



HAL
open science

Etude de la représentation géométrique et texturale de scènes 3D pour les services de visualisation dans un contexte télécommunicant

Pierre Alliez

► **To cite this version:**

Pierre Alliez. Etude de la représentation géométrique et texturale de scènes 3D pour les services de visualisation dans un contexte télécommunicant. Interface homme-machine [cs.HC]. Télécom ParisTech, 2000. Français. NNT: . tel-00006476

HAL Id: tel-00006476

<https://pastel.hal.science/tel-00006476>

Submitted on 16 Jul 2004

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Résumé

Ce mémoire examine le problème de la représentation des données contenues dans les scènes tridimensionnelles, ces dernières étant destinées à une finalité graphique et diffusées dans un contexte télécommunicant.

La réalité virtuelle représente aujourd'hui une technologie à part entière puisqu'elle autorise la simulation, la navigation et l'interaction avec un univers sensoriel synthétisé de toutes pièces. Les potentialités offertes par la réalité virtuelle ont été très bien perçues lorsque les ordinateurs graphiques ont été capables d'assurer un rendu des images à une cadence interactive. De plus, l'augmentation des capacités des machines coïncide aujourd'hui avec un développement vertigineux des réseaux, ce qui décuple littéralement le champ d'application de la réalité virtuelle. Plusieurs difficultés subsistent toutefois pour développer les services de télécommunications associés : le volume des données est considérable, l'affichage est coûteux en calculs et la nature des données nécessite une prise en compte spécifique dans le contexte client-serveur. Si l'on ajoute à cela l'inégalité des capacités des terminaux et la variabilité des réseaux, on obtient une synthèse du triple challenge de la thèse : le stockage, la transmission et la visualisation de scènes 3D complexes. On pourrait aussi résumer la problématique en endossant le temps d'une phrase, le rôle d'un opérateur de télécommunication : "quels que soient : le terminal, le volume de données à transmettre et le débit du réseau, l'utilisateur doit pouvoir visualiser à tout moment une image attractive".

Ce mémoire relève le défi sous la forme de briques technologiques susceptibles de s'assembler pour composer un moteur de transmission et d'affichage de scènes 3D en réseau. Il propose des solutions correspondant à un schéma courant en codage de l'information audiovisuelle : simplification, approximation, et codage échelonné. Il reste cependant une spécificité propre à la 3D : la visualisation. En effet, il est d'une part plus coûteux de projeter un polygone à l'écran plutôt que d'afficher un pixel, et d'autre part le point de vue hautement variable d'une caméra influence sensiblement la pertinence des données transmises.

Le premier chapitre, introductif, fournit une compréhension naturelle de la problématique et les trois autres suivent naturellement l'ordre des opérations appliquées sur les scènes 3D : la simplification et l'approximation, le codage et la visualisation. Plus précisément, les maillages composant les objets d'une scène sont simplifiés par fusion d'arête et approximés par minimisation de volume. Ils sont ensuite encodés par une technique de conquête sur les arêtes exploitant les valences des sommets. L'échelonnabilité est, quant à elle, obtenue sur les positions par une technique apparentée aux plans binaires de raffinement. Le codage des textures est adressé par intégration d'une technique développée au sein du laboratoire, cette dernière combinant la transformée en cosinus discrète et les éléments finis sur une hiérarchie de maillages triangulaires. Sur le terminal, la visualisation dépendant du point de vue est assurée par une technique de reconstruction adaptative des surfaces fonctionnant par subdivision successive des régions d'intérêt et des silhouettes. Nous montrons qu'une telle reconstruction s'applique en cours de transmission et s'adapte aisément à la puissance graphique d'un terminal. Enfin, les perspectives de développement concernent l'entrelacement du codage avec la technique de reconstruction adaptative et le contrôle de la déviation géométrique lors de la simplification.

Abstract

This document examines the problem of data representation of three-dimensional scenes, these ones being intended for a graphic purpose and streamed on various networks.

Virtual reality now represents a whole technology since it authorizes simulation, navigation and interaction with a fully synthesized universe. The potentialities offered by virtual reality were well understood when graphic computers were able to display synthetic images at an interactive rate. Moreover, the power increase of computers coincides now with a rapid development of networks, which increases tenfold the application field based on virtual reality. Nevertheless, a difficulty remains in developing these applications ; the associated telecommunication services appear delicate to elaborate for various reasons : the data volume may be huge, the display is CPU-intensive and the graphic nature of data requires a specific development adapted to the client-server framework. And more, terminals are not equal in term of capacities, and networks have limited and variable flows. One thus obtains the triple challenge of the thesis : storage, transmission and visualization of 3D scenes. One could also summarize the problematics by endorsing during one sentence, the role of a telecommunication operator : whatever : the terminal, the volume of data and the network flow, the user may be able to visualize an appealing image.

This document takes on the challenge and describes technological elements suitable for transmission and the display of 3D scenes. Proposed solutions correspond to a standard scheme in the field of audio-visual information : simplification, approximation, and scaleable coding. The visualization remains however 3D-specific and even represents an essential part. Indeed, it is obviously more expensive to display a polygon rather than a pixel, and the variable camera point of view highly influences the relevance of transmitted data.

The first chapter provides a natural understanding of the problematic and the three others naturally follow the operations applied to 3D scenes : simplification and approximation, coding and visualization. The meshes composing the objects of a 3D scene are simplified by edge collapse and are approximated by volume minimization. They are then encoded by an edge conquest technique exploiting the vertex degrees distribution, the scalability being obtained onto the positions by a bitplane-like technique. Texture coding is addressed by integration of a technique developed within the laboratory, this one combining discrete cosine transform and finite elements associated with a triangular mesh hierarchy. Visualization depending on the point of view of the client is ensured by successive subdivision of region of interest and silhouette areas. We show that such a technique also applies during transmission and reconstruction, and easily adapts geometric complexity to the graphic power of a client. Lastly, future work concerns the interlacing of coding with the adaptive subdivision technique, and the deviation control during mesh simplification.

Préambule

Le multimédia diffusé partout, tout le temps et pour tous, voici comment pourrait se résumer le leitmotiv des opérateurs de télécommunications. En pratique il s'agit d'un véritable défi, nécessitant la conjonction de techniques étroitement liées au codage et à la transmission des données.

Ce mémoire examine plus précisément les problèmes de la représentation de l'information graphique contenue dans les scènes tridimensionnelles. Il consigne les principaux travaux effectués après trois ans de thèse au sein de l'unité de recherche de codage et d'indexation de l'information multimédia au sens large (les images fixes, animées, synthétiques et aussi le son), le fil conducteur correspondant à la diffusion de l'information sur des réseaux hétérogènes.

Parallèlement à une augmentation vertigineuse de la quantité d'information circulant sur le réseau, une diversification tout aussi spectaculaire de nouveaux services à *forte valeur ajoutée* est proposée. Ce constat explique aussi pourquoi le rapport données / voix augmente de façon exponentielle. En pratique, chaque nouveau service nécessite un codage spécifique, même si l'on retrouve des réflexes communs entre les codeurs audio / vidéo et 3D. Ainsi, transmettre des données de nature graphique nous amène rapidement à consulter les travaux de la communauté de géométrie algorithmique, des infographistes, et même des traiteurs d'images puisque la finalité est commune : produire des images attractives.

Les noyaux durs communs aux données multimédia concernent principalement le volume de données à transmettre, mais aussi l'adaptation aux disparités des infrastructures de télécommunications. En effet, à partir d'une base de données unique, comment acheminer l'information d'une part à un utilisateur connecté sur un réseau à haut débit muni d'un ordinateur graphique dernier cri, et d'autre part à un utilisateur muni d'un terminal rudimentaire connecté par modem ? Nous avons trouvé quelques réponses à ce défi par le biais d'une étude sur la compression échelonnable des données et la reconstruction adaptative sur un terminal. L'objet de ce document consiste principalement à développer des méthodes algorithmiques élaborées afin de traduire les notions de compression et d'échelonnabilité sur les données de nature essentiellement géométrique contenues dans les scènes tridimensionnelles. Enfin, les techniques de reconstruction adaptative des surfaces par subdivision s'avèrent particulièrement bien adaptées à une visualisation dépendant du point de vue d'un observateur sur le terminal.

Table des matières

1	Introduction	7
1.1	Motivations	7
1.1.1	La réalité virtuelle à finalité graphique	7
1.1.2	Les applications potentielles	8
1.1.3	Les services télécommunicants	9
1.1.4	Positionnement de la thèse vis à vis des applications	11
1.2	Définition du problème	12
1.3	Travaux précédents	13
1.4	Positionnement des travaux	14
1.4.1	Simplification et approximation	14
1.4.2	Codage	15
1.4.3	Echelonnabilité du codage	15
1.4.4	Visualisation	16
1.5	Organisation du document	16
1.6	Préliminaires	16
1.6.1	Les scènes en réalité virtuelle	16
1.6.2	La représentation par maillages	17
	Du monde réel au graphisme sur ordinateur	17
	Acquisition et modélisation	18
	Les maillages surfaciques	21
1.6.3	Transmission et rendu sur écran	23
2	Simplification et approximation	27
2.1	Introduction	27
2.2	Travaux précédents	28
2.2.1	Décimation	29
2.2.2	Raffinement par subdivision	36
2.2.3	Echantillonnage	37
2.2.4	Positionnement des travaux	38
2.3	Approximation	38
2.3.1	Principe général	38
2.3.2	Métrie basée volume	38
2.3.3	Optimisation en position	40
	Variation élémentaire de volume	40
	Interpolation	41
	Evolution de surface	42

	Gradient à pas adaptatif	45
2.3.4	Résultats	46
2.4	Simplification par fusion d'arêtes	49
2.4.1	Principe général	49
2.4.2	Simplification géométrique	49
2.4.3	Double queue de priorité	49
2.4.4	Pré-positionnement	51
2.4.5	Résultats	52
2.5	Discussion et perspectives	52
2.5.1	Minimisation de volume	56
2.5.2	Suppression de la redondance	56
3	Codage	59
3.1	Introduction	59
3.2	La compression des données	60
3.2.1	Compression sans perte	60
	Supprimer la redondance	60
	Repérer les plages de valeurs	62
3.2.2	Compression avec pertes	62
	Réduire la précision	62
	Redistribuer l'énergie	63
3.3	La notion d'échelonnabilité	63
3.3.1	Du point de vue d'un opérateur	63
3.3.2	Différents types de scalabilité	64
3.3.3	Les données 3D	65
	Le multiplexage	65
	Le raffinement continu	67
3.4	Travaux précédents	67
3.4.1	Les formats de données	67
3.4.2	Les industriels	67
3.4.3	Les consortiums	68
3.4.4	Compression de scènes tridimensionnelles	68
3.5	Codage de maillages triangulaires	74
3.5.1	Principe général	77
3.5.2	Encodage	77
	Connectivité	78
	Géométrie	88
3.5.3	Décodage	97
	Connectivité	99
	Géométrie	103
3.5.4	Résultats	103
	Connectivité	105
	Géométrie	110
3.6	Codage des textures	112
3.6.1	Représentation	112
3.6.2	Compression	119
3.6.3	Résultats	119

3.6.4	Une piste vers la visualisation locale adaptative	121
3.7	Discussion et perspectives	121
4	Visualisation	125
4.1	Introduction	125
4.1.1	Un budget polygones par image	125
4.1.2	Les critères de réalisme	126
4.2	Travaux précédents	127
4.2.1	Détection des silhouettes	127
4.2.2	Raffinement adaptatif	129
4.2.3	L'analyse multirésolution	130
4.2.4	La subdivision de surfaces	130
4.2.5	La simplification adaptative	134
4.2.6	La piste choisie	134
4.3	Raffinement adaptatif de maillages triangulaires	134
4.3.1	Principe général	135
4.3.2	Détection des régions d'intérêt	135
4.3.3	Subdivision hybride	136
4.3.4	Filtrage de la géométrie	139
4.3.5	Ajout de contraintes	141
4.3.6	Continuité du raffinement	144
4.3.7	Optimisation et robustesse	150
4.3.7.1	Détection des régions d'intérêt	150
4.3.7.2	Structure de données	155
4.3.8	Résultats	158
4.3.8.1	Comparaison avec la subdivision de Loop	158
4.3.8.2	Raffinement global	158
4.3.8.3	Raffinement dépendant du point de vue	159
4.3.8.4	Reconstruction de surfaces courbes par morceaux	159
4.3.8.5	Reconstruction en cours de décodage	159
4.4	Discussion et perspectives	159
5	Conclusion et perspectives	173
5.1	Récapitulatif	173
5.2	Perspectives	174
5.3	Les prochains challenges	176

Bibliographie	179
Annexe A	195
Annexe B	197
Annexe C	203
Annexe D	205
Liste des figures	212
Liste des tableaux	230

Chapitre 1

Introduction

Ce mémoire concerne la réalité virtuelle à finalité graphique et ses applications sous la forme de services distribués en réseau : le principal enjeu consistant à faciliter la mise en relation des données avec les utilisateurs. Avant d'aborder la description des techniques mises en œuvre, décrivons les motivations qui nous ont amenés à relever le challenge.

1.1 Motivations

Cette section devrait nous assurer une compréhension naturelle et progressive de la problématique. Pour cela, il convient de définir dans un premier temps la notion de réalité virtuelle, les applications potentielles, puis de replacer cette technologie dans le contexte graphique et télécommunicant étroitement lié au sujet de ce mémoire, plus proche des préoccupations d'un opérateur de télécommunications.

1.1.1 La réalité virtuelle à finalité graphique

Définition

La réalité virtuelle représente aujourd'hui une technologie à part entière. Par le biais de la création d'environnements artificiels et de calculs informatiques, celle-ci permet à un utilisateur de vivre une expérience en trois dimensions. La clé de son succès réside essentiellement dans la sensation d'immersion qu'elle procure sans qu'aucun objet réel ne soit présent.

Les volontés des acteurs du domaine

Les acteurs de la réalité virtuelle cherchent soit à reproduire un réalisme confondant en générant un univers crédible d'un point de vue sensoriel, soit à nous faire approcher le *merveilleux technologique* d'un univers de science-fiction. En effet, dans notre vie réelle, nous (et les objets) ne saurions échapper aux lois élémentaires de la physique, tandis qu'un univers sensoriel synthétisé de toutes pièces supprime la notion de distance, d'échelle, de masse, et autorise l'ubiquité.

Une finalité graphique

Les ordinateurs graphiques actuels permettent de contempler et d'animer des images de scènes tridimensionnelles, éventuellement en relief. De tels systèmes s'adressent à notre système visuel, et la motivation de ce mémoire concerne uniquement cette finalité graphique sur un écran constituant dans ce contexte la *fenêtre sur le monde virtuel*. Pourtant, gardons à l'esprit que l'objectif final de la réalité virtuelle concerne l'ensemble de nos sens. Idéalement on devrait pouvoir toucher les objets, sentir des odeurs, ressentir les variations de température, etc... afin de ne plus se contenter de regarder une scène mais d'en faire partie (on parle aussi

d'immersion). Le succès rencontré par les ordinateurs graphiques et les images de synthèse accompagne l'art de l'illusion, qui connaît un développement exponentiel depuis que nous savons reproduire, créer et manipuler des images. Cette croissance explique pourquoi les efforts sont principalement concentrés sur la création des images (l'infographie) depuis plusieurs années : c'est en effet le moyen le plus efficace pour créer l'illusion.

Comment créer un monde virtuel ?

Créer un monde virtuel nécessite de modéliser des scènes, ses objets et ses comportements physiques dans un premier temps, puis d'en assurer l'affichage à une cadence interactive. La majorité des objets sont opaques, l'illusion peut donc être créée en se limitant seulement à la modélisation de la *peau* des objets, on parle alors de représentation surfacique (voir section 1.6.2). Des couleurs, des matériaux, et des textures sont ajoutés aux objets afin de gagner en réalisme, et un éclairage complexe peut être simulé. L'augmentation des détails rend délicat la phase de modélisation, le stockage de la base de données et surtout le rendu des images. En effet, suivre simplement les modifications de position d'un observateur peut nécessiter un nombre trop important de calculs entre chaque image : il s'agit du problème de l'*exploitation* d'un modèle virtuel. L'ordinateur se révèle souvent beaucoup trop lent pour animer une scène au rythme du rafraîchissement d'écran souhaité. De ce fait, il va falloir développer des algorithmes de restriction, de modification, de simplification, de compression et d'amélioration sur les modèles afin de rendre une scène réellement attractive. Cet éventail de procédés fait justement l'objet du travail présenté dans ce mémoire.

En pratique, et malgré un grand nombre d'heuristiques, les images sont rarement photo-réalistes. Toutefois, la puissance actuelle des ordinateurs suffit à rendre la technique utilisable. De plus le champ d'applications apparaît si vaste, qu'il rend impossible la création d'une liste exhaustive. Les sections 1.1.2, 1.1.3 et 1.1.4 présentent respectivement un éventail des applications potentielles, les services télécommunicants envisagés à plus court terme, puis le positionnement de la thèse vis à vis des applications.

1.1.2 Les applications potentielles

La réalité virtuelle permet la simulation, et touche de ce fait de très nombreux domaines. La liste suivante consigne quelques applications potentielles :

- les visites virtuelles : la simulation autorise la navigation à l'intérieur de sites existants et modélisés à partir de relevés géométriques et de photos prises sur le terrain. Des sites anciens peuvent également être virtuellement reconstruits à partir d'archives, de photos ou de toute autre information permettant le retour dans le temps. La réalité virtuelle peut ainsi jouer un rôle éducatif important dans ce contexte,
- la conception assistée par ordinateur : les professionnels de l'industrie et les architectes souhaitent travailler sur des modèles de plus en plus réalistes afin de gagner en clarté auprès de leurs interlocuteurs. Dans ce domaine, la réalité virtuelle évite bien souvent la construction onéreuse de maquettes miniatures,
- la cartographie : les données acquises par imagerie radar ou optique permettent de modéliser fidèlement des terrains dont la taille est souvent considérable. Un moteur de réalité virtuelle autorise ensuite la navigation à l'intérieur de telles scènes. La représentation des données joue alors un rôle crucial pour ces applications,
- les jeux : il s'agit d'un domaine très innovant en matière d'imagerie tridimensionnelle, les créateurs cherchant en permanence à plonger le joueur doté de pouvoirs extraordinaires dans un univers fantastique et captivant,
- le spatial : l'espace est un domaine privilégié pour la simulation en situation d'apesanteur ou de micro-apesanteur, les lois de la mécanique pouvant être adaptées à sa guise en réalité virtuelle,

- l’enseignement : un centre de documentation virtuel est accessible à toute heure, où le contenu peut être consigné d’une part dans des livres rapidement accessibles par de puissants outils de navigation, et d’autre part sous la forme de modèles tridimensionnels animés et interactifs. Il est en effet bien plus naturel de *prendre en main* un objet dont on veut comprendre le fonctionnement plutôt que de se contenter d’en consulter le plan ou les photos,
- le commerce électronique : de par son caractère très séduisant, la réalité virtuelle commence à prendre place dans des applications encore expérimentales de commerce électronique. Idéalement il serait possible de déambuler à l’intérieur de boutiques virtuelles, d’examiner et de visualiser le principe de fonctionnement d’un produit avant de prendre une décision d’achat,
- la simulation : une image de synthèse est élaborée afin d’imiter son référent : la réalité, qu’elle a pour mission de représenter. Elle permet d’accroître la précision des constatations en diminuant sensiblement les coûts. La réalité virtuelle permet ainsi d’*essayer* le réel, ce qui correspond assez bien au métier des ingénieurs, c’est à dire concevoir, prévoir et anticiper,
- la réalité augmentée : les images numériques peuvent se superposer à la vision naturelle. A titre d’exemple, cette technologie est cruciale pour un pilote d’engins de plus en plus rapides qui n’a plus le temps de glaner des informations sur son tableau de bord. On laisse ainsi mesurer des paramètres à la machine, qui sont ensuite projetés en surimpression sur un cockpit ou une visière,
- la médecine : les images jouent un rôle prépondérant et croissant lors des phases médicales : diagnostic, thérapie per et post-opératoire. Imaginons un corps humain virtuel associé à une modélisation géométrique, physique et physiologique. Ceci autoriserait la simulation d’interventions chirurgicales et doterait la chirurgie mini-invasive de moyens supplémentaires, pour laquelle l’enjeu consiste à permettre à un chirurgien spécialiste du domaine d’opérer un patient d’un autre pays. Le scénario suivant n’est pas irréaliste : le chirurgien pilote (à plus grande échelle pour être plus précis) des robots depuis un périphérique haptique à distance, tandis qu’il reçoit en retour une image virtuelle des organes réellement modifiés, acquise par un système d’imagerie volumique. L’atlas de ces organes est visualisé en fausses couleurs afin de mieux discerner les parties sensibles. L’immense avantage d’un simulateur réside dans sa capacité à reproduire à volonté des situations rarement rencontrées, les seules qui posent problème.

1.1.3 Les services télécommunicants

Nous ne nous étendrons pas sur la révolution promise par la société de l’information, mais nous pouvons toutefois affirmer que grâce aux télécommunications, la réalité virtuelle peut être maintenant distribuée à l’autre bout de la terre et de l’espace quasi instantanément, et être ainsi partagée entre plusieurs usagers. Toutes les applications précédemment citées peuvent être ainsi redéfinies sous la forme de services en réseau, mais la réalité virtuelle immersive ne correspond pas vraiment aux applications envisagées dans l’immédiat par l’entreprise. L’enjeu réside plutôt dans le développements d’interfaces homme-machine évoluées et surtout de services à forte valeur ajoutée. De telles applications mettent en œuvre la transmission de contenus interactifs et ciblent des services bien précis. La liste suivante consigne les applications télécommunicantes clairement envisagées :

- le télécommerce : identifié comme le service prioritaire. L’enjeu consiste à réaliser une présentation interactive de produits associée à une assistance en ligne comprenant des modes d’emploi visuels et interactifs, et les principes de fonctionnement. Toutefois, l’augmentation du trafic ¹ lié à ce genre de service ne représente pas un enjeu suffisant pour un opérateur. Il s’agit bien plus de proposer des technologies exclusives liées aux *outils serveur* : la transmission progressive, la compression ou encore la visualisation, afin de fournir un service à forte valeur ajoutée,

¹appelé aussi *delta-minutes* dans le jargon des opérateurs.

- les applications culturelles : plus immersives, tout en possédant un lien étroit avec le télécommerce et les activités marchandes. La visite d'un village antique grec concrétise notamment ce genre de service. Après avoir consulté les avis de groupes d'utilisateurs potentiels, il s'avère crucial de développer au cœur de ces applications l'interactivité, la notion de groupes guidés, d'intérêt commun (autour de l'histoire d'un site reconstruit à titre d'exemple) et surtout de sites virtuels dynamiques car habités et maintenus par une communauté regroupant à la fois les utilisateurs et les créateurs de contenus. A titre d'exemple, un site antique grec virtuel s'avère bien plus attractif lorsqu'il se construit progressivement, avec l'avis des utilisateurs regroupés en une communauté régie suivant les lois de l'époque. L'étude consultative met en évidence le fait que la simple visualisation de scènes statiques, même dessinées avec soin, possède un intérêt limité. Il s'agit donc de renouveler la curiosité de l'utilisateur. Nous approchons ici la notion de communautés virtuelles, jugée suffisamment attractive pour constituer une famille de services à part entière,
- les communautés virtuelles : les salons de discussion en ligne (les *chats*) représentent à eux seuls 50% du volume des communications. L'intérêt pour l'entreprise est toutefois moindre pour ce type d'application puisque le contenu est ici entièrement créé par les utilisateurs. Il s'avère donc difficile de générer une valeur ajoutée à un service évoluant en permanence au gré des volontés d'une communauté virtuelle. Le *portail* constitue néanmoins un service à valeur ajoutée sous la forme d'une interface entre l'opérateur et la communauté. Un service de mise en correspondance de communautés virtuelles a été notamment expérimenté sous la forme d'une visualisation tridimensionnelle des connectés. En pratique un tel service s'avère difficile à mettre en œuvre : la navigation est malaisée, et l'ergonomie y joue un rôle primordial. Enfin, les besoins ne sont pas encore clairement identifiés en terme de centres d'intérêts. A l'inverse, le jeu définit plus clairement les contours d'un centre d'intérêt commun à un groupe de joueurs connectés,
- le jeu : par essence interactif, habité, dynamique et regroupant une communauté de joueurs évoluant dans une base de données de grande taille. Les jeux évolutifs en ligne connaissent aujourd'hui un succès considérable, favorisant ainsi la création de nouveaux besoins et donc de services à valeur ajoutée : réseaux haut débit, portails dédiés, etc... On constate en pratique que le jeu réunit toutes les conditions pour favoriser l'innovation en matière de services en ligne. Si l'entreprise consacre une partie de ses développements autour des activités ludiques ou marchandes précédemment évoquées, la communication de groupe pour le travail collaboratif présente toutefois les perspectives de développement et les défis les plus intéressants,
- le travail collaboratif : lié à l'entreprise, à la mondialisation de l'économie et à la communication de groupe. Il s'agit d'un secteur en évolution rapide, où la demande s'avère très prononcée de la part des entreprises qui perçoivent la réalité virtuelle (associée à la vidéo) comme un outil de travail entre des personnes ou des décideurs distants. Un service de travail collaboratif en ligne réunirait la téléconférence, le bureau virtuel et les maquettes en une même entité où pourrait avoir lieu les discussions en situation autour de projets divers : problèmes localisés, règles de fabrication, décisions, etc... Une application très demandée correspond à la télé-maintenance ou la *réunion de crise*, où l'enjeu consiste à concentrer en un délai très court les compétences d'experts distants capables de prendre les décisions adaptées (un expert est en général rare et coûteux, et son temps doit être optimisé). Les solutions proposées aujourd'hui en vidéoconférence pour l'entreprise correspondent à des services à très forte valeur ajoutée : réseaux à débit élevé et garanti, terminaux et applications spécialisés, etc... La 3D viendrait ici naturellement augmenter la qualité de tels services puisqu'elle autorise une mise en situation réaliste, l'animation, l'interactivité et une adaptativité sensiblement accrue (absence de commutation vidéo envisageable, ubiquité, multiplicité, ajout de métaphores visuelles pour la prise de parole, plusieurs points de vue simultanés sur la scène, changement d'échelle des maquettes virtuelles, etc...). Le télé-bureau virtuel est encore à l'état de maquettes de services, et donne matière à réflexion en matière d'interface homme-machine. Une appli-

cation privilégiée et connexe concerne également l'accès à l'information de manière évoluée par le biais d'humanoïdes virtuels,

- les humanoïdes virtuels accompagnent la diffusion de l'information par une information visuelle, et permettent de développer un aspect conversationnel. Les clones [vhu] sont sérieusement envisagés pour des services très ciblés comme la lecture de courriers électroniques, les nouvelles distribuées en ligne, les forums de discussion ou les bornes interactives. Toutefois la technologie ne s'avère pas assez mature pour *humaniser* l'ordinateur. La dernière étude met en évidence le fait que la difficulté se situe plus au niveau de l'intelligence artificielle et de l'animation réaliste qu'au niveau du graphisme. En pratique l'animation s'avère aussi riche, voire plus, en terme de quantité d'information que le modèle graphique, et souvent plus délicate à modéliser. L'aspect synthèse / reconnaissance vocale pose également des défis non résolus à l'heure actuelle. Les premiers essais auprès d'utilisateurs font apparaître clairement l'aspect frustrant du décalage des technologies : un clone parfaitement photo-réaliste génère une attente inconsidérée en terme d'intelligence artificielle. Les humanoïdes virtuels sont donc destinés à des applications de service neutres, et l'on peut encore prédire un avenir radieux aux caméras interactives...

1.1.4 Positionnement de la thèse vis à vis des applications

La diversité des applications envisagées nous amène à privilégier un service télécommunicant particulier, tout en essayant de construire des briques technologiques génériques. Le télécommerce correspond ainsi à l'application type, ce dernier possédant un lien étroit avec les applications culturelles et le jeu. Le scénario suivant constitue un exemple d'application :

- les scènes sont composées d'objets et de décors correspondant soit aux terrains, soit aux bâtiments composant l'environnement. On considère uniquement la transmission d'objets modélisés par maillages surfaciques,
- le rôle de l'utilisateur consiste à examiner les objets au cours de la navigation, à déambuler dans les décors, et à interagir avec les composants de la scène,
- nous considérons seulement l'aspect transmission, décodage et visualisation des données.

Finalement, l'enjeu prioritaire consiste à faciliter l'acheminement des données depuis le serveur jusqu'au terminal d'un utilisateur. Il apparaît donc pertinent de connaître en premier lieu la nature de l'information.

