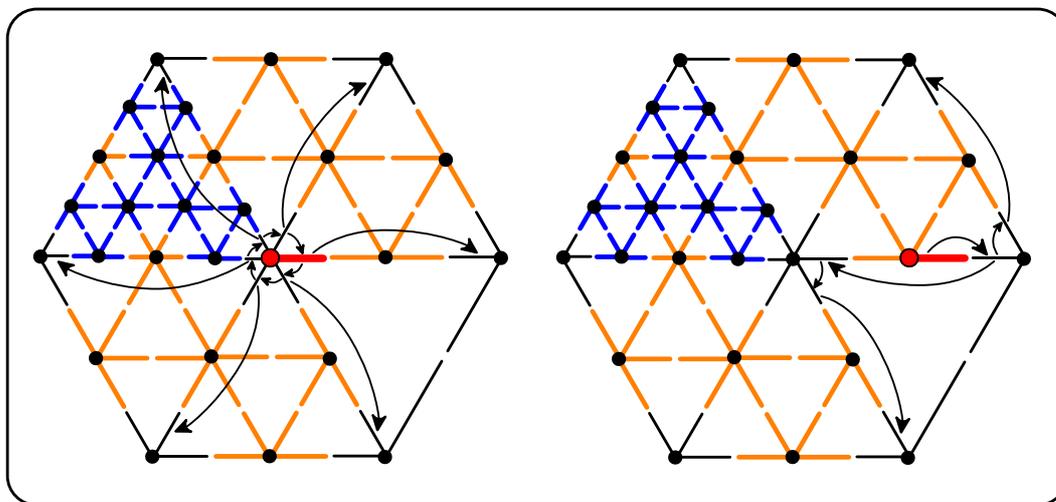
Figure 3.52 - Parcours des cellules voisines dans une 2-carte

solution. Les algorithmes de parcours peuvent même exploiter le fait que les brins disposent de liaisons à différents niveaux pour accélérer certains accès inter-niveaux.

La figure 3.53 illustre l'exécution de deux algorithmes au sein d'une 2-carte multirésolution. Cette 2-carte est subdivisée de manière adaptative jusqu'au niveau 2. On ne fait pas ici usage des informations de niveau 2 mais elles sont ici représentées afin de montrer que la complexité en temps de ces algorithmes n'est en rien dépendante des profondeurs de subdivision atteintes dans le maillage.

Le premier a pour objectif le calcul de la position d'un sommet pair au niveau 1 en utilisant le masque de subdivision du schéma de Loop (voir figure 3.14). Pour cela, partant d'un brin du sommet – en rouge dans le dessin – on atteint le sommet voisin au niveau 0 en traversant la liaison  $\beta_0^0$ . On passe ensuite à l'arête suivante dans le sommet en traversant la liaison  $\beta_1^0$ . Dès que l'on retombe sur le brin de départ, l'ensemble des sommets voisins a été visité. Le temps d'exécution est linéaire en la taille de ce voisinage.

Le deuxième algorithme illustré est celui qui calcule la position d'un sommet impair au niveau 1 toujours avec le masque de subdivision du schéma de Loop. Partant d'un brin du sommet, on traverse sa liaison  $\beta_0^1$  pour tomber sur un brin du sommet de droite. A partir de ce brin, on traverse d'une part les liaisons  $\beta_1^0$  et  $\beta_0^0$  pour arriver sur le sommet du haut, et d'autre part la liaison  $\beta_0^0$  pour arriver directement sur un brin du sommet de gauche. A partir de ce dernier brin, on combine à nouveau les liaisons  $\beta_1^0$  et  $\beta_0^0$  pour finir sur le sommet du



**Figure 3.53** - Exemples d'algorithmes exploitant les relations de voisinage présentes à différent niveaux de résolution dans une 2-carte multirésolution

bas. Le temps d'exécution est ici constant.

Ces algorithmes sont bien sûr applicables à n'importe quel niveau de résolution et ce en conservant exactement la même complexité en temps.

Des algorithmes similaires peuvent être écrits par exemple pour parcourir, au sein d'un niveau de résolution  $i$  donné, l'ensemble des sommets d'une face (par exemple dans le but de l'afficher ou de calculer un vecteur normal), l'ensemble des faces adjacentes à une face (par exemple dans le but de calculer les angles entre la normale d'une face et celles de ses voisines) ou l'ensemble des faces incidentes à un sommet (par exemple dans le but d'estimer un vecteur normal en un sommet).

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, puis en redescendant de l'autre côté de l'arête. Le voisinage entre les différents quadrees de la forêt est résolu explicitement au niveau de l'ensemble des racines qui constitue le niveau 0 de l'objet. Les faces étant les seules entités topologiques représentées dans cette structure, les adjacences entre les autres types de cellules doivent être résolues par l'intermédiaire des faces, ce qui peut engendrer un coût plus important que si elles étaient résolues directement.

Ces requêtes de voisinage entre faces sont exécutées au pire en  $O(\log(n))$ , avec  $n$  la taille de l'arbre. En pratique, en plus d'être la complexité au pire des requêtes

d'adjacence (la complexité moyenne est plutôt de l'ordre de  $O(\log(\log(n)))$ ), si on ne subdivise pas vers des niveaux trop élevées, cette valeur  $\log(n)$  n'atteint souvent pas une valeur très élevée. Néanmoins celle-ci n'est pas constante et le fait d'avoir besoin d'un temps qui dépende de la profondeur de l'arbre pour résoudre des adjacences qui concernent un niveau donné est en soi un défaut auquel il est important d'apporter une alternative. De plus, ces opérations d'adjacence étant les plus utilisées pour la mise à jour de l'objet durant la déformation d'un maillage, cette amélioration n'est pas négligeable d'un point de vue de l'efficacité générale d'une application d'édition multirésolution.

### 3.5.3 Complexité en espace

Nous comparons ici le coût mémoire d'une 2-carte multirésolution dotée d'un 0-plongement et celui d'une forêt de quadrees dans le cadre de la représentation de maillages obtenus par quadrisection de triangles. Comme aucune formulation générale ne peut être donnée dans le cas adaptatif et que la subdivision régulière est le cas le pire en ce qui concerne les besoins en mémoire, nous effectuons cette comparaison dans le cas régulier.

#### 2-carte multirésolution

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 maillage de niveau de résolution 0, et soit  $k$  le niveau de résolution maximum. Comme dans le cadre d'une quadrisection de triangles appliquée régulièrement sur tout le maillage le nombre de brins est multiplié par 4 à chaque pas de subdivision, on a :

$$|B| = b_0 \cdot 4^k \quad (3.1)$$

Pour calculer le coût mémoire total de l'information topologique dans une 2-carte multirésolution, il nous faut compter le nombre de pointeurs stockés par l'ensemble des brins de la carte.

Pour la relation  $\beta_0$ , il faut additionner les tailles des tableaux de pointeurs contenus dans chaque brin. La taille de ce tableau varie en fonction du niveau d'insertion du brin. On observe que 3/4 des brins ont été insérés au niveau de résolution maximum lors du dernier pas de subdivision et ne disposent donc que d'une seule liaison  $\beta_0$ ; le 1/4 de brins restant proviennent des niveaux

inférieurs et  $3/4$  de ces derniers, i.e.  $3/16$  du nombre de brins total, disposent de deux liaisons  $\beta_0$ ; ... Formellement, pour  $i$  entre 1 et  $k$ , il y a  $|B| \cdot \frac{3}{4^i}$  brins dont le tableau est constitué de  $i$  éléments. Les brins décrivant le niveau 0 ont eux  $k + 1$  éléments dans leur tableau de liaisons  $\beta_0$ .

Le nombre total de pointeurs stockés dans les tableaux de liaisons  $\beta_0$  de tous les brins de la carte est donc, après remplacement de  $|B|$  en utilisant l'équation (3.1) :

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

Pour la relation  $\beta_1$ , les choses sont beaucoup plus simples. En effet, comme on l'a vu précédemment dans ce chapitre, la valence des sommets n'est pas modifiée durant ce processus de raffinement. Les liaisons  $\beta_1$  ne sont donc jamais modifiées. En respectant le fait qu'un tableau de liaisons topologiques ne grandit pas si la liaison n'est pas modifiée, chaque brin ne dispose que d'une liaison  $\beta_1$ . En d'autres termes, il y a en tout  $|B|$  ou  $b_0 \cdot 4^k$  pointeurs stockés par l'ensemble des brins de la carte pour la relation  $\beta_1$ .

Le plongement géométrique est attaché aux brins par un tableau de pointeurs vers des points 3D. Dans le cas où on utilise un schéma de subdivision approximant, le plongement des sommets pairs est modifié à chaque niveau. En suivant le même principe que pour le stockage des relations topologiques, le tableau de plongement grandit à chaque pas de subdivision. Pour chaque brin, la taille du tableau de pointeurs vers le 0-plongement a exactement la même taille que celui de liaisons  $\beta_0$ .

Le nombre total de pointeurs stockés par l'ensemble des brins est donc :

$$b_0 \cdot 4^k + 2 \cdot \left( b_0 \cdot (k + 1) + b_0 \cdot 3 \cdot \sum_{i=1}^k i \cdot 4^{k-i} \right) \quad (3.2)$$

La somme de l'expression (3.2) peut être identifiée à la série entière :  $\sum_{n \geq 0} i \cdot x^i = \frac{x}{(1-x)^2}$ , définie pour  $|x| < 1$ . Après avoir sorti les termes constants de la somme, on substitue ici  $\frac{1}{4}$  à  $x$ . 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 (3.2) se simplifie alors en (en omettant les termes négligeables) :

$$\frac{33}{9} \cdot b_0 \cdot 4^k \quad (3.3)$$

## Quadrees

Les quadrees stockent cinq pointeurs par nœud pour l'information topologique : quatre vers les nœuds fils et un vers le nœud père. Les racines des quadrees stockent sept pointeurs : quatre vers les nœuds fils, et trois pour les relations d'adjacence entre les racines de la forêt. Pour l'information de plongement, 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 de l'arbre. 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 :

$$10 \cdot f_0 + 8 \cdot f_0 \cdot \sum_{i=1}^k 4^i \quad (3.4)$$

La somme de l'expression (3.4) 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 (3.4) se simplifie alors en (en omettant les termes négligeables) :

$$\frac{32}{3} \cdot f_0 \cdot 4^k \quad (3.5)$$

## Comparaison

On peut maintenant calculer le ratio entre le coût d'une 2-carte multirésolution (3.3) et celui d'une forêt de quadrees (3.5). Dans un maillage triangulaire,

chaque face est composée de trois brins. On a alors  $f_0 = \frac{b_0}{3}$ . On obtient donc le ratio suivant :

$$\frac{\frac{33}{9} \cdot b_0 \cdot 4^k}{\frac{32}{9} \cdot b_0 \cdot 4^k} = \frac{33}{32} \quad (3.6)$$

On voit que le modèle des 2-cartes multirésolution 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 devient encore plus faible.

Le fait d'utiliser un schéma de subdivision interpolant plutôt qu'approximant rend même 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'est plus modifiée d'un niveau au suivant. Les tableaux de pointeurs vers l'information de plongement contenus dans chaque brin n'augmentent donc plus d'un niveau à l'autre se réduisent donc, tout comme pour le stockage de la relation  $\beta_1$ , à un seul pointeur. Dans ce cas, le ratio entre 2-cartes multirésolution et quadrees est de  $\frac{30}{32}$ , ce qui ramène le coût des 2-cartes multirésolution à environ 6% d'espace mémoire en moins que le stockage d'un quadree.

### 3.6 Conclusion

On a vu dans ce chapitre comment les 2-cartes multirésolution peuvent servir de support topologique pour la représentation de surfaces de subdivision multirésolution. Le fait de disposer, pour chaque niveau de résolution, de la représentation complète de la partition cellulaire d'une 2-variété orientable conjointement à l'ensemble des relations d'incidence et d'adjacence entre ces cellules, donne à ce modèle de nombreux avantages dans ce contexte.

Du point de vue de la généralité de représentation, nous avons montré que de nombreux schémas de subdivision sont supportés. En effet, une 2-carte multirésolution peut représenter des maillages polygonaux et appliquer sur ces maillages des opérations de raffinement variées entre les niveaux. Nous avons illustré ici son utilisation dans le cadre des schémas de Loop [Loo87], Catmull-Clark [CC78], Doo-Sabin [DS78] ainsi que des schémas quad/triangle [SL03] et  $\sqrt{3}$  [Kob00].

Cette capacité à gérer des maillages polygonaux quelconques et le fait de représenter toutes les cellules de la subdivision est une différence fondamentale de notre modèle par rapports aux structures existantes. Cette propriété

permet non seulement un usage dans de nombreux cadres différents mais aussi et surtout de maintenir un maillage consistant — sans trous topologiques — dans le cadre d'une subdivision adaptative dans les schémas faisant usage d'un raffinement primal. Ces algorithmes de raffinement utilisent en effet des opérateurs de subdivision et d'insertion d'arêtes qui ne peuvent être correctement appliqués sur le maillage que si la structure sous-jacente représente effectivement ces entités topologiques.

Du point de vue de l'efficacité, les 2-cartes multirésolution permettent d'accéder directement au maillage correspondant à n'importe quel niveau de résolution sans avoir à calculer ni à reconstruire la moindre information d'adjacence entre cellules. Il suffit pour cela de fixer le paramètre de niveau et de parcourir la 2-carte correspondante. Chaque niveau étant une 2-carte, on accède aux relations d'incidence et d'adjacence entre les cellules en temps optimal, et ce quel que soit le niveau de résolution considéré. Dans le cadre des surfaces de subdivision, et notamment de leur édition multirésolution interactive, la géométrie associée aux sommets des maillages des différents niveaux de résolution doit être mise à jour constamment. Les positions des sommets étant calculées en fonction des positions de sommets appartenant à un certain voisinage, l'accès optimal aux cellules adjacentes, sans aucune dépendance au niveau considéré ni à la profondeur de la structure multirésolution, permet d'améliorer l'efficacité des algorithmes de calcul de la géométrie.

Enfin, on a vu que le coût mémoire d'une structure de donnée dérivée du modèle des 2-cartes multirésolution reste dans le même ordre de grandeur que celui d'une structure à base de quadtree, usuellement utilisée dans ce cadre applicatif.



---

# MAILLAGES PROGRESSIFS

---

## Sommaire

---

<b>4.1</b>	<b>Introduction . . . . .</b>	<b>153</b>
<b>4.2</b>	<b>Applications des maillages progressifs multirésolution . . . . .</b>	<b>155</b>
4.2.1	Filtrage . . . . .	155
4.2.2	Compression . . . . .	158
<b>4.3</b>	<b>Supports topologiques classiques des maillages progressifs . . . . .</b>	<b>161</b>
<b>4.4</b>	<b>Utilisation des cartes multirésolution . . . . .</b>	<b>162</b>
4.4.1	Gestion des ensembles de brins . . . . .	162
4.4.2	Gestion des coutures . . . . .	165
4.4.3	Vérification des contraintes topologiques . . . . .	167
4.4.4	Contractions groupées . . . . .	171
4.4.5	Economies de mémoire . . . . .	173
4.4.6	Application aux algorithmes de traitement de la géométrie	175
4.4.7	Accès au maillage . . . . .	180
4.4.8	Comparaison du coût mémoire . . . . .	182
<b>4.5</b>	<b>Conclusion . . . . .</b>	<b>186</b>

---

## 4.1 Introduction

De nombreux maillages sont aujourd'hui issus de processus d'acquisition 3D. Ceux-ci sont généralement très denses et leur connectivité ne présente pas de régularité particulière. De nombreux travaux ont été menés afin de pouvoir appliquer à ces objets les algorithmes issus des études sur les maillages multirésolution. Comme on l'a vu dans l'introduction du chapitre 3, une première

approche consiste à construire une surface de subdivision multirésolution approchant au mieux l'objet original en partant d'une version simplifiée du maillage d'origine [LSS<sup>+</sup>98]. La connectivité de départ est alors perdue lors de ce processus de remaillage.

D'autres approches ont été développées afin de conserver la connectivité d'origine des objets. Celles-ci prennent appui sur la notion de maillage progressif introduite dans [Hop96] et que l'on a présentée dans le chapitre 1. Les applications de ces représentations vont de l'édition multirésolution au filtrage de maillages bruités [GSS99] en passant par la compression de maillage [PR00, AG05, Ber07].

Pour rappel, un maillage progressif est construit en opérant une succession de contractions d'arêtes dans le maillage. Ces dernières peuvent ensuite être inversées en appliquant les éclatements de sommets correspondants. Les structures de données utilisées pour coder les maillages progressifs sont généralement conçues de la façon suivante. Le maillage le plus grossier est stocké ainsi qu'un ensemble d'opérations d'éclatement de sommet. Chacun de ces noeuds doit permettre d'identifier le lieu de l'éclatement et contenir l'information nécessaire à la reconstruction. Cet ensemble peut être une simple file inversant l'ordre des contractions d'arête, ou organisé en un arbre de dépendances permettant une reconstruction adaptative du maillage.

Tout comme les algorithmes de subdivision, ceux développés dans le cadre des maillages progressifs font grand usage des opérateurs topologiques de voisinage à des fins de calculs géométriques. La topologie des maillages intermédiaires doit donc être disponible afin de faire circuler l'information dans la hiérarchie.

Les structures classiques utilisées pour encoder de telles représentations progressives sont basées sur le codage d'une part du maillage courant dans une structure "monorésolution" classique, tout en maintenant d'autre part une file ou un arbre de mises-à-jour. Afin d'accéder aux maillages intermédiaires et ainsi permettre l'exécution des calculs géométriques, ces structures nécessitent d'appliquer constamment des opérateurs de modification topologique sur le maillage courant.

Après avoir présenté le fonctionnement de certaines de ces méthodes multirésolution sur les maillages progressifs, nous montrons dans ce chapitre comment les cartes multirésolution permettent d'améliorer l'efficacité des algorithmes appliqués sur ces maillages en fournissant un accès et des parcours topologiques efficaces de tous les maillages intermédiaires sans avoir besoin de reconstruire cette information.