L'information est immatérielle

Même si nous pouvons affirmer que l'information est immatérielle sur un réseau (sous une forme d'électrons dans un câble, de photons confinés dans une fibre optique ou d'ondes électromagnétiques dans l'atmosphère ou l'espace), il en va autrement pour son stockage et sa transmission puisque des limitations et des variations de capacité et de débit freinent le développement des services distribués. Il est en effet nécessaire de construire des supports physiques afin de stocker et de transmettre l'information, et de fournir des efforts pour l'acheminer. Il s'agit donc de trouver une représentation adaptée au contexte télécommunicant de l'information contenue dans les scènes tridimensionnelles.

Les contraintes réseau

Le fonctionnement d'un réseau partagé par de multiples utilisateurs peut s'avérer extrêmement complexe, et l'on ne s'attardera pas sur ces détails techniques dans ce mémoire. Nous résumons brièvement les principales contraintes d'un réseau : limitation et variabilité. En effet, le débit disponible par l'utilisateur d'un réseau est limité et variable dans le temps ; de plus, plusieurs réseaux peuvent s'interconnecter. Il s'agit donc de générer un *flux adaptable* sur les réseaux à partir de scènes tridimensionnelles. Nous verrons comment cela se traduit en terme de représentation ; mais nous pouvons déjà esquisser le terme d'échelonnabilité sur les données qui

permet d'assouplir l'interprétation des données lors de la réception et surtout d'afficher très tôt une version acceptable des données sur un terminal.

1.2 Définition du problème

Transmettre l'information ne suffit plus. Il s'avère aujourd'hui crucial pour un opérateur de télécommunications de développer une expertise sur les données véhiculées par les réseaux.

Depuis plusieurs dizaines d'années, le problème de la modélisation et du rendu réaliste de scènes tridimensionnelles correspond à la préoccupation principale des infographistes. Toutefois, celui de la représentation et du codage des données 3D à des fins de transmission en réseau, plus récent, préoccupe les chercheurs depuis l'accélération vertigineuse du développement des réseaux. Le triple objectif de cette thèse consiste donc à développer des méthodes facilitant le stockage, la transmission et le rendu de scènes, en vue d'une application finale de diffusion graphique sur des réseaux et des terminaux hétérogènes en terme de débits et de capacités.

D'un point de vue plus technique, les méthodes de codage et de représentation doivent s'adapter au contexte télécommunicant, mais également à la finalité graphique des données. De plus, devant une contrainte réseau élevée en limitation et en variabilité de débit, une compression de l'information contenue dans les scènes, allée à un flux adaptable sur les réseaux, représente l'enjeu de cette thèse. Ces deux tâches se traduisent en la simplification, l'approximation et le codage échelonné des maillages constituant les objets d'une scène. A cette première réalisation s'ajoute la finalité graphique des données qui nécessite une prise en compte du point de vue d'un observateur virtuel sur le terminal, des parties cachées et de tout autre paramètre susceptible d'augmenter le ratio qualité perceptuelle / quantité de données.

Plus précisément, soit M un maillage formant le support géométrique d'un objet 3D. Il s'agit :

1. de simplifier et d'approximer la géométrie de M tout en conservant une bonne fidélité visuelle aux données originales afin de supprimer la géométrie non perceptuellement pertinente de l'objet (et abusivement générée lors de la phase d'acquisition ou de modélisation), et de coïncider avec le débit des réseaux ou les capacités des terminaux,
2. de réduire le volume de données décrivant le maillage M en supprimant les redondances contenues dans les formats de données usuels (la compression de l'information),
3. d'organiser le train binaire résultant de la compression de M en un flux adaptable sur les réseaux et les terminaux ; ce qui se traduit en une échelonnabilité de granularité fine et une organisation des données par ordre d'innovation² décroissante,
4. de développer sur le terminal tous les algorithmes susceptibles de raffiner la géométrie de M en fonction du point de vue d'un observateur virtuel, ou du moins de recréer l'illusion d'une géométrie complexe, participant au réalisme. Les silhouettes et les données proches de l'observateur doivent notamment faire l'objet d'un soin tout particulier étant donné leur forte contribution visuelle, et les singularités géométriques devront être conservées au cours du raffinement des surfaces.

²l'innovation est reliée à la notion de quantité d'information

1.3 Travaux précédents

Exploiter une scène complexe de réalité virtuelle dans un contexte client-serveur nécessite un large éventail de briques technologiques à développer : depuis la représentation des scènes, la compression, en passant par les protocoles de transmission, le décodage progressif, jusqu'à la visualisation interactive. En pratique le *moteur ultime* de réalité virtuelle n'existe pas encore, et ce mémoire contribue à cette recherche. Notre choix s'est donc naturellement orienté vers le développement de quatre briques technologiques clés, à savoir l'approximation, la simplification, le codage échelonné et la visualisation adaptative de maillages.

Même si le but final consiste à assembler judicieusement ces techniques, il est possible d'étudier scrupuleusement chacune d'elles indépendamment, en conservant un *œil à l'horizon* d'une future intégration dans un système complet.

Une étude bibliographique par chapitre

Plutôt qu'une longue étude bibliographique préalable, ce mémoire présente un état de l'art par chapitre, avec un premier panorama global des travaux existants pour chaque brique technologique :

- Simplification et approximation : ces algorithmes représentent souvent le prélude au codage, et surtout un moyen efficace de ramener le volume de données à une quantité coïncidant avec les débits d'un réseau ou les capacités d'un ordinateur graphique. Plus précisément, l'enjeu consiste à réduire le nombre de triangles des maillages tissant la surface des objets 3D. Intuitivement, il est naturel de commencer par enlever les éléments les moins significatifs afin de conserver la meilleure approximation perceptuelle de l'objet initial. Et puisque modifier les positions des sommets ou permuter les arêtes ne change pas le nombre de triangles, les auteurs s'autorisent à procéder ainsi lorsque cette opération améliore la qualité de l'approximation.

Nous venons de décrire en quelques mots la technique de la *décimation*, la plus répandue, qui consiste à retirer itérativement un ou plusieurs éléments (on parle alors de *décimation par fauchée*). Il existe aussi des techniques de simplification fonctionnant par échantillonnage ou par raffinement. Dans ce dernier cas, des éléments sont progressivement ajoutés à un maillage initial très grossier, aux endroits où la distorsion mesurée s'avère la plus élevée,

- Codage :
 - sans perte (ou bijectif) : il consiste à retirer toute la redondance d'un volume de données, tout en permettant une reconstruction parfaitement fidèle à l'original. Les acteurs du multimédia sont passés maîtres dans cet art, puisqu'il est une condition essentielle à leur développement. Les scènes tridimensionnelles n'étant après tout qu'un média de plus, les auteurs s'inspirent souvent des réflexes développés par les *chasseurs de bits* (codage différentiel, entropique, par plages, etc...), avec toutefois des méthodes spécifiques à la nature géométrique des données. En pratique le codage sans perte est souvent utilisé pour coder la connectivité³ des maillages, et l'on remarque principalement la famille des *éplucheurs*, qui découpent les maillages en bandes de triangles afin de coder sous une forme compacte l'information de marche en zigzag sur les sommets. On trouve aussi la famille des *tourneurs*, qui marchent autour des sommets plutôt qu'en zigzag. Chaque conférence du domaine voit une nouvelle série d'innovations permettant d'abaisser les coûts et les temps de codage,
 - avec pertes : les auteurs acceptent de supprimer une certaine quantité de données jugée insuffisamment pertinente au regard de la place qu'elle occupe. Cette technique est appliquée notamment sur

³La connectivité représente l'information de maillage des sommets pour former les faces, tandis que la géométrie représente les positions des sommets.

les positions, les normales et les couleurs des maillages puisque la précision initiale des données n'est pas toujours indispensable pour des données à finalité graphique,

- échelonnable : en théorie, toute partie tronquée d'un train binaire échelonnable permet de reconstruire une information exploitable. Il en résulte un format emboîté, capable de s'adapter aux variations de débits d'un réseau ou aux capacités d'un terminal. Naturellement, les données doivent être classées par ordre d'innovation décroissante. On trouve principalement deux familles de méthodes : l'analyse multirésolution et l'approche par décimation inverse. La première, appelée aussi technique par ondelettes, consiste à représenter un maillage sous la forme d'une succession de maillages emboîtés afin de stocker sous une forme compacte les raffinements d'un niveau à l'autre. La seconde consiste à exploiter une technique de décimation afin d'enregistrer à rebours la séquence de simplifications élémentaires, et permet ainsi de reconstruire un maillage avec une résolution définie au sommet près,
- Visualisation : elle représente l'ensemble des techniques développées afin d'améliorer au niveau du terminal, soit le rendu des objets 3D, soit la vitesse de rafraîchissement des images. Ces techniques concernent principalement deux familles d'algorithmes : la simplification ou le raffinement adaptatif, dépendant du point de vue d'une caméra virtuelle et éventuellement des sources d'éclairages. La simplification adaptative intervient lorsque l'ordinateur n'est pas assez puissant pour supporter une certaine complexité géométrique, ou lorsque les objets ne sont pas visuellement prioritaires : loin du point de vue, placés dans l'obscurité, ou masqués (même partiellement). L'amplification géométrique (ou raffinement) intervient lorsque l'ordinateur possède une marge de calcul largement suffisante pour traiter les données, ou lorsque les régions d'un objet sont finement observées. Ce raffinement est souvent combiné à un déplacement des positions des sommets afin de répondre à une contrainte différentielle sur la surface, notamment sur les silhouettes d'un objet qui constituent des régions d'attention visuelle. Les auteurs travaillent ainsi d'une part pour détecter rapidement les régions d'intérêt, et d'autre part pour raffiner continûment les maillages sur ces mêmes régions.

1.4 Positionnement des travaux

L'objectif de cette section consiste à répondre aux questions suivantes : pourquoi avoir choisi une méthode plutôt qu'une autre ? les résultats sont-ils significatifs ? où se situe la *valeur ajoutée* ? Cette section propose des esquisses de réponses, en sachant que l'objectif de ce mémoire consiste à argumenter en détail les algorithmes susceptibles de répondre aux problèmes rencontrés.

1.4.1 Simplification et approximation

Le choix d'une métrique

Le choix d'une métrique s'avère essentiel pour la phase d'approximation des maillages [CRS98]. Partant du constat qu'une métrique basée volume donne des résultats significatifs en approximation d'images [LSL97], celle-ci a donc été adaptée aux maillages 3D dans ce mémoire. Cela apparaît d'autant plus motivant que les auteurs du domaine proposaient de nombreuses mesures de distorsion (enveloppes de simplification [CVM⁺96], préservation de volume [LT98], métrique L_2 [Hop96], métrique L_∞ [ESV98], distance de Hausdorff [CKS98] [Kle98] [Bor99] [GK98], etc...) sans réellement utiliser la métrique basée volume afin de *guider le processus* d'optimisation des positions. L'adaptation aux maillages tridimensionnels s'est effectuée avec succès, et l'on constate notamment une prise en compte naturelle des singularités géométriques par la métrique, telles que les arêtes vives, les coins, etc... Cette caractéristique est très intéressante vis à vis d'une métrique L_2 , qui nécessite

notamment une détection préalable additionnelle des singularités, et la définition de seuils correspondants (voir [Hop96]). Enfin, le processus d'approximation constitue également un post-traitement bien adapté aux techniques de simplification fonctionnant suivant la métrique de Hausdorff (voir [Bor99]).

Combiner des critères

La géométrie est simplifiée par décimation puisque cette technique s'avère efficace pour conserver en permanence une référence aux données. Dans plusieurs méthodes de simplification, le processus de décimation est piloté par une queue de priorité combinant un [SZL92] [CKS98] ou plusieurs critères [GHJ⁺97] [Hop96] [GH97], en général reliés à une notion de distorsion géométrique. La phase de simplification peut être aussi précédée d'une étape de classification [AS96] [SZL92], afin d'interdire la simplification sur certaines régions. Partant du principe qu'il s'agit d'atteindre tous les niveaux de complexité géométrique, et qu'une pondération entre de nombreux critères sans rapports directs nécessite des réglages trop fastidieux des coefficients de pondération (courbure, compacité, valence) [GHJ⁺97], nous avons fait le choix d'une double queue de priorité combinant deux critères évoluant au cours du temps, dont l'un est prioritaire. Cette méthode permet d'éviter le réglage des pondérations d'une part, et d'accélérer sensiblement la vitesse de décimation d'autre part puisque seul un sous-ensemble des candidats est traité à chaque itération. Un simple réglage des priorités est aussi plus intuitif qu'une pondération. Il est ainsi possible de privilégier les faces ayant un bon rapport d'aspect plutôt que la préservation de la courbure, afin de faciliter l'entrelacement avec un processus d'optimisation basé volume.

1.4.2 Codage

Concernant le codage de la connectivité des maillages, le challenge est assez audacieux, puisqu'en 1997 les coûts étaient déjà proches de 1 bit par face pour des maillages très réguliers [THLR97], tandis que Touma et Gotsman abaissaient ce même plancher à 0.2 bit par face en 1998 [TG98]. Après quelques essais insatisfaisants en terme de coûts de codage à partir de la famille des algorithmes fonctionnant par bandes de triangles [THLR97] et tampons d'indices [Dee95] [BYG96], il a été choisi d'explorer la voie ouverte par Touma et Gotsman [TG98]. Le défi consiste alors à redistribuer l'énergie, c'est à dire à ne plus représenter la connectivité sous la forme de références (même indirectes) à des sommets, mais plutôt sous la forme d'une liste de valences, afin d'en exploiter la distribution. En pratique, nous avons trouvé le moyen d'améliorer de 35 % en moyenne les coûts de codage de la connectivité, et même beaucoup plus sur des maillages très réguliers. La robustesse et la vitesse de décodage ont été également sensiblement améliorées, et l'accent a été mis sur le coût mémoire de la structure de données nécessaire au décodage.

1.4.3 Echelonnabilité du codage

Le gain apporté par l'échelonnabilité est double : elle permet d'une part l'obtention d'un flux adaptable sur les réseaux et les terminaux, et d'autre part la reconstruction progressive des données sur le terminal. Il s'agirait d'obtenir idéalement une échelonnabilité sur tous les types de données. Les auteurs ont principalement concentré leurs efforts sur l'échelonnabilité en connectivité [TGH98] [Hop96] [PH97] [AdRL⁺99] [VTK⁺99], alors que l'échelonnabilité sur les positions est beaucoup plus rare [LK98a] [DG99b]. Les positions après compression occupant en moyenne dix fois plus de volume que la connectivité, nous avons fait le choix de concentrer cette fonctionnalité sur ce type de données. Les positions des sommets sont ainsi codées indépendamment de la connectivité et de façon échelonnable par quantification successive [LK98a] après prédiction multiple et représentation par plans de bits de taille variable. Il en résulte des coûts de codage réduits, ainsi que la possibilité de raffiner continûment les positions sur le terminal.

1.4.4 Visualisation

La visualisation concerne l'ensemble des techniques mises en œuvre pour assurer à chaque instant une qualité optimale des images obtenues à partir des données transmises sur le réseau. Nous avons choisi d'examiner les méthodes de reconstruction de surfaces par subdivision de surfaces. L'originalité provient notamment de la prise en compte du point de vue d'un observateur sur le terminal et du comportement adaptatif de l'algorithme.

Raffinement adaptatif

Lors de la conférence *Pacific Graphics 99*, Peter Schröder [Sch99] a présenté les opportunités offertes par les surfaces de subdivision [DKT98] [Loo87] [Kob99]. Les techniques de subdivision sont par essence rapides, robustes et efficaces : trois qualités cruciales pour un affichage dépendant du point de vue sur le terminal où on dispose seulement d'un fragment de seconde durant lequel la géométrie doit être mise à jour afin d'embellir globalement le rendu de la scène. La technique de subdivision élaborée dans ce mémoire met en œuvre un opérateur de raffinement hybride combinant subdivision barycentrique 1 vers 3 des faces et permutation d'arête. Cet opérateur est utilisé à l'origine par Kobbelt [Kob99] dans le but de définir un opérateur robuste 1 vers 9 par le biais de deux itérations successives, le raffinement étant opéré à l'origine sur les régions de forte courbure ou dans le but de minimiser une erreur d'approximation. Nous montrons dans ce mémoire que cet opérateur s'avère particulièrement bien adapté à un raffinement réversible localisé sur les régions d'intérêt et les silhouettes. De plus, une reconstruction de surfaces courbes par morceaux peut en être adaptée [HDD⁺94]. Enfin, une structure de données optimisée autorise un raffinement réversible, continu, rapide et robuste, ainsi qu'une phase de détection rapide des régions d'intérêt.

1.5 Organisation du document

L'organisation de ce document suit la *ligne de vie* d'une scène tridimensionnelle, et respecte ainsi l'ordre des opérations effectuées successivement dans une chaîne de création, de représentation, de transmission et d'affichage de scènes tridimensionnelles (figure 1.1).

Les étapes de design (ou modélisation), de structuration et de transmission ne sont pas traitées dans cette étude, tandis que les étapes de simplification et approximation, de codage et de visualisation font l'objet des chapitres principaux 2, 3, 4. Cet ordre correspond à l'ordre chronologique du travail effectivement réalisé au cours de la thèse, les méthodes d'approximation et de simplification pouvant être entrelacées. Une section de discussion conclut chaque chapitre afin de présenter les évolutions possibles et/ou nécessaires.

1.6 Préliminaires

Avant d'examiner le problème de la représentation des données contenues dans les scènes tridimensionnelles, convenons d'abord de formaliser la notion de scènes en réalité virtuelle et de leurs objets constitutifs : les maillages.

1.6.1 Les scènes en réalité virtuelle

Une scène est constituée entre autres d'objets tridimensionnels fixes ou animés, d'un ensemble d'effets liés à l'éclairage, d'une atmosphère, et des propriétés d'animation et d'interaction associées aux objets. On s'intéressera aux scènes statiques dans ce mémoire, et plus particulièrement aux objets les constituant puisqu'ils

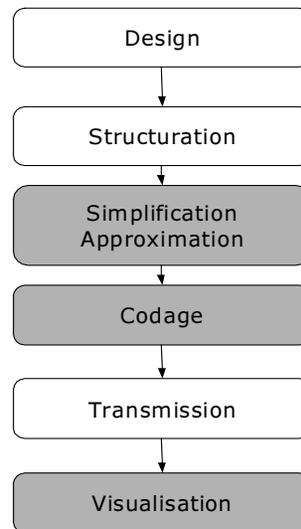


FIG. 1.1 : La chaîne de création (design), de représentation (structuration, simplification / approximation, codage), de transmission et de visualisation d'une scène tridimensionnelle distribuée. Le plan de ce mémoire suit le chemin emprunté par une scène, depuis le serveur jusqu'au terminal.

représentent la majorité des données. En pratique, les infographistes utilisent une forme de représentation bien adaptée à leur manipulation sur ordinateur, c'est à dire les *maillages*.

1.6.2 La représentation par maillages

Dans cette section, nous allons décrire la phase de traduction d'objets réels en objets virtuels manipulables par un ordinateur, leur synthèse par modélisation directe ou par acquisition, une définition plus formalisée des maillages et enfin leur rendu sur écran.

Du monde réel au graphisme sur ordinateur

Le monde est composé d'objets tridimensionnels, c'est à dire d'éléments volumiques. Si l'on essaie de représenter les objets sous une forme adaptée à l'ordinateur, on est rapidement confronté à des problèmes de surabondance et surtout de mauvaise exploitation des données. En effet, la plupart des objets sont opaques, l'immense majorité de leurs données est invisible car ces données sont masquées par les éléments de la surface. De plus, les données ayant une finalité graphique, il s'avère raisonnable de se limiter à la frontière géométrique des objets, qui caractérise la représentation surfacique. Ainsi un objet peut être vu comme une variété différentielle de dimension 2 plongée dans \mathbb{R}^3 ou comme un accolement de ces variétés.

En pratique, une variété différentielle trop abstraite pour une représentation informatique, est stockée sous une forme discrète. On approxime ainsi la surface idéale par un tissage (ou réseau) de mailles, appelé aussi *maillage surfacique*. La figure 1.2 montre que ce concept peut aboutir à un réalisme convaincant. Toutefois, nous verrons par la suite que la représentation tridimensionnelle, aussi détaillée soit-elle, n'est qu'un premier pas vers le réalisme.

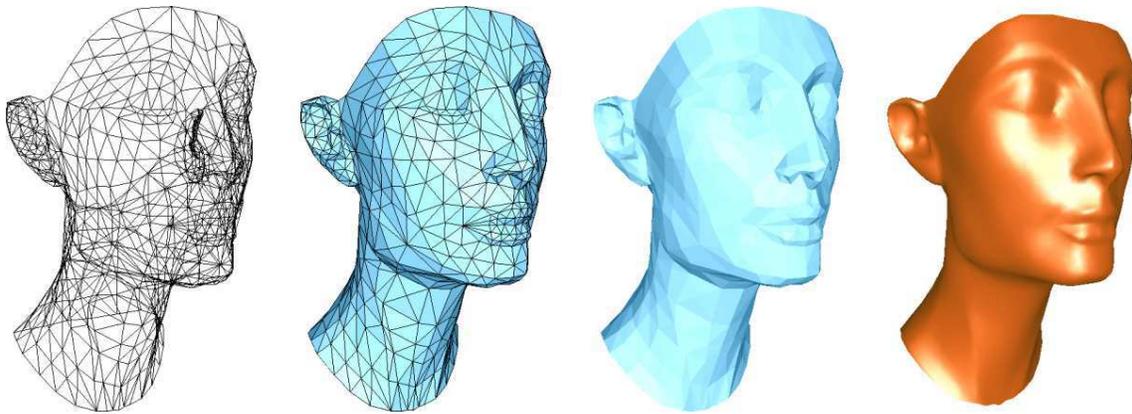


FIG. 1.2 : Le concept de réseau maillé, ou maillage surfacique. Les infographistes se limitent à la frontière géométrique des objets, et approximent la surface d'un objet par un réseau de mailles. De gauche à droite : nous visualisons les arêtes du réseau maillé, les faces et arêtes superposées, les faces seules et un maillage muni d'un matériau.

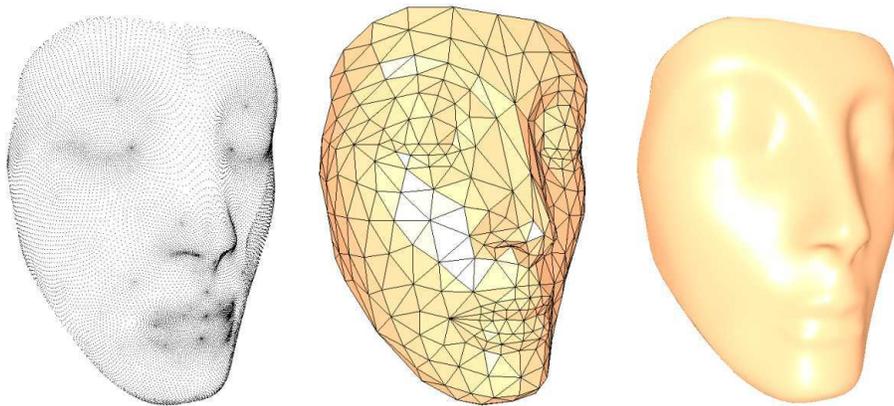


FIG. 1.3 : Principe de reconstruction de maillages à partir d'un nuage de points non organisés, le plus souvent acquis par scanner tridimensionnel à base de laser.

Acquisition et modélisation

La tâche du créateur de contenu consiste à générer des maillages à partir de données tridimensionnelles. Cette tâche est loin d'être aisée, même si le graphiste dispose d'outils de plus en plus puissants concernant les phases d'acquisition, de modélisation et d'édition (les outils de saisie, de visualisation et de modélisation fonctionnent encore majoritairement en 2D). Les données peuvent être soit synthétisées de toutes pièces, soit acquises par les différents dispositifs listés ci-dessous avant d'être éditées et transformées en maillages.

Scanner 3D

Des techniques d'acquisition par laser permettent de connaître les positions d'une multitude de points échantillonnés sur un objet. Il s'agit ensuite de convertir cet ensemble de points non organisés en un maillage [HDD⁺92] [SBD86], définissant ainsi des relations de connectivité entre les points (voir figure 1.3). L'échantillonnage étant souvent effectué par balayage régulier, il en résulte souvent une surabondance des données.

Imagerie radar

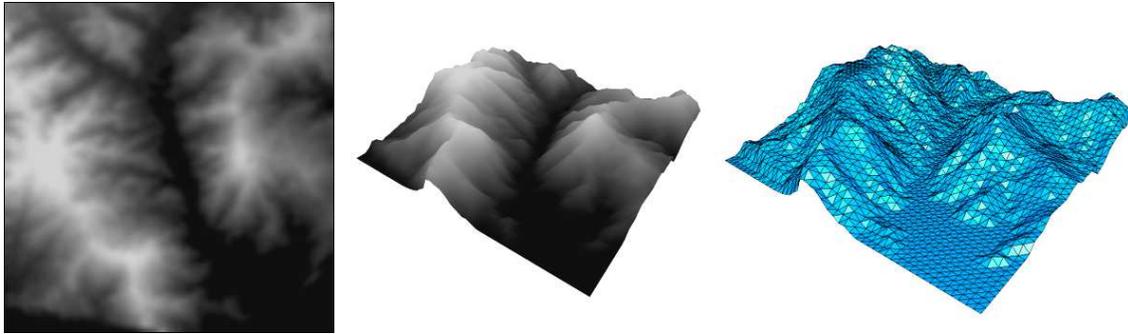


FIG. 1.4 : Les radars produisent des cartes d'altitude de terrains (à gauche et au milieu), à partir desquelles une triangulation est reconstruite (à droite).

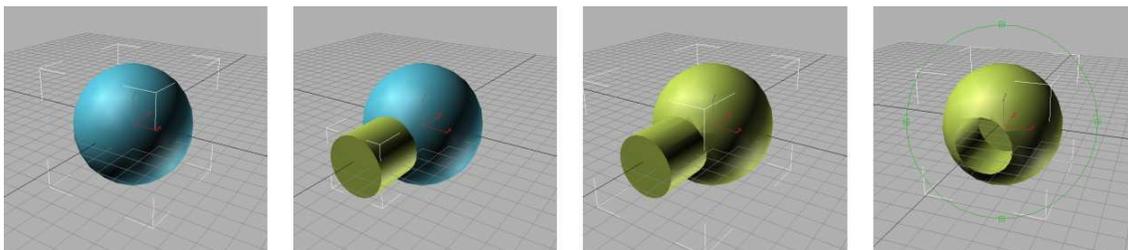


FIG. 1.5 : Des primitives géométriques paramétrables (rayon, longueurs, diamètres, etc...) sont combinées par des opérateurs booléens pour construire des modèles.

Les radars produisent des cartes d'altitude de terrains : on parle aussi de MNT (Modèles Numériques de Terrain). A partir de ces images, il est possible de générer une triangulation (régulière ou non) afin de permettre la simulation de navigation sur des données réelles (voir figure 1.4). Des photos satellites ou aériennes peuvent aussi habiller la surface afin de gagner en réalisme.

Vision par ordinateur

Les techniques de vision par ordinateur tendent à simuler la vision humaine avec la notion de perspective offerte par notre vision binoculaire. La géométrie épipolaire [Fau93] et les techniques de traitement d'images permettent ainsi de retrouver une partie de la géométrie d'un objet [Che92] à partir de plusieurs images acquises depuis différents points de vue ou depuis une séquence vidéo. On parle aussi de perception tridimensionnelle d'une scène à partir de la stéréoscopie et du mouvement.

Primitives graphiques

Des primitives graphiques élémentaires et paramétrables peuvent être directement manipulées par les outils de modélisation. Les concepteurs, architectes et graphistes sont familiers de ces techniques de modélisation, et combinent les primitives (voir figure 1.5) afin d'aboutir à un résultat semblable au modèle préalablement construit dans leur imaginaire. On peut également citer les surfaces de type NURBS (Non-Uniform Rational B-Splines [Lav99] [nur]) paramétrables par des points de contrôle et des coefficients de pondération par sommet. En CAO (Conception Assistée par Ordinateur), les propriétés différentielles de telles surfaces coïncident avec les contraintes géométriques lors de la fabrication.

Imagerie médicale

L'imagerie médicale produit des données volumiques (ou voxels) par des techniques d'imagerie par résonance

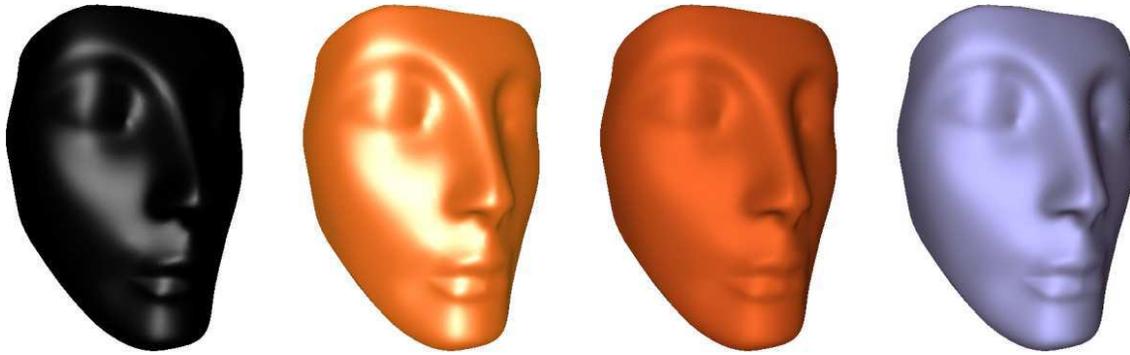


FIG. 1.6 : Différents matériaux : plastique noir, laiton, cuivre et étain. On remarque une interaction lumière/surface différente suivant le type de matériau défini.

magnétique (IRM), par tomographie d'émission, imagerie nucléaire, ultrasonore, etc... ces données sont ensuite converties en surfaces d'iso-valeurs [LC87] et produisent là encore une surabondance de données liée à une fine discrétisation des données source. Dans le cas général, les créateurs de contenu combinent les techniques d'acquisition entre elles (scanner et photos stéréoscopiques), ajoutent une séquence d'édition et éventuellement de traduction en une base de données interprétable par l'ordinateur.

Des attributs additionnels et des astuces

Afin de gagner en réalisme, les infographistes s'attaquent au problème de la lumière et des ombres sur les matières. Cette question est primordiale pour l'art pictural. L'infographiste remplace alors le coup de pinceau d'un artiste par le seul génie de la machine : la puissance de calcul. On ajoute ainsi sur les faces, les sommets ou les coins des attributs d'apparence, tels que :

- des couleurs,
- des normales : elles permettent de calculer l'interaction lumière/surface, et peuvent simuler visuellement la géométrie, on parle aussi de *bump mapping*,
- des matériaux : ils définissent le comportement photométrique de la surface, ou plus exactement les interactions surface/lumière, et nécessitent de ce fait la définition préalable des normales. On peut caractériser sommairement un matériau par quatre réflectivités (diffuse, spéculaire, ambiante, émission) et un terme de luminosité (voir figure 1.6),
- la transparence : plus rarement utilisée, elle traduit les coefficients de transmission de la lumière,
- et une coordonnée de texture : associée à un sommet et à une image de texture, elle permet d'*habiller* une face à l'aide d'une portion d'image. Cette technique améliore considérablement le réalisme d'un objet sans en raffiner la géométrie.

Malgré la définition d'attributs sur les maillages, le rendu direct de maillages nécessite une complexité géométrique très élevée afin de rendre indécélable la structure polygonale du modèle. Aussi les infographistes définissent des post-traitements lors du rendu afin d'améliorer l'aspect visuel. La méthode la plus répandue est l'interpolation linéaire de l'ombrage entre les sommets d'un triangle (voir figure 1.7). Ce traitement est couramment câblé sur les cartes des ordinateurs graphiques actuels.

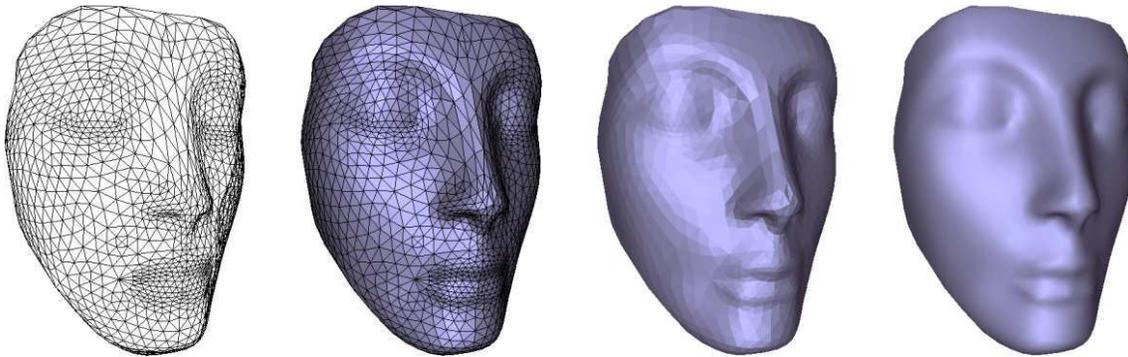


FIG. 1.7 : Un exemple du post-traitement le plus couramment utilisé : l'interpolation linéaire de l'ombrage ou lissage de Gouraud. De gauche à droite : le maillage original en mode ligne, en mode face, la version ombrée avec une normale définie par face, puis la version avec interpolation de l'ombrage de Gouraud avec une normale définie par sommet.

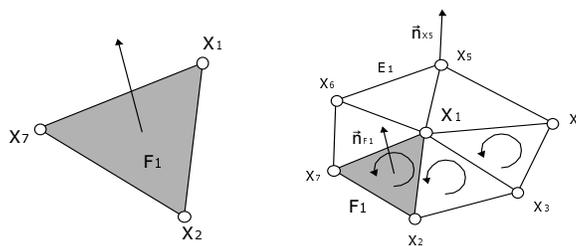


FIG. 1.8 : La géométrie et la connectivité définissent un maillage. Sur cet exemple la face triangulaire F_1 est définie par l'ensemble ordonné $\{X_1, X_7, X_2\}$, tandis que le sommet X_1 est défini par les coordonnées $\{x_1, y_1, z_1\}$. La normale \vec{n}_{F_1} associée à la face F_1 peut être soit déduite de la géométrie, soit définie explicitement par l'infographiste. Il en est de même pour la normale \vec{n}_{X_5} associée au sommet X_5 .

Les maillages surfaciques

Définition

Un maillage se réfère à deux modèles mathématiques : un graphe (connectivité des sommets) et une surface triangulaire par morceau (sous-variété de \mathbb{R}^3). Dans les formats graphiques usuels, un maillage est défini par un ensemble de sommets dans \mathbb{R}^3 , un ensemble de faces orientées et des attributs associés aux faces, aux sommets ou aux coins. Les faces décrivent la *connectivité* des sommets du maillage, tandis que les sommets en définissent la *géométrie*. Les arêtes sont directement déduites des faces et partagent au moins une face du maillage. Notons toutefois que les arêtes définissent aussi la géométrie puisqu'elles caractérisent notamment la convexité locale. Plus précisément, la géométrie est définie par un ensemble de N vecteurs tridimensionnels X_1, X_2, \dots, X_N représentant les coordonnées des positions des sommets. La connectivité décrit la manière dont sont *maillés* les sommets pour décrire des faces. Chaque face F_i s'appuie donc sur un ensemble ordonné de k_i sommets coplanaires, avec un minimum de trois éléments qui forment une face triangulaire. On définit ainsi une face F_i par un k_i -uplet de références à des sommets qui peuvent être exprimées sous la forme d'indices (voir figure 1.8). Les normales sont associées à un sommet, une face ou un coin (doublet face/sommet). Elles peuvent être soit déduites de la géométrie, soit directement lues dans un fichier si l'infographiste les a explicitement définies.

Un maillage peut être soit composé de polygones, soit purement triangulaire, et l'on peut affirmer qu'au-

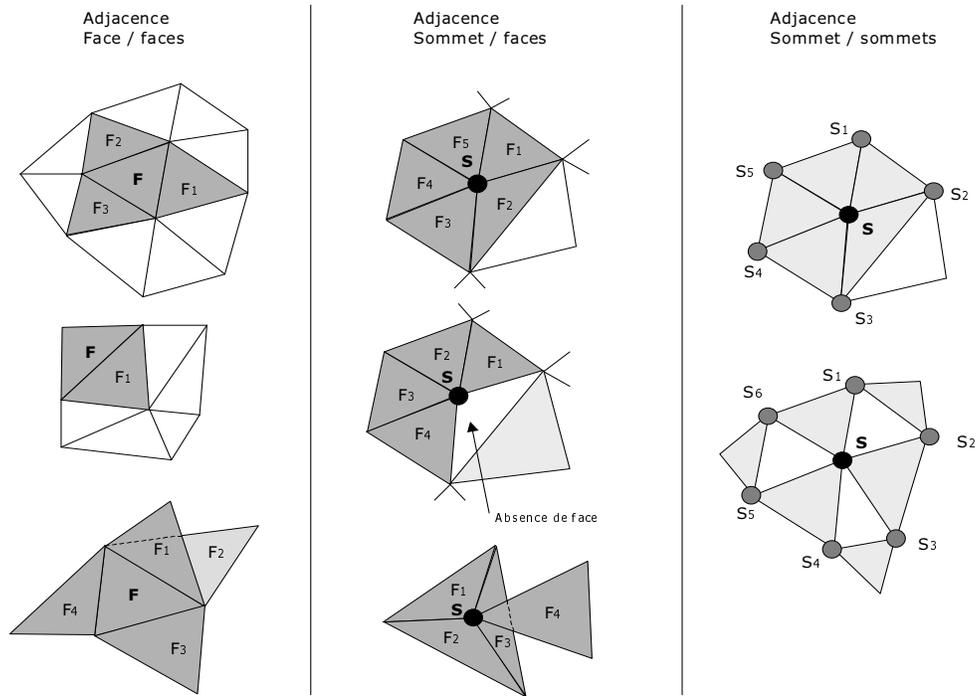


FIG. 1.9 : L'adjacence est définie entre les éléments de type sommet, face ou arête d'un maillage. Adjacence face / face : deux faces triangulaires sont adjacentes dès qu'elles partagent une arête. Adjacence sommet / faces : un sommet S connaît la liste des faces le contenant. Adjacence sommet / sommet : un sommet S connaît la liste des sommets partageant une arête. Le nombre de sommets voisins d'un sommet est appelé *valence* ou degré. Adjacence arête / face : une arête connaît la liste des faces qu'elle partage.

aujourd'hui, la suprématie du triangle est incontestable. Le triangle est un mode de représentation proche du matériel : la coplanarité est garantie, le volume de données est prédictible, l'interpolation linéaire entre trois sommets est directe, et sa simplicité permet d'élaborer des algorithmes très robustes. Enfin, la contrainte de coplanarité imposée sur les faces polygonales permet de convertir ces dernières en un groupe de triangles, comme cela est effectué en pratique au moment de l'affichage par les accélérateurs graphiques.

Adjacence

L'adjacence est une information additionnelle construite à partir de la connectivité du maillage. On parle aussi de *voisinage*. Elle intervient lors de la construction des normales à partir de la géométrie et surtout pour la manipulation algorithmique des maillages. Intuitivement, on peut dire qu'un élément E de type sommet, face ou arête est adjacent à un ensemble d'éléments S si celui-ci partage un ou plusieurs éléments avec S (voir figure 1.9). Il faut garder à l'esprit que la structure d'adjacence nécessite d'une part un espace mémoire additionnel et d'autre part un soin tout particulier pour la mise à jour d'une telle structure lors de la modification algorithmique des modèles. On essaie dans le cas général de ne construire que la structure d'adjacence strictement nécessaire et minimale afin d'optimiser la vitesse et la robustesse des algorithmes. En pratique, même s'il existe plusieurs niveaux d'adjacence pour les maillages triangulaires, notre implémentation en utilise une version réduite pour les raisons précédemment évoquées. Le chapitre 4 explicite notamment le choix d'une structure de données allégée des arêtes et de l'adjacence associée.

Topologie et invariants

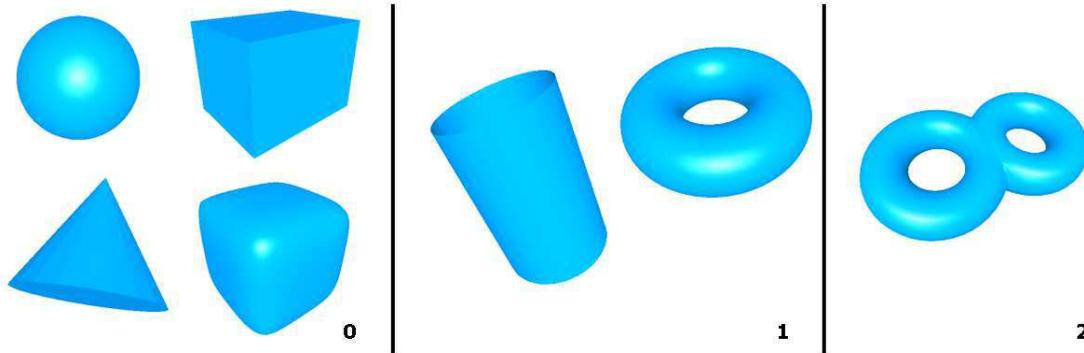


FIG. 1.10 : Le genre d'une surface correspond au nombre maximal de coupures applicables à une surface tout en la conservant connexe.

La topologie est la partie des mathématiques qui étudie la notion de continuité et de limite, et permet de classifier les espaces en fonction de leur forme globale, cette dernière devant être conservée au cours de la simplification ou du codage avec pertes de maillages. Par définition, les espaces d'une même classe peuvent se déduire les uns des autres par déformation continue, sans découpage ni déchirure. Dans le cas des surfaces, le plan est de topologie différente de la sphère (et toute surface homéomorphe à celle-ci) puisqu'aucune déformation continue ne peut le transformer en une sphère.

On appelle *genre* d'une surface le nombre maximal de coupures successivement applicables tout en conservant une surface connexe. Ainsi le genre d'une sphère et de tout objet homéomorphe à celle-ci est égal à zéro, le genre d'un tore vaut un (voir figure 1.10).

Citons aussi la caractéristique d'Euler-Poincaré utilisée pour classifier les surfaces simples : $c = F + S - A$, F , S et A étant respectivement le nombre de faces, de sommets et d'arêtes. Tous les polyèdres convexes et d'une façon générale les objets homéomorphes à une sphère vérifient cette relation avec $c = 2$.

Singularités géométriques

Les arêtes vives régulières, les coins, et les bords d'un maillage appartiennent au groupe des singularités géométriques (figure 1.11). La section 2.4 traitera d'algorithmes gérant les particularités représentatives de ces zones. En effet, on remarque une sensibilité particulière de la vision humaine aux singularités. Enfin, la courbure mesurée sur la surface peut également être prise en compte.

1.6.3 Transmission et rendu sur écran

Nous venons de définir la notion de maillages à des fins de transmission et de rendu sur un périphérique graphique. Suivons maintenant un objet tridimensionnel au cœur de la chaîne de transmission et de rendu afin de mieux évaluer les enjeux des méthodes mises en œuvre au cours de ce mémoire. Cette incursion nous permettra de mieux comprendre où se situent les goulets d'étranglement, et de justifier la stratégie choisie. Si l'objectif final est constant (il s'agit d'afficher le plus rapidement et avec la meilleure qualité possible une scène tout en encombrant au minimum les réseaux et les ordinateurs), le réalisme des scènes (fortement relié à la quantité de données) augmente en permanence puisque les infographistes disposent d'outils de modélisation de plus en plus puissants et de volumes de stockage croissants.

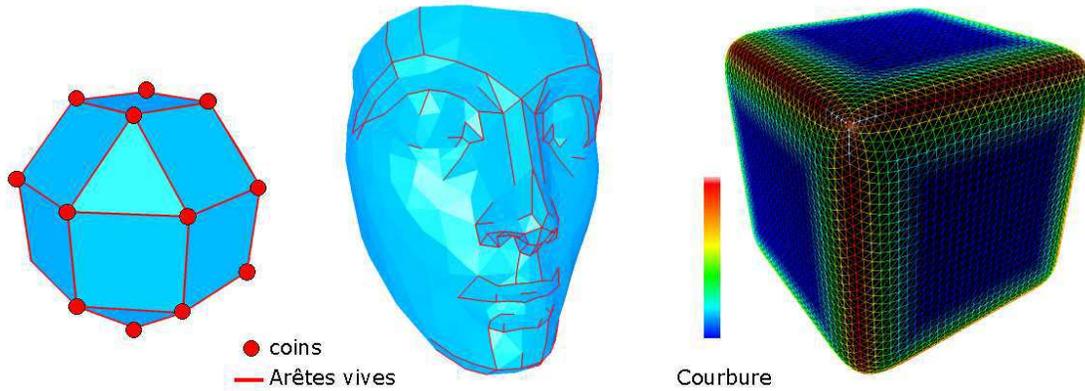


FIG. 1.11 : Les coins et les arêtes vives appartiennent au groupe des singularités, visuellement pertinentes. La courbure fait également partie des critères à prendre en compte lors du développement des algorithmes.

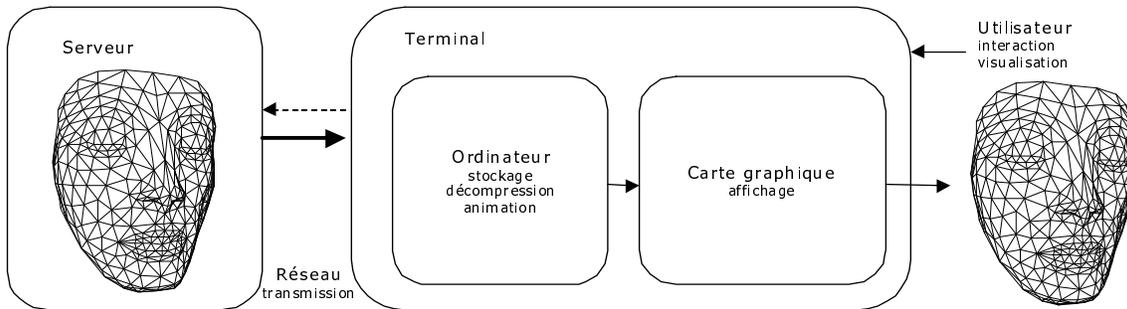


FIG. 1.12 : La chaîne de transmission et de rendu d'une scène 3D.

La figure 1.12 illustre sommairement la chaîne de transmission et de rendu, qui comprend en fin de chaîne un utilisateur susceptible d'interagir avec les scènes. Un seul terminal a été représenté sur cet exemple, mais les conditions réelles d'exploitation concernent un ensemble d'utilisateurs connectés à partir de terminaux et de réseaux divers et variés.

Le réseau

On commence à localiser d'ores et déjà le principal goulet d'étranglement, représenté par le réseau. En effet, son débit est variable, limité, et considérablement plus lent qu'un bus local de transfert de données sur le terminal. Par ailleurs son rôle est spécialisé à la transmission des données, et on suppose que cette dernière s'effectue sans perte. Les grandeurs théoriques des bandes passantes sont de l'ordre de 10 à 100 Mbits/s pour un réseau interne et de 50 kbits/s pour un réseau externe (modem). Notre rôle consiste donc à réduire le volume de données occupé par un objet 3D et à construire un flux binaire adaptable sur les différents réseaux : il s'agit donc bien de *compression* et de *représentation* des données.

Le terminal

Les données sont acheminées sur le terminal, il s'agit alors de décompresser, de stocker et de régénérer une scène tridimensionnelle compréhensible par le moteur de réalité virtuelle, avant d'en assurer l'animation et l'interaction avec l'utilisateur. L'objectif consiste alors à représenter les données sous une forme limitant les échanges avec la carte graphique, et à optimiser la vitesse de mise à jour des paramètres d'animation et d'interaction avec la scène.

La carte graphique

Les données convergent vers la carte graphique par l'intermédiaire d'une interface logicielle (ou API [ope] [SA98] [McR98]), dans un format proche de la structure du maillage décrite en section 1.6.2. La carte est chargée d'assurer une très grande part des calculs nécessaires au rendu d'un objet 3D. Précisons que la carte possède une mémoire propre afin de stocker localement des données géométriques et des textures, et que la plupart des algorithmes de base sont aujourd'hui câblés sur les accélérateurs de dernière génération (z-buffer, culling, transform, lighting, Gouraud, etc...). Dans la version théorique (naïve) d'un algorithme de rendu, et pour chaque position de caméra ou pour chaque modification de la scène, une image est entièrement produite à partir de la totalité des données décrivant la scène. L'accélérateur balaie alors l'intégralité des triangles pour le rendu d'une image ; cela explique pourquoi les concepteurs expriment les capacités d'une carte graphique en millions de polygones par seconde, ces derniers étant colorés, éclairés, et texturés. Malgré la puissance de tels calculateurs, la complexité des scènes générées peut alors dépasser les capacités de traitement pour assurer une cadence interactive de l'affichage, et l'alimentation même de la carte en données devient problématique.

En pratique tous les triangles d'un objet 3D n'ont pas la même contribution au rendu une fois projetés à l'écran, certains sont en dehors du champ de vision, d'autres tournent le dos à la caméra et ne sont pas affichés par convention avec les infographistes. En résumé, la problématique concerne une *alimentation optimisée* de la carte graphique de sorte qu'un maximum de données injectées dans le pipeline de rendu soient visuellement pertinentes. Bien entendu, nous verrons que tous les artifices sont utilisés pour donner l'illusion de la présence d'une géométrie complexe, synonyme de réalisme.

Maintenant munis d'une vision plus claire de la problématique, nous pouvons enchaîner directement sur le prochain chapitre : la simplification et l'approximation de maillages.

Chapitre 2

Simplification et approximation

Ce chapitre concerne les méthodes de simplification et d'approximation de maillages triangulaires. La section 2.1 introduit le domaine concerné, et une revue des principaux travaux est proposée dans la section 2.2. Afin d'améliorer les performances des techniques actuelles ou de remédier à certaines déficiences, la section 2.3 présente une méthode d'approximation pouvant être appliquée alternativement ou successivement à une méthode générique de simplification, suivie par une méthode de simplification fonctionnant par décimation dans la section 2.4. On termine alors par une discussion des algorithmes et une présentation des perspectives d'évolution dans la section 2.5.

2.1 Introduction

La problématique principale de ce mémoire concerne la représentation de maillages triangulaires à des fins de transmission et de rendu. Or, comme décrit dans le chapitre introductif, les coûts de stockage et de rendu s'expriment en nombre de triangles, ou nombre de polygones par seconde pour un accélérateur graphique. Il convient donc d'évaluer soigneusement la pertinence visuelle d'une donnée avant de la transmettre sur un réseau, puis sur un bus graphique. De plus, la simplification est une phase préalable naturelle à l'encodage des données. Ceci nous conduit naturellement à regarder de plus près les algorithmes de *simplification* et d'*approximation* de maillages. Il faut toutefois distinguer deux objectifs lors de la simplification : retirer la redondance géométrique, ou préparer un format progressif en simplifiant le maillage jusqu'à l'obtention d'un triangle unique.

Définition

La simplification d'un maillage M consiste à trouver un maillage \hat{M} de plus faible complexité géométrique et conservant une distorsion visuelle minimale avec le maillage original M . On peut donc découper ce problème en deux phases successives ou alternatives : l'optimisation et la simplification géométrique.

Le sujet est brûlant depuis que les ordinateurs graphiques sont capables d'afficher les scènes tridimensionnelles à une cadence interactive. Nous verrons dans la section 2.2 les principales innovations des acteurs du domaine, exploitant les nombreux degrés de liberté offerts par les phases de simplification et d'optimisation.

2.2 Travaux précédents

Les acteurs de l'infographie ont rapidement compris que le degré de complexité géométrique peut faire la différence entre un affichage lent et saccadé, et un rafraîchissement *temps réel* des images synthétiques au cours de la navigation dans une scène tridimensionnelle. Par ailleurs, les opérateurs de télécommunications ont constaté qu'un délai d'attente trop long entre une requête sur le réseau et l'affichage d'une première image constitue assurément un frein au développement du multimédia. La simplification de maillages tridimensionnels est ainsi rapidement devenue incontournable pour propulser la réalité virtuelle hors des laboratoires de recherche et des entreprises de création de contenus.

Un problème pour plusieurs applications

En pratique le problème de la simplification de maillages intervient dans tous les domaines où la surabondance des données maillées représente un inconvénient majeur. Citons quelques exemples :

- la visualisation : le coût de calcul d'une image augmente sensiblement en fonction du nombre d'éléments constituant une scène,
- les éléments finis : la durée des calculs augmente avec la complexité des maillages, il s'agit donc de simplifier la géométrie tout en conservant une précision numérique satisfaisante,
- l'imagerie médicale : les surfaces sont construites à partir d'éléments volumiques élémentaires (les voxels), introduisant une redondance élevée,
- la cartographie : les maillages sont générés à partir de modèles numériques de terrain, ou de cartes d'altitude régulièrement échantillonnées avec une précision très élevée.

La notion de *surabondance* ou de redondance s'avère souvent fortement liée aux applications. En ce qui concerne les applications liées à réalité virtuelle à finalité graphique, il semble raisonnable de penser qu'un élément redondant ne contribue en rien à l'amélioration du rendu d'une image de synthèse. Or cette notion dépend de nombreux paramètres subjectifs tels que le point de vue, la distance d'observation, l'éclairage de la scène, etc... De tels paramètres sont impossibles à prendre en compte lors de l'encodage réalisé hors ligne, aussi nous relierons la notion de redondance à une mesure purement géométrique indépendante du point de vue. Nous verrons qu'il en est toutefois autrement lors de la visualisation décrite dans le chapitre 4.

La revue suivante concerne seulement les algorithmes traitant des réseaux maillés tridimensionnels, toutefois il convient de mentionner les solutions alternatives :

- l'approche vidéo : on ne transmet pas la géométrie à un terminal pour des calculs de rendu. Un serveur de forte capacité de stockage et de rendu stocke la scène, en assure le rendu, puis transmet l'image résultante après compression. Seul un flux vidéo transite alors sur le réseau, et le rendu est direct. Aucune scène n'étant transmise, chaque nouveau point de vue nécessite la transmission d'une image, entraînant un encombrement rapide des réseaux.
- l'approche hybride : le fond de la scène est rendu sur le serveur, transmis et réutilisé par déformation durant plusieurs images sur le terminal. Seuls les objets au premier plan de la scène voient leur géométrie transmise et rendue par des calculs.
- le rendu basé images : les positions de la caméra virtuelle d'un observateur sont contraintes à un ensemble de chemins prédéfinis, et l'image de synthèse finale dépend d'une combinaison d'images précalculées sur le serveur et transmises au cours de la navigation. Seul un flux d'images est alors transmis sur le réseau.

Ces approches sont séduisantes par certains aspects, toutefois la transmission de la géométrie présente des avantages décisifs : une fois transmise, l'observateur peut interagir indéfiniment avec les objets d'une scène 3D

et changer à sa guise un point de vue.

Il est assez délicat d'établir une taxinomie pour les méthodes de simplification, le but de cette section n'étant pas de rédiger un état de l'art de plus [Eri96] [KGBT96] [HG97] [Lue97] [HRH⁺97] [KBGT98]. Les méthodes sont principalement classées par leur relation aux données originales :

- la décimation : on s'appuie en permanence sur les données initiales, puisque la décimation consiste à supprimer itérativement un ou plusieurs éléments du maillage afin d'atteindre une certaine complexité géométrique visée, ou une complexité géométrique minimale pour une erreur tolérée maximale,
- le raffinement : il représente l'opération inverse de la décimation. On commence par construire un modèle de base, puis les régions de plus grande distorsion géométrique sont itérativement raffinées jusqu'à l'obtention de la complexité géométrique souhaitée ou d'une erreur de distorsion maximale autorisée. L'intérêt provient du type de raffinement, avantageux dans un but de codage ou d'allègement de la structure de données. En contrepartie le problème consiste à construire ce niveau de base (c'est encore un problème de simplification),
- et l'échantillonnage : seule une partie des données originales nous intéresse. Un ensemble d'échantillons est prélevé sur la surface (régulièrement ou aléatoirement), puis une phase optionnelle d'optimisation est appliquée afin d'approximer au mieux la surface originale.

Commençons par décrire le principe de la décimation, le plus couramment utilisé.

2.2.1 Décimation

La décimation s'appuie en permanence sur les données à chaque itération, et constitue de ce fait une technique très efficace pour la simplification. Elle peut s'opérer sur les sommets, les arêtes ou les faces par le biais d'opérateurs élémentaires de simplification. On essaie de trouver les opérations les plus simples possibles afin de limiter les degrés de liberté résultants (coûteux en calculs), ces mêmes degrés de liberté autorisant par ailleurs des qualités d'approximation satisfaisantes. Enumérons maintenant les opérations les plus utilisées.