## 4.2 Applications des maillages progressifs multirésolution

### 4.2.1 Filtrage

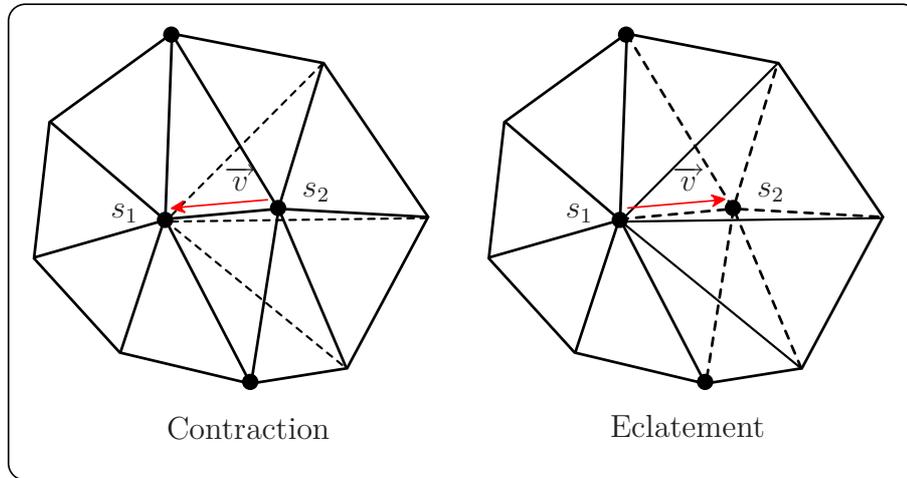
Les méthodes proposées dans [GSS99] ont pour but d'adapter des techniques telles que le lissage ou l'édition multirésolution à des maillages triangulaires irréguliers en utilisant une hiérarchie de maillages progressifs et des filtres non-uniformes. Quelques définitions sont nécessaires afin d'éclaircir le sens des termes "irrégulier" et "non-uniforme".

On appelle triangulation *régulière*, un maillage triangulaire dont tous les sommets sont de valence 6. On appelle triangulation *irrégulière* un maillage triangulaire dont les sommets ont une valence quelconque. On appelle triangulation *semi-régulière*, un maillage triangulaire obtenu par quadrisections successives des faces d'un maillage triangulaire irrégulier de départ : les sommets issus du maillage de départ conservent leur valence quelconque tandis que tous les autres sommets du maillage sont de valence 6. C'est ce que l'on a appelé "connectivité de subdivision" dans le chapitre précédent dans le cadre des surfaces de subdivision.

Les filtres appliqués sur un voisinage donné pour le calcul de la position des sommets lors des opérations d'analyse et de synthèse de la géométrie dans des maillages multirésolution sont de natures différentes suivant le type de connectivité du maillage utilisé. On appelle filtre *uniforme* un filtre utilisant les mêmes coefficients pour tous les sommets du maillage. Un tel filtre ne peut être utilisé que sur des maillages réguliers. On appelle filtre *semi-uniforme* un filtre dont les coefficients dépendent de la connectivité locale du maillage. Ce type de filtre est typiquement utilisé au sein de maillages semi-réguliers (par exemple le filtre utilisé dans le schéma de subdivision de Loop illustré dans la figure 3.14). Finalement on appelle filtre *non-uniforme* un filtre dont les coefficients dépendent à la fois de la connectivité et de la géométrie locales. Les coefficients de tels filtres sont donc propres à chaque sommet du maillage auquel il s'applique. Dans [GSS99], les coefficients des filtres utilisés dépendent à la fois des aires des triangles incidents et des longueurs des arêtes voisines.

Nous n'allons pas entrer ici dans les détails de la méthode utilisée afin de calculer les coefficients locaux des filtres non-uniformes, mais uniquement donner une description des algorithmes mettant en œuvre ces filtres.

Tout d'abord un maillage progressif est construit à partir du maillage fin de départ. Une succession de contractions d'arêtes est donc effectuée, et le sommet



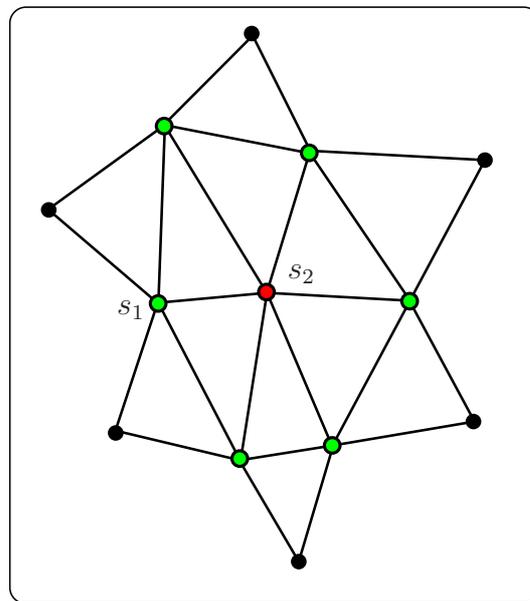
**Figure 4.1** - Illustration du vecteur à conserver pour reconstruire la position du sommet lors de l'éclatement de sommet

résultant de la contraction est ici à chaque fois positionné aux coordonnées de l'un des deux sommets constituant les extrémités de l'arête contractée. Comme on le voit dans la figure 4.1, l'arête constituée des sommets  $s_1$  et  $s_2$  est contractée sur le sommet  $s_1$ .

Dans les maillages progressifs classiques, l'objectif est simplement de pouvoir reconstruire le maillage fin original à l'identique en inversant la séquence de contractions d'arêtes. Aucun filtre particulier n'est utilisé ici, et pour chaque contraction d'arête, un vecteur est simplement conservé afin de pouvoir repositionner les sommets lors de l'éclatement de sommet correspondant. Dans le cas d'une contraction d'arête effectuée de la manière que l'on vient de décrire, c'est le vecteur  $\vec{v}$  qui est stocké pour être ajouté à  $s_1$  afin de positionner le sommet  $s_2$  lors de la reconstruction du maillage.

On cherche ici à se munir d'un outil qui permette non seulement de reconstruire à l'identique le maillage  $M$  de départ, mais également de pouvoir effectuer sur ce dernier des traitements géométriques tels que le lissage ou l'édition. A chaque contraction d'arête, on ne va donc pas stocker le vecteur  $\vec{v}$ , mais calculer les coefficients du filtre non-uniforme local qui, appliqué à un voisinage lors de l'éclatement de sommet correspondant, vont permettre de calculer une position pour le sommet  $s_2$ . La figure 4.2 illustre le voisinage utilisé pour le calcul de la position du sommet  $s_2$  après un éclatement de sommet. Ce voisinage appelé "1-voisinage ailé" inclut l'ensemble des sommets voisins ainsi que l'ensemble des sommets étant à une distance topologique 2 de  $s_2$ . Chacun de ces sommets participe à hauteur de son coefficient au calcul d'une position pour  $s_2$ .

La position calculée à partir de ce voisinage n'est pas la position originale



**Figure 4.2** - Voisinage utilisé pour le calcul de la position d'un sommet après un éclatement de sommet

du sommet  $s_2$  mais une position qui permet de lisser localement la géométrie du maillage tout en minimisant la distortion des arêtes voisines. Les sommets voisins de  $s_2$  (sommets verts de la figure 4.2) sont également déplacés après l'éclatement de sommet en poursuivant le même objectif. Les positions de ces derniers sont également calculées comme une combinaison affine des positions des sommets de leur 1-voisinage ailé qui inclut le sommet  $s_2$  que l'on vient tout juste d'insérer dans le maillage.

Afin d'être en mesure de reconstruire le maillage de départ à l'identique, ce sont les vecteurs exprimant la différence entre les nouvelles positions calculées par le filtre pour le sommet  $s_2$  et ses voisins et les positions originales de ces sommets qui sont conservés. Ces vecteurs appelés vecteurs de détail sont exprimés dans des repères locaux.

On se retrouve en définitive avec le maillage grossier  $M^0$  et un ensemble d'opérations d'éclatement de sommet. A chacune de ces opérations sont rattachés les coefficients du filtre local ainsi qu'un ensemble de vecteurs de détails. L'ordre dans lequel les contractions d'arêtes sont effectuées à partir du maillage de départ est ici choisi en fonction de deux critères : d'une part on cherche à minimiser l'erreur commise par le maillage grossier par rapport au maillage de départ, et d'autre part on privilégie la contraction des arêtes les plus longues du maillage.



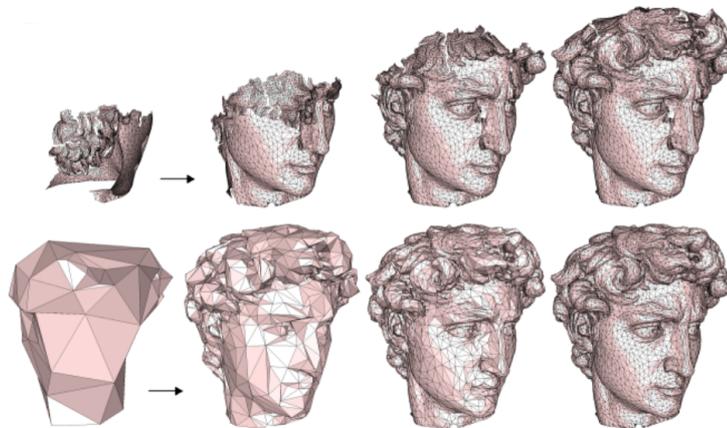
**Figure 4.3** - Différentes opérations de filtrage appliquées à un modèle  
(image tirée de [GSS99])

En utilisant une terminologie issue du traitement du signal, il est dit que les premiers éclatements de sommet de la séquence sont relatifs aux basses fréquences de l'objet, en allant jusqu'au hautes fréquences pour les derniers. La "porteuse" du signal est assimilée au maillage grossier  $M^0$ . Lors de la reconstruction du maillage fin à partir du maillage grossier, si on ajoute l'ensemble des vecteurs de détails à chaque éclatement de sommet, on se retrouve en présence du maillage  $M$  de départ (figure 4.3 gauche). Si on n'en ajoute aucun, on génère une version lisse du maillage  $M^0$  disposant de la même connectivité que le maillage  $M$ . Si on ajoute les vecteurs de détail jusqu'à un certain point et qu'on les ignore par la suite, on applique une sorte de filtre "passe-bas" qui lisse uniquement les détails fins de l'objet (figure 4.3 milieu gauche et droit). Un effet inverse au lissage permettant l'exagération des caractéristiques d'un objet peut être obtenu en amplifiant des vecteurs de détail lors de la reconstruction. Tout un éventail d'effets peut être obtenu en combinant ces différents effets.

L'édition multirésolution consiste ici simplement, comme dans le cadre des maillages semi-réguliers, à éditer les positions des sommets d'un maillage intermédiaire. La géométrie des niveaux plus fins est alors à nouveau synthétisée en ajoutant les vecteurs de détail au fur et à mesure.

#### 4.2.2 Compression

Les méthodes de compression de maillages triangulaires sont usuellement classifiées en deux approches appelées *compression incrémentale* et *compression progressive*. L'image 4.4 illustre parfaitement la différence entre ces deux approches. On voit des étapes intermédiaires dans le décodage d'un maillage



**Figure 4.4** - Décodage d'un maillage triangulaire compressé avec une méthode directe (en-haut) et progressive (en-bas) (image tirée de [AG05])

triangulaire compressé avec une méthode incrémentale (en-haut) et avec un méthode progressive (en-bas). La première décode le maillage en partant d'un triangle donné et reconstruit le maillage complet en ajoutant peu à peu des triangles. La seconde part d'une version grossière du maillage et le raffine progressivement en donnant accès à des résolutions intermédiaires de l'objet durant le processus.

Dans les deux cas, deux types d'informations doivent être considérés pour la compression : la connectivité du maillage et le plongement géométrique. Nous allons nous intéresser ici aux méthodes progressives. Celles-ci peuvent encore une fois être classifiées en deux approches suivant que l'on souhaite ou non conserver la connectivité d'origine du maillage à compresser. Si on ne souhaite pas la conserver, de la même manière que pour le filtrage, on peut utiliser une technique de remaillage afin de se retrouver avec un maillage semi-régulier [LSS<sup>+</sup>98] et utiliser des techniques issues des surfaces de subdivision multirésolution [KSS00].

Dans le cas où l'on souhaite conserver la connectivité du maillage d'origine, les méthodes diffèrent selon les opérateurs topologiques utilisés pour simplifier et reconstruire progressivement le maillage. Certaines méthodes telles que [TGHL98] ou [BPZ99] utilisent des opérateurs complexes qui mettent en jeu des petites zones du maillage. D'autres comme [COLR99] ou [AD01] utilisent l'opération de suppression de sommet – qui crée un trou dans le maillage en supprimant un sommet avec toutes ses faces incidentes – suivie d'une retriangulation. Enfin des méthodes comme celle présentée dans [PR00] exploitent directement la représentation en maillage progressif utilisant les opérations de contraction d'arête et d'éclatement de sommet. Dans cette dernière, le maillage grossier  $M^0$  est encodé avec une méthode de compression incrémentale, et les

opérations de raffinement progressif sont regroupés en ensembles d'éclatement de sommets indépendants.

En ce qui concerne le traitement de la géométrie, les positions des sommets sont manipulées à deux moments : lors d'une contraction d'arête, deux sommets sont fusionnés en un seul dont la position est assignée par un opérateur que nous appelons *approximant*; lors d'un éclatement de sommet, un sommet donne naissance à deux sommets et nous appelons *reconstructeur* l'opérateur qui assigne une position à ces deux sommets.

Comme on l'a vu dans le chapitre 1, dans les maillages progressifs classiques, l'approximant a le choix entre le milieu de l'arête à contracter ou chacun de ses deux sommets incidents. En fonction du choix effectué, un vecteur de différence est stocké. Ce vecteur est donné au reconstructeur lors de l'éclatement de sommet afin de redonner aux sommets leurs positions d'origine. On a donc autant de vecteurs à stocker que d'arêtes contractées. A des fins de compression, ces vecteurs peuvent être quantifiés puis encodés en utilisant un codage entropique dans des tables de Huffman [Huf52].

On obtient de meilleurs taux de compression si les vecteurs à encoder sont les plus petits et réguliers possibles. C'est dans cet objectif qu'on a développé des méthodes basées sur des opérateurs de prédiction de la géométrie. Le reconstructeur utilise alors en premier lieu un opérateur que l'on appelle *prédicteur* pour assigner des positions aux sommets issus de l'éclatement de sommet. C'est l'erreur commise par ce prédicteur qui va être codée sous forme d'un vecteur à stocker. Si le prédicteur est bon, les vecteurs à stocker sont plus petits et réguliers, entraînant un meilleur taux de compression. Dans le cadre d'une compression avec perte, on peut alors également appliquer un seuil sur ces vecteurs correcteurs d'erreur et ne conserver que les plus significatifs d'entre eux.

Dans [PR00] l'approximant consiste simplement à positionner le sommet issu de la contraction d'arête au milieu de cette arête. Le prédicteur est quant à lui conçu pour estimer les positions des sommets issus d'un éclatement de sommet en interpolant les sommets de leur voisinage en s'inspirant du schéma de subdivision interpolant Butterfly [DLG90]. Pour reprendre les termes précédemment employés, on peut parler ici de filtre semi-uniforme pour le prédicteur. Les filtres non-uniformes tel que proposé dans [GSS99] pour les applications de filtrage et d'édition ne sont pas de bons candidats pour la compression. En effet, étant différents pour chaque sommet, tous les coefficients devraient être stockés.

### 4.3 Supports topologiques classiques des maillages progressifs

Comme on l'a vu dans la section précédente, de nombreuses applications utilisent des filtres géométriques appliqués sur un voisinage donné afin de calculer des positions aux sommets des maillages progressifs. Afin de récupérer les positions des sommets voisins impliqués dans ces calculs, l'information topologique du maillage doit être disponible.

Comme on l'a vu dans le chapitre 1 les structures classiquement utilisées pour la représentation de maillages progressifs [Hop98] stockent d'une part le maillage grossier  $M^0$  et d'autre part l'ensemble des opérations d'éclatement de sommets. Le maillage est stocké dans une structure contenant un tableau de sommets et un tableau de faces, chacune des faces contenant un tableau permettant de retrouver ses trois sommets ainsi qu'un tableau indiquant ses trois triangles adjacents. L'encodage d'une opération d'éclatement de sommet consiste lui d'une part en l'information nécessaire pour localiser le lieu et déterminer l'orientation de l'éclatement, et d'autre part en l'information nécessaire au positionnement géométrique des sommets issus de l'éclatement.

Pour passer du maillage  $M^i$  au maillage  $M^{i+1}$  on exécute l'une de ces opérations d'éclatement de sommet. Une fois le sommet identifié, un nouveau sommet est inséré dans le tableau de sommets ainsi que deux nouvelles faces dans le tableau de faces. L'information de voisinage de ces des deux faces doit ensuite être positionnée, et celle de leurs quatre faces voisines doit être mise à jour. Lors d'une contraction d'arête, c'est l'opération inverse qui est effectuée sur le maillage courant.

Suivant les applications des maillages progressifs, une telle structure nécessitant de mettre à jour la structure topologique représentant le maillage courant à chaque opération de raffinement ou de simplification est plus ou moins appropriée. Si l'objectif est d'effectuer une transmission progressive du maillage par un réseau, alors on le maillage est obligatoirement mis-à-jour au fil de la réception des opérations de raffinement. En revanche, quand la représentation progressive est le support topologique à l'application d'algorithmes de traitement de la géométrie tels qu'on vient de les décrire, qui opèrent, à la moindre modification des paramètres de filtrage, un recalcul des positions des sommets du maillage à tous les niveaux, alors des structures de données plus efficaces peuvent être utilisées.