Les opérations atomiques

On définit principalement quatre opérations élémentaires de décimation sur les maillages triangulaires :

1. la suppression de sommet : un sommet est supprimé, ainsi que l'ensemble de ses arêtes et des ses faces adjacentes, ce qui génère un trou polygonal devant être alors retriangulé (figure 2.1) [SZL92]. A chaque opération élémentaire un ensemble de n triangles est ainsi remplacé par $n - 2$ triangles, dont l'arrangement conditionne en partie la qualité de l'approximation,
2. la fusion d'arête : les deux sommets d'une arête sont fusionnés, entraînant la suppression d'un sommet et d'au plus deux faces adjacentes à l'arête considérée (figure 2.2). Le degré de liberté résultant concerne la position du sommet après contraction [AS96] [LT98] [Hop96] [Bor99]. Sa version généralisée peut fusionner deux sommets proches, sans qu'ils soient reliés par une arête [PH97] [EM99] [GH97] [GH98]. Un tel opérateur permet ainsi de connecter des parties disjointes du maillage original au cours de la simplification. Enfin, on peut étendre l'opérateur à une forêt de fusions d'arêtes sur le maillage [TGHL98] [Tau98] [GTLH99],
3. la demi-fusion d'arête : elle représente un cas particulier de la fusion d'arête, puisque la position du sommet résultant de la contraction est contrainte à l'un ou l'autre des sommets initiaux (figure 2.3). Ainsi les sommets du maillage après simplification sont un sous-ensemble des sommets du maillage initial [KCS98] [CKS98],

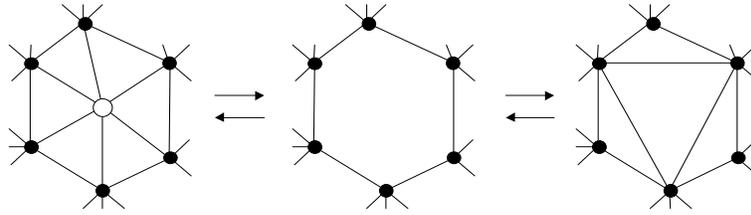


FIG. 2.1 : La suppression d'un sommet entraîne la suppression de ses faces adjacentes, et nécessite de ce fait la retriangulation du trou résultant. A chaque opération élémentaire, un ensemble de n triangles est ainsi remplacé par $n - 2$ triangles, dont l'arrangement conditionne en partie la qualité de l'approximation.

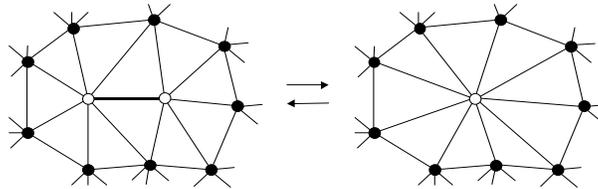


FIG. 2.2 : La fusion d'arête agglomère deux sommets et supprime de ce fait un ou deux triangles adjacent(s) à l'arête concernée. Une phase d'optimisation locale ou une heuristique détermine la position optimale du sommet résultant de la fusion.

4. la fusion de faces : les trois sommets d'une face triangulaire sont fusionnés en un seul, supprimant de ce fait deux sommets, et 4 faces dans le cas général (moins sur les bords du maillage) (figure 2.4) [GHJ⁺97].

Il n'existe pas d'*opérateur parfait*, puisque chacun possède des atouts et des inconvénients :

- la suppression d'un sommet nécessite une retriangulation coûteuse, mais source d'optimisation. Elle fournit le moyen de définir la résolution d'un maillage au sommet près et ne déplace pas les sommets initiaux,
- la fusion d'arête représente un cas particulier de la suppression de sommet et ne nécessite pas de retriangulation. Le positionnement du sommet résultant de la fusion peut être perçu soit positivement dans un but d'optimisation en position, soit négativement dans une perspective de codage ou d'optimisation en

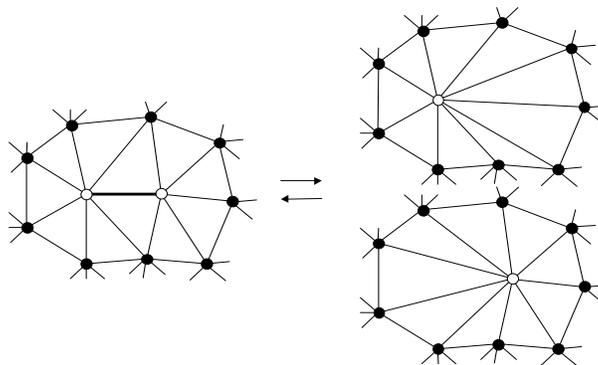


FIG. 2.3 : La demi-fusion d'arête est un cas particulier de la fusion d'arête puisque l'un des sommets est choisi comme accumulateur, tandis que les faces dégénérées par l'opération sont supprimées. Cette opération réduit le degré de liberté offert par la fusion d'arête, c'est à dire la conservation de la position du sommet.

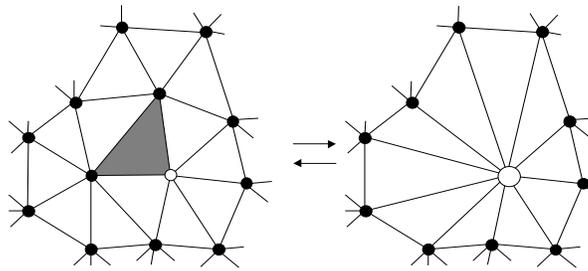


FIG. 2.4 : La fusion de face supprime une face en agglomérant deux de ses sommets en un sommet, appelé accumulateur. Deux sommets et au plus quatre faces sont alors supprimées au cours de l'opération. Il en résulte un degré de liberté important : la position du sommet accumulateur, ou le choix de ce dernier si l'on contraint les sommets à conserver leur position initiale.

vitesse,

- la demi-fusion d'arête limite le degré de liberté offert par la simple fusion d'arête et simplifie les coûts de codage en position,
- la fusion de face autorise une simplification rapide puisqu'elle supprime au plus quatre faces à chaque itération, et ne nécessite pas de structure de données basée arêtes.

D'une façon générale, un ou plusieurs éléments sont ainsi décimés en fonction d'un critère de distorsion géométrique. Soit l'approche est basée seuil et tous les éléments dont la suppression entraînent une distorsion en deçà du seuil sont supprimés en une *fauchée*, soit l'élément entraînant la distorsion minimale est supprimé en priorité. On peut définir également des contraintes ou des comportements spécifiques sur les bords ou les singularités géométriques, et une phase d'optimisation géométrique peut être appliquée dans le but d'approximer au mieux soit le maillage original, soit le maillage juste avant décimation. D'une façon générale les diverses méthodes d'optimisation minimisent la distorsion géométrique. Les ingrédients d'un algorithme de simplification par décimation se résument en trois étapes : l'opérateur de décimation, la métrique reliée à la notion de distorsion, et le critère de validité [KCS98]. En pratique, même s'il est assez aisé de définir ces ingrédients, la *recette* aboutissant à la méthode idéale de décimation n'existe pas. Avant de présenter et de justifier nos choix, retournons quelques années en arrière afin d'avoir une vision plus nette du panorama historique.

Classification, suppression de sommets et distance au plan moyen

Schröder et al [SZL92] ont proposé en 1992 une méthode de décimation bien adaptée aux triangulations générées par l'algorithme des *marching cubes* [LC87]. Les sommets sont d'abord classés en 5 catégories (simple, complexe, bordure, arête intérieure et coin), puis un comportement spécifique est défini pour chaque classe. Les sommets simples sont décimés par *fauchée* si leur distance au plan moyen défini par ses sommets adjacents n'excède pas un certain seuil. Le trou résultant de la décimation est retriangulé suivant un critère relié à la compacité des triangles. Un comportement spécifique est ensuite défini sur les sommets de type bordure, arête ou coin, et les sommets complexes ne sont pas décimés. Cette approche est rapide, toutefois la mesure de la distorsion ne fait pas référence au maillage initial.

Partitionnement, fusion d'arêtes, coplanarité des faces et colinéarité des arêtes

Algorri et Schmitt [Alg95] [AS96] ont développé à partir de 1995 un algorithme adapté aux triangulations générées par reconstruction de surfaces. Le maillage est tout d'abord partitionné suivant un critère de coplanarité des facettes, puis une phase de lissage par permutation d'arêtes est appliquée. Les sommets sont ensuite classifiés en fonction de leurs arêtes vives incidentes, et seules les arêtes joignant deux sommets situés sur une

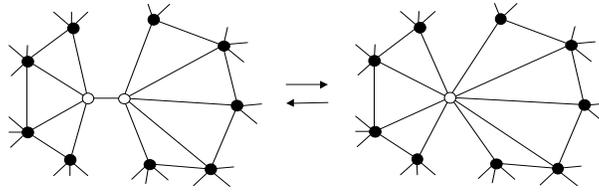


FIG. 2.5 : La fusion d'arêtes généralisée agglomère deux sommets séparés par une distance inférieure à un seuil prédéfini. L'intérêt d'un tel opérateur réside en sa capacité à simplifier par fusion des régions non connexes sur le maillage.

région plane sont fusionnées. Une phase de fusion des arêtes colinéaires est ensuite appliquée pour simplifier le maillage le long des arêtes vives régulières.

Fusion d'arête

Suite à ses travaux en reconstruction de surfaces [Hop93] [Hop94], Hoppe a introduit en 1996 un format progressif [Hop96] [Hop98] en exploitant l'opérateur réversible de fusion d'arête, définissant ainsi un maillage sous la forme d'une séquence de raffinements. La méthode fonctionne avec une queue de priorité, et optimise la position d'un sommet en fonction d'une distance L_2 à la surface originale et d'un terme de régularisation assimilé à un ressort de longueur nulle attaché aux arêtes du sommet résultant de la fusion. Afin d'éviter le lissage des arêtes vives par l'effet de la métrique et du terme de ressort, l'algorithme développe une approximation sommet / courbe entre un sommet appartenant au groupe des singularités et l'ensemble des arêtes vives du maillage original. Cet algorithme prend aussi en compte les attributs d'apparence, le point de vue de l'observateur [Hop97], et a été étendu à la fusion d'arête généralisée avec Popovic [PH97]. Il en résulte une méthode donnant de bons résultats, au détriment des coûts de calculs et d'une complexité algorithmique conséquente. Enfin, les phases de détection des singularités et d'optimisation nécessitent une expertise certaine pour le réglage des paramètres.

Garland et Heckbert [GH97] ont proposé en 1997 une métrique originale de simplification combinée à l'opérateur de fusion d'arête généralisée (figure 2.5).

Afin de sélectionner une paire de sommets à fusionner, on associe à cette opération élémentaire la notion de coût. Une matrice 4×4 symétrique Q est ainsi définie pour chaque sommet, et l'erreur associée au sommet $v = [v_x v_y v_z 1]^T$ s'exprime sous la forme quadratique $\delta(v) = v^T Q v$ (en pratique la surface d'erreur de niveau $\delta(v) = \varepsilon$ est une quadrique). Le coût d'une contraction élémentaire $(v_1, v_2) \rightarrow \bar{v}$ s'exprime ensuite sous la forme d'une nouvelle matrice $\bar{Q} = Q_1 + Q_2$, cette dernière étant associée au sommet résultant de la fusion. Les sommets accumulent ainsi les erreurs au cours de la simplification. L'algorithme se résume sous la forme de la séquence suivante :

1. Calculer les matrices Q pour chaque sommet,
2. Lister toutes les paires à contracter,
3. Calculer la position optimale \bar{v} après contraction de chaque paire (v_i, v_j) ,
4. Calculer le coût $\bar{v}^T (Q_1 + Q_2) \bar{v}$ résultant de la contraction de chaque paire,
5. Ordonner les paires dans une queue de priorité triée par ordre de coût croissant,
6. Retirer itérativement la paire (v_i, v_j) de moindre coût de la pile, la contracter et mettre à jour dans la pile les coûts des paires affectées par cette contraction.

Cette méthode s'avère efficace, générale, rapide et robuste et des améliorations ont été apportées en 1998 par les mêmes auteurs [GH98] afin de prendre en compte les bords, les attributs d'apparence (normales, couleurs)

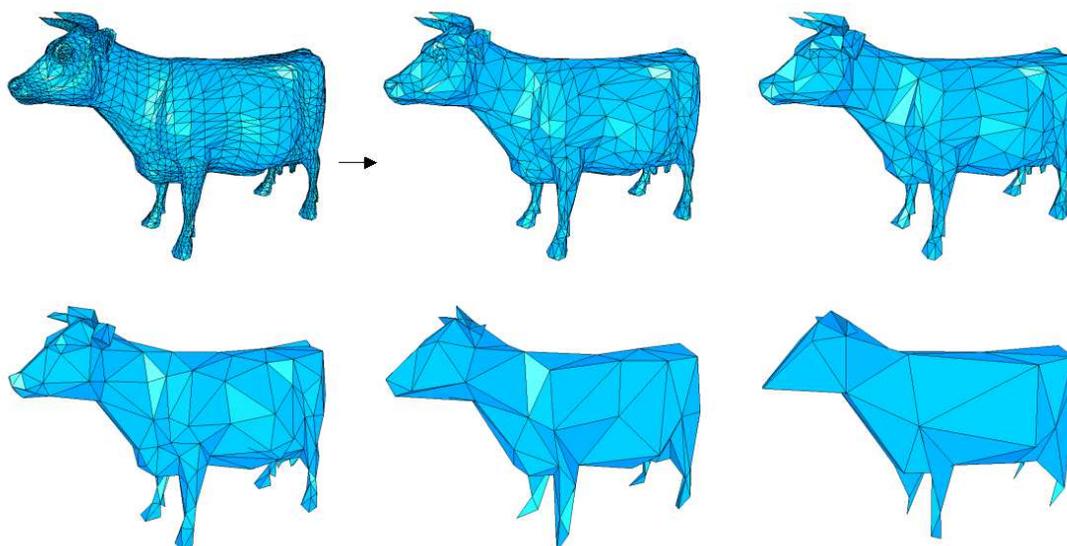


FIG. 2.6 : Simplification de Garland-Eckbert appliquée au maillage *cow*. Respectivement de gauche à droite et de haut en bas : le maillage original (5804 faces) puis les versions géométriquement simplifiées à 2000, 1000, 500, 200 et 100 faces. L'opérateur de fusion d'arête généralisé autorise la fusion de régions disjointes du maillage.

et les coordonnées de texture (figure 2.6). Les auteurs ont également constaté qu'un placement sous-optimal par demi-fusion d'arête s'avère très efficace et économe en mémoire. En 1999, et suite aux travaux de Garland et Heckbert [GH97] [GH98], Hoppe [Hop99a] a encore amélioré la métrique quadrique intégrant sous une forme unifiée la géométrie et les attributs d'apparence. Les principales améliorations présentées par Hoppe concernent notamment l'absence d'accumulation des coûts au cours de la simplification et la préservation du volume de Lindstrom et Turk [LT98]. Cette technique présente aujourd'hui un très bon ratio qualité / coût calcul pour des maillages dotés d'attributs.

Enveloppes de simplification et volume de tolérance

Cohen et al [CVM⁺96] définissent deux enveloppes, intérieure et extérieure, et contraignent le maillage à l'intérieur de ces enveloppes au cours de la décimation. La déviation géométrique en terme de volume est ainsi bornée. L'algorithme construit initialement deux surfaces en élevant de manière adaptative chaque triangle suivant deux directions normales de sens opposés. Cette opération produit ainsi des prismes fondamentaux dont l'offset est défini de telle sorte qu'aucune auto-intersection n'apparaisse. L'approche fonctionne ensuite soit globalement par décimation d'ensembles connexes de triangles et retriangulation du trou résultant, soit localement par décimation de sommets tant que les opérations élémentaires n'entraînent pas un débordement hors des frontières fixées par les enveloppes.

Guézic [Gué97] définit une erreur tolérée paramétrable sur chaque sommet. Un volume d'erreur en est ensuite déduit par interpolation linéaire entre les sommets. Le calcul du volume s'effectue par l'union des boules de rayon variable interpolées entre les sommets. Le maillage est contraint à l'intérieur de ce volume d'erreur au cours de la simplification, bornant ainsi la déviation géométrique.

Distance de Hausdorff

La distance de Hausdorff est couramment utilisée pour mesurer la distorsion au cours de la décimation [CKS98] [Kle98]. Sa version unilatérale (non-symétrique) présente quelques défauts majeurs, toutefois Gumhold et Klein [GK98] la modifient pour la rendre symétrique dans un contexte de multirésolution.

Une récente publication de Borouchaki [Bor99] présente une technique de simplification des maillages surfaciques combinant la notion d'enveloppe de simplification par le biais de la distance de Hausdorff et un contrôle très efficace de la qualité en forme des éléments. La simplification opère par décimation avec l'opérateur de fusion d'arête, couplé avec une optimisation de la forme des triangles par permutation d'arête et déplacement des sommets. Selon l'auteur, une triangulation adéquate d'une surface doit vérifier deux propriétés fondamentales : la *proximité* et la *régularité*. La proximité traduit le contrôle de l'écart maximal entre le maillage et la surface, tandis que la régularité traduit localement une continuité d'ordre G^1 de la surface. Le critère de qualité (ou *compacité*) des triangles, essentiel pour de nombreuses applications, est scrupuleusement respecté au cours de la décimation. La technique s'inspire de Cohen et al [CVM⁺96] en définissant deux types de zones de tolérance : une enveloppe globale autour de la surface à l'intérieur de laquelle le maillage est contraint au cours de la simplification, et un cône centré en chaque sommet pour respecter la contrainte de régularité définie par l'ouverture angulaire du cône. La méthode consiste à décimer et à optimiser alternativement les arêtes et les sommets du maillage surfacique, et ce tant que ces opérations respectent les contraintes exprimées par un triplet de contrôle (δ, θ, β) donné :

- δ traduit l'erreur géométrique maximale tolérée exprimée en pourcentage de la diagonale de la boîte englobante du maillage,
- θ traduit le niveau de régularité du maillage exprimé en degrés (l'angle du cône de tolérance précédemment mentionné),
- et β permet de contrôler la dégradation en qualité des triangles.

En pratique le calcul de la distance de Hausdorff est optimisé par majoration successive à chaque décimation ou déplacement élémentaire, la permutation d'arête n'est autorisée que si les faces concernées sont quasi coplanaires et si l'optimisation en position conduit à une amélioration de la qualité des triangles. Les résultats montrent des qualités en forme et une régularité parfaitement contrôlées, ainsi que des temps de calcul fulgurants : respectivement 70, 45 et 23 secondes pour passer de 134000 à 22000, 8000 et 3400 triangles (figure 2.7). Les qualités d'une telle méthode sont nombreuses :

- la déviation est contrôlée suivant plusieurs critères,
- le paramétrage par le biais du triplet (δ, θ, β) est aisé,
- et la vitesse de décimation augmente lorsque le nombre de triangles cible diminue.

Fusion de faces et optimisation en position

Gieng et al [GHJ⁺97] utilisent la fusion de face comme opérateur atomique de décimation, et optimisent la position du sommet résultant. Les deux courbures principales sont calculées localement sur la surface avant décimation de la face, puis le sommet est placé sur le point situé à l'intersection entre une droite de direction la normale \vec{n} à la face concernée et passant par le centroïde déduit des courbures principales, et la surface approximante. La distorsion liée à la fusion d'un triangle T est proportionnelle à sa surface, et à la somme de trois termes respectivement liés à sa courbure, sa compacité et la valence du sommet résultant de la fusion. Il en résulte un réglage fastidieux des paramètres, et un temps d'exécution important.

Simplification topologique

El-Sana et Varshney [ESV98] proposent une méthode originale de simplification topologique utilisant la métrique L_∞ et les alpha prismes.

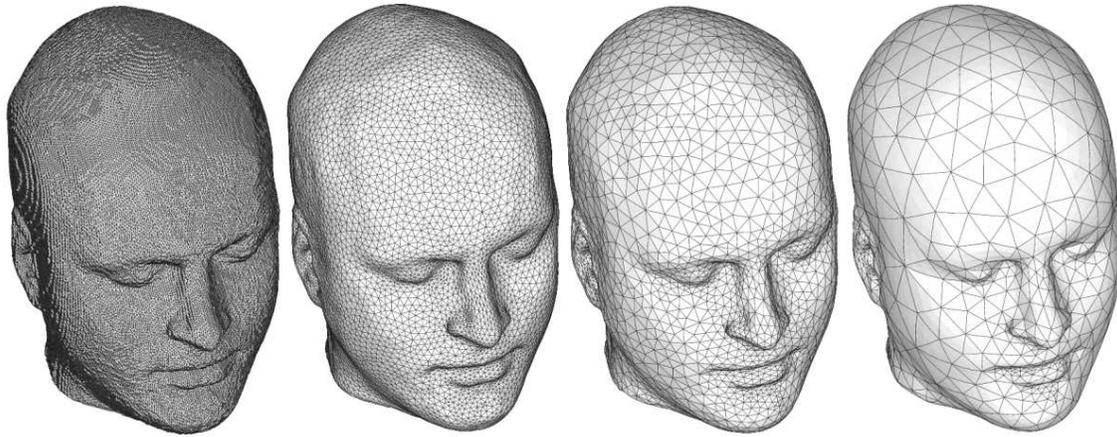


FIG. 2.7 : Simplification de Borouchaki (avec l'autorisation de l'auteur). Respectivement de gauche à droite : le maillage original (134000 faces) puis les versions géométriquement simplifiées à 22000, 8000, et 3400 faces. La qualité en forme des triangles est parfaitement contrôlée, ainsi que la distorsion géométrique exprimée en pourcentage de la diagonale de la boîte englobante.

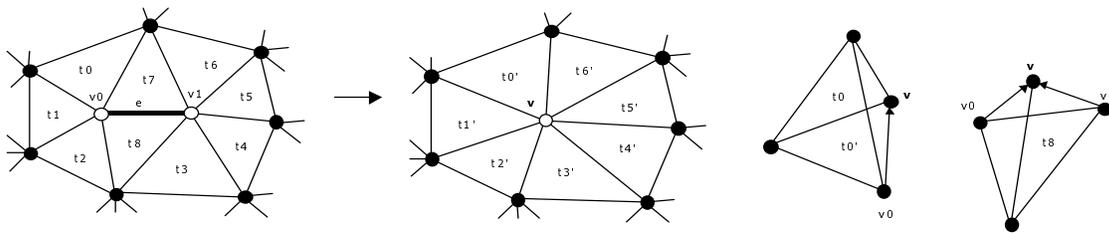


FIG. 2.8 : L'opération de fusion de l'arête e agglomère les sommets v_0 et v_1 en un sommet cible v . Les triangles t_7 et t_8 sont supprimés à cette occasion et l'opération de fusion forme des volumes élémentaires signés (deux exemples à droite). Le sommet v est placé de telle sorte que la variation élémentaire de volume soit nulle afin de préserver le volume du maillage au cours de la décimation.

Une distorsion basée écran

Cohen et al [COM98] ont proposé en 1998 de découpler les couleurs et les normales de la géométrie, et de générer plutôt une texture et une carte des normales. L'erreur est mesurée en équivalent écran pour ces données, tandis que la géométrie est simplifiée par fusion d'arêtes.

Préservation de volume

Lindstrom et Turk [LT98] proposent une méthode rapide de décimation basée sur la fusion d'arêtes. Le principe est très astucieux : imaginons la trace laissée par un triangle adjacent lors d'une fusion d'arête, elle forme un volume signé, positif si ce dernier est situé en dehors du modèle et négatif sinon (figure 2.8). En sommant les contributions des triangles adjacents on obtient la variation de volume générée localement par une fusion d'arête. La résolution du système aboutit à contraindre sur un plan la position du sommet résultant de la fusion, puis une maximisation de la compacité des triangles permet de trouver sa position définitive sur le plan. Une approche basée sur la conservation de surface agit ensuite pour préserver les bords.

Les derniers papiers cités et les travaux de Cignoni et al [CRS98] montrent une activité importante autour du choix des métriques et des critères de distorsion : distance L_2 Hausdorff, préservation de volume, enveloppes

de simplification, etc... et nous verrons dans la section 2.3 notre choix pour une métrique basée volume, afin de guider le processus d'optimisation. Voyons maintenant une approche inverse : le raffinement par subdivision.

2.2.2 Raffinement par subdivision

Les approches basées décimation nous ont montré que le choix d'une opération inversible permet de représenter un maillage sous la forme d'une séquence plus ou moins ordonnée de raffinements élémentaires (voir respectivement [Hop96] et [Hop97]). On peut ainsi définir la résolution d'un maillage au sommet près. Toutefois, il peut être extrêmement intéressant dans une finalité de codage ou de raffinement rapide (par niveaux) d'enregistrer un maillage sous la forme d'une séquence de raffinements obtenus par subdivision régulière [EDD⁺95] [CPD⁺96] [SS95] [LDW97] [SGW98] [LSS⁺98] [GSS99] [KCVS98] [DGSS99] [Gio99] ou même irrégulière [VTK⁺99].

L'approche de Eck et al [EDD⁺95] consiste à partitionner un maillage, puis à remailler les régions correspondantes suivant une organisation imbriquée par subdivision régulière 1 vers 4. L'analyse multirésolution aboutit ensuite à un ensemble de coefficients d'ondelettes contenant l'information de raffinement servant à passer d'un niveau de résolution à l'autre. Les améliorations consistent entre autres à prendre en compte les attributs d'apparence [CPD⁺96], et à développer des outils pour les maillages, analogues à ceux utilisés en traitement de signal [GSS99].

La décomposition en ondelettes de Valette et al [VTK⁺99] effectue une analyse sur des maillages quelconques, levant ainsi la contrainte de structure emboîtée 1 vers 4. Les auteurs démontrent que seulement 4 types de subdivision sont suffisants pour bâtir la structure via un ensemble de permutations d'arêtes préalables.

La principale difficulté d'un mode de représentation multi-échelle réside en la paramétrisation de la surface associée au remaillage. Gioia [Gio99] propose un algorithme en trois étapes : partitionnement polygonal de la surface, triangulation et paramétrisation du maillage de base résultant. Le partitionnement est obtenu par propagation récursive sur les triangles suivant un critère de courbure mesuré à différentes échelles. On obtient alors un découpage en régions défini par union de polygones, ces dernières étant séparées par les courbes de discontinuité du maillage original. La triangulation s'opère ensuite par ajout progressif de sommets classifiés en trois catégories : les sommets frontières à au moins trois régions, les sommets de bords nécessaires à une bonne approximation des bordures et les sommets nécessaires à une triangulation homogène. La paramétrisation est ensuite obtenue par relaxation et traitement spécifique sur les bords des régions. Une telle approche opère de la sorte un partitionnement efficace des maillages et prépare naturellement la phase de codage des coefficients d'ondelettes.

Récemment, Guskov et al [GVSS00] ont défini le concept de *maillages normaux* comme nouveau mode de description des surfaces, celui étant fortement inspiré de la géométrie différentielle. Un maillage normal est une structure multirésolution où chaque niveau s'exprime en fonction du niveau plus grossier et d'un ensemble d'offsets normaux à la surface. Un maillage de base est généré au préalable par la dernière mouture de la technique de simplification de Garland-Heckbert [GH98], puis une paramétrisation MAPS [LSS⁺98] est élaborée sur ce niveau afin de construire une hiérarchie de maillages obtenue par subdivisions successives. Cette technique ne garantit pas une restitution sans perte de la connectivité mais s'avère très performante pour une application de codage progressif [KG00] [KSS00], aussi nous reviendrons sur cette méthode dans le chapitre 3.

Il existe également une famille de méthodes ne fonctionnant ni par décimation, ni par raffinement, mais par



FIG. 2.9 : Représentation progressive de Yemez-Schmitt sur la statuette *Coignard* : le niveau de détail augmente progressivement avec le nombre de particules, lui-même lié au nombre de triangles : 164, 795, 3232, 12914, 51253, et 201792.

échantillonnage.

2.2.3 Echantillonnage

Turk [Tur92] commence par échantillonner un grand nombre de sommets sur la surface du maillage original, puis un processus de relaxation les déplace sur les régions de forte courbure. Une triangulation conjointe est ensuite bâtie en utilisant les sommets du maillage original, puis itérativement simplifiée en fonction d'une erreur évaluée localement.

Rossignac et Borrel [RB92] découpent l'espace en une grille régulière de résolution spécifiée par l'utilisateur, puis agglomèrent les sommets contenus dans la même cellule en un sommet unique. Cette méthode est appelée *vertex clustering*. Un processus d'élimination supprime ensuite les arêtes et les triangles dégénérés. La méthode s'avère très rapide, mais au prix d'une qualité inférieure aux méthodes fonctionnant par décimation et contrôle.

Il existe des variantes basées sur l'agglomération de sommets [ESV99] [SS97], avec la contrainte d'une grille régulière en moins, et la possibilité de générer des niveaux de résolution continus.

Yemez et Schmitt [YS00] proposent une méthode originale adaptée à la transmission progressive d'objets 3D complexes et texturés (figure 2.9). La géométrie et la texture sont représentées sous une forme unifiée de particules surfaciques associées à un découpage de l'espace en *octrees*. L'ordonnement judicieux des particules permet ensuite une reconstruction progressive à partir d'un volume de données compact. En pratique le volume englobant minimal de l'objet est partitionné en *octree*, et les cubes sont étiquetés en trois classes : dans, hors ou sur la surface. Seuls les cubes classés sur la surface sont intéressants pour la structure, et sont ainsi subdivisés tout en pré-codant une structure de type *surface octree*. Une telle organisation associée à un précalcul de la triangulation autorise la génération (et la commutation) au vol de niveaux de résolution d'une manière très robuste.

2.2.4 Positionnement des travaux

Nombres de méthodes de simplification exploitent la métrique de Hausdorff, la métrique L_2 , des heuristiques de préservation de volume ou des enveloppes de simplification, mais aucune n'utilise vraiment la métrique basée volume, c'est à dire le volume enfermé entre le modèle et la surface originale, pour *guider* un processus d'optimisation de la géométrie. Cela provient du fait qu'il n'existe pas de formule analytique simple exprimant le volume enfermé entre deux maillages triangulaires, d'autant plus lorsque ceux-ci comportent des trous et/ou des bords. Pourtant, une telle méthode a été développée pour l'approximation des images par Sanson, Lechat et Labelle [LSL97] [SL98] [Lec99], et l'on constate que cette métrique permet d'approximer efficacement les contours d'une image vue comme une carte d'altitude composée d'un réseau de mailles. La prochaine section présente donc le travail d'adaptation à la troisième dimension d'une telle méthode, qui peut être utilisée indépendamment de la méthode de simplification.

2.3 Approximation

L'approximation d'un maillage \hat{M} se définit par rapport à un maillage référent M , et consiste à minimiser la distorsion géométrique entre \hat{M} et M en agissant sur la géométrie de \hat{M} . Elle représente une phase préalable au codage, puisqu'elle minimise directement la distorsion visuelle liée à la simplification.

2.3.1 Principe général

Soit M un maillage original et \hat{M} sa version géométriquement simplifiée appelée aussi modèle. L'algorithme d'optimisation présenté dans cette section agit sur les positions des sommets d'un maillage et permet d'obtenir la meilleure approximation au sens d'une métrique basée sur le volume compris entre le modèle et le maillage original M . On peut ainsi générer un maillage progressif en alternant décimation atomique et optimisation locale, ou plus simplement un maillage optimisé globalement après une séquence de décimation (figure 2.10).

Il reste à caractériser la notion de distorsion géométrique entre deux maillages, on choisit pour cela une métrique basée sur le volume.

2.3.2 Métrique basée volume

Cette section présente le contexte théorique de l'approximation de maillages 3D afin de lier le calcul de l'approximation avec la métrique choisie pour la mesure de la déviation géométrique. On montre l'expression de la déviation géométrique entre deux maillages et la formulation variationnelle du problème d'optimisation où les positions des sommets sont progressivement optimisées. Le choix d'une métrique est un point crucial du problème d'approximation de maillages 3D puisque celle-ci permet de relier la notion de distorsion visuelle aux données et de guider le processus d'optimisation. Soit M le maillage original et \hat{M} sa version géométriquement simplifiée à optimiser. On choisit d'utiliser un critère purement géométrique pour caractériser la déviation : le volume compris entre les deux surfaces décrites par les maillages M et \hat{M} (figure 2.11).

Le volume est noté $V(M, \hat{M})$ et sa valeur est donnée par la formule de Lebesgue :

$$V(M, \hat{M}) = \int_{\mathbb{R}^3} I_{M, \hat{M}}(\vec{q}) d^3 \vec{q} \quad (2.1)$$

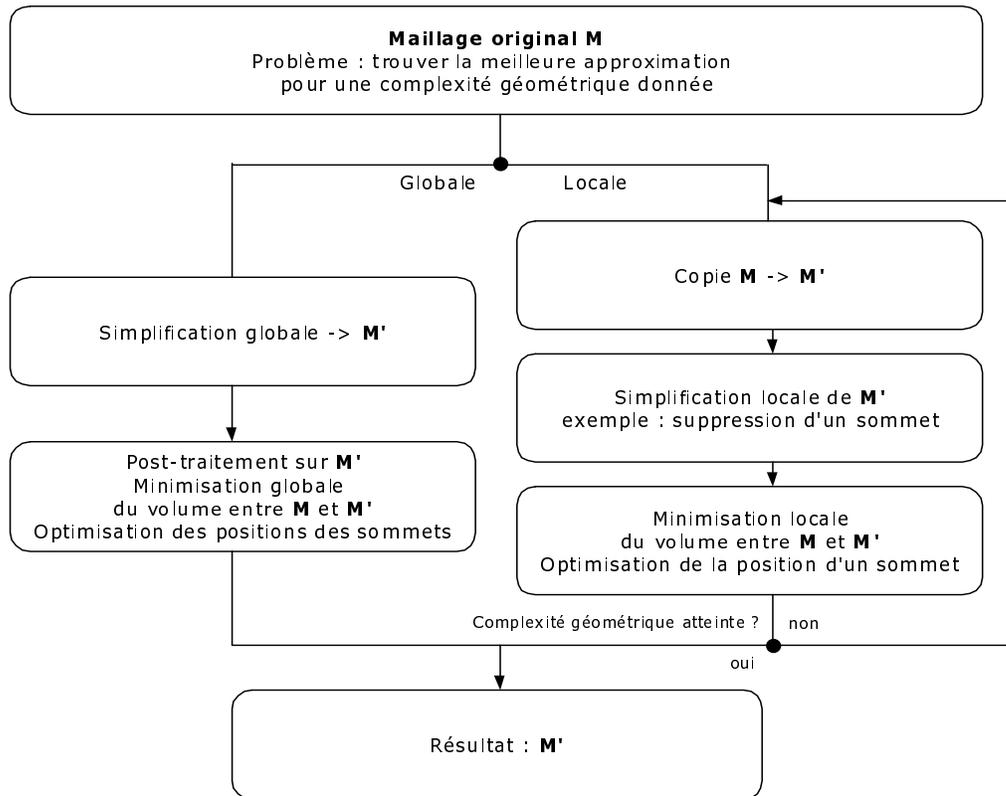


FIG. 2.10 : Le processus d'optimisation peut être appliqué globalement après une méthode de simplification géométrique ou localement en alternant décimation atomique et optimisation locale. Dans les deux cas l'optimisation fait référence au maillage original.

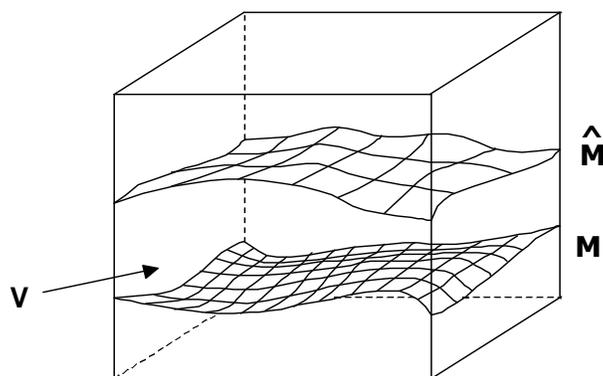


FIG. 2.11 : La déviation géométrique est définie par le volume compris entre les deux surfaces M et \hat{M} .

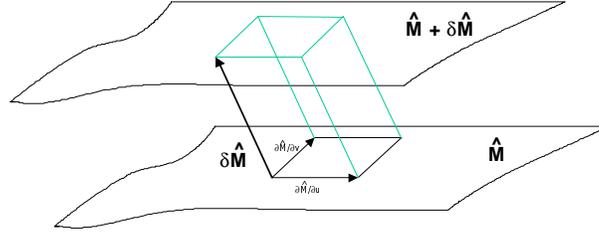


FIG. 2.12 : La variation élémentaire de volume correspond au parallélépipède généré par $\delta\hat{M}(u, v)$ sur l'élément de surface correspondant à $dudv$. Une telle variation est signée : négative lorsque le volume décroît et positive dans le cas inverse. Cette constatation nous amène ensuite à remplacer la notion de volume intérieur au modèle \hat{M} par le volume enfermé entre \hat{M} et M .

où $I_{M, \hat{M}}(\vec{q})$ représente la fonction indicatrice de l'intérieur du volume, \vec{q} est un point de \mathbb{R}^3 et $d^3\vec{q}$ est la mesure de Lebesgue de \mathbb{R}^3 . Notons ici que l'intérieur du volume définit le volume compris entre les maillages M et \hat{M} . Voyons maintenant si la phase d'optimisation en position est possible à partir d'une telle expression.

2.3.3 Optimisation en position

Un maillage étant partiellement défini par les positions de ses sommets $\{X_1, X_2, \dots, X_N\}$ et un ensemble de faces orientées, on peut déduire une fonction de coût à minimiser dépendant de ces variables à partir de la métrique basée volume :

$$E(X_1, X_2, \dots, X_N) = V(M, \hat{M}(X_1, X_2, \dots, X_N)) \quad (2.2)$$

L'algorithme d'optimisation développé consiste donc à trouver les positions optimales des sommets $\{X_1, X_2, \dots, X_N\}$ qui minimisent le volume $V(M, \hat{M})$. Toutefois, à partir de maillages triangulaires, il n'existe pas d'expression analytique simple du volume (équation 2.1) en fonction des variables $\{X_1, X_2, \dots, X_N\}$. Ainsi, on ne peut pas en déduire directement la dérivée afin de construire un algorithme de minimisation par gradient. Rappelons que nous cherchons en fait à *minimiser* le volume, et ne cherchons en aucun cas à le calculer. Nous utilisons plutôt l'expression de la variation élémentaire de volume afin d'en déduire une expression du gradient.

Variation élémentaire de volume

L'algorithme d'optimisation peut être vu comme un processus d'évolution du maillage, calculé dans le but de minimiser le volume $V(M, \hat{M})$. Pour cela, nous devons trouver l'expression de la variation de volume pour une évolution arbitraire du maillage \hat{M} . Cette évolution est définie par un champ de vecteur $\delta\hat{M}$ sur le modèle approximant. Si l'on considère une paramétrisation arbitraire de la surface par u et v , le champ de vecteur s'exprime sous la forme : $\delta\hat{M}(u, v)$. Considérons maintenant une approximation locale de la surface par son plan tangent (une surface G^1 -continue en somme), la variation élémentaire de volume représente ainsi le parallélépipède généré par $\delta\hat{M}(u, v)$ sur l'élément de surface correspondant à $dudv$ (figure 2.12). Notons que cette variation de volume est signée, négative lorsque le volume décroît, et positive dans le cas inverse. Cela nous amène à définir la notion de volume intérieur pour le modèle \hat{M} , impossible à définir pour des surfaces ouvertes. En pratique, la notion de volume intérieur est traduite par le volume compris entre \hat{M} et M .

On peut ainsi écrire la valeur absolue de la variation élémentaire de volume induite par $\delta\hat{M}$ sur l'élément de surface $dudv$ par le produit mixte :

$$|\delta_2 V(\hat{M}, \delta\hat{M}(u, v))| = \left| \delta\hat{M}(u, v) \cdot \left[\frac{\partial\hat{M}(u, v)}{\partial u} \times \frac{\partial\hat{M}(u, v)}{\partial v} \right] \right| dudv \quad (2.3)$$

Considérons maintenant l'expression suivante :

$$\left[\frac{\partial\hat{M}(u, v)}{\partial u} \times \frac{\partial\hat{M}(u, v)}{\partial v} \right] dudv = \vec{n}(u, v) d\sigma(u, v) \quad (2.4)$$

où $\vec{n}(u, v)$ représente le vecteur normal unité sur $\hat{M}(u, v)$ et $d\sigma(u, v) = \left\| \frac{\partial\hat{M}}{\partial u} \times \frac{\partial\hat{M}}{\partial v} \right\| dudv$ l'élément de surface.

On réintroduit maintenant le maillage original M , le but étant de minimiser le volume entre M et \hat{M} . La variation de volume est négative lorsque $\delta\hat{M}(u, v)$ ramène \hat{M} sur M . On peut ainsi écrire la variation élémentaire de volume :

$$\delta_2 V(M, \hat{M}, \delta\hat{M}(u, v)) = \eta(M, \hat{M}, \vec{n}(u, v)) (\delta\hat{M}(u, v) \cdot \vec{n}(u, v)) d\sigma(u, v) \quad (2.5)$$

où $\eta(M, \hat{M}, \vec{n}(u, v)) = \eta(u, v) = -1$ lorsque la normale $\vec{n}(u, v)$ est orientée vers le volume compris entre M et \hat{M} , et $+1$ sinon. Ce terme prend maintenant explicitement en compte la dépendance aux deux maillages M et \hat{M} .

On peut ensuite en déduire la variation de volume pour le maillage entier \hat{M} :

$$\delta V(M, \hat{M}, \delta\hat{M}) = \int \int_{u,v} \eta(u, v) (\delta\hat{M}(u, v) \cdot \vec{n}(u, v)) d\sigma(u, v) \quad (2.6)$$

Notez que seule l'hypothèse de G^1 -continuité a été émise sur l'expression de \hat{M} .

Interpolation

Rappelons maintenant que \hat{M} est une surface définie par un maillage, ce qui nous permet de retourner à un problème d'optimisation en dimension finie puisque l'on agit seulement sur les positions d'un nombre borné de nœuds du maillage :

$$\hat{M}(u, v) = \sum_{i=1}^N \lambda_i(u, v) X_i \quad (2.7)$$

où les X_i décrivent les sommets définis sur \mathbb{R}^3 et les $\lambda_i(u, v)$ sont les fonctions de forme déduites des coordonnées barycentriques. On utilise ici une interpolation affine (figure 2.13).

Nous sommes maintenant munis d'un interpolateur, il reste donc à exprimer l'expression de l'évolution globale de la surface.

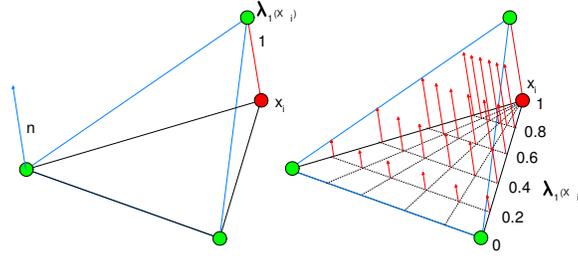


FIG. 2.13 : Les fonctions de forme sont déduites à partir d’une interpolation linéaire sur les sommets du triangle. Pour tout point (u, v) à l’intérieur du triangle, exactement trois des valeurs des coefficients sont différentes de zéro. Ces valeurs sont les coordonnées barycentriques du point à l’intérieur du triangle. En pratique ces valeurs sont calculées par un rapport de surfaces (voir [ZT91]).

Evolution de surface

L’évolution globale du maillage peut s’écrire sous la forme :

$$\delta E(M, \hat{M}, \delta X_1, \delta X_2, \dots, \delta X_N) = \sum_{i=1}^N \int \int_{u,v} \eta(u, v) \lambda_i(u, v) (\delta X_i \cdot \vec{n}(u, v)) d\sigma(u, v) \quad (2.8)$$

On peut ainsi en déduire l’expression de la dérivée partielle de la fonction de coût pour un sommet X_i :

$$\frac{\partial E}{\partial X_i} = \int \int_{u,v \in \text{Supp}(\lambda_i)} \eta(u, v) \lambda_i(u, v) \vec{n}(u, v) d\sigma(u, v) \quad (2.9)$$

où $\text{Supp}(\lambda_i)$ est le support du domaine de la fonction de forme λ_i , dans notre cas ce support correspond aux faces adjacentes du sommet X_i (figure 2.14). Cette expression est évaluée par discrétisation de l’équation 2.9, qui se traduit en un échantillonnage régulier sur les triangles (figure 2.15). Un nombre de points par face est paramétré par l’utilisateur, et la contribution vectorielle résultante d’une face est pondérée par sa surface. Ici $\vec{n}(u, v)$ traduit le vecteur normal unité sur la face courante. $\lambda_i(u, v)$ définit la fonction de forme, sa valeur vaut 1 sur X_i et décroît linéairement vers zéro sur les sommets voisins (figure 2.13).

Plusieurs méthodes peuvent être utilisées afin de déterminer le terme d’orientation relative entre les deux surfaces $\eta(u, v)$, ce qui constitue un réel problème théorique. On utilise toutefois une heuristique qui représente un bon compromis temps / qualité. Pour chaque point X_{uv} échantillonné sur une face voisine F de X_i , on considère une ligne L issue de X_{uv} et de direction la normale \vec{n} à F (figure 2.16). On cherche ensuite la plus proche intersection notée X_M entre L et le maillage original M , avec une restriction au voisinage de X_i par le biais d’une structure de partitionnement spatial définie par Wyvill [WMW86]. Notons F_M la face intersectée de M et \vec{n}_M le vecteur normal unité sur F_M . On déduit ensuite l’orientation relative locale entre \hat{M} et M par la loi suivante : Si l’intersection existe et $(\vec{n} \cdot \vec{n}_M > 0)$, alors $\eta(u, v) = -\text{signe}(\vec{n} \cdot \overrightarrow{X_{uv}X_M})$, sinon $\eta(u, v) = 0$.

Cette loi est illustrée par la figure 2.16. La contribution du *rayon* issu de la face F est ainsi inhibée lorsqu’aucune intersection n’est détectée dans le voisinage de X_i ou lorsque les orientations des faces correspondantes sont opposées. En pratique cette heuristique fonctionne lorsque \hat{M} est une version décimée de M puisque la décimation garantit une relative conservation des orientations, elle permet aussi d’éviter la formation de plis sur la surface et rend le comportement de l’algorithme cohérent.

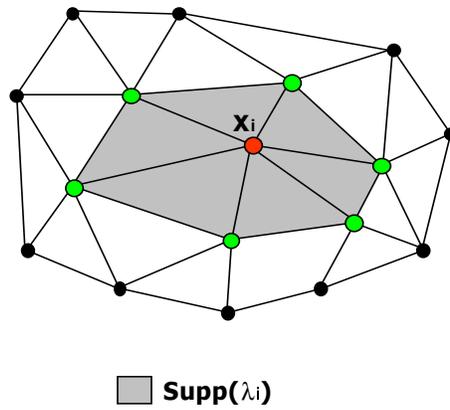


FIG. 2.14 : Le support du domaine de λ_i concerne seulement les faces adjacentes du sommet X_i , sur lesquelles on applique une méthode d'échantillonnage pour évaluer l'équation 2.9.

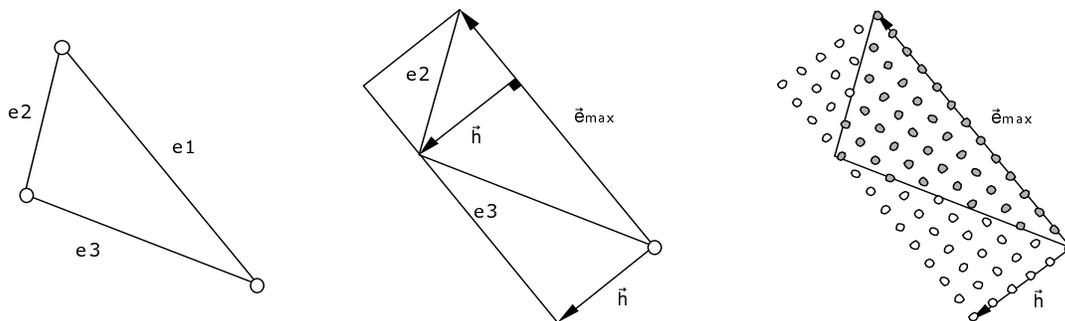


FIG. 2.15 : L'expression 2.9 est évaluée par échantillonnage régulier sur les triangles. Un repère d'échantillonnage est localement déterminé par l'arête de longueur maximale dénotée \vec{e}_{max} et la hauteur \vec{h} du triangle. Seuls les points localisés sur le triangle sont retenus pour le calcul (à droite), puis la contribution totale est pondérée par la surface du triangle.

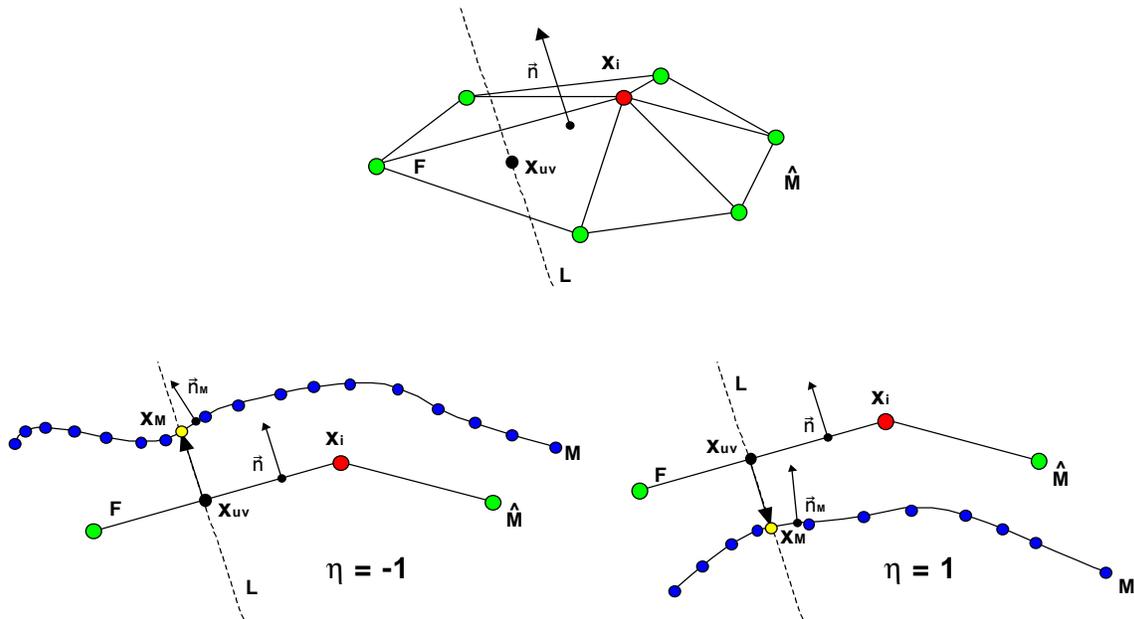


FIG. 2.16 : Cette figure décrit le calcul local de l'orientation relative entre M et \hat{M} . Soit X_{uv} un point échantillonné sur une face F adjacente au sommet X_i , on considère une ligne L issue de X_{uv} et de direction le vecteur normal \vec{n} à F . On cherche la plus proche intersection notée X_M entre L et le maillage original M . Si l'intersection est trouvée dans le voisinage de X_{uv} , et que les normales \vec{n} et \vec{n}_M ont la même orientation, on compare alors l'orientation entre \vec{n} et $X_{uv}\vec{X}_M$ afin d'en déduire l'orientation relative locale entre M et \hat{M} , autrement dit le terme η .

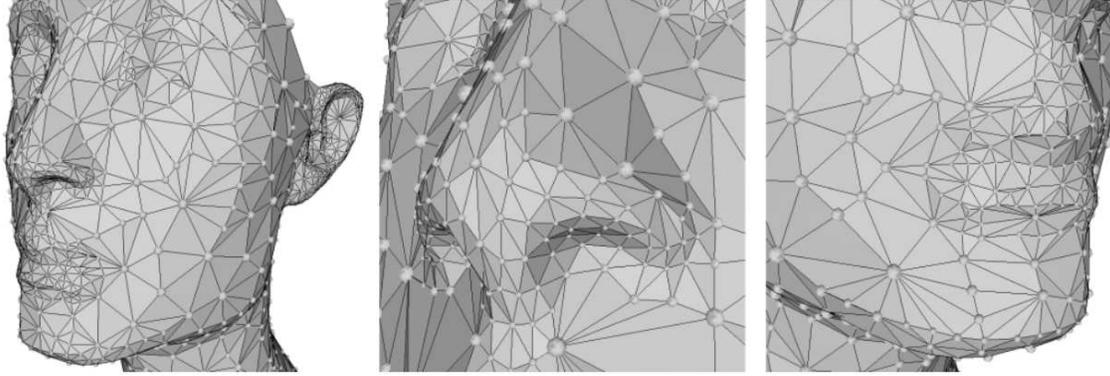


FIG. 2.17 : Le pas du gradient est initialisé par une fraction de la longueur moyenne des arêtes autour de chaque sommet ($\gamma_0 = 0.1$ sur cet exemple). Cette heuristique correspond à une grandeur géométrique locale.

L'équation 2.9 indique seulement la direction et le sens du mouvement devant être appliqué au sommet X_i afin de minimiser localement le volume, mais cela ne donne aucune indication quant à la norme du vecteur résultant. La section suivante étudie ce problème, ainsi que la description de l'algorithme de gradient mis en œuvre.

Gradient à pas adaptatif

Un gradient à pas adaptatif a été mis en œuvre afin de résoudre le problème d'optimisation. Il s'agit de minimiser la fonction de coût $E(X_1, \dots, X_N)$. Pour ce faire la méthode itérative génère un ensemble de vecteurs X^0, X^1, \dots (où X^k décrit l'ensemble des sommets à la k^{ieme} itération), qui approxime progressivement l'optimum local de E . Pendant la minimisation, les positions des sommets évoluent de manière à réduire le volume compris entre les deux surfaces. Cette évolution nécessite le calcul du gradient à l'itération k (voir équation 2.9).

Après la k^{ieme} itération la position courante du sommet X_i peut ainsi s'écrire :

$$X_i^{k+1} = X_i^k - \gamma_i^k \frac{\frac{\partial E^k}{\partial X_i}}{\left\| \frac{\partial E^k}{\partial X_i} \right\|} \quad (2.10)$$

où γ_i^k représente un pas adaptatif borné : $\gamma_i^k \in [\gamma_{min}, \gamma_0]$. γ_0 correspond à la valeur initiale de γ_i^k , fonction de la dynamique géométrique locale de \hat{M} . On utilise pour cela une fraction de la valeur moyenne des longueurs d'arêtes adjacentes au sommet courant X_i (image 2.17). Durant le processus d'optimisation, le pas est réduit lorsqu'une oscillation est détectée entre deux itérations successives, i.e. lorsque $(\frac{\partial E^{k+1}}{\partial X_i} \cdot \frac{\partial E^k}{\partial X_i}) < 0$. Cette heuristique fonctionne bien puisque les sommets oscillent autour de leur optimum, de plus la convergence du processus est ainsi assurée. Dans l'approche traditionnelle de l'algorithme de gradient à pas adaptatif, c'est le calcul de variation d'énergie $\delta E(M, \hat{M}, \delta X_i)$ qui pilote l'évolution du pas. Or l'énergie correspond ici au volume compris entre les deux surfaces décrites par les maillages M et \hat{M} , et on rappelle que l'on ne calcule jamais cette expression.

L'optimisation est stoppée lorsqu'un nombre donné d'itérations est atteint, où lorsque les mouvements des sommets deviennent inférieurs à un seuil prédéfini. Le processus d'optimisation peut ainsi être considéré comme un ensemble de vecteurs de mouvement ramenant progressivement \hat{M} sur M , tout en minimisant le volume compris entre ces deux maillages. La section suivante présente les résultats de l'optimisation en position.