## 4.4 Utilisation des cartes multirésolution

Le modèle des cartes multirésolution permet une représentation des maillages progressifs autorisant un accès direct à toutes les versions intermédiaires du maillage. Une information topologique complète accessible à chaque niveau conduit alors à une efficacité optimale des algorithmes de traitement de la géométrie exploitant un tel support.

Dans cette approche où les niveaux de résolution sont obtenus par simplifications successives du maillage en opérant des contractions d'arêtes, on se rapproche ici du cadre des pyramides de cartes combinatoires présentées dans le chapitre 1.

La construction des niveaux se passe ici en sens inverse par rapport à ce qui était fait dans le cadre des surfaces de subdivision multirésolution. Chaque étape de simplification consiste à appliquer des opérations de simplification topologique sur le maillage le plus grossier actuel, c'est-à-dire celui de niveau 0. Afin de rester cohérent dans la numérotation des niveaux de résolution, on commence donc par effectuer un décalage vers le haut des niveaux existants : chaque maillage de niveau  $i$  avec  $i \geq 0$  devient le nouveau maillage de niveau  $i + 1$ . On crée ensuite le nouveau maillage de niveau 0 par simplification de celui qui est maintenant le maillage de niveau 1. On se retrouve finalement avec une hiérarchie où, comme dans le cadre des surfaces de subdivision, le maillage de niveau 0 est le plus grossier, et le maillage de niveau maximum est le plus fin.

La figure 4.5 montre les applications successives de deux contractions d'arêtes au sein d'une 2-carte. Le point de départ pour construire la hiérarchie est la 2-carte représentant le maillage fin  $M$ . Etant le seul niveau existant au départ, cette 2-carte constitue le niveau 0 de la hiérarchie, à la fois le plus fin et le plus grossier. Tous les brins appartiennent à l'ensemble  $N^0$  et disposent d'une liaison par la permutation  $\phi_1^0$  et d'une liaison par l'involution  $\phi_2^0$ . Par exemple le brin  $a$  est lié au brin  $b$  par  $\phi_2^0$ .

### 4.4.1 Gestion des ensembles de brins

Dans le premier dessin de la figure 4.5, on voit en rouge une arête sélectionnée pour la contraction. Les six brins correspondant aux deux faces supprimées lors de la contraction d'arête sont dessinées en orange dans le deuxième dessin. Ces brins sont supprimés de l'ensemble de brins  $N^0$  et envoyés dans l'ensemble  $N^1$ .

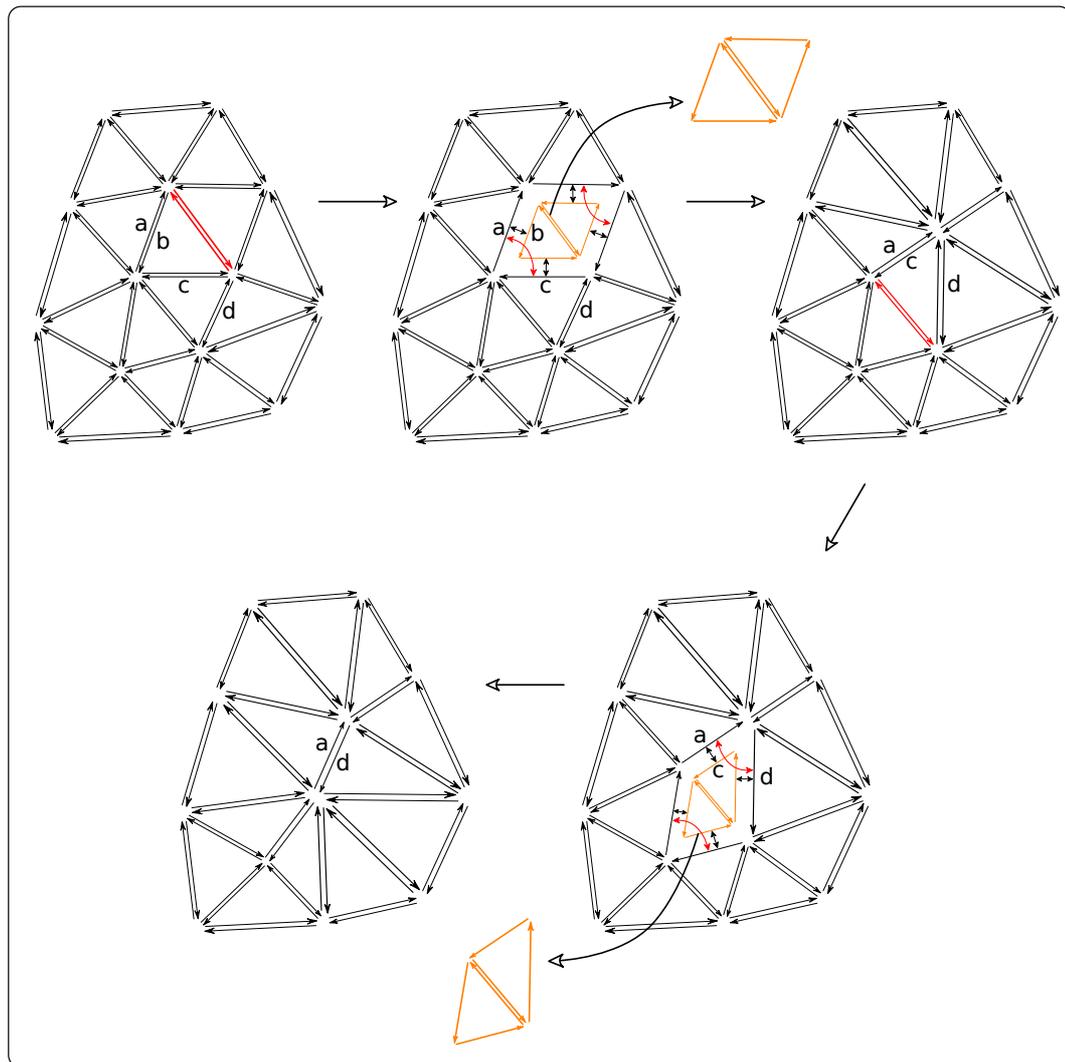
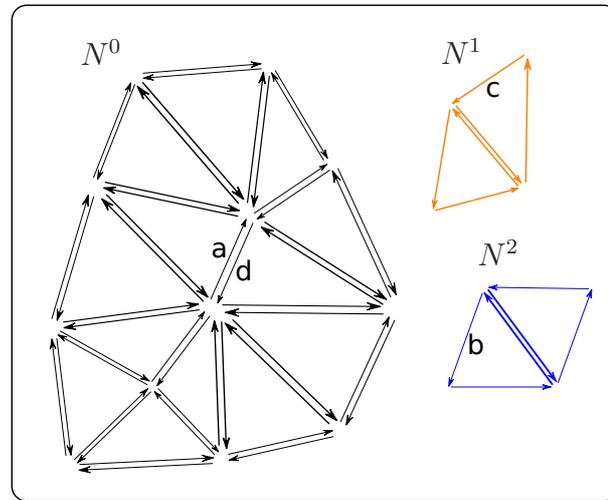


Figure 4.5 - Exécution de contractions d'arêtes successives dans une 2-carte multirésolution



**Figure 4.6** - Ensembles de brins après deux contractions d'arêtes dans une 2-carte multirésolution

En suivant la définition donnée dans le chapitre 2 concernant l'imbrication des ensembles de brins, on a  $B^0 = N^0$  et  $B^1 = N^0 \cup N^1$ . L'ensemble de brins appartenant à la 2-carte décrivant le niveau de résolution 0 est donc maintenant composé de six brins de moins que celui de la 2-carte décrivant le niveau de résolution 1.

Si on suit la procédure que l'on a décrit plus haut qui consiste à décaler tous les niveaux vers le haut, puis à construire le nouveau niveau 0 par simplification du niveau 1, on aurait aussi pu commencer par déplacer tous les brins de  $N^0$  dans l'ensemble  $N^1$ , puis repeupler l'ensemble  $N^0$  avec tous les brins n'étant pas supprimés lors de l'application de l'opérateur de simplification. Le résultat est cependant le même si on commence par créer un ensemble  $N^1$  vide et que l'on y déplace les brins de  $N^0$  concernés par la simplification.

Lors de la contraction d'arête suivante, on va construire un nouveau niveau 0 plus grossier. On commence donc par décaler les niveaux existants strictement supérieurs à 0 vers le haut : pour tout  $i$  tel que  $1 \leq i \leq \max Niv$ , on a  $N^{i+1} = N^i$ . On se retrouve avec un ensemble  $N^0$  contenant toujours le même ensemble de brins, et un ensemble  $N^1$  vide, dans lequel on va déplacer les six brins sélectionnés correspondant aux deux faces supprimées par la contraction d'arête effectuée au sein de la 2-carte de niveau 0.

La figure 4.6 illustre l'état des ensembles de brins de la 2-carte multirésolution après l'application de ces deux contractions d'arêtes.

#### 4.4.2 Gestion des coutures

Dans la définition des 2-cartes multirésolution, pour tout  $i \geq 0$ , les relations  $\phi_1^i$  et  $\phi_2^i$  doivent être définies pour chaque brin de l'ensemble  $B^i$ . De la même manière que dans le cadre des surfaces de subdivision, chaque brin de l'ensemble  $N^i$  disposera donc de  $1 + \max Niv - i$  liaisons pour chaque relation,  $\max Niv$  étant le niveau maximum défini dans la structure.

Comme on le voit dans la figure 4.5, les coutures par la permutation  $\phi_1$ , qui forment les cycles de brins décrivant les faces de la subdivision, ne sont pas modifiées par l'application d'une contraction d'arête. La couture d'un brin par  $\phi_1$  est donc identique au sein de toutes les 2-cartes auxquelles il appartient, depuis son niveau d'insertion jusqu'au niveau maximum de la structure multirésolution. D'un point de vue pratique lors de l'implantation, chaque brin ne va donc contenir qu'un seul pointeur en ce qui concerne cette couture.

Les liaisons par l'involution  $\phi_2$  qui relient les faces entre elles sont quant à elles modifiées lors d'une contraction d'arête ou d'un éclatement de sommet. C'est la gestion de ces liaisons qui va nous intéresser ici.

Lors de la définition d'un nouveau maillage de niveau 0, on a vu que l'on commençait par décaler vers le haut les ensembles de brins correspondant aux niveaux strictement supérieurs à 0. Les coutures stockées dans ces brins subissent implicitement le même décalage. En effet, les niveaux auxquels correspondent les coutures contenues dans les tableaux de coutures sont alignés sur le niveau d'insertion du brin. Quand ce dernier est incrémenté, les niveaux qui correspondent aux coutures stockées sont également incrémentés.

Après le décalage des ensembles de brins, l'ensemble  $N^1$  est vide et  $N^0$  contient toujours le même ensemble de brins. Ces brins contiennent chacun  $1 + \max Niv$  coutures pour la relation  $\phi_2$ . Or le niveau maximum de la structure vient d'être incrémenté et tous les autres ensembles de brins décalés. Non seulement il manque une couture à ces brins, mais les coutures encore présentes ne correspondent plus au niveau adéquat. Afin de rétablir la cohérence, une nouvelle case est ajoutée au tableau, et toutes les coutures existantes sont décalées vers le haut. On a pour tout brin  $b \in N^0$  et tout  $i$  tel que  $1 \leq i \leq \max Niv$ ,  $\phi_2^{i+1}(b) = \phi_2^i(b)$ . La couture de niveau 0 reste inchangée durant l'opération.

Après cette opération, les deux 2-cartes  $C^0$  et  $C^1$  sont équivalentes. En effet, l'ensemble  $N^1$  ne contient aucun brin, pour chaque brin de l'ensemble  $N^0$ , les deux premières cases des tableaux de coutures sont égales.

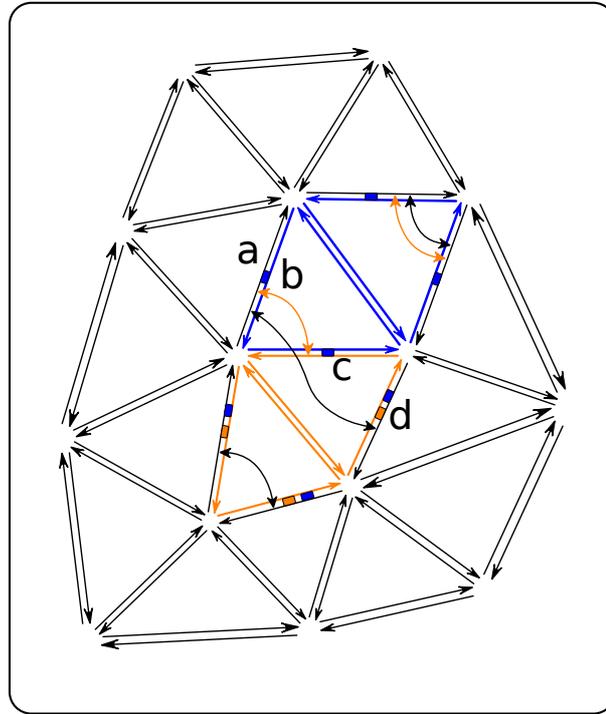


Figure 4.7 - Modification des coutures par  $\phi_2$  au fil des niveaux de résolution

Une fois que l'on a sélectionné une arête pour la contraction au sein du maillage de niveau 0, on envoie depuis l'ensemble  $N^0$  vers l'ensemble  $N^1$  les six brins correspondant aux deux faces supprimées. Ces derniers contiennent alors une couture de trop qui est celle que l'on vient d'ajouter en début du tableau de couture. On supprime alors cette liaison, et on décale vers le bas toutes les autres liaisons du tableau.

Il est bien entendu possible de diminuer le nombre de décalages de tableaux effectués lors de ces opérations. Une autre méthode consiste à commencer par ne pas toucher aux tableaux de coutures des brins de  $N^0$ . La structure hiérarchique est alors momentanément inconsistante. On sélectionne ensuite l'arête à contracter, puis on envoie les brins correspondant dans l'ensemble  $N^1$ . C'est seulement à ce moment là que l'on augmente la taille des tableaux de coutures des brins restant dans  $N^0$  et que l'on procède à leur décalage.

Finalement, les coutures par  $\phi_2^0$  des quatre brins de  $N^0$  adjacents au "trou" créé par la suppression des deux faces sont modifiées afin de refermer correctement le maillage.

La figure 4.7 illustre les modifications de couture par  $\phi_2$  au cours des éclatements de sommets correspondant aux contractions d'arêtes de l'exemple précé-

dent. Les brins en noir sont ceux de l'ensemble  $N^0$ , en orange ceux de  $N^1$  et en bleu ceux de  $N^2$ . La 2-carte de niveau 0 est composée des brins de  $N^0$  (en noir) cousus par les coutures  $\phi_2^0$  matérialisées par des flèches noires. On a par exemple  $\phi_2^0(a) = d$ . Les liaisons  $\phi_2^0$  qui ne sont pas dessinées dans la figure sont celles qui ne seront pas modifiées dans les niveaux supérieurs.

Au niveau 1, on ajoute les brins de  $N^1$  (en orange). Les six brins ajoutés disposent chacun de leur couture par  $\phi_2^1$  qui les relie soit entre eux (pour deux d'entre eux), soit à des brins déjà présents avant leur ajout (pour les quatre autres). Les coutures les liant à des brins déjà présents impliquent donc pour ces derniers une modification des coutures au niveau 1. On a par exemple  $\phi_2^1(a) = c$ . Ces coutures sont ici matérialisées par des rectangles ou des flèches oranges. Pour tous les autres brins de l'ensemble  $B^1$ , la couture de niveau 1 est égale à celle de niveau 0.

Au niveau 2, on ajoute les brins de  $N^2$  (en bleu). A nouveau, les six brins ajoutés disposent de leur couture par  $\phi_2^2$  et celles qui concernent des brins préexistants entraînent chez ceux-ci une modification de la couture au niveau 2. On a par exemple  $\phi_2^2(a) = b$ . Les coutures modifiées au niveau 2 sont ici matérialisées par des rectangles bleus. Pour tous les autres brins de l'ensemble  $B^2$ , la couture de niveau 2 est égale à celle de niveau 1.

Jusqu'à présent, les contractions d'arêtes sont sélectionnées et exécutées une par une. Cela aboutit à une hiérarchie de maillages comportant autant de niveaux que l'on a exécuté de contractions d'arête. Les maillages de deux niveaux consécutifs ont à chaque fois un sommet de différence. Les ensembles de brins  $N^i$ , avec  $i \geq 1$  sont tous composés d'exactly six brins. A chaque fois qu'un niveau est créé, tous les brins restés dans l'ensemble  $N^0$  voient leur tableau de couture par  $\phi_2$  s'agrandir, et pour la majorité d'entre eux (tous sauf quatre), la valeur située en tête de tableau est dupliquée. Nous verrons par la suite quelles stratégies peuvent être appliquées afin de réduire cette redondance dans le stockage de l'information et ainsi de minimiser les besoins en mémoire.

### 4.4.3 Vérification des contraintes topologiques

Quand on souhaite sélectionner une arête pour la contraction, certaines contraintes topologiques doivent être vérifiées afin de ne pas créer de cellules dégénérées lors de l'application de la contraction. Ces vérifications peuvent tirer parti de la présence d'une information topologique complète dans le modèle des 2-cartes multirésolution.