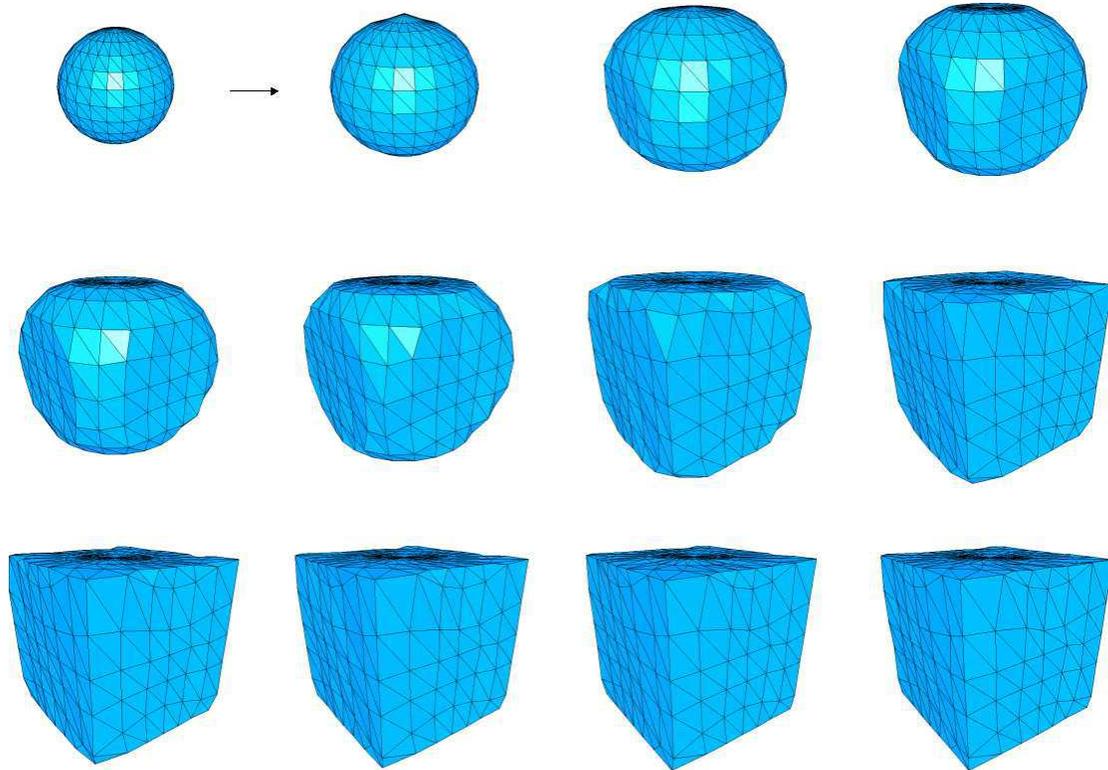


FIG. 2.18 : Minimisation de volume entre une sphère et un cube. Deux maillages différents ont été choisis volontairement sur cet exemple afin de mettre en évidence les qualités intrinsèques de la métrique basée volume. On constate sur cet exemple une parfaite approximation de la surface originale, notamment sur les coins et les arêtes vives régulières. Les sommets convergent par oscillation autour de leur optimum local, et l'optimisation s'arrête soit lorsqu'un nombre d'itérations est atteint, soit lorsque les mouvements deviennent infimes par rapport aux pas initiaux du gradient. A condition de conserver une compacité raisonnable sur les triangles, l'optimum est atteint lorsque la solution initiale est valide et lorsque le volume enfermé entre les deux surfaces est nul, ce qui correspond à une parfaite approximation des singularités géométriques. Le sous-optimum serait atteint si le maillage initial ne disposait pas de la géométrie suffisante pour approximer correctement le maillage de référence.

2.3.4 Résultats

Sur la figure 2.18, le processus d'optimisation minimise le volume entre deux maillages triangulaires. On choisit volontairement deux maillages différents (une sphère pour le modèle, et un cube pour le maillage de référence) afin de mettre clairement en évidence le comportement de l'algorithme. Sur cet exemple 10 itérations sont effectuées entre deux maillages successifs, pour un total de 110 itérations. Aucune contrainte géométrique n'est imposée, et les singularités géométriques telles que les arêtes vives et les coins sont correctement approximates sans processus additionnel d'approximation de courbes ni de seuil de détection d'arêtes vives décrits par Hoppe [Hop96]. Les qualités de la métrique choisie concernent notamment la prise en compte naturelle des singularités, à l'inverse de la métrique L^2 . Voyons maintenant un résultat correspondant à l'utilisation finale de la méthode : après une phase de simplification par décimation.

La figure 2.19 illustre l'algorithme d'optimisation entre un rhombicuboctaèdre et sa version géométriquement simplifiée. Les positions des sommets sont progressivement optimisées de manière à minimiser le

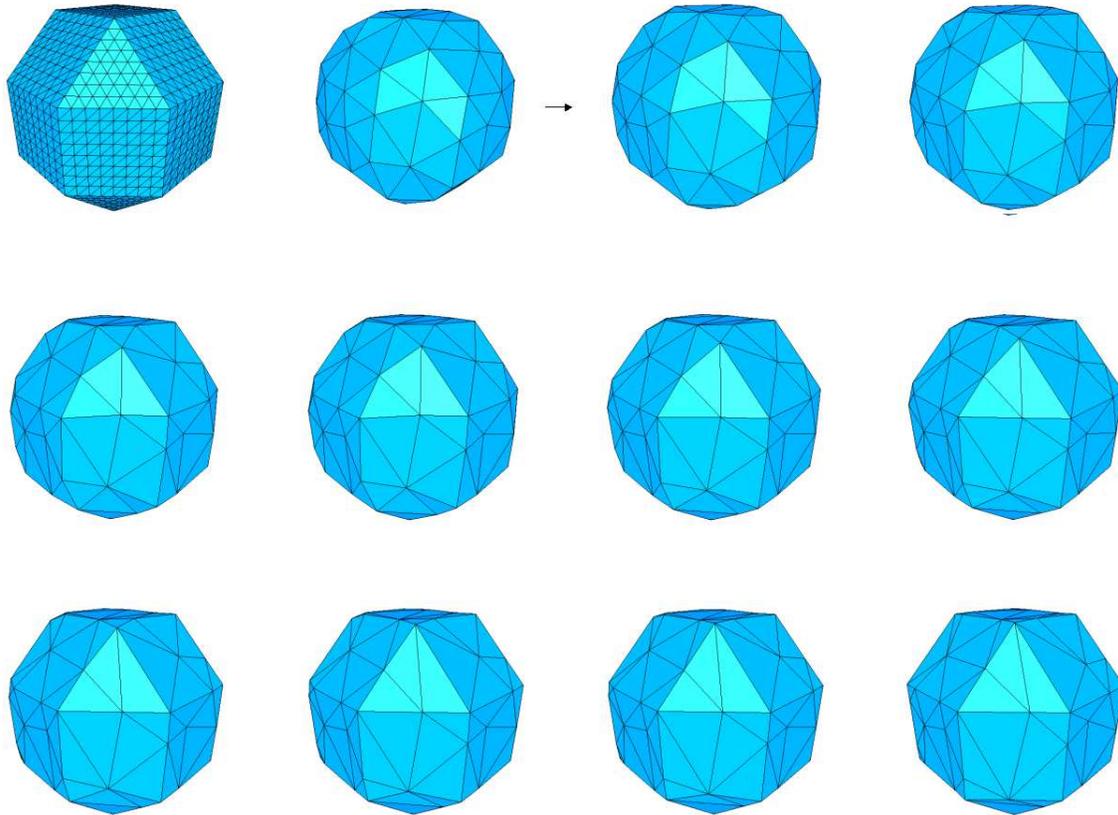


FIG. 2.19 : Minimisation de volume entre un rhombicuboctaèdre et une version géométriquement simplifiée par fusion d'arêtes suivant un critère de longueur seul. 10 itérations sont effectuées entre deux maillages successifs sur cet exemple, et la version finale approxime parfaitement le maillage original.

volume compris entre les deux maillages, on peut constater une approximation naturelle des coins et des arêtes vives. A l'inverse, la figure 2.20 illustre une mauvaise approximation des singularités géométriques générée par l'utilisation de la métrique L_2 sans détection additionnelle des arêtes vives. De plus, cette métrique nécessite l'ajout d'un terme de régularisation ou de ressort dont le paramétrage est délicat et influence sensiblement le résultat.

La figure 2.21 illustre la restauration de volume appliquée sur le maillage *horse*, après une phase de simplification. La phase d'optimisation déplace les sommets dans le but de restaurer le volume original modifié lors de la phase de simplification, et d'améliorer ainsi la qualité visuelle. Notez que sur la version 4500 triangles, le ventre et le dos du cheval voient leur volume restauré. Ces améliorations sont notamment intéressantes pour l'opérateur topologique de fusion d'arête, qui contracte localement le modèle dans les régions convexes, et augmente localement le volume dans les régions concaves.

Les résultats démontrent l'intérêt d'utiliser une métrique basée volume pour optimiser les positions des sommets d'un maillage, et son adaptation à la simplification de maillages par fusion d'arête. De plus, une telle phase d'optimisation peut être appliquée après un large éventail de méthodes de simplification, la seule contrainte résidant dans la nécessité de conserver une compacité raisonnable sur les éléments. Par ailleurs, un entrelacement dans un contexte d'optimisation locale est également possible. La section suivante étudie la phase

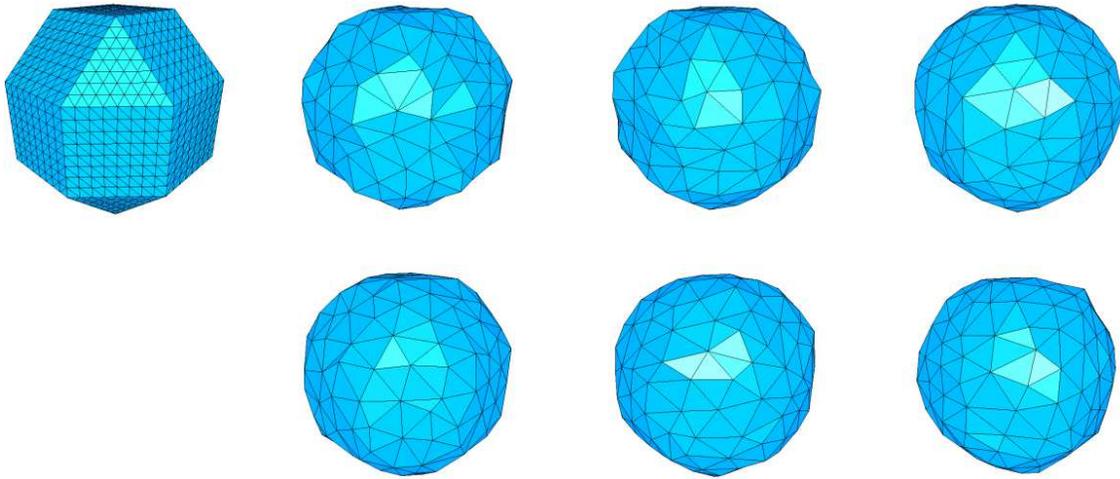


FIG. 2.20 : Approximation L_2 entre un rhombicuboctaèdre et sa version géométriquement simplifiée. Le maillage original, puis les versions simplifiées puis optimisées pour différentes pondérations croissantes du terme de ressort associé à la métrique L_2 . On constate un effet de lissage global plus ou moins important en fonction de la pondération du ressort, ainsi que la mauvaise approximation des singularités comme les coins et les arêtes vives.

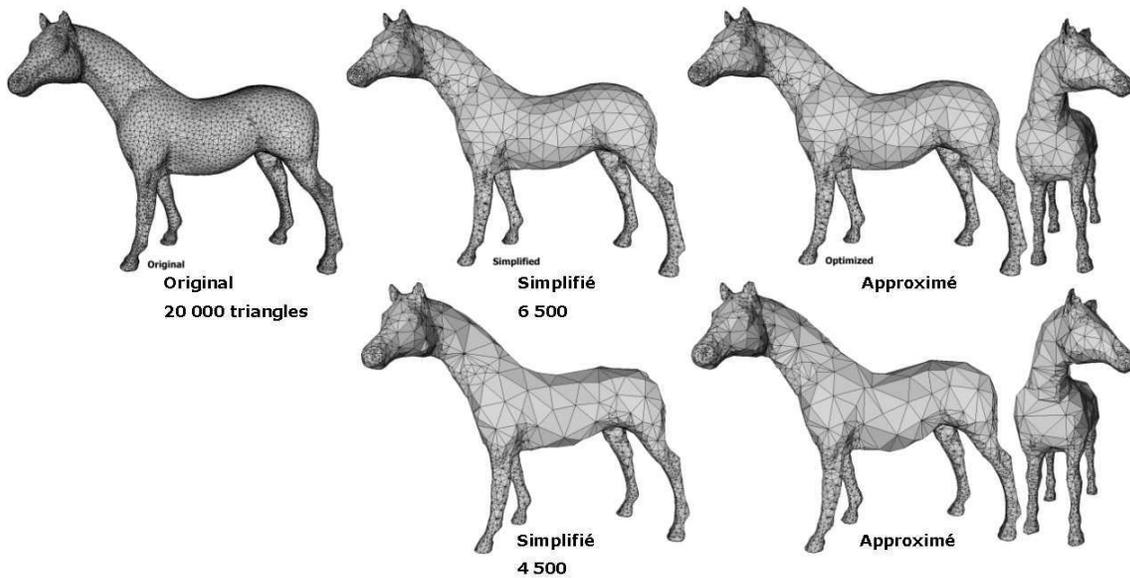


FIG. 2.21 : Le maillage *horse* (20 000 triangles), ses versions respectivement simplifiées et approximées (en haut : 6500 triangles, en bas : 4500 triangles). La phase d'optimisation en position restaure le volume du maillage initial, et on note une amélioration sensible sur la version 4500 triangles, le ventre et le dos voyant leur forme rétablie après la phase de décimation.

de simplification, et plus particulièrement une méthode adaptée à la suppression de la redondance géométrique des maillages triangulaires.

2.4 Simplification par fusion d'arêtes

Pour remédier au problème généralement constaté des réglages délicats de paramètres, et afin d'obtenir une méthode de simplification bien adaptée à l'approche d'optimisation précédemment décrite, cette section étudie une méthode de simplification par fusion d'arêtes.

2.4.1 Principe général

La méthode de simplification proposée utilise la fusion d'arête comme opération atomique de décimation, une double queue de priorité imbriquée basée sur deux critères hiérarchiques, ainsi qu'une méthode de pseudo-optimisation exploitant le degré de liberté offert par la fusion d'arête. Le critère prioritaire est relié à une notion de courbure calculée dans le voisinage, tandis que le critère secondaire est lié à une grandeur géométrique locale.

2.4.2 Simplification géométrique

Afin de réduire la complexité géométrique d'un maillage, l'opérateur topologique élémentaire de décimation utilisé correspond à la fusion d'arête (figure 2.22), grâce à ses bonnes propriétés : l'orientation et la topologie est localement conservée (dans une certaine mesure), aucun trou n'est créé sur la surface, et il n'est pas nécessaire de trianguler après décimation atomique, contrairement à la suppression de sommet. On peut de plus exploiter deux degrés de liberté au cours de la décimation : l'ordre des opérations élémentaires de transformation à appliquer sur le maillage et la position du sommet résultant de la contraction d'une arête. Cette section montre que l'on peut préserver des propriétés géométriques intéressantes sur le maillage, telles que la courbure locale et le rapport d'aspect des triangles. Il est à noter que ce rapport d'aspect constitue une caractéristique importante à conserver pour des raisons de stabilité numérique de l'algorithme d'optimisation présenté précédemment. La section suivante détaille l'exploitation des deux critères dans une double queue de priorité.

2.4.3 Double queue de priorité

Afin d'exploiter le premier degré de liberté mentionné concernant l'ordre des arêtes à fusionner en priorité, une queue de priorité est construite. Convenons que les singularités géométriques doivent être préservées aussi longtemps que possible au cours de la décimation. En effet, les régions de forte courbure sont souvent hautement informatives, puisqu'elles peuvent correspondre à des arêtes vives régulières ou à des coins. On utilise ainsi un critère simple et rapide à évaluer autour d'un sommet : l'angle maximal du dièdre formé par les normales à deux de ses faces adjacentes. Le critère de courbure utilisé pour une arête candidate à la suppression correspond à la moyenne de ce critère évalué sur chacun des deux sommets correspondants. Mentionnons que d'autres critères plus sophistiqués d'estimation de la courbure existent [FB96] et pourraient être utilisés, cependant le critère mentionné fournit de bonnes performances en terme de ratio qualité / complexité. Ce critère de courbure locale est utilisé comme critère prioritaire durant la phase de simplification, et considère une arête candidate valide pour la fusion lorsque sa courbure associée est mesurée en deçà d'un certain seuil. Toutefois,

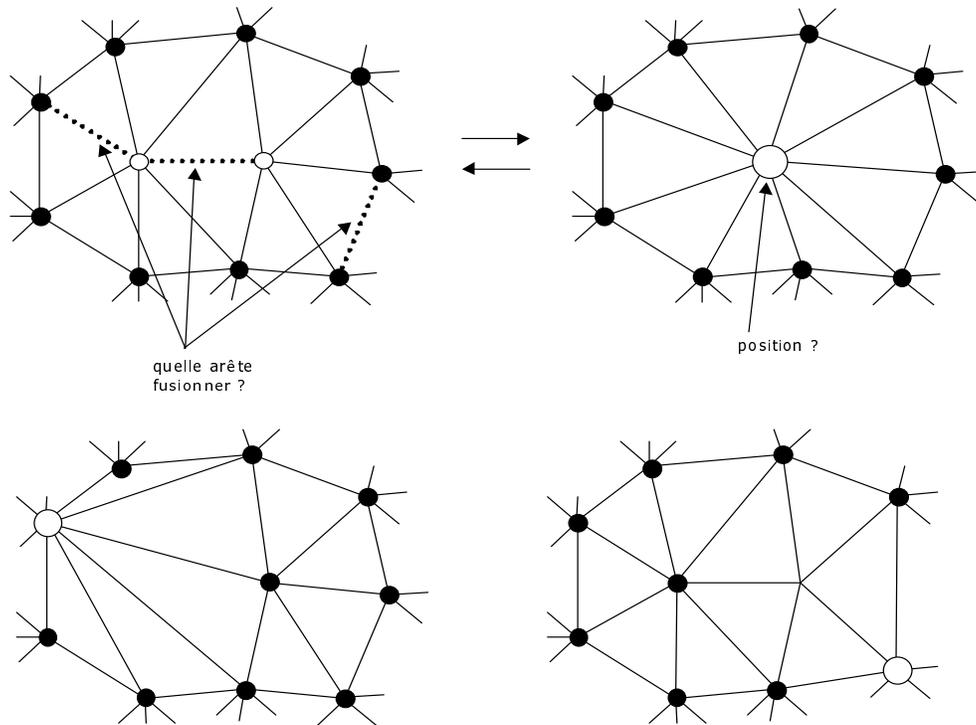


FIG. 2.22 : L'opération élémentaire de décimation choisie correspond à la fusion d'arête. A chaque décimation élémentaire un sommet et une ou deux faces sont supprimées. Cette opération ne nécessite pas de retriangulation, et correspond à une granularité fine de décimation puisque la résolution peut être choisie au sommet près. On obtient donc deux degrés de liberté à exploiter : l'ordre des arêtes à décimer en priorité et la position du sommet résultant de la fusion. Ce positionnement peut correspondre soit à un pré-positionnement avant optimisation, soit à une pseudo-optimisation qui placerait le sommet sur un optimum probable.

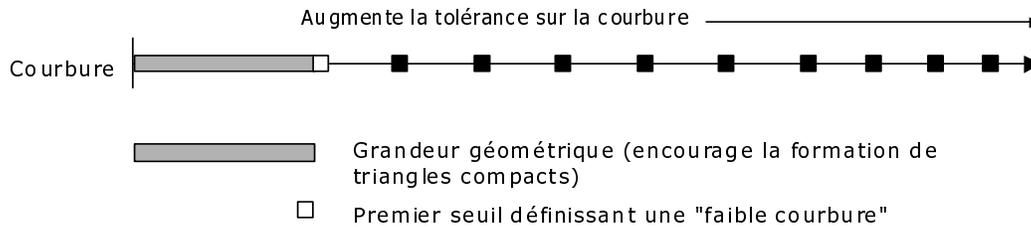


FIG. 2.23 : La queue de priorité combine deux critères organisés en une hiérarchie : la courbure locale et une grandeur géométrique (la longueur d'arêtes dans le cas présent). Au départ un seuil définissant la notion de faible courbure est paramétré, et les arêtes de longueur minimale sont décimées en priorité sur le sous-ensemble correspondant. Au cours du processus de décimation, la tolérance sur la courbure est relâchée afin d'atteindre la complexité géométrique visée. Le critère de longueur d'arête diminue la densité géométrique d'une part, et encourage la formation de triangles compacts sur les régions de faible courbure d'autre part.

ce critère seul ne fournit pas une stabilité numérique satisfaisante lors de la phase d'optimisation précédemment décrite, puisque cette dernière utilise une méthode d'interpolation par éléments finis. C'est pourquoi un autre critère encourageant la conservation de triangles de rapport d'aspect satisfaisants (i.e. compacité maximale) est ajouté. Un critère de compacité pourrait être utilisé, cependant il s'agit de réduire la densité géométrique tout en conservant des éléments de bon aspect. Dans le cas présent il serait préférable de décimer les petits éléments en priorité, puisque leur contribution visuelle est *en moyenne* moins importante après projection à l'écran. Un critère relié à une grandeur géométrique s'impose alors, l'un deux correspondant à la longueur de l'arête candidate à la fusion. Ce critère tend aussi à former des triangles équilatéraux, préservant ainsi leur rapport d'aspect afin de garantir la robustesse de la phase d'optimisation. Les deux critères cités (courbure et rapport d'aspect) sont combinés de la façon suivante : la densité géométrique est réduite sur les régions de faible courbure tout en préservant la compacité des triangles, et la tolérance de courbure est relâchée au cours de la décimation afin d'atteindre la complexité géométrique visée (figure 2.23). La figure 2.24 illustre le comportement du processus de décimation suivant les deux critères.

Une fois l'arête sélectionnée pour la fusion, il reste un degré de liberté essentiel à exploiter : la position du sommet après contraction.

2.4.4 Pré-positionnement

Après la contraction d'arête, le sommet résultant de la fusion est placé sur la position supposée proche de l'optimum. Cette position constitue la solution initiale avant optimisation et permet donc une accélération conséquente de la méthode d'approximation précédemment décrite lorsque cette dernière est utilisée conjointement. Cette phase peut être également vue comme une pseudo-optimisation si aucune optimisation des positions n'est effectuée.

Intuitivement, une fusion d'arête doit entraîner le moins de distorsion possible sur la surface, les cas sensibles intervenant essentiellement sur les coins, les *dards* et les arêtes vives régulières. On introduit maintenant la notion de nombre d'arêtes vives autour d'un sommet afin de différencier les cas : une arête est dite vive lorsque l'angle du dièdre formé par les normales de ses deux faces adjacentes excède un seuil prédéfini. Le nombre d'arêtes vives autour des deux sommets de l'arête vive candidate à la suppression est évalué, puis deux cas en sont déduits :

- les nombres sont différents (figure 2.25, colonne de gauche) : on place le sommet résultant sur le sommet

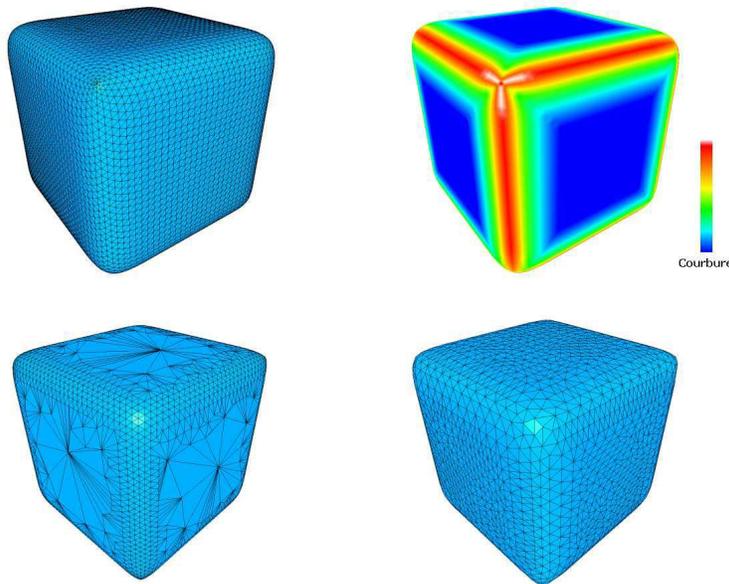


FIG. 2.24 : De gauche à droite et de bas en haut respectivement : le maillage original, la courbure affichée en fausses couleurs, le maillage simplifié uniquement sur les régions de faible courbure, puis le maillage simplifié suivant le critère de longueur d’arête seul.

de plus grand nombre d’arêtes vives. Cette heuristique place les sommets sur les coins, ou sur les arêtes vives lorsqu’une arête est située sur une région plane à une extrémité, et sur une arête vive à l’autre extrémité. La forme est ainsi localement préservée,

- les nombres sont identiques (figure 2.25, colonne de droite) : on place le sommet résultant au milieu de l’arête. Cette heuristique permet de conserver un bon rapport d’aspect des triangles situés soit sur une région plane, soit de part et d’autre d’une arête vive régulière.

L’annexe A résume l’algorithme de simplification sous la forme d’un pseudo-code, et la section suivante illustre les performances de la méthode.

2.4.5 Résultats

La figure 2.26 illustre le processus de simplification appliqué sur le maillage *fandisk*. La troisième image montre une concentration des nœuds sur la région de forte courbure, les arêtes vives régulières et les coins. Cette région est ensuite simplifiée lorsque la contrainte sur la courbure est relaxée par la double queue de priorité. On note que la phase de pré-positionnement place automatiquement les sommets sur les coins et les arêtes vives, jusqu’à un certain niveau de décimation. La figure 2.27 montre une concentration des sommets sur les régions de plus forte courbure, tandis qu’une compacité satisfaisante est obtenue sur les triangles.

2.5 Discussion et perspectives

Dans cette section nous allons résumer chaque technique, décrire sa valeur ajoutée, son intégration dans un schéma global de diffusion de scènes et les évolutions souhaitables.

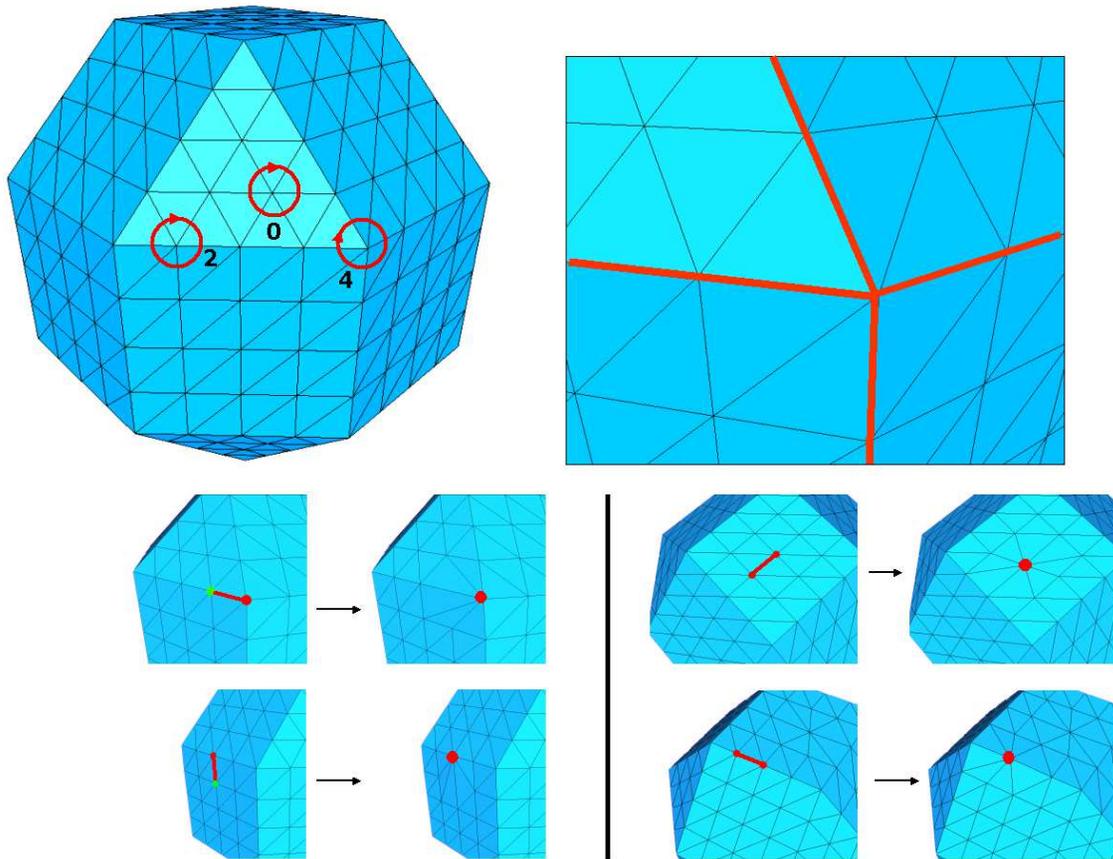


FIG. 2.25 : La phase de pré-positionnement peut être vue comme une sorte de prédiction de la position optimale. Les arêtes vives sont dénombrées autour des sommets d'une arête candidate à la fusion. Si les deux nombres sont différents, le sommet conservé après contraction est placé sur le sommet de plus grand nombre d'arêtes vives (colonne de gauche). Dans le cas inverse, le sommet est placé au milieu de l'arête (colonne de droite). On remarque sur cet exemple que les coins et arêtes vives sont correctement conservés, ainsi que la compacité des triangles situés soit sur une région plane, soit de part et d'autre d'une arête vive régulière.

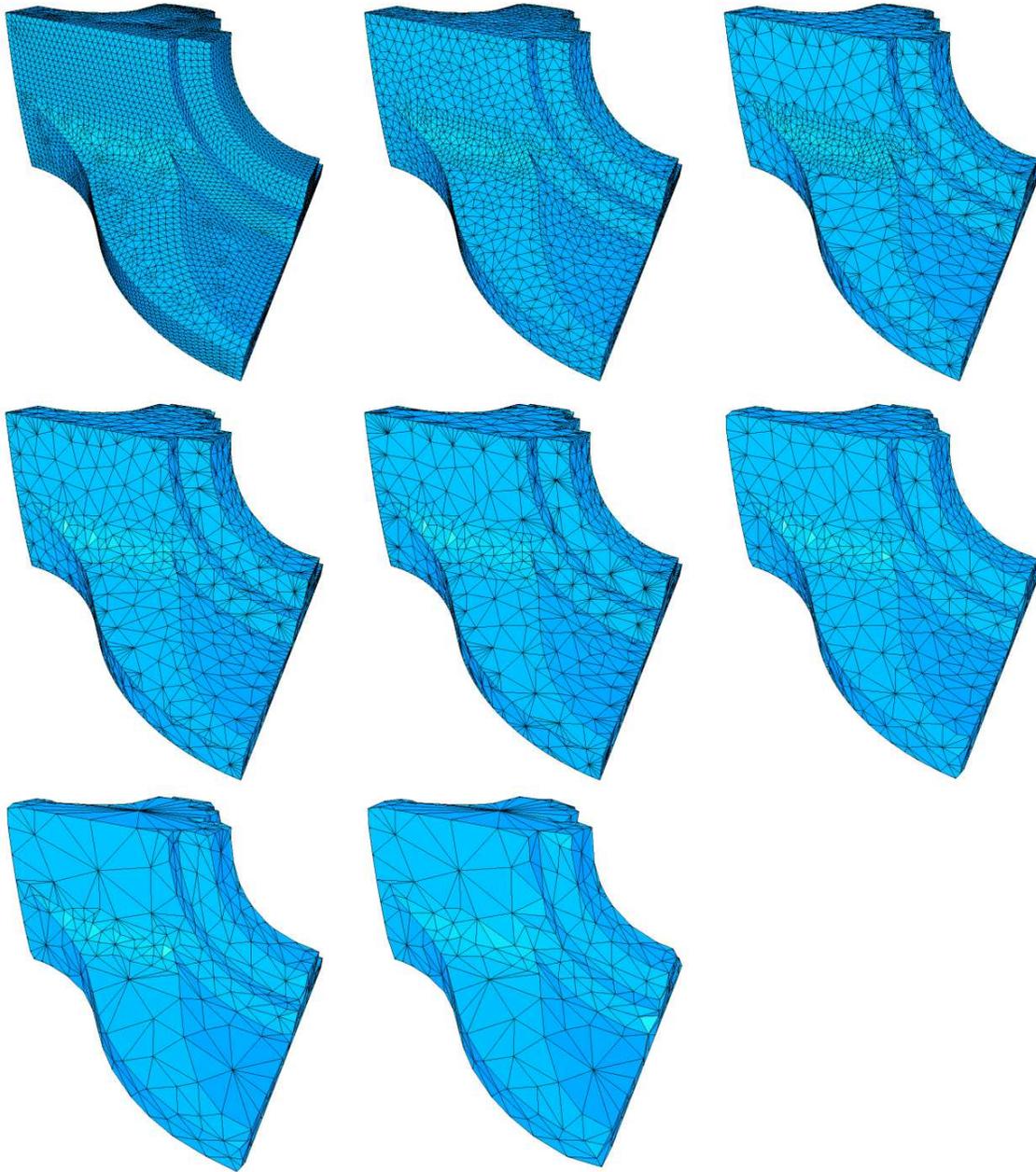


FIG. 2.26 : Décimation du maillage *fandisk*. De gauche à droite, puis de haut en bas : respectivement 12946, 6500, 4000, 3000, 2600, 2000, 1000, et 800 faces. Temps d'exécution : environ 60 secondes sur un PII 266 MHz. L'algorithme est mis en œuvre par vagues successives de décimation en agissant sur le seuil de courbure. Les vagues de décimation retirent itérativement les arêtes de plus faible longueur sur les régions de faible courbure, tout en relâchant progressivement la tolérance en terme de courbure. La contrainte de courbure est ensuite quasi nulle afin d'atteindre la complexité géométrique souhaitée et de décimer le maillage le long des arête vives régulières.

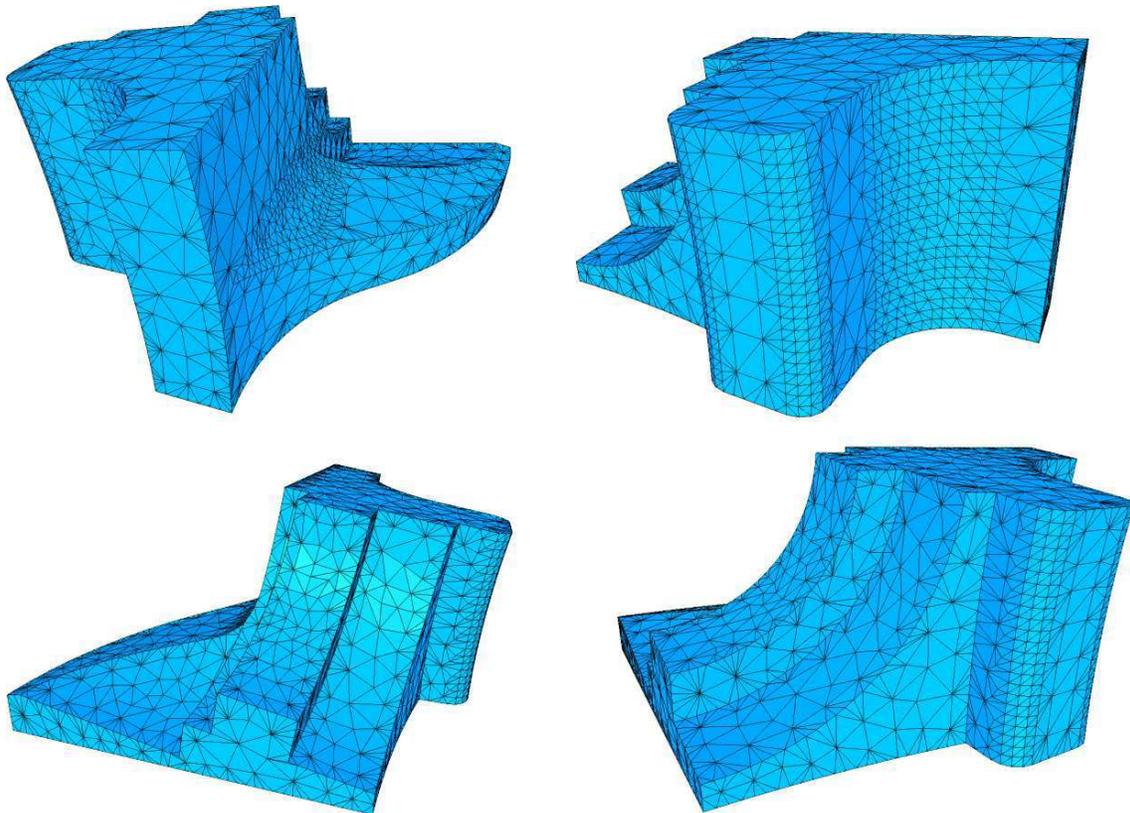


FIG. 2.27 : Détail du maillage *fundisk* au cours de la simplification. On constate que seules les régions de faible courbure sont décimées, sur lesquelles la compacité des triangles est conservée, le critère de longueur d'arête tendant à générer des triangles équilatéraux. Le pré-positionnement des sommets après fusion permet de conserver les coins et de préserver une compacité satisfaisante de part et d'autre des arêtes régulières. Il est à noter que seule une augmentation de la tolérance en terme de courbure permet de décimer par la suite le maillage le long d'une arête vive.

2.5.1 Minimisation de volume

Valeur ajoutée

Dans la section 2.3, nous avons présenté une nouvelle technique d'approximation de maillages tridimensionnels, l'originalité de cette technique provenant du choix de la métrique basée volume, et de l'élaboration du processus d'optimisation des positions associé. Une telle métrique possède naturellement de bonnes qualités d'un point de vue perception visuelle, puisque les discontinuités sont naturellement prises en compte (figure 2.19), et la forme globale est restaurée lorsque la complexité géométrique le permet (figure 2.21). La méthode d'optimisation a fait l'objet d'un dépôt de brevet [ALS98] et de deux publications [ALS98a] [ALSS99].

Intégration

Le processus d'approximation autorise une intégration dans un schéma d'optimisation aussi bien global que local. Dans sa version globale, la méthode peut être considérée comme un post-traitement après toute méthode de décimation. Dans son autre version, le volume est localement minimisé après chaque opération élémentaire de décimation. Quelques contraintes subsistent toutefois, en terme de rapport d'aspect sur les triangles pour assurer une stabilité numérique satisfaisante, et en terme d'orientation relative des surfaces. Enfin, les temps de calcul se comptent en minutes, voire quelques dizaines de minutes sur des maillages complexes (10 000 triangles), avec 100 points échantillonnés par triangle. Une telle optimisation est donc plutôt adaptée à un traitement *off-line* après un processus de décimation appliqué dans le but de supprimer les redondances géométriques.

Evolutions souhaitables

Le noyau dur concerne assurément les temps de calcul, et ce malgré l'implémentation d'une structure de données optimisée définie par Wyvill [WMW86] et quelques heuristiques mises en place afin d'accélérer le calcul des orientations relative des surfaces. Enfin, l'intégration de la permutation d'arête comme opérateur atomique d'optimisation est aussi souhaitable. Enfin, il est raisonnable de penser que la technique semble bien adaptée comme traitement successif à la technique de Borouchaki [Bor99] puisque cette dernière contrôle scrupuleusement la qualité des triangles tout en bornant la déviation géométrique par le biais de la distance de Hausdorff. La minimisation de volume viendrait ainsi compléter le processus de simplification mis en œuvre par l'auteur.

La technique d'approximation doit être couplée à une technique de simplification géométrique, la première méthode consistant à supprimer la redondance géométrique des maillages.

2.5.2 Suppression de la redondance

Valeur ajoutée

La simplification par fusion d'arêtes présentée exploite une double queue de priorité basée sur deux critères (courbure et grandeur géométrique), dont l'un est prioritaire. Cette simplification est associée à une phase de pseudo-optimisation ou de pré-positionnement, suivant que l'on entrelace une phase d'optimisation en position. Cette étape consiste à placer le sommet résultant de la fusion d'une arête sur une position probablement optimale. Il en résulte un processus de décimation efficace (environ une minute pour décimer le maillage *fandisk* de 12 000 à 800 faces). La technique de simplification a fait l'objet d'un dépôt de brevet [AL99b] et de deux publications [ALSS99] [ALS98a] décrivant simultanément la méthode d'optimisation.

Intégration

Ce processus de décimation nous semble bien adapté à un processus *off-line* préalable au codage, et nous le réservons plutôt pour la suppression de la redondance, ou pour ramener la complexité géométrique à un ordre de grandeur coïncidant avec les capacités graphiques des terminaux.

Evolutions souhaitables

Même si la double queue de priorité évite un réglage délicat des pondérations entre les critères, on ne peut éviter une paramétrisation minimale, notamment le premier critère relié à la courbure, puis la séquence de relaxation de ce critère au cours de la décimation. En pratique, ces paramètres sont masqués par le biais de la définition d'une jauge de préférence entre la compacité des triangles et la préservation de la courbure, plus intuitive. L'algorithme de simplification idéal ne nécessiterait aucun réglage, si ce n'est la complexité géométrique visée. Enfin, même si un tel algorithme s'avère bien adapté à la suppression de la redondance, il l'est moins pour une réduction drastique de la complexité. On ne l'utilisera pas pour développer un format progressif pour cette dernière raison, mais aussi car il nécessite une structure de données complète (faces / sommets / arêtes), coûteuse en mémoire et en maintenance algorithmique. Enfin, une amélioration d'une telle technique pourrait concerner le contrôle de la déviation géométrique et de la qualité des éléments telle qu'on peut les trouver dans la technique de Borouchaki [Bor99], ainsi que la prise en compte des trous et bords des maillages.

Chapitre 3

Codage

3.1 Introduction

Dans ce chapitre nous considérons l'étude de maillages dont la redondance géométrique a été supprimée par simplification et approximation. L'intérêt de l'étude consiste à générer un train binaire compact afin de faciliter le stockage et la transmission des données sur des réseaux hétérogènes. Le domaine du codage des données concerne principalement deux finalités :

1. le cryptage, qui représente l'art de coder une information de manière à la rendre incompréhensible de tous, excepté par un destinataire privilégié. Le développement massif des télécommunications développe le sentiment de vulnérabilité des informations gardées secrètes pour des raisons stratégiques. Le cryptage réside donc au cœur même du secret de telles informations et met en œuvre un algorithme de codage complexe utilisant une clé confidentielle afin de les rendre indéchiffrables sans cette dernière.
2. la compression de données qui représente l'art de minimiser le volume de données occupé par une quantité d'information donnée [Faq00]. En ce qui nous concerne, le taux de compression s'exprime ainsi en nombre de bits par polygone. Il s'agit de répondre à la contrainte de stockage et de transmission de scènes tridimensionnelles complexes, le débit d'un réseau étant limité d'une part et soumis à des variations, voire des congestions d'autre part.

On parle de codage ou de compression sans perte lorsque le code est bijectif et permet de retrouver exactement les données originales après décompression. On parle aussi de codage avec pertes lorsque l'algorithme de compression introduit une distorsion entre les données originales et les données reconstruites après décompression. C'est justement cette distorsion, qui se traduit en terme de réduction de la qualité perceptuelle après affichage des scènes sur un périphérique graphique, que nous essaierons de minimiser ou du moins de rendre visuellement indécélable car le codage avec pertes autorise des taux de compression nettement supérieurs au codage sans perte. On peut également introduire des degrés de liberté en terme d'interprétation du train binaire par le biais de la notion d'échelonnabilité, qui se traduit aussi en terme de rendu progressif au cours de la réception des données sur un terminal.

Dans ce chapitre, les sections 3.2 et 3.3 formalisent respectivement les notions de compression et d'échelonnabilité sur les données. Afin de mieux situer notre apport technique, la section 3.4 présente les travaux de différentes équipes ayant travaillé sur le domaine directement concerné ou connexe, et la section 3.5 explicite notre méthode de codage échelonnable, suivie par des études comparatives présentées sous la forme de résultats en section 3.5.4. Enfin, la section 3.7 termine par une discussion présentant les perspectives envisageables.

3.2 La compression des données

Depuis la diversification de l'information diffusée en réseau et les enjeux économiques sous-jacents, les techniques de compression de données font l'objet de très nombreux efforts de recherche [NG96]. Les acteurs du multimédia ont d'ailleurs très bien compris l'aspect vital de la compression pour le type de données qu'ils génèrent (texte, images fixes ou animées, son, etc...). Ce constat nous incite à observer scrupuleusement les réflexes développés par cette communauté afin d'appliquer les mêmes principes pour les données graphiques à véhiculer sur les réseaux. Toutefois, et même si certains grands principes sont systématiquement utilisés aujourd'hui, on peut affirmer que toutes les innovations sont bienvenues. Il suffit de surveiller les publications du domaine afin de se convaincre des capacités d'innovation des acteurs concernés.

La compression des données combine les domaines des mathématiques et de la géométrie : la théorie de l'information, les probabilités, les ondelettes, la géométrie fractale, l'algorithmique, etc... et pour toutes les données à finalité graphique, cette technologie possède un lien avec le système visuel humain. Nous n'essaierons pas de dresser la liste exhaustive des ouvrages et des publications concernant le domaine de la compression de données, mais esquisserons plutôt quelques principes propres aux deux familles de méthode : la compression sans perte (section 3.2.1) ou avec pertes (section 3.2.2).

3.2.1 Compression sans perte

Dans certains cas, il s'avère absolument crucial de reconstruire l'information exacte (texte, images médicales, connectivité d'un maillage, etc...). L'entropie indique alors la borne minimale nécessaire au codage sans perte de l'information [Sha48] [Mac95]. Elle traduit le nombre minimal de bit(s) nécessaire(s) par symbole :

$$\text{entropie} = \sum_1^N p_i \log_2 \frac{1}{p_i}$$

où

- N représente le nombre de symboles à coder,
- et p_i la probabilité du symbole S_i .

Il faut toutefois noter que ce calcul considère la probabilité des symboles pris séparément, et ne tient pas compte des corrélations inter-symboles. La suppression de la redondance correspond à la première étape de la compression sans perte.

Supprimer la redondance

Compresser une information sans perte de façon optimale consiste à en retirer toute la redondance, celle-ci ayant été introduite soit volontairement afin de réduire une certaine ambiguïté ou des erreurs d'interprétation, soit involontairement suivant les modes de représentation utilisés en informatique. La redondance est notamment assez facilement repérable dans le format VRML [vrmb], puisque chaque face adjacente à un sommet S_1 de degré n fait n fois référence à S_1 (Figure 3.1). Cette remarque nous laisse entrevoir l'idée qu'il faudrait idéalement obtenir sur cet exemple : une référence unique par sommet.

Afin de supprimer la redondance dans une séquence de données, diverses approches ont été élaborées :

L'approche statistique

La méthode la plus connue pour la réduction de la redondance correspond certainement au codage de Huffman

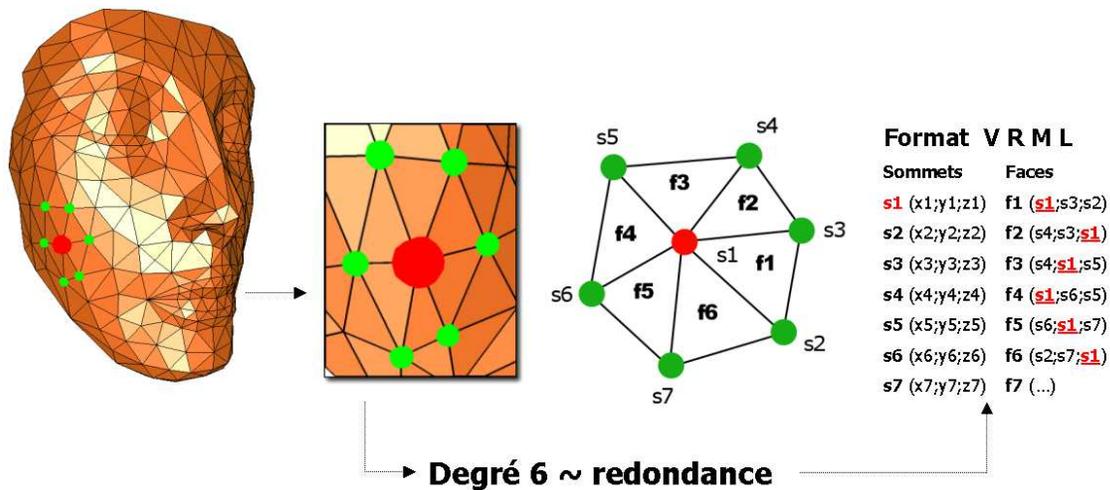


FIG. 3.1 : La redondance principale contenue dans un fichier VRML97. Le nombre 6 correspond à la valence moyenne des sommets d'un maillage triangulaire. Sur cet exemple, chaque face f_i adjacente au sommet s_1 fait référence à ce sommet (souligné à droite). Il en résulte une redondance du même ordre pour la représentation de la connectivité.

[Huf52] développé en 1952, qui génère un code par symbole dont la longueur est inversement proportionnelle à sa probabilité d'apparition. Dans un format non compressé, nous avons l'habitude de stocker sur le même nombre de bits les éléments à coder (8 bits par caractère, 8 ou 24 bits par pixel), or il est parfois plus efficace de générer des codes courts pour les symboles fréquents, et des codes plus longs pour les symboles plus rares. La distribution des symboles agit alors directement sur l'efficacité du codage. La difficulté consiste à générer les codes préfixés correspondants, dont la longueur est contrainte à une valeur entière. Nous verrons que cette contrainte a été ensuite supprimée.

L'approche par dictionnaire statique

Les chercheurs Abraham Lempel et Jacob Ziv [ZL78] ont développé en 1978 une approche basée sur un dictionnaire. L'idée consiste à considérer une séquence de données sous la forme de motifs, et d'observer la répétition de ces derniers. Afin de mieux cerner l'intérêt de la méthode, prenons en exemple la langue française. En théorie il est possible de former un très grand nombre de mots à partir des 26 lettres de l'alphabet (26^{10} pour des mots de 10 lettres, soit environ 140 000 milliards de mots). En pratique il n'existe qu'environ 60 000 mots, donc seul un échantillon des 140 000 milliards de mots pouvant être formés ont un sens. Il serait donc judicieux d'énumérer les mots de la langue française afin d'en dresser un dictionnaire, chaque mot serait ensuite remplacé par un index dans la liste. L'inconvénient d'une telle méthode, réside dans la nécessité de transmettre le dictionnaire avec les données (idem pour Huffman), la taille de ce dernier pouvant s'avérer significative.

L'approche par dictionnaire dynamique

Terry Welch [Wel84] améliore l'algorithme précédent en 1984 en supprimant la contrainte de transmission du dictionnaire. L'algorithme, nommé LZW, utilise un dictionnaire dynamique contenant les motifs de la séquence à compresser. Il en résulte un fonctionnement adaptatif puisqu'il existe un dictionnaire par séquence transmise, ce dernier étant fonction des données précédemment lues. On gagne en souplesse puisque la dimension et l'initialisation du dictionnaire sont autorisés en cours de codage. Le dictionnaire est alors reconstruit au décodeur au fur et à mesure de la transmission des données.

Le codage arithmétique

Le codage arithmétique, développé à partir de 1979 [RL79] [Ris83] [WNC87], code également un symbole en fonction de sa probabilité d'occurrence, mais sur une longueur de bits fractionnaire, ce qui lui confère un avantage indéniable par rapport au codage de Huffman. Le gain en terme de taux de compression est sensiblement amélioré. Le principe d'un tel codeur est très astucieux, il consiste à coder une séquence de symboles avec un unique nombre réel contenu dans l'intervalle ouvert $[0; 1[$. Cet intervalle est successivement subdivisé à chaque apparition d'un symbole S_i et en fonction de sa probabilité d'occurrence p_i . Après le codage de la séquence on obtient un sous-intervalle de $[0; 1[$ qui permet une reconstruction sans perte de la séquence initiale. Chaque élément S_i de probabilité p_i est ainsi codé sur $\log_2(\frac{1}{p_i})$ bit(s), plus une quantité négligeable devant $\log_2(\frac{1}{p_i})$. Telle que présentée ici, la méthode nécessite la connaissance statistique des symboles, toutefois il en existe une variante avec un modèle adaptatif. Cette dernière initialise le codeur avec un modèle d'équiprobabilité (ou de probabilité moyenne) et met à jour ce dernier au fur et à mesure du codage, bien entendu le même principe est appliqué au cours du décodage. On verra par la suite que cette dernière version a principalement fait l'objet de notre choix [MNW95] [Wit97].

La famille des codeurs entropiques autorise une combinaison avec le codage par plages lorsque le signal s'y prête.

Repérer les plages de valeurs

Si le signal à coder contient des plages de valeurs ou des motifs identiques, il devient rentable d'utiliser le codage RLE (Run-Length Encoding) qui préfixe chaque symbole de son nombre d'occurrences. Le codage *Stack-Run* [TVC96] entrelace le codage par plages et le codage entropique. Il autorise notamment un gain sensible en vitesse lors du décodage.

Toutes les méthodes précédemment citées permettent une restitution parfaite des données, mais il est parfois plus intéressant d'autoriser une légère distorsion afin d'obtenir des ratios de compression plus élevés.

3.2.2 Compression avec pertes

La compression avec pertes permet d'obtenir des taux de compression élevés au prix d'une distorsion : l'enjeu consistant à optimiser le compromis débit / distorsion. Cette technique est couramment utilisée dans le domaine de la compression des images : nous pouvons accepter la dégradation d'une image si elle est indiscernable à l'œil ou suffisamment faible pour être visuellement acceptable, en contrepartie d'un taux de compression bien plus intéressant. Ce même principe peut être indirectement appliqué aux images produites par un moteur de réalité virtuelle.

La première technique concerne la réduction de la précision.

Réduire la précision

Suivant les applications, la précision des données originales n'est pas toujours requise. Dans ce cas, on peut alors procéder à l'équivalent d'un arrondi sur les valeurs par l'intermédiaire d'une phase de *quantification*. A titre d'exemple, le format VRML [vrmb] stocke les coordonnées des sommets dans le format IEEE flottant sur 32 bits, et fournit une précision de $3.4e^{-38}$ à $3.4e^{+38}$. Cette précision n'est pas indispensable pour les applications de visualisation. On s'autorise ainsi à arrondir les valeurs des positions pour les raisons suivantes :

- les périphériques graphiques ont une précision finie, il est inutile de définir une précision inférieure à l'erreur projetée sur l'écran ou dans un casque de réalité virtuelle,
- la vision humaine ne discerne plus au delà d'une certaine précision,
- et les objets sont rarement modélisés avec deux ordres de longueur extrêmes, afin d'éviter l'effet de *masquage* (quid du micron sur un objet de plusieurs mètres?).

Par contre, les applications de calcul scientifique ou de conception assistée par ordinateur requièrent une excellente précision, voire la précision originale pour une certaine classe de calculs (comme l'étude du comportement mécanique au niveau des discontinuités). Cette constatation justifie d'autant plus le développement de méthodes de codage échelonnable. Il est à noter que le réglage des niveaux de quantification fait l'objet d'un soin tout particulier dans les normes de codage actuelles de l'information audiovisuelle [mpea].

Redistribuer l'énergie

Cependant, avant de procéder à une quantification, il peut être judicieux de transformer le mode de représentation des données. Le modèle de perception visuelle dans le domaine transformé permet ensuite de déterminer les niveaux de quantification autorisés. Dans tous les cas, la transformation doit être réversible, la plus connue d'entre elles correspond à la DCT (Discrete Cosinus Transform) pour le codage des images.

Décrivons maintenant la notion d'échelonnabilité, qui représente la clé de voûte de l'adaptation aux infrastructures de télécommunications.

3.3 La notion d'échelonnabilité

Cette notion, aussi définie par l'anglicisme *scalabilité* ou représentation multi-échelle, est pratiquement devenue aujourd'hui une technologie à part entière puisqu'elle s'avère incontournable aux yeux des opérateurs de télécommunications. Endossons maintenant le rôle d'un opérateur afin de mieux comprendre pourquoi la scalabilité a atteint une telle importance dans le domaine des télécommunications.

3.3.1 Du point de vue d'un opérateur

L'univers se réduit alors à un ensemble de sources, de canaux et de récepteurs. Ces éléments sont hétérogènes car les maillons de cette chaîne télécommunicante sont inégaux en terme de capacité de stockage et de débit, de plus les utilisateurs n'expriment pas toujours les mêmes besoins. Il est ainsi impossible de prévoir très précisément le nombre d'utilisateurs connectés simultanément, ni leurs velléités en terme de consultation de services à chaque instant. Enfin, les opérateurs de télécommunications doivent cohabiter, ce qui augmente encore les difficultés. Malgré ces contraintes, l'objectif prioritaire (et même le *métier*) consiste à satisfaire les utilisateurs, et donc à acheminer en temps voulu par le réseau une information attractive, et surtout réellement exploitable ; le terme *exploitable* signifiant ici un type et une quantité de données correspondant aux capacités locales du terminal en terme :

- de stockage : le volume de données peut dépasser l'espace disque ou la mémoire du terminal, surtout après décompression,
- de traitement : la quantité de données réceptionnée nécessite suivant le type d'information des capacités de calcul supérieures aux performances parfois très limitées d'un terminal mobile. La phase de décompression requiert à elle seule une série de calculs parfois non négligeable,

- d’affichage : les capacités d’affichage pour un ordinateur graphique s’expriment en résolution et en nombre d’images par seconde en ce qui concerne des images fixes ou animées, et en nombre de polygones par seconde pour des scènes tridimensionnelles (quelques millions aujourd’hui). Un polygone nécessite en effet un certain nombre de calculs (positionnement dans l’espace, masquage, projection à l’écran, éclairage, coloration, placage de texture, transparence, etc...) avant d’être affiché.

La scalabilité possède un lien avec la progressivité. A titre d’exemple, le succès du format JPEG progressif développé pour les images fixes de l’Internet résume à lui seul l’intérêt de la scalabilité obtenue sur des images, puisque l’utilisateur peut visualiser très tôt au cours de la réception une image approximative. Un tel format permet ainsi de naviguer plus efficacement dans l’océan d’information que représente la toile.

Après avoir défini la notion d’échelonnabilité, regardons maintenant à quels niveaux celle-ci doit être obtenue.