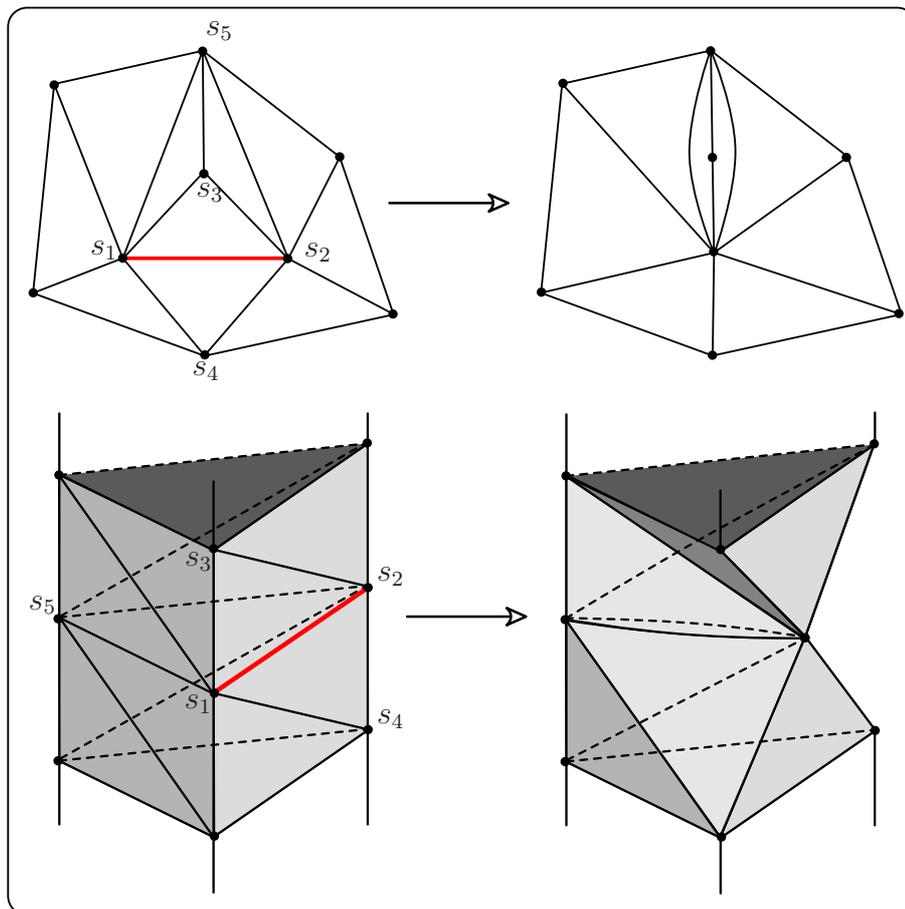


Figure 4.8 - Différents cas de dégénérescence du maillage après application d'une contraction d'arête

La figure 4.8 illustre deux cas de figure où la contraction d'une arête (en rouge dans les dessins de gauche) donne naissance à des cas dégénérés. Dans l'exemple du haut, après contraction de l'arête, deux faces se retrouvent confondues en une arête et on se retrouve avec un sommet de valence 2. Dans l'exemple du bas, la contraction de l'arête conduit au pincement d'un tube à section triangulaire.

Pour le premier cas, on connaît l'évolution de la valence des sommets mis-en-jeu lors d'une contraction d'arête. La valence des sommets  $s_3$  et  $s_4$  diminue de 1. Il suffit alors d'interdire une contraction d'arête si l'un des sommets  $s_3$  ou  $s_4$  a une valence strictement inférieure à 4. Si on souhaite éviter de créer des sommets de valence 3 au sein du maillage, on peut également calculer au préalable la valence du sommet résultant de la contraction, et interdire cette dernière si le résultat est inférieur à 4. La valence du sommet résultant de la contraction est donné par la formule suivante :  $val(s_1) + val(s_2) - 4$  où  $val(s_1)$  et  $val(s_2)$  sont les valences des sommets  $s_1$  et  $s_2$ .

Dans une 2-carte multirésolution, la valence d'un sommet à un niveau donné est calculée tel qu'illustré dans le listing 4.4.1. On parcourt simplement l'orbite du sommet et on incrémente un compteur pour chaque brin traversé. Les fonctions `beta1` et `beta_1` permettent de passer d'un brin `b` au brin respectivement suivant et précédent dans l'orbite sommet de ce brin.

Cette technique n'est pas suffisante pour éviter le cas de l'exemple du bas de la figure 4.8. Une autre méthode plus générale met en jeu la notion de chemin de connexion entre deux sommets. Un chemin de longueur  $n$  est une série d'arêtes sans doublons  $\{a_1, a_2, \dots, a_n\}$  telles que si  $a_i$  est incidente aux sommets  $s_i^1$  et  $s_i^2$ , alors on a pour tout  $i \in [0, n - 1]$ ,  $s_i^2 = s_{i+1}^1$ . Ce chemin relie les sommets  $s_1^1$  et  $s_n^2$ .

Après une contraction d'arête, tous les couples d'arêtes composant un chemin de longueur 2 reliant les sommets  $s_1$  et  $s_2$  se retrouvent confondues. Pour les arêtes composant les deux chemins passant par les sommets  $s_3$  et  $s_4$ , ce résultat est exactement l'effet recherché et les triangles dégénérés  $s_1s_2s_3$  et  $s_1s_2s_4$  sont supprimés. Si il en existe d'autres, alors ces chemins donneront naissance à d'autres dégénérescences non voulues telles que celles illustrées dans la figure 4.8.

Si l'exemple du haut de la figure 4.8 pourrait également être résolu en supprimant les deux autres triangles dégénérés  $s_1s_3s_5$  et  $s_2s_3s_5$  apparus lors de la contraction, celui de l'exemple du bas ne peut être résolu de la sorte car il ne s'agit pas ici de l'apparition de triangles dégénérés supplémentaires (il n'y a pas de triangle  $s_1s_3s_5$  dans le maillage), mais du pincement d'un volume

**Listing 4.4.1** Exemple d'implantation du calcul de la valence d'un sommet

---

```

int Carte::valenceSommet(Brin b, int niv)
{
    int val = 0;
    Brin bb = b;
    do
    {
        val++;
        bb = beta1(beta2(bb, niv), niv);
    } while (bb != b);
    return val;
}

Brin Carte::beta1(Brin b, int niv)
{
    return phi1(phi2(b, niv), niv);
}

Brin Carte::beta_1(Brin b, int niv)
{
    return phi2(phi_1(b, niv), niv);
}

```

---

prismique. Pour éviter ce cas, il convient alors simplement d'interdire une telle contraction d'arête.

De manière générale, pour chaque contraction d'arête envisagée, on va appliquer l'algorithme suivant : si il existe entre  $s_1$  et  $s_2$  un chemin de longueur 2 autre que ceux passant par  $s_3$  et  $s_4$ , alors la contraction de l'arête  $s_1s_2$  n'est pas autorisée. Afin de maintenir l'intégrité du maillage, ces restrictions doivent être respectées quelle que soit la structure de données sous-jacente utilisée. Le listing 4.4.2 illustre un exemple d'implantation d'un algorithme permettant de détecter de telles configurations.

Cette fonction, extrêmement simple à coder au sein d'une structure topologique telle que les 2-cartes, commence par marquer l'ensemble des plongements de sommets attachés aux sommets voisins de  $s_1$  en excluant les sommets  $s_3$  et  $s_4$ . Ensuite, pour chaque sommet voisin de  $s_2$  (toujours en excluant les sommets  $s_3$  et  $s_4$ ) on vérifie si son plongement est déjà marqué. Si c'est le cas, c'est qu'il existe un chemin de longueur 2 entre  $s_1$  et  $s_2$  passant par ce sommet. En termes de complexité, cet algorithme effectue tout d'abord  $val(s_1) - 2$  marquages de plongements, puis  $val(s_2) - 2$  tests au pire (c'est-à-dire si le résultat de la recherche d'un tel chemin se révèle négatif) afin de vérifier le marquage des

---

**Listing 4.4.2** Exemple d'implantation de la détection de l'existence d'un chemin de longueur 2 excédentaire entre deux sommets

---

```

bool Carte::existeChemin2(Brin b, int niv)
{
    bool res = false;
    Brin bb = beta1(beta1(b, niv), niv);
    do
    {
        marquer(bb->p);
        bb = beta1(bb, niv);
    } while(bb != beta_1(b, niv));
    Brin c = phi2(b, niv);
    Brin cc = beta1(beta1(c, niv), niv);
    do
    {
        if(estMarque(cc->p))
            res = true;
        cc = beta1(cc, niv);
    } while(cc != beta_1(c, niv) && res == false);
    return res;
}

```

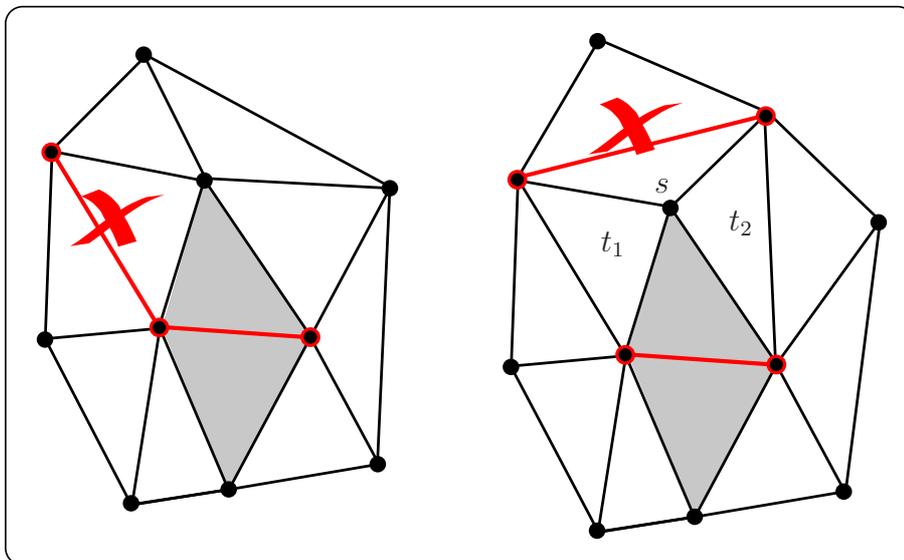
---

plongements. Chacune de ces opérations de marquage et de vérification n'est qu'une simple opération de masquage bits à bits dans un entier permettant d'encoder 32 marqueurs.

#### 4.4.4 Contractions groupées

Dans le cadre de méthodes comme l'édition ou l'application de filtres géométriques, on souhaite effectuer des traitements sur la géométrie du maillage le plus fin en prenant appui sur la hiérarchie de maillages. L'accès à l'ensemble des versions intermédiaires du maillage progressif n'est pas obligatoire. Les contractions d'arêtes peuvent alors être exécutées par "paquets" de contractions *indépendantes* comme cela est fait dans [PR00] dans le cadre de la compression de maillages progressifs. Nous allons détailler par la suite ce que l'on entend par indépendance de deux contractions d'arêtes.

L'effet au niveau de la 2-carte multirésolution sous-jacente est que le nombre de niveaux créés n'est plus égal au nombre de contractions effectuées, mais largement inférieur, suivant le nombre de contractions regroupées dans chaque paquet. Les ensembles de brin  $N^i$ , avec  $i \geq 1$  ne contiennent donc plus seule-



**Figure 4.9** - Exemples de cas interdits lors de la sélection de contractions d'arêtes groupées

ment 6 brins chacuns, mais  $6 * c$ ,  $c$  étant le nombre de contractions exécutées lors de la création du niveau.

Si l'on n'effectue qu'une seule contraction à la fois, tous les brins restés dans l'ensemble  $N^0$  voient leur tableau de coutures par  $\phi_2$  s'agrandir par la tête alors que cette couture de niveau 0 n'est différente que pour quatre d'entre eux. Une fois les contractions d'arêtes exécutées en parallèle, cette nouvelle couture de niveau 0 est différente pour  $4 * c$  brins, diminuant d'autant la duplication des données.

Pour que plusieurs contractions puissent être exécutées en parallèle, il faut qu'elles soient indépendantes. Des contraintes topologiques supplémentaires sont alors à vérifier lors de la sélection des arêtes à contracter en plus de celles vues précédemment. La figure 4.9 illustre deux cas interdits lors de la sélection de contractions d'arêtes multiples. Le cas de gauche consiste à contracter deux arêtes partageant un même sommet. Trois sommets fusionneraient alors en un seul et on ne saurait pas où positionner ce sommet. Le cas de droite consiste à contracter deux arêtes dont les sommets sont deux à deux voisins directs. Dans ce cas, ces contractions simultanées feraient du sommet  $s$  un sommet de valence 2, et des triangles  $t_1$  et  $t_2$  des triangles dégénérés. Si ces contractions d'arêtes étaient exécutées l'une après l'autre, on serait en réalité ici en présence d'un cas équivalent au premier cas de la figure 4.8.

Dans le cadre d'outils de traitement de la géométrie tels que le filtrage ou

```

pour chaque arête  $a$  dans la carte de niveau 0 faire
  | si elle n'est pas marquée alors
  | | si elle respecte les contraintes topologiques alors
  | | | Sélectionner  $a$  pour la contraction;
  | | | Marquer un voisinage de  $a$ ;
  | | fin
  | fin
fin

```

**Algorithme 9** : Algorithme de simplification

l'édition multirésolution présentés au début du chapitre [GSS99], l'indépendance entre deux contractions d'arêtes implique, en plus de ces contraintes topologiques, une distance minimum afin d'éviter tout conflit lors de l'application des filtres géométriques. En effet, si on se réfère à la figure 4.2, lors de l'application d'un éclatement de sommet, la position du sommet  $s_2$  que l'on vient d'insérer, mais également celle de l'ensemble de ses voisins est modifiée. Si deux éclatements de sommets effectués au même niveau insèrent des sommets qui partagent des voisins, alors il y a conflit pour le calcul de la position de ces derniers, car les masques non-uniformes utilisés sont propres à chaque opération d'éclatement de sommet.

Afin de lever ces ambiguïtés, la simplification d'un maillage en vue de la création d'un niveau de résolution plus grossier est réalisée tel que présenté dans l'algorithme 9. On considère l'ensemble des arêtes du maillage à simplifier. Pour chacune, si elle n'est pas encore marquée, on vérifie que sa contraction respecte bien les contraintes topologiques vues ci-dessus. Si oui, on sélectionne cette arête pour la contraction, et on marque un voisinage afin d'interdire toute contraction qui provoquerait des conflits. L'ordre dans lequel on considère les arêtes peut également dépendre de critères relatifs à l'erreur commise par le maillage simplifié par rapport au maillage fin initial.

#### 4.4.5 Economies de mémoire

Afin d'éliminer totalement la redondance dans le stockage des coutures, on peut également mettre en place un mécanisme particulier, différent de celui appliqué dans le cadre des surfaces de subdivision multirésolution. Dans ce dernier, le tableau de coutures était tronqué à partir du dernier niveau auquel cette couture était modifiée. Les requêtes concernant un niveau supérieur à la taille du tableau renvoyaient simplement la dernière couture stockée. Ce mécanisme simple et efficace fonctionne très bien dans ce cadre car les modifications de

couture sont régulières et qu'il n'y a, par construction, pas de trou à combler dans ces tableaux. En effet, une face n'est subdivisée vers le niveau  $i + 1$  que si elle est elle-même déjà une face de niveau  $i$ . Ainsi, quand une couture cesse d'être modifiée pour un brin à un niveau donné, elle ne l'est plus pour tous les niveaux supérieurs.

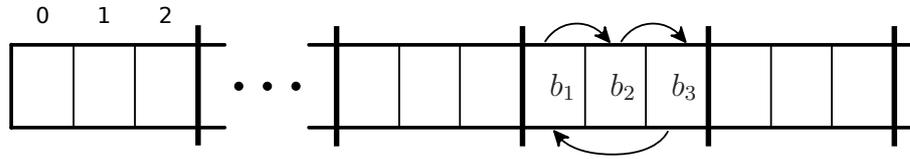
Les choses se présentent différemment dans le cadre des maillages progressifs. Premièrement les duplications de coutures ont lieu en début de tableau et non en fin. Ensuite, le fait que la couture par  $\phi_2$  d'un brin ne change pas lors d'une phase de simplification n'entraîne pas le fait qu'elle ne changera pas par la suite, lors de futures contractions d'arêtes. Ainsi, un brin de l'ensemble  $N^0$  peut tout à fait avoir un tableau de coutures similaire au suivant [3; 6; 6; 6; 5; 5; 5; 5; 7; 7] où chaque valeur identifie un brin auquel il est cousu entre les niveaux 0 et 9. En 10 niveaux dans lesquels ce brin est présent, sa couture par  $\phi_2$  est modifiée 3 fois, aux niveaux de résolution 1, 4 et 8.

Une solution pour économiser de l'espace mémoire consiste à ne stocker qu'une fois l'identifiant de chaque brin auquel ce brin est lié, conjointement au niveau à partir duquel cette liaison est effective. On se retrouve pour notre exemple avec l'information suivante : [(0, 3); (1, 6); (4, 5); (8, 7)]. Au lieu d'avoir dix identifiants à stocker, on se retrouve ici avec quatre couples, dont chacun est composé d'un niveau et d'un identifiant. Si l'on stocke les niveaux sur moins de bits que les identifiants de brins, on peut, sauf dans les cas atypiques où la couture change à chaque niveau, économiser de la mémoire. Typiquement, on va coder les niveaux sur 1 octet, et les identifiants (des pointeurs) sur 4 octets. Dans notre exemple, le gain en place est de 50%.

Dans le même temps, la complexité en temps d'une requête de couture à un niveau donné pour un brin augmente. Si l'on stocke le tableau au complet, cette requête, qui consiste simplement à accéder dans un tableau indexé par le niveau, est effectuée en temps constant. Si l'on stocke un ensemble de couples (niveau;identifiant) cette requête demande plusieurs tests successifs sur les niveaux jusqu'à trouver l'intervalle auquel appartient le niveau demandé et donc la couture correspondante.

Concernant la permutation  $\phi_1$  permettant de tourner autour des faces du maillage, celle-ci peut simplement ne pas être stockée du tout. En effet, dans le cadre des maillages progressifs, on manipule des maillages composés uniquement de triangles. De plus, ceux-ci ne sont jamais subdivisés, c'est-à-dire que pour tout brin de la carte, sa couture par  $\phi_1$  n'est jamais modifiée dans les niveaux de résolution supérieurs à son niveau d'insertion.

Chaque triangle étant composé d'exactly trois brins, on peut alors stocker



**Figure 4.10** - Illustration de l'implantation de la relation  $\phi_1$  par opération sur les index au sein d'un tableau de brins

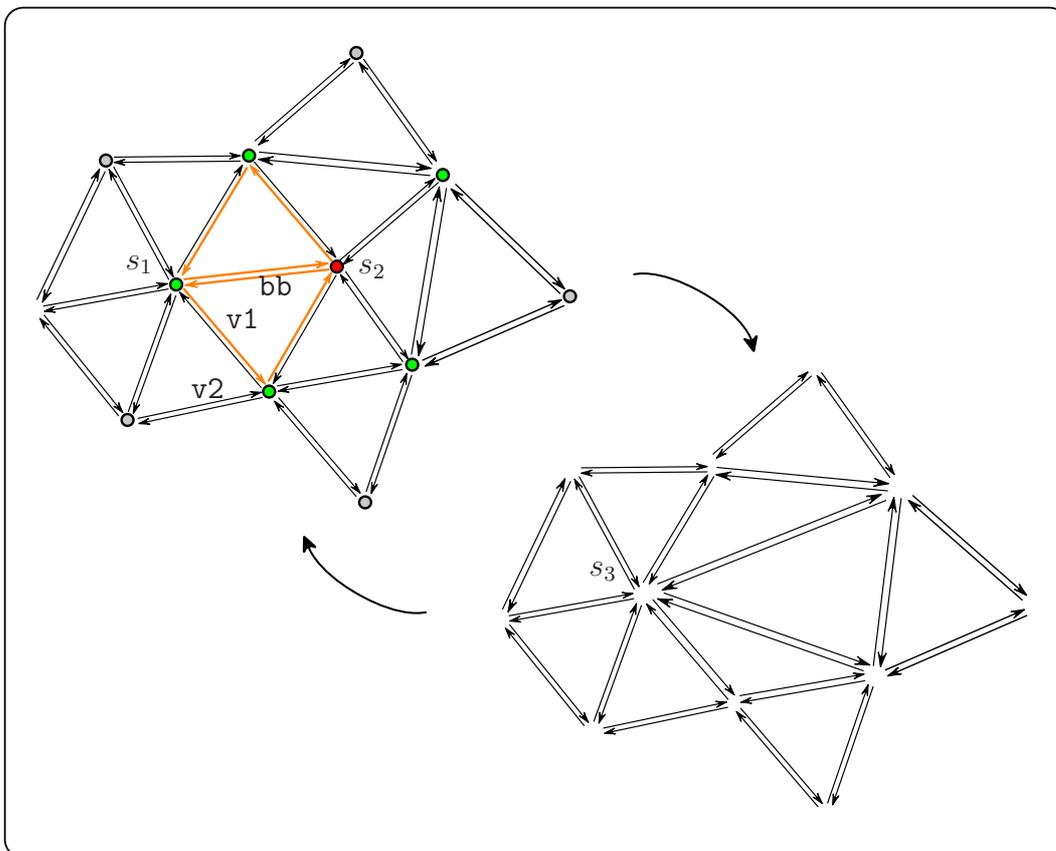
les brins de chaque ensemble  $N^i$  dans des tableaux alignés sur des ensembles de trois brins, tel qu'illustré dans la figure 4.10. A partir du brin  $b$  d'indice  $k$  dans le tableau de brins, son image par la permutation  $\phi_1$  est le brin contenu dans la case d'indice  $3 * \lfloor \frac{k}{3} \rfloor + (k \% 3 + 1) \% 3$  du même tableau.

#### 4.4.6 Application aux algorithmes de traitement de la géométrie

Une fois que l'on a mis en place le processus de création de la hiérarchie de maillages, l'application aux outils de traitement de la géométrie est directe. Dans le cadre des méthodes présentées dans [GSS99], on construit les niveaux de résolution successifs en effectuant des contractions d'arêtes par groupes en respectant les contraintes exposées ci-dessus.

Dans la méthode originale, les contractions d'arêtes sont effectuées dans un ordre dicté à la fois par l'erreur commise lors de la simplification et par la taille des arêtes. On s'efforce donc ici de respecter le même ordre tout en regroupant des contractions à chaque niveau. L'ordre strict qui aurait été utilisé si on effectuait les contractions une à une n'est donc ici plus forcément respecté. Cependant, rien dans le modèle n'oblige à effectuer à chaque niveau le nombre maximum de contractions indépendantes possibles relativement aux contraintes topologiques. En effet, même si il reste des arêtes candidates valides pour la contraction mais que celles-ci ne sont pas un bon choix pour la contraction à ce niveau en regard des critères géométriques, on peut alors simplement les ignorer, et elles seront contractées plus tard lors de la création d'un autre niveau de résolution.

En ce qui concerne la gestion de la géométrie, pour chaque contraction d'une arête formée des sommets  $s_1$  et  $s_2$ , le sommet résultant  $s_3$  est plongé sur la position du sommet  $s_1$ . On procède dans le même temps au calcul des coefficients du filtre non-uniforme ainsi qu'à celui des vecteurs de détail associés au sommet  $s_2$  ainsi qu'à l'ensemble de ses sommets voisins. Un exemple de fonction réalisant le calcul du vecteur de détail pour le sommet  $s_2$  est donné dans le listing 4.4.3.



**Figure 4.11** - Illustration de l'algorithme d'application du filtre de calcul de la position d'un sommet issu d'un éclatement de sommet

---

**Listing 4.4.3** Exemple d'implantation d'une fonction de calcul du vecteur de détail associé à un sommet supprimé lors d'une contraction d'arête

---

```

void Carte::calculDetail(Brin b, int niv)
{
    Point3D res;
    Brin bb = b;
    do
    {
        Brin v1 = phi1(bb, niv);
        res += pos(v1, niv-1) * coeff(v1, niv-1);
        Brin v2 = phi_1(phi2(v1, niv), niv);
        res += pos(v2, niv-1) * coeff(v2, niv-1);
        bb = beta1(beta2(bb, niv), niv);
    } while(bb != b);
    Vecteur3D detail = pos(b, niv) - res;
    affecteVecteurDetail(b, niv, detail);
}

```

---

La fonction `calculDetail` réalise le calcul du vecteur de détail associé au sommet du brin `b` au niveau `niv`. Celui-ci est le résultat de la différence entre la position actuelle du sommet au niveau `niv` et le résultat de l'application du filtre d'éclatement de sommet non-uniforme sur le 1-voisinage ailé du sommet. La fonction commence donc par calculer le résultat du filtre. C'est le `Point3D` `res` qui va contenir le résultat de l'application du filtre. On parcourt avec le Brin `bb` l'ensemble des brins de l'orbite du sommet. Pour chaque brin de cette orbite, on récupère le plongement au niveau `niv-1` du sommet voisin, que l'on multiplie par le coefficient attribué à ce sommet par le filtre non-uniforme. On récupère ensuite le plongement au niveau `niv-1` du sommet correspondant à l'une des ailettes du 1-voisinage ailé du sommet.

La figure 4.11 illustre à un moment donné de l'algorithme les brins utilisés dans le listing 4.4.3. Enfin, le vecteur de détail est calculé comme résultat de la différence entre la position courante du sommet et celle que l'on vient de calculer (cette opération est en réalité réalisée dans un repère local relatif au maillage grossier). On affecte ensuite au sommet le vecteur de détail que l'on vient de calculer.

En ce qui concerne le calcul des vecteurs de détails associés aux sommets voisins de  $s_2$ , la fonction qui réalise ce calcul est quasiment la même à la seule différence suivante : étant donné que le sommet  $s_2$  fait partie du 1-voisinage ailé de ces sommets, et que  $s_2$  ne dispose pas de plongement au niveau `niv-1`, on utilise dans le calcul la position de  $s_2$  que l'on vient de calculer au niveau

---

**Listing 4.4.4** Exemple d'implantation d'une fonction de synthèse de la géométrie d'un sommet issu d'un éclatement de sommet

---

```

void Carte::synthese(Brin b, int niv)
{
    Point3D res;
    Brin bb = b;
    do
    {
        Brin v1 = phi1(bb, niv);
        res += pos(v1, niv - 1) * coeff(v1, niv - 1);
        Brin v2 = phi_1(phi2(v1, niv), niv);
        res += pos(v2, niv - 1) * coeff(v2, niv - 1);
        bb = beta1(beta2(bb, niv), niv);
    } while(bb != b);
    if(eclat_courant < seuil_filtrage)
        res += detail(b, niv);
    plongeSommet(b, niv, res);
}

```

---

`niv` en tant que résultat de l'application du filtre.

Une fois l'ensemble des niveaux construits et l'ensemble des coefficients de filtres et de vecteurs de détails calculés, on peut ensuite modifier les paramètres de filtrage ou éditer la position de certains sommets aux niveaux de résolution grossiers et observer le résultat sur la géométrie de l'objet fin. Pour cela, après chaque modification, on synthétise à nouveau la géométrie des niveaux de résolution fins.

Là où dans une structure de données classique il fallait reconstruire la topologie de l'ensemble des maillages intermédiaires en appliquant effectivement les éclatements de sommets parallèlement au calcul de la géométrie, dans une 2-carte multirésolution, il suffit de traverser les niveaux de résolution du plus grossier au plus fin afin de faire remonter les modifications de l'information géométrique. Le calcul de la géométrie est la seule et unique tâche qu'il reste à exécuter, la topologie de tous les maillages intermédiaires étant déjà contenue au sein de la 2-carte multirésolution.

Afin de synthétiser la géométrie d'un niveau de résolution  $i$  donné, il suffit de parcourir l'ensemble des brins de  $N^i$ . En effet, cet ensemble contient tous les brins ajoutés lors des éclatements de sommets effectués à ce niveau. Pour chaque groupe de 6 brins qui ont été ajoutés dans  $N^i$  lors des contractions d'arêtes correspondantes, deux brins sont cousus par  $\phi_2$ . Ceux-ci correspondent

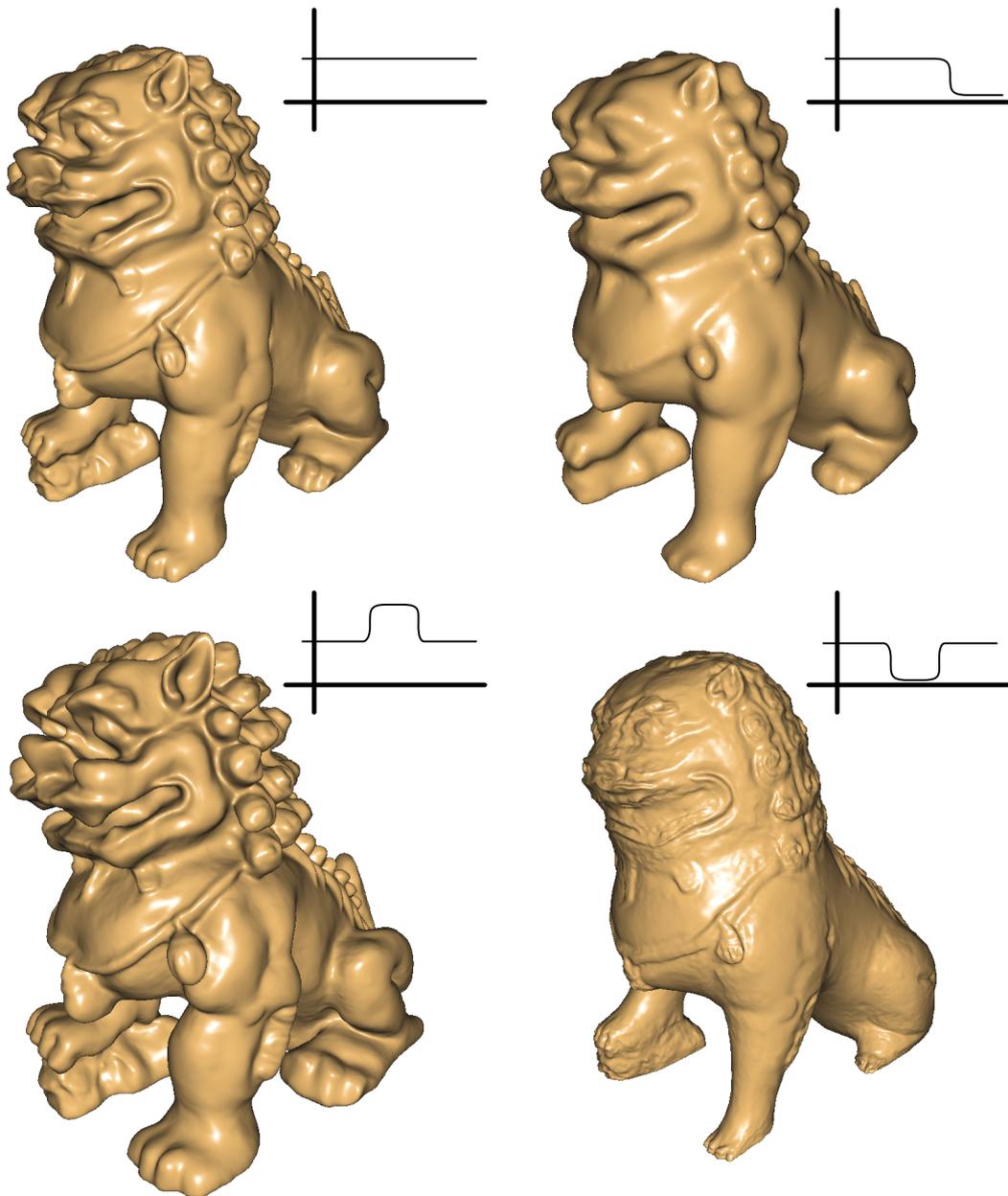


Figure 4.12 - Exemple d'application des outils de filtrage de la géométrie au sein d'une 2-carte multirésolution

à l'arête qui a été contractée, et ces brins appartiennent pour l'un à l'orbite du sommet  $s_1$ , et pour l'autre à l'orbite du sommet  $s_2$ . Comme dans notre cadre applicatif on se doit de distinguer ces deux sommets, on peut par exemple utiliser un marqueur pour les différencier. On calcule alors ensuite simplement la position du sommet  $s_2$  et de ses voisins en appliquant le filtre non-uniforme local et en ajoutant ou non, suivant les paramètres du filtrage souhaité sur l'objet, les vecteurs de détail associés aux sommets.

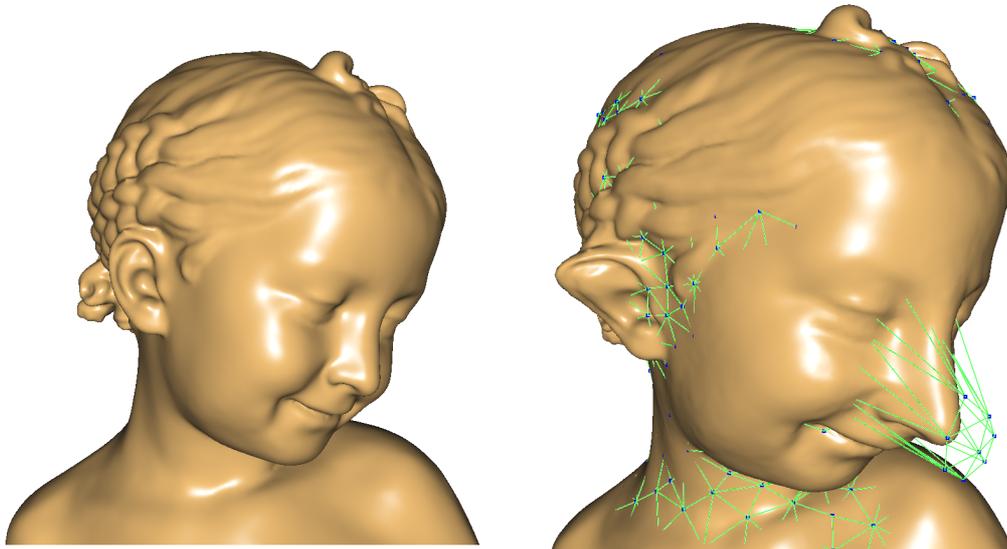
Considérons l'exemple d'un filtre passe-bas où pour un certain nombre des premiers éclatements de sommets, on ajoute les vecteurs de détail au résultat du filtre, tandis que pour les suivants, on plonge directement les sommets à la position calculée par le filtre non-uniforme en ignorant les vecteurs de détail. On parcourt alors successivement les ensembles de brins  $N^i$  avec  $1 \leq i \leq \max Niv$ . A chaque niveau, pour chacun des sommets  $s_2$  insérés dans le maillage, on applique la fonction `synthese` du listing 4.4.4 qui calcule le résultat du filtre, et n'ajoute le vecteur de détail correspondant que si le compteur `eclat_courant` du nombre d'éclatements de sommets déjà traités ne dépasse pas le seuil fixé `seuil_filtrage`. Comme nous l'avons expliqué précédemment, la fonction utilisée pour le calcul de la position des sommets voisins de  $s_2$  est légèrement différente afin d'intégrer dans le calcul la position que l'on vient de calculer pour le sommet  $s_2$  à ce niveau.

La figure 4.12 illustre des exemples de maillages filtrés au sein d'une 2-carte multirésolution. L'objet de départ est illustré en-haut à gauche. En-haut à droite, les fréquences hautes ont été coupées résultant en un lissage des détails de la surface. En-bas à gauche, les fréquences moyennes ont été amplifiées produisant une exagération des caractéristiques de l'objet. Enfin, en-bas à droite, ces mêmes fréquences moyennes ont été supprimées ce qui a pour effet de coller les détails de l'objet sur une surface sous-jacente plus lisse.

Dans la figure 4.13 c'est l'édition multirésolution qui est illustrée. On voit ici à gauche l'objet de départ, et à droite le même objet après un certain nombre d'édition à des niveaux de résolution différents.

#### 4.4.7 Accès au maillage

Comme on l'a rappelé au début de ce chapitre, l'objectif dans le cadre des outils de traitement de la géométrie sur des maillages progressifs est de conserver la connectivité initiale du maillage tout en appliquant des traitements sur le plongement qui prennent appui sur la hiérarchie de maillages. En général, afin d'observer les effets du filtrage sur les détails de l'objet, on souhaite visualiser



**Figure 4.13** - Exemple d'édition multirésolution basé sur une hiérarchie de maillages progressifs au sein d'une 2-carte multirésolution

la version la plus fine du maillage, c'est-à-dire celle qui présente la connectivité originale de l'objet. Cependant, au moment de l'affichage de l'objet, on peut tout de même souhaiter accéder aux versions intermédiaires de ce maillage, ou effectuer un affichage adaptatif présentant des densités de faces variables suivant les zones du maillage. De même, si l'on veut effectuer des éditions multirésolution sur l'objet, on doit pouvoir accéder aux maillages de contrôle intermédiaires.

L'accès régulier à un niveau de résolution  $i$  se fait ici exactement comme on l'a décrit au chapitre précédent dans le cadre des surfaces de subdivision, en fixant le paramètre de niveau  $i$  et en considérant simplement la 2-carte  $C^i = (B^i, \phi_1^i, \phi_2^i)$ . Les maillages auxquels on accède présentent alors d'un niveau au suivant autant de nouveaux sommets que l'on a regroupé d'éclatements de sommets à ce niveau lors de la construction de la hiérarchie.

On peut également définir un accès adaptatif au sein de la structure. Dans ce cas, le principe est similaire à l'accès adaptatif que l'on effectuait dans le cadre des surfaces de subdivision. Un algorithme en deux temps commence tout d'abord par parcourir la structure en profondeur à partir du niveau le plus grossier afin de marquer les brins à conserver pour l'affichage ainsi que le niveau à utiliser pour leur couture. Dans un deuxième temps, le maillage correspondant aux brins sélectionnés est affiché.

#### 4.4.8 Comparaison du coût mémoire

Nous allons donner ici une estimation du coût mémoire de la représentation d'un maillage progressif dans une 2-carte multirésolution par rapport au coût d'une structure classique. On ne compare ici que le coût de la structure topologique, l'information de plongement occupant dans les deux cas le même espace mémoire.

Soit  $nb_S(M)$  le nombre de sommets du maillage de départ. Dans un maillage triangulaire, on a en moyenne un nombre de triangles  $nb_T(M) = 2 \cdot nb_S(M)$ . Soit  $m$  le nombre d'étapes de simplification appliquées au maillage et soit  $t \in ]0; 1]$  le taux de simplification. On appelle taux de simplification la proportion de sommets survivants entre deux niveaux de résolution consécutifs. Plus ce taux est élevé, moins on supprime de sommets lors de l'application d'un groupement de contractions d'arêtes. Dans la pratique, suivant les restrictions de distance entre les contractions imposées par les applications ce taux peut se situer entre 0.6 et 0.8.

Le nombre de sommets des différents maillages de niveau  $i \geq 0$  est égal à  $nb_S(M^i) = t^{m-i} \cdot nb_S(M)$ . Le nombre de sommets ajoutés au niveau  $i$  est égal à la différence entre le nombre de sommets présents dans le maillage au niveau  $i$  et au niveau  $i + 1$ , ou autrement dit :  $nb_S(N^i) = (t^{m-i} - t^{m-i+1}) \cdot nb_S(M)$ .

#### 2-carte multirésolution

Afin de calculer le coût d'une 2-carte multirésolution, on va compter le nombre de pointeurs stockés dans l'ensemble des brins de la carte. Comme on l'a vu précédemment, les coutures par  $\phi_1$  peuvent ne pas être stockées. On va donc compter uniquement le nombre de coutures par  $\phi_2$ . Pour cela, on va additionner les tailles des tableaux de pointeurs contenus dans chaque brin, sachant que cette taille varie en fonction du niveau d'insertion du brin.

Le nombre de brins contenus dans l'ensemble  $N^0$ , c'est-à-dire l'ensemble de brins nécessaires à la représentation du maillage le plus grossier, est égal à  $nb_B(N^0) = 6 \cdot nb_S(M^0) = 6 \cdot t^m \cdot nb_S(M)$ . En effet, chaque face triangulaire est composée de 3 brins et on a  $nb_T = 2 \cdot nb_S$ . On retrouve ici le fait qu'en moyenne dans un maillage triangulaire, les sommets ont une valence égale à 6. Le nombre de brins contenus dans les ensembles  $N^i$  avec  $i \geq 1$  est donc égal à  $nb_B(N^i) = 6 \cdot nb_S(N^i) = 6 \cdot (t^{m-i} - t^{m-i+1}) \cdot nb_S(M)$ . On ajoute bien 6 brins pour chaque couple de deux triangles insérés lors d'un éclatement de sommet, ce qui correspond également au fait que l'on a ajouté un sommet dans

le maillage.

Chaque brin inséré au niveau  $i \geq 0$  contient  $m - i + 1$  coutures par l'involution  $\phi_2$ . L'ensemble des brins de  $N^i$  stockent donc en tout  $nb_{\phi_2}(N^i) = (m - i + 1) \cdot nb_B(N^i)$ . Le nombre total de coutures par l'involution  $\phi_2$  stockées par l'ensemble des brins de la 2-carte multirésolution est donc égal à :

$$\begin{aligned}
nb_{\phi_2} &= nb_{\phi_2}(N^0) + \sum_{i=1}^m nb_{\phi_2}(N^i) \\
&= (m+1) \cdot 6 \cdot t^m \cdot nb_S(M) + \sum_{i=1}^m (m-i+1) \cdot 6 \cdot (t^{m-i} - t^{m-i+1}) \cdot nb_S(M) \\
&= 6 \cdot nb_S(M) \cdot \left( (m+1) \cdot t^m + \sum_{i=1}^m (m-i+1) \cdot (t^{m-i} - t^{m-i+1}) \right) \\
&= 6 \cdot nb_S(M) \cdot \left( (m+1) \cdot t^m + (m+1) \cdot \sum_{i=1}^m (t^{m-i} - t^{m-i+1}) \right. \\
&\quad \left. - \sum_{i=1}^m i \cdot (t^{m-i} - t^{m-i+1}) \right)
\end{aligned}$$

On a  $\sum_{i=1}^m (t^{m-i} - t^{m-i+1}) = (1 - t^m)$  et  $\sum_{i=1}^m i \cdot (t^{m-i} - t^{m-i+1}) = m - \sum_{i=1}^m t^{m-i+1}$ .

On peut donc écrire :

$$\begin{aligned}
nb_{\phi_2} &= 6 \cdot nb_S(M) \cdot \left( (m+1) \cdot t^m + (m+1) \cdot (1 - t^m) - m + \sum_{i=1}^m t^{m-i+1} \right) \\
&= 6 \cdot nb_S(M) \cdot \left( 1 + \sum_{i=1}^m t^i \right)
\end{aligned}$$

Comme on a  $\sum_{i=1}^m t^i = \sum_{i=0}^m t^i - 1$ , et que pour  $t < 1$ ,  $\sum_{i=0}^m t^i$  tend vers  $\frac{1}{1-t}$ , on peut finalement écrire :

$$nb_{\phi_2} \simeq 6 \cdot nb_S(M) \cdot \left( \frac{1}{1-t} \right)$$

## Structure classique

Dans les structures classiques [Hop98] le maillage le plus grossier  $M^0$  est encodé avec l'ensemble des opérations d'éclatement de sommets. Cet encodage compact est utilisé à des fins de compression ou de transmission progressive. Cependant, lors d'une utilisation locale dans le cadre d'une application de traitement de la géométrie où tous les maillages intermédiaires doivent être accessibles rapidement, on a en mémoire une représentation du maillage le plus fin  $M$ , et les maillages intermédiaires sont accédés en maintenant pour chaque face et chaque sommet son statut de face active ou inactive.

Le maillage  $M$  est donc encodé dans une structure contenant un tableau de sommets et un tableau de faces, chacune des faces contenant un tableau permettant de retrouver ses trois sommets ainsi qu'un tableau indiquant ses trois triangles adjacents. On se focalise ici sur le coût de l'information topologique. On ne va donc pas compter le coût du tableau de sommets, mais uniquement le coût de stockage des faces et de l'adjacence entre ces dernières.

Le nombre de sommets dans le maillage le plus fin est  $nb_S(M)$ . Le nombre de triangles dans ce même maillage est donc  $nb_T(M) = 2 \cdot nb_S(M)$ . Comme chaque triangle stocke 6 pointeurs (3 sommets et 3 triangles voisins), le nombre total de pointeurs stockés pour le maillage est :  $nb_{ptr}(M) = 12 \cdot nb_S(M)$ .

Conjointement à ce maillage est encodé l'ensemble des opérations d'éclatement de sommet. Le nombre d'éclatement de sommets est simplement égal à la différence entre le nombre de sommets du maillage le plus grossier et le nombre de sommets du maillage le plus fin, c'est-à-dire :  $nb_{ecl} = nb_S(M) - nb_S(M^0) = (1 - t^m) \cdot nb_S(M)$ .

Chaque éclatement de sommet doit pouvoir identifier le sommet à éclater ainsi que deux de ses voisins afin de savoir où insérer les deux nouvelles faces. Afin de maintenir la structure d'arbre binaire, chacun doit également maintenir des pointeurs vers un sommet père et deux sommets fils. Avec l'encodage présenté dans le chapitre 1, on peut compter 5 pointeurs pour le stockage de chaque opération d'éclatement de sommet.

Le nombre de pointeurs à stocker pour l'ensemble des éclatements de sommet est donc  $5 \cdot (1 - t^m) \cdot nb_S(M)$ . Au total, le nombre de pointeurs à stocker pour l'ensemble de la structure est égal à :

$$\begin{aligned}
nb_{ptr} &= 12 \cdot nb_S(M) + 5 \cdot (1 - t^m) \cdot nb_S(M) \\
&= 5 \cdot nb_S(M) \cdot \left( \frac{12}{5} + (1 - t^m) \right) \\
&= 5 \cdot nb_S(M) \cdot \left( \frac{17}{5} - t^m \right)
\end{aligned}$$

Comme  $t^m$  tend vers 0, on a finalement :

$$nb_{ptr} \simeq 17 \cdot nb_S(M)$$

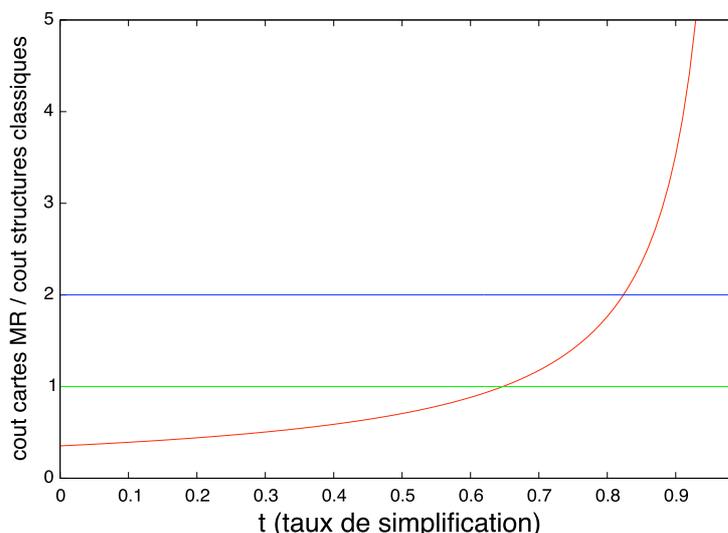
## Comparaison

Si on calcule le ratio entre le coût de ces deux structures, on obtient :

$$\begin{aligned}
ratio &= \frac{6 \cdot nb_S(M) \cdot \left( \frac{1}{1-t} \right)}{17 \cdot nb_S(M)} \\
&= \frac{6}{17} \left( \frac{1}{1-t} \right)
\end{aligned}$$

Cette expression exprime le coût lié à l'utilisation d'une 2-carte multirésolution par rapport à une structure classique. Plus le taux de simplification  $t$  est éloigné de 1, c'est-à-dire plus on supprime de sommets lors de la création d'un niveau plus grossier, plus le coût de notre modèle se révèle faible. En effet, plus on effectue de contraction d'arêtes groupées, moins il y a de brins pour lesquels la couture par  $\phi_2$  reste identique et doit donc être dupliquée. À l'inverse, si l'on ne regroupe que très peu de contractions d'arêtes à chaque pas de simplification (donc pour un taux de simplification proche de 1), les duplications sont nombreuses dans une 2-carte multirésolution et son coût devient alors assez réhibitore.

La figure 4.14 illustre l'évolution du coût mémoire d'une 2-carte multirésolution par rapport à celui d'une structure classique en fonction du taux de simplification. On voit que, pour un taux de simplification de 0.6, c'est-à-dire si on conserve 60% des sommets à chaque pas de simplification, une 2-carte multirésolution nécessite environ 0.88 fois l'espace mémoire occupé par les structures classiques. Avec un taux de simplification de 0.8, où chaque pas de simplification fait disparaître 20% des sommets du maillage, le coût mémoire est environ



**Figure 4.14** - Ratio entre le coût mémoire d'une carte multirésolution et des structures classiques en fonction du taux de simplification

1.76 fois plus important. C'est pour un taux de simplification d'environ 0.65 que le coût des deux structures est équivalent.

Il est à noter que ce rapport est indépendant de la taille du maillage et du nombre de pas de simplification effectués. De plus, si on intègre le coût lié au stockage de la géométrie qui est équivalent quel que soit la structure utilisée, le rapport entre les coûts des deux structures se resserre encore.

On peut réduire le coût lié à la duplication de coutures dans les 2-cartes multirésolution en exploitant la méthode présentée précédemment qui consiste à stocker une seule fois chaque couture conjointement avec le niveau à partir duquel cette couture est valide. Ainsi, au prix d'une légère perte de performance, due au fait que l'accès à la couture d'un brin à un niveau de résolution donné ne s'effectue plus en temps constant, on diminue encore le coût mémoire lié à l'utilisation d'une 2-carte multirésolution.

## 4.5 Conclusion

Nous avons vu dans ce chapitre comment les 2-cartes multirésolution peuvent servir de support topologique pour l'application d'outils de traitement de la géométrie sur une hiérarchie de maillages construite en utilisant le principe des maillages progressifs.

Le principal intérêt de notre modèle est de fournir ici directement l'information topologique de l'ensemble des maillages correspondant à chaque niveau de résolution de la hiérarchie. Contrairement aux structures de données classiquement utilisées dans ce cadre, qui stockent d'une part le maillage courant et d'autre part un ensemble de modifications à appliquer au maillage, une fois la 2-carte multirésolution construite, il n'est plus nécessaire d'effectuer la moindre opération topologique sur le maillage.

Lors de la modification de paramètres de filtres géométriques ou au cours d'édifications multirésolution, la complexité de la mise-à-jour du maillage se réduit à celle des calculs relatifs à la géométrie. L'efficacité de ces outils est alors accrue, sans pour autant que cela se répercute sur un coût en espace mémoire beaucoup plus élevé. En effet, si l'on écarte les configurations où peu de contractions d'arêtes sont regroupées par niveau, celui-ci est en général comparable à celui des structures existantes.

Dans le cadre de la compression de maillages, les 2-cartes multirésolution sont moins intéressantes. En effet, l'objectif étant ici d'économiser un maximum de mémoire, les structures très spécialisées telle que celle présentée dans [PR00] sont plus performantes en terme de compacité mémoire. En revanche, ces structures ne sont pas du tout conçues pour être utilisées directement au sein d'outils de visualisation et de manipulation, mais uniquement à des fins de stockage ou de transmission.

De son côté une 2-carte multirésolution fournit plus d'information de manière explicite et a donc nécessairement besoin de plus d'espace pour être stockée. Cependant, nous en avons tout de même fait usage dans le cadre de la compression de maillages surfaciques, au sein d'un outil permettant de tester et de comparer la qualité de différents opérateurs d'approximation et de prédiction de la géométrie utilisés pour compresser cette dernière. Notre modèle apporte alors ici son efficacité en terme de complexité en temps.



# CONCLUSION

---

## 1 Conclusion

Dans cette thèse, notre objectif était de concevoir un nouveau modèle de représentation de maillages multirésolution pouvant répondre aux exigences des applications qui prennent appui sur ce genre de maillages, et proposer des alternatives face aux défauts que présentent les structures existantes. A ces fins, nous avons choisi de partir des cartes combinatoires, modèle déjà éprouvé en modélisation géométrique, qui présente nombre de caractéristiques intéressantes propres à satisfaire nos besoins.

Nous avons présenté une extension multirésolution du modèle des cartes combinatoires. Une carte multirésolution est définie comme un ensemble de cartes imbriquées dans lesquelles des brins sont ajoutés progressivement afin de raffiner d'un niveau au suivant la subdivision de l'objet représenté. Chaque niveau de résolution est alors représenté par une carte accessible de manière directe, permettant de bénéficier des propriétés de ce modèle au sein de chaque maillage de la hiérarchie. Nous avons d'abord défini les cartes multirésolution dans le cadre très général des hypercartes de dimension  $n$ , puis nous avons spécialisé cette définition en dimension 2, c'est-à-dire au cadre de la représentation de maillages surfaciques, et donné un ensemble d'opérateurs permettant de les manipuler.

Parmi les applications qui exploitent des maillages multirésolution, nous avons choisi deux cadres applicatifs différents : les surfaces de subdivision multirésolution et les maillages progressifs. Les structures de données classiquement utilisées dans ces contextes présentent des défauts dont la plupart sont liés à l'absence d'une représentation complète de l'information topologique associée aux maillages. Pour chacune de ces applications, nous avons montré comment les cartes multirésolution peuvent être mises en œuvre. Nous présentons ci-après les avantages de notre modèle.

## Généralité et souplesse

Les structures usuelles représentent de manière figée un certain type de cellule et sont optimisées pour l'application d'un certain type de raffinement entre les niveaux. Cela impose alors le développement et la maintenance de nombreuses structures de données différentes spécifiques et n'autorise pas naturellement à utiliser des raffinements basés sur des opérations topologiques diverses.

La généralité fournie par les cartes permet la représentation de maillages polygonaux à tous les niveaux de la hiérarchie. Cette propriété permet non seulement d'appliquer notre modèle à des cadres applicatifs exploitant des types de maillages différents, mais aussi de résoudre des problèmes liés à la rigidité des structures existantes. De plus, des opérations de raffinement variées peuvent être appliquées sur ces maillages entre les niveaux, ouvrant alors la voie à l'utilisation de méthodes de raffinement généralement peu utilisées en multirésolution.

Dans le cadre des surfaces de subdivision multirésolution, la souplesse du modèle autorise le support, au sein de la même structure, de nombreux schémas de subdivision. Nous avons illustré ici son utilisation dans le cadre des schémas classiques de Loop [Loo87], Catmull-Clark [CC78] et Doo-Sabin [DS78]. L'application au schéma  $\sqrt{3}$  [Kob00] est rendue possible par l'utilisation d'opérations de basculement d'arêtes entre les niveaux. L'application au schéma quad/triangle [SL03] montre également qu'il est possible ici d'utiliser, de manière directe et cohérente, des opérateurs de raffinement différents au sein d'un même objet.

Des opérations topologiques telles que la contraction d'arête et l'éclatement de sommet sont également applicables entre deux niveaux d'une 2-carte multirésolution. Cela nous a permis d'appliquer notre modèle au cadre des maillages progressifs. Cela montre que celui-ci n'est pas limité à un seul cadre applicatif mais est suffisamment souple pour s'adapter à différents modes de génération

de la hiérarchie de maillages.

## Cohérence topologique

Les structures arborescentes classiques ne gèrent pas l'ensemble des cellules de la subdivision mais représentent uniquement les faces. Les cartes multirésolution ont quant à elles la capacité de gérer des maillages polygonaux quelconques et de représenter l'ensemble des cellules de la subdivision et l'ensemble de leurs relations topologiques. Cela permet de maintenir pour chaque niveau de résolution un maillage consistant.

Les schémas de subdivision faisant usage d'opérateurs primaires utilisent des opérateurs de raffinement topologique agissant sur les arêtes du maillage. Dans le cadre d'un raffinement adaptatif, cette absence de gestion des arêtes dans les structures classiques engendre des trous dans la topologie des maillages représentés. Les solutions proposées pour ces structures mettent en jeu des post-traitements qui demandent une vérification de la configuration des faces voisines ou limitent l'éventail des maillages auxquels on peut accéder dans la hiérarchie.

Au sein d'une 2-carte multirésolution, la gestion fine des arêtes lors de l'application des opérateurs de raffinement permet de faire évoluer le nombre de côtés des faces du maillage de manière cohérente avec le voisinage. Aucune vérification supplémentaire ni aucun ajout de cellule temporaire n'a plus alors à être effectué, et on a la garantie que les maillages auxquels on accède représentent effectivement la partition cellulaire d'une 2-variété orientable.

Cette gestion indépendante et cohérente de la subdivision des cellules au sein d'un maillage est rendue possible par la définition d'une nouvelle notion : le niveau de subdivision d'une cellule. Nous avons donné un algorithme purement combinatoire pour le calcul de cette propriété, puis des variantes optimisées spécialisées à certains modes de construction des maillages.

## Efficacité

Dans tous les cadres applicatifs expérimentés (analyse et édition multirésolution, filtrage de la géométrie, compression), la géométrie des sommets est calculée en fonction des positions des sommets voisins. La mise-à-jour de cette information nécessite de nombreuses requêtes de voisinage et engendre de nom-

breux parcours topologiques.

Pour les surfaces de subdivision, les structures arborescentes répondent à ces requêtes en un temps dépendant de leur profondeur (dans le cadre adaptatif). Pour les maillages progressifs, les structures classiques encodent séparément le maillage courant et l'arbre de modifications. L'accès à la topologie correspondant à un maillage d'un niveau donné nécessite d'appliquer les modifications le séparant du maillage courant. Dans les deux cas, ces opérations ne sont pas optimales.

Dans les cartes multirésolution, chaque maillage de la hiérarchie est représenté par une carte accessible directement et à tout moment. Cela permet de répondre aux requêtes de voisinage en des temps optimaux quel que soit le niveau de résolution considéré.

## Coût mémoire

Les gains en efficacité obtenus pour les maillages progressifs provient de l'encodage explicite et simultané de l'ensemble des maillages de la hiérarchie, alors que seules les modifications sont encodées dans les structures classiques. Pour les surfaces de subdivision, l'amélioration de la cohérence topologique et de l'efficacité des parcours provient d'une part de la représentation de l'ensemble des cellules et de leurs relations de voisinage, et d'autre part de la capacité à modéliser des maillages quelconques. Les structures classiques à base d'arbres sont quant à elles optimisées pour des schémas de subdivision spécifiques. Ces deux aspects pourraient laisser craindre un coût mémoire beaucoup plus important pour les cartes multirésolution.

Nous avons proposé une étude comparative du coût mémoire des implantations de notre modèle et des autres structures mentionnées. Nous avons montré que ces coûts restent dans le même ordre de grandeur. Sauf dans certains cas de figures, comme la construction d'un maillage progressif où le taux de simplification serait particulièrement défavorable, les avantages liés à l'utilisation d'une carte multirésolution ne sont donc pas obtenus aux dépens d'un coût de stockage prohibitif.

## 2 Perspectives

Nos perspectives de recherche s'articulent autour de plusieurs points. Tout d'abord, concernant les surfaces de subdivision multirésolution, il existe encore de nombreux autres schémas de subdivision qui utilisent d'autres méthodes de raffinement tels que les schémas  $\sqrt{2}$  [VZ04], 4-8 [VZ01] ou  $\sqrt{5}$  [ISD05]. Ces schémas peuvent également être implantés au cœur des 2-cartes multirésolution. Comme on l'a vu dans l'application du schéma quad/triangle, la généralité du modèle des cartes multirésolution autorise l'application d'opérateurs de raffinement différents suivant les zones de l'objet. Pour peu que ces opérateurs de raffinement soient naturellement compatibles ou rendus compatibles au niveau de la gestion des frontières entre les zones, on peut imaginer l'utilisation d'algorithmes de raffinement variés au sein d'un même objet.

Dans le même ordre d'idée, nous avons déjà vu comment des algorithmes de subdivision primaux et duaux peuvent s'appliquer sur un maillage de manière parallèle en exploitant le fait que l'information combinatoire des maillages primal et dual est contenue au sein d'une 2-carte. Nous souhaitons étudier s'il est également possible d'appliquer ces algorithmes dans des zones différentes d'un même objet.

Les deux approches que nous avons présentées pour la génération des niveaux, à savoir la subdivision et la simplification progressive, ne sont pas incompatibles pour la construction d'une carte multirésolution. Nous pensons alors à mélanger ces deux approches afin de permettre dans une seule application, de partir d'un maillage donné et d'en obtenir d'une part des versions plus grossières, par l'application d'opérateurs de contraction d'arêtes, et d'autre part des versions plus détaillées, par l'application d'opérateurs de subdivision.

Une autre évolution naturelle de ce travail est de monter en dimension et de s'intéresser à la représentation topologique de maillages multirésolution volumiques. Les cartes multirésolution étant définies en dimension quelconque, de tels maillages sont en effet aisément supportés. Les avantages de notre modèle, notamment sa souplesse en ce qui concerne le type des objets représentés et la manière de les construire, et son efficacité lors des parcours des objets, pourraient également se révéler intéressants dans des applications ayant pour support ce type de maillage.

Au niveau de la définition même du modèle multirésolution, nous voulons mettre en place un mécanisme permettant à un objet, entre deux niveaux successifs, non seulement de se raffiner dans certaines zones, mais également dans le même temps de se simplifier dans d'autres zones. Cela implique de

pouvoir ajouter des brins à certains endroits mais aussi de pouvoir en retirer à d'autres, ce qui n'est pas possible dans la définition actuelle du modèle.

Une manière de mettre ce mécanisme en place sans pour autant opérer de modifications trop profondes dans le modèle serait simplement que chaque brin, au lieu de ne connaître que son niveau d'insertion dans la structure, dispose également d'un niveau de suppression. Ainsi, l'ensemble de brins  $B^i$  décrivant la carte de niveau  $i$  serait défini comme l'union des brins des ensembles  $N^j$  tels que  $j \leq i$ , privé des brins dont le niveau de suppression est inférieur ou égal à  $i$ . Pour chaque relation, un brin inséré au niveau  $i$  ne doit alors plus stocker  $m - i + 1$  coutures, avec  $m$  le niveau maximum de la structure, mais  $s - i + 1$  avec  $s$  le niveau de suppression du brin.

Une autre manière de voir les maillages représentés pourrait alors consister à considérer chaque niveau comme l'état topologique d'un objet à un moment donné. L'évolution topologique d'un objet au cours d'une animation pourrait ainsi être encodée efficacement par une telle représentation.

# TABLE DES FIGURES

---

1	Exemple d'application d'une représentation multirésolution surfacique : plus l'objet est distant, plus le maillage utilisé pour l'affichage est grossier . . . . .	2
1.1	Graphe d'incidence . . . . .	12
1.2	Exemples d'objets variété et non-variété de dimension 1 . . . . .	13
1.3	Exemples d'objets variété et non-variété de dimension 2 . . . . .	13
1.4	Exemples de surfaces orientable (une sphère) et non-orientable (une bande de Moebius) . . . . .	14
1.5	Ambigüité de la représentation en fil de fer . . . . .	15
1.6	Représentation par arêtes ailées . . . . .	15
1.7	Représentation par demi-arêtes . . . . .	16
1.8	La boule pincée : exemple de quasi-variété de dimension 3 . . . . .	17
1.9	Tuples de cellules d'un graphe d'incidence . . . . .	18
1.10	Adjacences entre les tuples de cellules de la subdivision d'une 2-variété . . . . .	19
1.11	Exemple de 2-G-carte . . . . .	20
1.12	Orbites d'une 2-G-carte . . . . .	20
1.13	Couture de deux faces le long d'une arête dans une 2-G-carte . . . . .	21
1.14	Les deux orientation d'une 2-G-carte orientable (brins noirs ou brins blancs) . . . . .	22
1.15	2-carte correspondant à une des deux orientations d'une 2-G-carte orientable . . . . .	23
1.16	Exemple de 2-carte primale . . . . .	25
1.17	Exemple de plongement des sommets d'une 2-carte duale . . . . .	26

1.18	Exemple de représentation d'une surface à bord avec une 2-carte duale . . . . .	27
1.19	Illustration des algorithmes permettant de parcourir l'ensemble des faces incidentes à un sommet, et l'ensemble des sommets adjacents à un sommet sur une 2-carte duale . . . . .	28
1.20	Exemple d'un maillage polygonal et de son maillage dual . . . . .	29
1.21	Passage d'une 2-carte primale à la 2-carte duale représentant le dual du maillage de départ . . . . .	30
1.22	Exemple d'un maillage triangulaire et de son maillage dual . . . . .	30
1.23	L'opérateur de contraction d'arête et son inverse, l'éclatement de sommet . . . . .	34
1.24	Forêt d'arbres binaires de sommets permettant un accès adaptatif dans un maillage progressif . . . . .	36
1.25	Graphe acyclique orienté encodant les dépendances entre des opérations de mise-à-jour d'un maillage polygonal . . . . .	38
1.26	Maillage non-conforme généré par une coupe dans le graphe . . . . .	39
1.27	Exemples de quadtrees de faces à quatre et à trois côtés . . . . .	40
1.28	Exemples de maillages issus de restricted quadtrees et leurs maillages conformes correspondants . . . . .	41
1.29	Exemples de pyramide de 2-G-cartes . . . . .	43
2.1	Ensembles de brins imbriqués dans une carte multirésolution . . . . .	49
2.2	Indexation des relations dans une carte multirésolution primale . . . . .	50
2.3	Opérations de couture dans une 2-carte primale . . . . .	54
2.4	Insertion / Suppression d'un sommet dans une arête dans une 2-carte primale . . . . .	56
2.5	Coupe / Fusion de faces dans une 2-carte primale . . . . .	57
2.6	Basculement d'arête dans une 2-carte primale . . . . .	58
2.7	Opération d'insertion de sommets entre deux niveaux consécutifs d'une 2-carte multirésolution primale . . . . .	60
2.8	Opération d'insertion d'arête entre deux niveaux consécutifs d'une 2-carte multirésolution primale . . . . .	60
2.9	Opération de basculement d'arête entre deux niveaux consécutifs d'une 2-carte multirésolution primale . . . . .	61
2.10	Différentes opérations de couture entre deux niveaux consécutifs d'une 2-carte multirésolution primale . . . . .	62
3.1	Diagramme d'exécution d'un schéma de subdivision interpolant . . . . .	75
3.2	Subdivision de la topologie du maillage et masques de subdivision dans le schéma de subdivision de Catmull-Clark . . . . .	76
3.3	Premiers pas de subdivision d'un cube et surface limite avec le schéma de Catmull-Clark . . . . .	77
3.4	Edition d'une surface de subdivision de Catmull-Clark . . . . .	78

3.5	Edition multirésolution d'une surface de subdivision de Catmull-Clark . . . . .	79
3.6	Diagramme d'exécution d'un schéma de subdivision interpolant modifié pour intégrer des perturbations à chaque niveau . . . . .	80
3.7	Diagramme d'exécution de l'analyse associée à un schéma de subdivision interpolant . . . . .	80
3.8	Diagramme d'exécution de l'analyse (à gauche) et de la synthèse (à droite) associées à un schéma de subdivision interpolant . . . . .	81
3.9	Insertion d'un vecteur de détail dans la hiérarchie multirésolution. Les déformations appliquées par après aux niveaux plus grossiers conservent la déformation fine. . . . .	82
3.10	Exemple de maillage obtenu par une subdivision régulières . . . . .	86
3.11	Quadrisection de triangles . . . . .	87
3.12	Quadrisection de polygones . . . . .	88
3.13	Quadrisection de triangles au sein d'une 2-carte multirésolution . . . . .	89
3.14	Masques utilisés dans le schéma de subdivision de Loop et illustration des algorithmes de calcul au sein d'une 2-carte multirésolution . . . . .	90
3.15	Quadrisection de quadrangles au sein d'une 2-carte multirésolution . . . . .	92
3.16	Subdivision duale de maillage polygonal . . . . .	93
3.17	Subdivision duale de maillage polygonal au sein d'une 2-carte multirésolution duale . . . . .	94
3.18	Masque utilisé dans le schéma de subdivision de Doo-Sabin et illustration des algorithmes de calcul au sein d'une 2-carte multirésolution . . . . .	95
3.19	Représentation d'un maillage polygonal donné par la 2-carte primale encodant son maillage dual . . . . .	97
3.20	Subdivision de Doo-Sabin sur un maillage polygonal par l'application du schéma de Catmull-Clark sur la 2-carte primale encodant son maillage dual . . . . .	98
3.21	Exemple de maillage obtenu par une subdivision adaptative . . . . .	99
3.22	Subdivision adaptative primale de maillage triangulaire : une seule face a été subdivisée entre les deux niveaux . . . . .	100
3.23	Mise-à-jour d'une subdivision adaptative primale de maillage triangulaire en tenant compte du voisinage déjà subdivisé . . . . .	101
3.24	Exemple de subdivision adaptative primale de maillage quadrangulaire . . . . .	103
3.25	Exemple de subdivision adaptative obtenue avec un raffinement dual sur un maillage polygonal . . . . .	104
3.26	Subdivision adaptative de Doo-Sabin sur un maillage polygonal par l'application adaptative du schéma de Catmull-Clark sur la 2-carte primale encodant son maillage dual . . . . .	106

3.27	Une face subdivisée par quadrisection de triangles ne peut être simplifiée directement que si ses sous-faces ne sont pas subdivisées vers des niveaux supérieurs . . . . .	107
3.28	Simplification d'une face subdivisée par quadrisection de triangles	109
3.29	Exemple de maillage subdivisé à l'aide d'un critère uniforme (à gauche) ou variable (à droite) . . . . .	110
3.30	Illustration pour la formulation générale du calcul du niveau d'une face au sein d'un maillage obtenu par subdivision adaptative à l'aide d'un schéma primal . . . . .	116
3.31	Illustration pour la formulation générale du calcul du niveau d'un sommet au sein d'un maillage obtenu par subdivision adaptative à l'aide d'un schéma dual . . . . .	117
3.32	Cas particuliers du calcul du niveau d'une face au sein d'un maillage obtenu par subdivision adaptative de triangles . . . . .	118
3.33	Exemple de subdivision adaptative obtenue avec un raffinement consistant en une quadrisection de quadrangles . . . . .	119
3.34	Schématisme d'un accès régulier dans une 2-carte multirésolution construite de façon régulière (figure du haut) ou adaptative (figure du bas) . . . . .	121
3.35	Schématisme d'un accès adaptatif dans une 2-carte multirésolution construite de façon régulière (figure du haut) ou adaptative (figures du bas) . . . . .	122
3.36	Déroulement de l'algorithme d'accès adaptatif dans une 2-carte multirésolution . . . . .	123
3.37	Résultat de l'accès adaptatif dans une 2-carte multirésolution . . . . .	124
3.38	Gestion de la triangulation du maillage lors de l'affichage d'une 2-carte générée par l'application adaptative du schéma de Loop	125
3.39	Gestion de la triangulation du maillage lors de l'affichage d'une 2-carte générée par l'application adaptative du schéma de Catmull-Clark . . . . .	127
3.40	Application régulière du schéma de subdivision quad/triangle . . . . .	129
3.41	Application adaptative du schéma de subdivision quad/triangle	131
3.42	Exemple de maillage obtenu par une subdivision adaptative avec le schéma quad/triangle . . . . .	132
3.43	Processus de raffinement de la topologie dans le schéma de subdivision $\sqrt{3}$ . . . . .	133
3.44	Masques utilisés dans le schéma de subdivision $\sqrt{3}$ . . . . .	133
3.45	Deux pas de subdivision avec le schéma $\sqrt{3}$ au sein d'une 2-carte multirésolution duale . . . . .	134
3.46	Insertion de nouvelles arêtes dans un triangle dans le processus de raffinement du schéma $\sqrt{3}$ . . . . .	135
3.47	Subdivision $\sqrt{3}$ en deux temps en passant par le maillage dual . . . . .	136

3.48	Deux pas de subdivision adaptative avec le schéma $\sqrt{3}$ au sein d'une 2-carte multirésolution duale . . . . .	138
3.49	Détail d'un maillage lors d'une subdivision adaptative avec le schéma $\sqrt{3}$ . . . . .	139
3.50	Exemple de maillage subdivisé de manière adaptative et édité avec le schéma $\sqrt{3}$ . . . . .	141
3.51	Dans un maillage adaptatif des trous topologiques apparaissent dans le quadtree aux frontières entre les niveaux, alors qu'il reste topologiquement fermé avec une 2-carte multirésolution . . . . .	143
3.52	Parcours des cellules voisines dans une 2-carte . . . . .	145
3.53	Exemples d'algorithmes exploitant les relations de voisinage présentes à différent niveaux de résolution dans une 2-carte multirésolution	146
4.1	Illustration du vecteur à conserver pour reconstruire la position du sommet lors de l'éclatement de sommet . . . . .	156
4.2	Voisinage utilisé pour le calcul de la position d'un sommet après un éclatement de sommet . . . . .	157
4.3	Différentes opérations de filtrage appliquées à un modèle (image tirée de [GSS99]) . . . . .	158
4.4	Décodage d'un maillage triangulaire compressé avec une méthode directe (en-haut) et progressive (en-bas) (image tirée de [AG05]) . . . . .	159
4.5	Exécution de contractions d'arêtes successives dans une 2-carte multirésolution . . . . .	163
4.6	Ensembles de brins après deux contractions d'arêtes dans une 2-carte multirésolution . . . . .	164
4.7	Modification des coutures par $\phi_2$ au fil des niveaux de résolution	166
4.8	Différents cas de dégénérescence du maillage après application d'une contraction d'arête . . . . .	168
4.9	Exemples de cas interdits lors de la sélection de contractions d'arêtes groupées . . . . .	172
4.10	Illustration de l'implantation de la relation $\phi_1$ par opération sur les index au sein d'un tableau de brins . . . . .	175
4.11	Illustration de l'algorithme d'application du filtre de calcul de la position d'un sommet issu d'un éclatement de sommet . . . . .	176
4.12	Exemple d'application des outils de filtrage de la géométrie au sein d'une 2-carte multirésolution . . . . .	179
4.13	Exemple d'édition multirésolution basé sur une hiérarchie de maillages progressifs au sein d'une 2-carte multirésolution . . . . .	181
4.14	Ratio entre le coût mémoire d'une carte multirésolution et des structures classiques en fonction du taux de simplification . . . . .	186



# BIBLIOGRAPHIE

---

- [AD01] P. Alliez and M. Desbrun. Progressive encoding for lossless transmission of 3d meshes. In *Proceedings of SIGGRAPH '01*, pages 198–205, 2001.
- [AG05] P. Alliez and C. Gotsman. Recent advances in compression of 3d meshes. In N.A. Dodgson, M.S. Floater, and M.A. Sabin, editors, *Advances in Multiresolution for Geometric Modelling*, pages 3–26. Springer-Verlag, 2005.
- [Bau75] B.G. Baumgart. A polyhedron representation for computer vision. In *Proceedings of AFIPS*, volume 44 of *National Computer Conference*, pages 589–596, 1975.
- [BD94] Y. Bertrand and J.F. Dufourd. Algebraic specification of a 3d modeler based on hypermaps. *Computer vision, graphical model, and image processing*, 56(1) :29–60, 1994.
- [BDFL93] Y. Bertrand, J.F. Dufourd, J. Françon, and P. Lienhardt. Algebraic specification and development in geometric modeling. In *Proceedings of TAPSOFT'93, Orsay, France*, 1993.
- [Ber98] Y. Bertrand. Topofil : un modeleur interactif d'objets 3d à base topologique. *T.S.I.*, 17(4) :443–484, 1998.
- [Ber04] M. Bertram. Biorthogonal loop-subdivision wavelets. *Computing*, 72(1-2) :29–39, 2004.

- [Ber07] M. Bertram. Wavelet analysis for progressive meshes. In *Proceedings of SCCG'07*, 2007.
- [BK03a] L. Brun and W. Kropatsch. Combinatorial pyramids. In *IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 33–36, Barcelona, Spain, 2003.
- [BK03b] L. Brun and W. Kropatsch. Construction of combinatorial pyramids. In *Workshop on Graph Based Representation in Pattern Recognition*, volume LNCS 2726, pages 1–12, 2003.
- [BK06] L. Brun and W. Kropatsch. Contains and inside relationships within combinatorial pyramids. *Pattern Recognition*, 39(4) :515–526, 2006.
- [BKZ01] H. Biermann, D. Kristjansson, and D. Zorin. Approximate boolean operations on free-form solids. In *Proceedings of SIGGRAPH'01*, pages 185–194, 2001.
- [BLZ00] H. Biermann, A. Levin, and D. Zorin. Piecewise smooth subdivision surfaces with normal control. In *Proceedings of SIGGRAPH'00*, pages 113–120, 2000.
- [BMBZ02] H. Biermann, I. Martin, F. Bernardini, and D. Zorin. Cut-and-paste editing of multiresolution surfaces. In *Proceedings of SIGGRAPH '02*, pages 312–321, 2002.
- [BMZB02] H. Biermann, I. Martin, D. Zorin, and F. Bernardini. Sharp features on multiresolution subdivision surfaces. *Graphical Models*, 64(2) :61–77, 2002.
- [BPZ99] C.L. Bajaj, V. Pascucci, and G. Zhuang. Progressive compression and transmission of arbitrary triangular meshes. In *IEEE Visualization Conference Proceedings*, pages 307–316, 1999.
- [Bri89] E. Brisson. Representing geometric structures in  $d$  dimensions : Topology and order. In *5th ACM Conference on Computational Geometry*, pages 218–227, Saarbrücken, Germany, 1989.
- [Bri93] E. Brisson. Representing geometric structures in  $d$  dimensions : Topology and order. *Discrete & Computational Geometry*, 9 :387–426, 1993.
- [BS05] T. Boubekur and C. Schlick. Generic mesh refinement on gpu. In *Proceedings of Graphics Hardware '05*, 2005.

- [BS08] T. Boubekeur and C. Schlick. A flexible kernel for adaptive mesh refinement on gpu. *Computer Graphics Forum*, 27(1) :102–114, 2008.
- [BSBK02] M. Botsch, S. Steinberg, S. Bischhoff, and L. Kobbelt. Openmesh - a generic and efficient polygon mesh data structure. In *Proceedings of OpenSG Symposium*, 2002.
- [CC78] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6) :350–355, 1978.
- [CD99] D. Cazier and J.F. Dufourd. A formal specification of geometric refinements. *The Visual Computer*, 15 :279–301, 1999.
- [COLR99] D. Cohen-Or, D. Levin, and O. Remez. Progressive compression of arbitrary triangular meshes. In *IEEE Visualization Conference Proceedings*, pages 67–72, 1999.
- [DD04a] C. Dehlinger and J.F. Dufourd. Formalizing generalized maps in coq. *Theoretical Computer Science*, 23 :351–397, 2004.
- [DD04b] C. Dehlinger and J.F. Dufourd. Formalizing the trading theorem in coq. *Theoretical Computer Science*, 23 :399–442, 2004.
- [DFS04] N. Dodgson, M. Floater, and M. Sabin, editors. *Advances in Multiresolution for Geometric Modelling, Series in Mathematics and Visualization*. Springer-Verlag, 2004.
- [DKP04] L. DeFloriani, L. Kobbelt, and E. Puppo. A survey on data structures for level-of-detail models. *Advances in Multiresolution for Geometric Modelling, Series in Mathematics and Visualization*, pages 49–74, 2004.
- [DL03] G. Damiand and P. Lienhardt. Removal and contraction for n-dimensional generalized maps. In *International Conference on Discrete Geometry for Computer Imagery*, volume LNCS 2886, pages 408–419, 2003.
- [DLG90] N. Dyn, D. Levin, and J. A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9(2) :160–169, 1990.
- [DM02] L. DeFloriani and P. Magillo. Multiresolution mesh representation : Models and data structures. In M. Floater, A. Iske, and E. Quak, editors, *Tutorials on Multiresolution in Geometric Modelling*, pages 363–418. Springer-Verlag, 2002.

- [DMP98] L. DeFloriani, P. Magillo, and E. Puppo. Efficient implementation of multi-triangulations. In *IEEE Visualization*, pages 43–50, Research Triangle Park, NC (USA), 1998.
- [Doo78] D. Doo. A subdivision algorithm for smoothing down irregularly shaped polygons. In *Proceedings on Interactive Techniques in Computer Aided Design*, pages 157–165, Bologna, 1978.
- [DP95] L. DeFloriani and E. Puppo. Hierarchical triangulation for multiresolution surface description. *ACM Transactions on Graphics*, 14(4) :363–411, 1995.
- [DPM97] L. DeFloriani, E. Puppo, and P. Magillo. *A formal approach to multiresolution modeling*. Geometric Modeling : Theory and Practice. R. Klein, W. Strasser, R. Rau. Springer-Verlag, 1997.
- [DS78] D. Doo and M. Sabin. Analysis of the behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 10(6) :356–360, 1978.
- [Duf97] J.F. Dufourd. Algebras and formal specifications in geometric modeling. *The Visual Computer*, 13 :131–154, 1997.
- [Duf07] J.F. Dufourd. Design and formal proof of a new optimal image segmentation program with hypermaps. *Pattern Recognition*, 40(11) :2974–2993, 2007.
- [Edm60] J. Edmonds. A combinatorial representation for polyhedral surfaces. In *Notices American Mathematical Society*, volume 7, 1960.
- [ESV99] J. El-Sana and A. Varshney. Generalized view-dependant simplification. *Computer Graphics Forum*, 18(3) :83–94, 1999.
- [FIQ02] M. Floater, A. Iske, and E. Quak, editors. *Tutorials on Multiresolution in Geometric Modelling*. Springer-Verlag, 2002.
- [GSDL06] C. Grasset-Simon, G. Damiand, and P. Lienhardt. nd generalized map pyramids : Definition, representations and basic operations. *Pattern Recognition*, 39(4) :527–538, 2006.
- [GSS99] I. Guskov, W. Sweldens, and Peter Schröder. Multiresolution signal processing for meshes. In *Proceedings of SIGGRAPH '99*, pages 325–334, 1999.
- [HB87] B. Von Herzen and A.H. Barr. Accurate triangulations of deformed, intersecting surfaces. *Computer Graphics*, 21(4) :103–110, 1987.

- [Hop96] H. Hoppe. Progressive meshes. In *Proceedings of SIGGRAPH '96*, pages 99–108, 1996.
- [Hop97] H. Hoppe. View-dependant refinement of progressive meshes. In *Proceedings of SIGGRAPH '97*, pages 189–198, 1997.
- [Hop98] H. Hoppe. Efficient implementation of progressive meshes. *Computers & Graphics*, 22(1) :27–36, 1998.
- [HSB05] S. Hahmann, B. Sauvage, and G.P. Bonneau. Area preserving deformation of multiresolution curves. *Computer Aided Geometric Design*, 22(4) :349–367, 2005.
- [Huf52] D.A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the Institute of Radio Engineers*, 40(9) :1098–1102, 1952.
- [ISD05] I. Ivriissimtzis, M.A. Sabin, and N.A. Dodgson.  $\sqrt{5}$  Subdivision. Advances in Multiresolution for Geometric Modelling. N.A. Dodgson, M.S. Floater, M.A. Sabin. Springer-Verlag, 2005.
- [KCVS98] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of SIGGRAPH'98*, pages 105–114, 1998.
- [Ket99] Lutz Kettner. Using generic programming for designing a data structure for polyhedral surfaces. *Computational Geometry*, 13(1) :65–90, 1999.
- [Kob00] L. Kobbelt.  $\sqrt{3}$  subdivision. In *Proceedings of SIGGRAPH'00*, pages 103–112, 2000.
- [KSS00] A. Khodakovsky, P. Schröder, and W. Sweldens. Progressive geometry compression. In *Proceedings of SIGGRAPH '00*, pages 271–278, 2000.
- [LDW97] M. Lounsberry, T. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 16(1) :34–73, 1997.
- [Lev00] B. Levy. *Topologie Algorithmique - Combinatoire et Plongement*. PhD thesis, Institut National Polytechnique de Lorraine, Nancy, 2000.
- [Lie88] P. Lienhardt. Extension of the notion of map and subdivisions of a 3d space. In *5th symposium on Theoretical Aspects in Computer Science*, pages 301–311, Bordeaux, France, 1988.

- [Lie89] P. Lienhardt. Subdivisions of n-dimensional spaces and n-dimensional generalized maps. In *5th ACM Conference on Computational Geometry*, pages 228–236, Saarbrücken, Germany, 1989.
- [Lie91] P. Lienhardt. Topological models for boundary representation : a comparison with n-dimensional generalized maps. *Computer-Aided Design*, 23(1) :59–82, 1991.
- [Lie94] P. Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal on Computational Geometry and Applications*, 4(3) :275–324, 1994.
- [LM99] B. Levy and J.L. Mallet. Cellular modelling in arbitrary dimension using generalized maps. *Technical report - ISA-GOCAD - INRIA Lorraine*, 1999.
- [Loo87] C. Loop. Smooth subdivision surfaces based on triangles. Master’s thesis, University of Utah, 1987.
- [LS00] M. Lee and H. Samet. Navigating through triangle meshes implemented as linear quadtrees. *ACM Transactions on Graphics*, 19(2) :79–121, 2000.
- [LSS<sup>+</sup>98] A.W.F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin. MAPS : Multiresolution adaptive parameterization of surfaces. In *Proceedings of SIGGRAPH’98*, pages 95–104, 1998.
- [Paj98] R. Pajarola. Large scale terrain visualization using the restricted quadtree triangulation. In *Proceedings of the 9th IEEE Visualization ’98*, pages 19–26, Research Triangle Park, USA, 1998.
- [PR97] J. Peters and U. Reif. The simplest subdivision scheme for smoothing polyhedra. *ACM Transactions on Graphics*, 16(4) :420–431, 1997.
- [PR00] R. Pajarola and J. Rossignac. Compressed progressive meshes. *IEEE Transactions on Visualization and Computer Graphics*, 6(1) :79–93, 2000.
- [PS04] J. Peters and L.-J. Shiue. Combining 4- and 3-direction subdivision. *ACM Transactions on Graphics*, 23(4) :980–1003, 2004.
- [Pup98] E. Puppo. Variable resolution triangulations. *Computational Geometry Theory and Applications*, 11(3-4) :219–238, 1998.
- [Sam84] H. Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys*, 16(2) :187–260, 1984.

- [Sam90] H. Samet. *Applications of Spatial Data Structures : Computer Graphics, Image Processing and GIS*. Addison Wesley, Reading, MA, 1990.
- [Sch92] G. Schrack. Finding neighbors of equal size in linear quadtrees and octrees in constant time. *CVGIP : Image Understanding*, 55(3) :221–230, 1992.
- [SHB07] B. Sauvage, S. Hahmann, and G.P. Bonneau. Volume preservation of multiresolution meshes. *Computer Graphics Forum (Proceedings Eurographics 2007)*, 26(3) :275–283, 2007.
- [SL03] J. Stam and C. Loop. Quad/triangle subdivision. *Computer Graphics Forum*, 22(1) :79–85, 2003.
- [SW05] S. Schaefer and J. Warren. On  $c^2$  triangle/quad subdivision. *ACM Transactions on Graphics*, 24(1) :28–36, 2005.
- [Swe97] W. Sweldens. The lifting scheme : A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2) :511–546, 1997.
- [Tau95] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of SIGGRAPH'95*, pages 351–358, 1995.
- [Tau02] G. Taubin. Detecting and reconstructing subdivision connectivity. *The Visual Computer*, 18(5-6) :357–367, 2002.
- [TGHL98] G. Taubin, A. Guézic, W. Horn, and F. Lazarus. Progressive forest split compression. In *Proceedings of SIGGRAPH '98*, pages 123–132, 1998.
- [The00] S. They. *Plongement de complexes cellulaires*. PhD thesis, Université Louis Pasteur, Strasbourg, 2000.
- [The07] S. They. Implémentation objet générique d'une plateforme de modélisation géométrique. In *Groupe de Travail en Modélisation Géométrique (GTMG)*, Valenciennes, 2007.
- [Vin83] A. Vince. Combinatorial maps. *Journal of Combinatorial Theory*, pages 1–21, 1983.
- [VZ01] L. Velho and D. Zorin. 4-8 subdivision. *Computer Aided Geometric Design*, 18(5) :397–427, 2001.
- [VZ04] L. Velho and D. Zorin.  $\sqrt{2}$  subdivision for quadrilateral meshes. *The Visual Computer*, 20(2-3) :180–198, 2004.

- 
- [Wei85] K. Weiler. Edge-based data structures for modeling in curved-surface environments. *Computer Graphics and Applications*, 5(1) :21–40, 1985.
- [Zor05] D. Zorin. Modeling with multiresolution subdivision surfaces. *Eurographics'05 Tutorial*, 2005.
- [ZS01] D. Zorin and P. Schröder. A unified framework for primal/dual quadrilateral subdivision schemes. *Computer Aided Geometric Design*, 18(5) :429–454, 2001.
- [ZSD<sup>+</sup>00] D. Zorin, P. Schröder, T. DeRose, L. Kobbelt, A. Levin, and W. Sweldens. Subdivision for modeling and animation. *SIGGRAPH'00 Course Notes*, 2000.
- [ZSS96] D. Zorin, P. Schröder, and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. In *Proceedings of SIGGRAPH'96*, pages 189–192, 1996.
- [ZSS97] D. Zorin, P. Schröder, and W. Sweldens. Interactive multiresolution mesh editing. In *Proceedings of SIGGRAPH'97*, pages 259–268, 1997.