3.3.2 Différents types de scalabilité

L’échelonnabilité peut être obtenue lors de :

- la représentation : soit une source de données unique (un train binaire), la représentation échelonnée permet la restitution partielle ou progressive de l’information. Idéalement, on devrait pouvoir reconstruire une information exploitable à partir de toute partie tronquée du train binaire, même si en pratique l’on génère un niveau de base et une certaine granularité (il s’avère impossible de reconstruire une information exploitable dès le premier bit). L’innovation des données (au sens de théorie de l’information) doit être organisée suivant un ordre décroissant (figure 3.2), et la finesse du grain fait partie des critères d’évaluation,
- la transmission : en plus des problèmes de congestion et de réduction de bande passante par le nombre de flots de données, un même réseau ne définit pas un niveau unique de ressource en terme de débit, mais organise une hiérarchie de qualités et de priorités de services. La diffusion multicast¹, les services à débit garanti et la notion de qualité de service sont quelques exemples de la tendance actuelle : de plus en plus d’intelligence est injectée au niveau des réseaux. L’enjeu consiste alors à déporter l’intelligence sur le réseau² et sur le terminal. Concrètement, et dans un contexte de diffusion multicast, une passerelle³ peut être amenée à tronquer les trains binaires afin de satisfaire des contraintes de débit et/ou de qualité de service,
- et la réception : la qualité de service visée, le terminal cible et ses capacités de traitement et de restitution sont hautement hétérogènes. Il en résulte la nécessité d’interpréter un même flot de données différemment en fonction des contraintes énumérées. Il peut s’agir de scalabilité spatiale (images et vidéo), temporelle (vidéo et 3D), géométrique (3D) et même de scalabilité en terme de complexité algorithmique ; aussi commence-t-on à voir apparaître des algorithmes échelonnés en complexité lors du codage des images fixes par exemple, avec une application sur les appareils de photographie numérique. On peut ainsi choisir un mode *rafale* qui compresse une séquence d’images à une cadence élevée au détriment du taux de compression, tandis que le mode *cliché unique* autorise une méthode de compression plus complexe pour une qualité accrue.

¹la diffusion multicast définit la notion de partage de la bande passante pour des données synchronisées à destination de récepteurs multiples et variés.

²"l’ordinateur, c’est le réseau", SUN microsystems©

³une passerelle est un périphérique utilisé afin de connecter des réseaux de différents types, tels que ceux utilisant des architectures ou protocoles différents. Ils fonctionnent au niveau de la couche réseau, déterminent le chemin le plus approprié pour l’acheminement des paquets de données et filtrent le trafic de diffusion au niveau local.

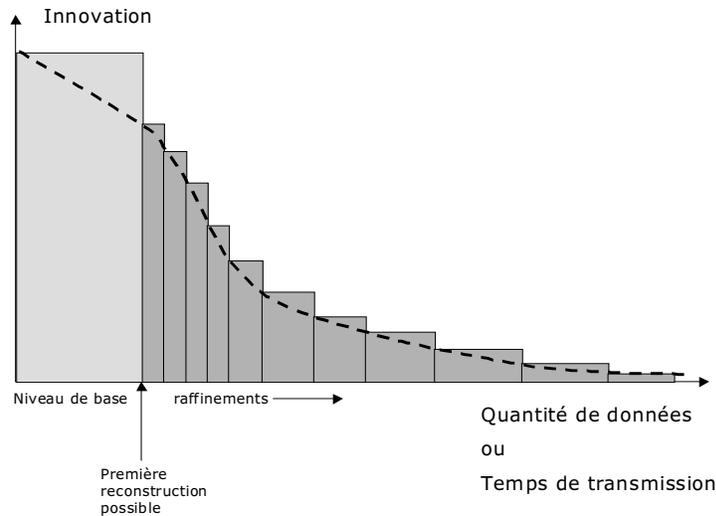


FIG. 3.2 : Dans un train binaire échelonnable bien conçu, les données sont organisées suivant l'ordre décroissant de l'innovation. Idéalement, on devrait ainsi obtenir la meilleure qualité perceptuelle pour une certaine quantité de données, correspondant au fameux *ratio débit / distorsion*.

La notion d'échelonnabilité est finalement illustrée par la figure 3.3. Essayons maintenant d'esquisser les premières incidences sur les données contenues dans les scènes 3D.

3.3.3 Les données 3D

L'obtention de l'échelonnabilité sur les maillages tridimensionnels n'est pas directe, et les problèmes de multiplexage et de raffinement continu constituent en pratique les noyaux durs de la problématique. Plus intuitivement, la notion de multiplexage correspond à l'art d'entrelacer des données de différents types tandis que le raffinement continu concerne essentiellement la progressivité d'une transition entre deux niveaux de détail différents lors de l'affichage.

Le multiplexage

Les maillages étant principalement constitués de sommets (la géométrie), de polygones (la connectivité) et d'attributs, la scalabilité de représentation consiste donc à générer en premier lieu un niveau de base de faible complexité géométrique avec peu d'attributs, puis des raffinements sur ces mêmes données. On commence à percevoir l'embryon de la difficulté : quel type de données faut-il insérer après le niveau de base pour respecter la courbe d'innovation décroissante de la figure 3.2 ? plutôt du raffinement géométrique ? (c'est à dire des positions plus précises), de la connectivité ? (de nouveaux triangles), des attributs ? (couleurs, normales, matériaux plus précis). Cette interrogation représente le problème cœur du *multiplexage* de données, que l'on retrouve par exemple en codage des images fixes : entre la précision des couleurs, la résolution, et le contenu fréquentiel. De plus, il est à noter que ces données sont souvent fortement corrélées.

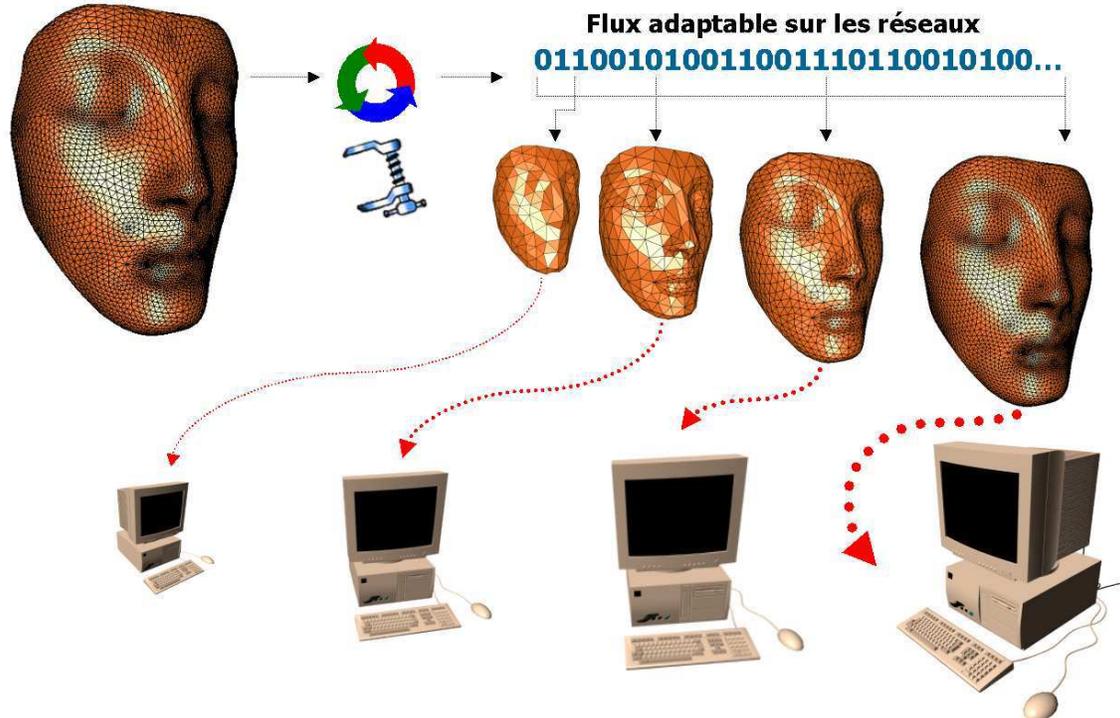


FIG. 3.3 : L'enjeu consiste à transformer un objet 3D sous la forme d'un flux adaptable sur les réseaux. Idéalement, une image peut être reconstruite en tronquant une partie du train binaire. La scalabilité autorise ainsi la double adaptation réseau / terminal.

Le raffinement continu

Un autre critère essentiel en scalabilité concerne l'incidence du raffinement d'une information sur la perception visuelle. Idéalement la transition entre deux niveaux de qualité doit être quasiment imperceptible, afin d'aboutir à un résultat visuel progressivement amélioré. On parle alors de raffinement continu, et le choix des opérations atomiques de raffinement ainsi qu'un certain nombre de principes liés à l'interpolation sont cruciaux pour la contrainte de continuité visuelle.

3.4 Travaux précédents

Le développement vertigineux des machines, des réseaux et des accélérateurs graphiques permet d'envisager depuis peu la réalité virtuelle diffusée pour tous publics à une cadence interactive. De par son côté extrêmement attractif et les perspectives d'applications qu'elle promet, la 3D a d'abord provoqué l'euphorie des équipes de recherche, avant que celle-ci se propage vers le grand public, en passant par les industriels qui ont rapidement discerné le formidable moyen de simulation dont ils pouvaient disposer.

Examinons tout d'abord le problème des formats de données 3D, puis les acteurs du domaine avant de décrire succinctement les solutions proposées par les équipes de recherche.

3.4.1 Les formats de données

Il existe aujourd'hui un nombre impressionnant de formats liés à la réalité virtuelle [wot], chacun possédant son domaine de prédilection :

- VRML : Virtual Reality Modeling Language [vrml] : description de scènes tridimensionnelles et navigation,
- 3DS, MAX : 3D Studio Max : logiciel de modélisation et de rendu d'images de synthèses,
- OBJ Alias Wavefront, modéleur et renderer⁴,
- IV : Open Inventor, Silicon Graphics,
- 3DMF : 3D Metafile Format,
- LIGHTWAVE : modéleur et renderer,
- POV : Persistence Of Vision : lancer de rayon,
- RENDERMAN : renderer,
- PDS : Planetary Data System format,
- IMAGINE : objets et textures,
- MSDL : Manchester Scene Description Language.

Cette pléthore de formats est compréhensible. En effet, la diversité des données contenues dans une scène 3D est à l'égal de l'univers à simuler. On trouve pratiquement un format par application visée ou par entreprise du domaine.

3.4.2 Les industriels

La génération et la structuration de telles données font partie intégrante du savoir-faire d'une entreprise, c'est pourquoi nombres de ces formats sont non documentés, on parle alors de format propriétaire (citons MetaStream pour le plus récent [met00] [AdRL⁺99]).

⁴anglicisme pour désigner un logiciel de rendu

3.4.3 Les consortiums

Il ressort toutefois qu'il peut devenir extrêmement fructueux de proposer un format ouvert et puissant de description de scène, puisque c'est un critère indispensable à une large diffusion. Aussi les entreprises et les universités se réunissent régulièrement pour proposer des solutions susceptibles d'aboutir à une norme.

Le langage VRML [vrml] est le fruit d'un consortium d'acteurs de l'infographie, il s'agit certainement du format le plus répandu sur le réseau. La plupart des navigateurs Internet sont capables d'interpréter et d'afficher ce format via des *plugins*⁵. Le consortium s'appelle aujourd'hui Web3D.

La norme MPEG [mpea] se décline en 6 versions, correspondant chacune à une famille d'applications. MPEG4 [mpeb] est certainement la plus intéressante en ce qui nous concerne puisqu'elle représente aujourd'hui le standard officiel pour les applications multimédia. La version 2 a été approuvée en décembre 1999, et les versions 3 et 4 sont actuellement dans la phase d'élaboration.

La concurrence entre les partisans de la normalisation et ceux du format propriétaire engendre des transformations fulgurantes du panorama. Cependant, intéressons-nous plutôt aux innovations techniques provenant de la course au *format ultime*.

3.4.4 Compression de scènes tridimensionnelles

Dans cette section, les données sont supposées correctement structurées et représentées. L'enjeu consiste alors à réduire le volume de données occupé par les maillages, et à organiser ces données sous une forme échelonnée, représentant le cœur de notre problématique.

VRML et d'autres formats sont écrits dans un format ASCII, et véhiculent de ce fait une quantité surabondante de données. Le format binaire BIFS⁶ [bif] correspond notamment à une alternative binaire intéressante au format VRML puisque ce format tient compte des excursions des données et utilise des codes de longueur variable. On pourrait également s'intéresser aux méthodes génériques de compression de données (zip, lha, lzh, winzip), mais il est notoirement plus efficace d'observer scrupuleusement la nature des données. Une scène étant principalement constituée d'objets, la majeure partie des données réside dans leur description en terme de connectivité, de géométrie et d'attributs d'apparence. Cette revue classe les méthodes suivant le principe employé pour le codage de la connectivité, et ne concerne que les maillages triangulaires. Mentionnons le cours dispensé au *Siggraph* en 1998, constituant une excellente référence [TRD⁺98].

La tendance

La connectivité d'un maillage constitue une partie très volumineuse d'un format ASCII ou binaire brut⁷. De plus, les gains en compression peuvent couramment atteindre un facteur 50 sur ces données. La majeure partie des auteurs exploitent alors l'ordre fourni par le codeur de la connectivité pour compresser les positions des maillages. Des taux de compression importants sont ainsi obtenus par l'exploitation de la corrélation spatiale. La technique la plus répandue concerne le codage par différences, associé à une étape de quantification, et il s'avère judicieux de le combiner à une méthode de prédiction.

⁵L'anglicisme *plugin* désigne une brique logicielle additionnée à un navigateur Internet, ce dernier représentant à l'heure actuelle l'interface homme-machine universelle.

⁶MPEG4 Binary Format for Scene Description.

⁷Un format brut décrit les données linéairement sans se soucier des problèmes de redondance.

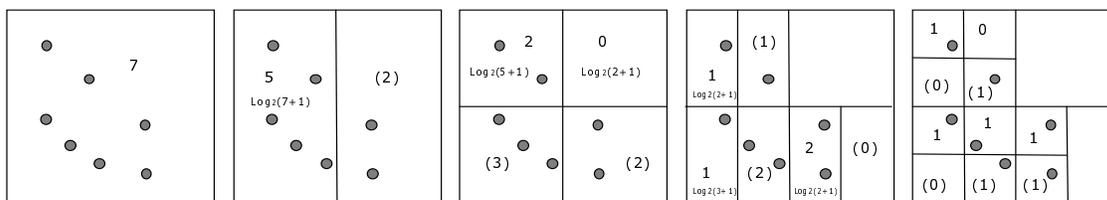


FIG. 3.4 : Un exemple bidimensionnel de codage des positions : la succession du nombre de points compris dans les sous-ensembles est transmise par le codeur, sur une longueur exprimée en nombre de bits. Les occurrences déductibles (donc non transmises) sont écrites entre parenthèses. Le code échelonnable résultant s'exprime sous la forme de la séquence suivante : $\{7,5,2,0,1,1,2,1,0,1,1,1\}$, chaque symbole étant codé sur un nombre de bits décroissant.

Un coût optimal

Certains auteurs [Ame99] [DG99a] [DG99b] [KPhyJgC99] s'affranchissent tout simplement du codage de la connectivité puisqu'ils font l'hypothèse d'une densité spatiale élevée sur les sommets. La connectivité des sommets n'est ainsi pas codée et on fait l'hypothèse que cette dernière est reconstruite par triangulation de Delaunay [GB97] [Dev98] [Soh99] ou par une méthode de reconstruction de surface plus sophistiquée comme le *Crust* [Ame99] ou la notion de *voisinage naturel* défini par Boissonnat et Cazals [eFC00]. L'ordre des sommets est exploité afin de comprimer leurs coordonnées, tandis que la topologie sera reconstruite à partir de ces dernières. Notons toutefois que la triangulation de Delaunay dépend étroitement de la géométrie, ces méthodes sont donc souvent réservées à la triangulation des maillages de terrains.

La méthode de codage des positions de Devillers et Gandoïn [DG99a] [DG99b] est particulièrement bien adaptée aux modèles numériques de terrain. L'algorithme fait astucieusement l'économie du codage de l'ordre sur les sommets. Un exemple 1D a été utilisé afin de faciliter la compréhension de ce dernier : soit S un ensemble de points situés sur un segment de droite, entre 0 et 2^b (coordonnées codées sur b bits). L'algorithme génère un flux binaire échelonnable de la façon suivante : le nombre total de sommets (sur 32 bits par exemple) est généré, puis le segment courant est récursivement subdivisé en deux demi-segments et l'on code seulement le nombre de sommets situés sur un seul des deux segments (celui de gauche par exemple), sur un nombre de bits optimal. Si le segment courant contient p points, le nombre de points dans le segment sera codé sur $\log_2(p+1)$ bits. Seuls les segments non-vides sont subdivisés, et l'algorithme termine lorsqu'il n'y a plus de segments supérieurs à 1. On vient ainsi encadrer progressivement les sommets par subdivision récursive, en n'encodant que le nombre de sommets relevés sur les segments. Le décodeur redispose ensuite les sommets sur les segments (régulièrement ou plus astucieusement), la méthode pouvant être étendue à toutes les dimensions (la figure 3.4 illustre le principe bidimensionnel). Il pourrait être ainsi possible de coder les attributs d'apparences, et une méthode de prédiction basée sur les occurrences des sommets peut en être directement adaptée. Nous venons de décrire un exemple de scalabilité à grain fin non basée sur des plans de bits.

En pratique, on s'intéresse surtout aux méthodes qui garantissent une reconstruction sans perte de la connectivité, la famille des méthodes la plus répandue concerne le découpage en bandes.

Le découpage en bandes

Un maillage constitue en fait un tissage de la surface à modéliser, dont les mailles sont triangulaires. De nombreux auteurs exploitent donc logiquement cette organisation afin de réduire le nombre de références aux sommets. L'objectif consiste alors à découper le maillage en bandes de triangles [XHM99] afin d'optimiser soit le coût de codage, soit l'accès rapide aux données par un bus graphique [Hop99b].

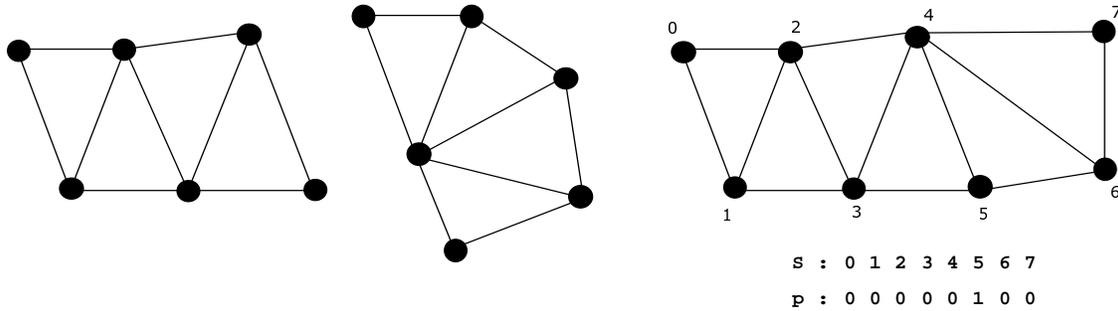


FIG. 3.5 : De gauche à droite et respectivement : une bande de triangles (triangle strip), une bande étoilée (triangle fan) et une bande de triangles généralisée pour le codage : les sommets sont référencés dans l'ordre de parcours sur le graphe, tandis qu'une chaîne de bits de permutation organise le maillage des sommets.

Deering [Dee95] commence par représenter un maillage sous une forme linéaire (*generalized triangle strip*), c'est à dire un ensemble de bandes de triangles générés par chaînage en zigzag des sommets de proche en proche (figure 3.5). Un buffer codé sur 4 bits stocke les 16 sommets précédemment codés, afin de réduire sensiblement les excursions des valeurs de référence aux sommets dans la plupart des cas. L'auteur fait ensuite appel au codage par différence, à la quantification et au codage de Huffman pour les positions. En 1998, Deering [Dee98] a intégré sa méthode après amélioration dans l'API JAVA 3D.

Gumhold et Strasser [GS98] utilisent également un buffer de sommets, tandis que Bar-Yehuda et Gotsman [BYG96] ont réalisé une étude approfondie de l'utilisation de buffers d'indices, afin d'aboutir à la conclusion suivante : Si n représente le nombre de sommets d'un maillage, alors un buffer de taille $12.72\sqrt{n}$ est optimal pour obtenir une référence unique à chaque sommet. C'est d'une façon générale une méthode bien adaptée aux contraintes matérielles d'un accélérateur graphique.

Taubin et al [THLR97] construisent une structure composée principalement de deux arbres de faces et de sommets entrelacés pour coder la connectivité. Les maillages sont en fait *pelés* comme des oranges (figure 3.6), puis les auteurs codent les *épluchures* suivant des séquences de marche sur les triangles, associées à un arbre de sommets. On obtient ainsi 1 bit/face pour des maillages très réguliers. Les auteurs combinent la prédiction, la quantification et le codage entropique pour les positions. Une extension échelonnable a été introduite par ces mêmes auteurs en 1998 [TGHL98], où un maillage est représenté par un modèle de base et une forêt d'insertions de sommets. Lors du raffinement, le maillage est découpé suivant les arêtes le long des arbres de la forêt, les sommets originaux sont écartés et les trous résultants sont retriangulés afin de former un niveau de résolution plus élevé. Cette méthode a fait l'objet d'une soumission à MPEG4 [Tau98], et un papier de Guézic et al [GTLH97] permet de rendre des maillages *manifold* à partir de surfaces *non-manifold*, cette contrainte étant imposée pour de nombreux algorithmes de codage de la connectivité. Les derniers travaux de Guézic et al [GBTS99] permettent ainsi la compression de maillages quelconques.

Rossignac [Ros99] garantit un coût de codage de la connectivité à moins de 2 bits/face pour tout maillage homéomorphe à une sphère, et légèrement supérieur pour les maillages comprenant des trous et des bords. La connectivité étant compressée indépendamment des positions, l'auteur propose à chacun d'utiliser sa propre méthode de prédiction. Une structure de demi-arêtes est utilisée, et le maillage est parcouru de triangle en triangle tout en générant 5 codes d'opération.

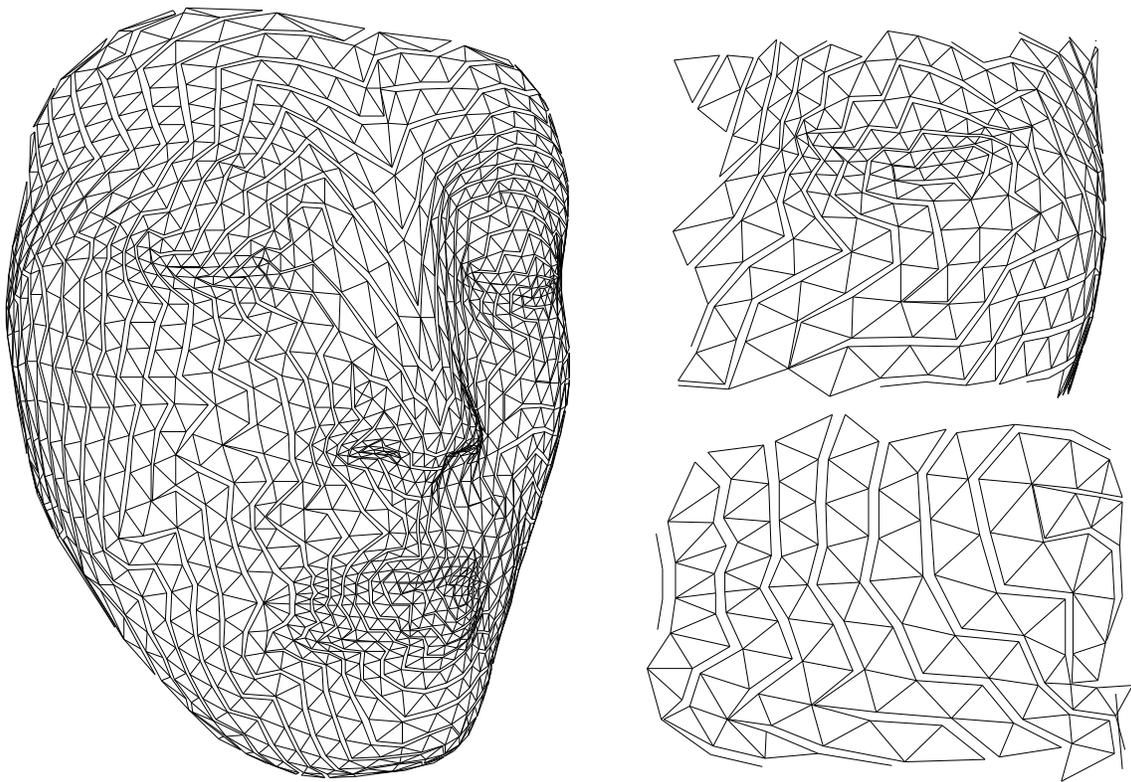


FIG. 3.6 : Un exemple de découpage du maillage *nefertiti* en bandes de triangles. La connectivité représente alors la triangulation des sommets sur chaque bande, la marche en zigzag correspondant au maillage par défaut.

Le partitionnement en couches

Bajaj et al [BCPZ99][BPZ99] partitionnent les maillages en couches de sommets et en bandes de faces. Chaque couche est ensuite encodée indépendamment afin de résister aux erreurs de transmission, ainsi une erreur sera confinée à la région de la couche concernée. Les positions sont codées par prédiction, quantification et codage entropique. Les normales sont indirectement prédites à partir de la géométrie, tandis que les attributs de couleur et les coordonnées de texture sont codées par prédiction suivie d'une quantification. L'application principale concerne la diffusion de fichiers VRML sur Internet.

Exploitation des valences

Touma et Gotsman [TG98] ont proposé en 1998 une méthode de codage exploitant la distribution des valences des sommets d'un maillage. Une conquête sur les arêtes est effectuée à partir de sommets pivots successifs, générant ainsi des codes de valence compressés par codage entropique. Les auteurs obtiennent ainsi des taux de compression allant jusqu'à 0.2 bit/face pour la connectivité. Les coordonnées des sommets sont compressées en utilisant une méthode linéaire unique de prédiction, s'appuyant sur la quasi-coplanarité supposée de 4 sommets successifs dans la liste des sommets réordonnés par le codeur de topologie. Les résidus sont ensuite quantifiés et associés à une table de 32 codes de longueur variable. La courbure de la surface est également prise en compte pour la prédiction, ce qui nécessite l'entrelacement connectivité / positions lors du codage. La méthode de Isenburg et Snoeyink [IS99] possède des similarités avec la méthode de Touma et Gotsman, mais réduit progressivement le maillage en un point via un jeu de transformations élémentaires essentiellement basées sur la fusion des arêtes.

Le graphe dual

Li et Kuo [LK98b] utilisent le graphe dual d'un maillage pour coder sa connectivité. Ils représentent un tel graphe sous la forme d'un arbre binaire dégradé. Chaque nœud du graphe est classé en deux types de cellule : un point de fusion ou une branche, et la longueur de la boucle adjacente à chaque nœud est fortement corrélée à sa probabilité de correspondre à l'un ou l'autre des types de nœud. Cette corrélation est exploitée afin d'encoder la connectivité avec 1 bit par face en moyenne. Les positions sont encodées progressivement par prédiction locale, quantification successive et codage entropique.

Décomposition en ondelettes

Les méthodes de représentation échelonnée basées ondelettes [SGW98] [VTK⁺99] se prêtent à un codage efficace puisque la distribution des coefficients d'ondelettes présente la formation d'un pic autour de la valeur zéro. La plupart des méthodes utilisent une organisation hiérarchique emboîtée à partir d'un maillage de base irrégulier. Le modèle résultant est ainsi appelé maillage semi-régulier. Toutefois, Valette et al [VTK⁺99] proposent de bâtir une décomposition en ondelettes par subdivision irrégulière, modulo quelques permutations d'arêtes mémorisées pour être restaurées après décodage du niveau le plus fin.

Le concept de *maillages normaux* récemment introduits par Guskov et al [GVSS00] [KG00] [KSS00] constitue le niveau d'exploitation le plus abouti de la décomposition en ondelettes pour le codage progressif des maillages. Un maillage normal consiste en une représentation multirésolution composée d'un maillage de base généré suivant la technique de Garland et Heckbert [GH97] et d'une séquence de raffinements (les détails) obtenus par subdivision régulière emboîtée. La technique suppose que tous les détails s'expriment suivant la direction normale à la surface, la géométrie étant stockée sous la forme d'un scalaire par sommet. Une telle représentation se prête idéalement au codage progressif dès lors que l'on s'autorise un codage avec pertes de la connectivité. Après l'étude de la paramétrisation MAPS [LSS⁺98], Khodakovsky et Guskov [KG00] constatent qu'une représentation par maillage comprend trois composantes : la géométrie, la connectivité et la paramétrisation. Les deux dernières composantes ayant une influence non prépondérante pour la définition

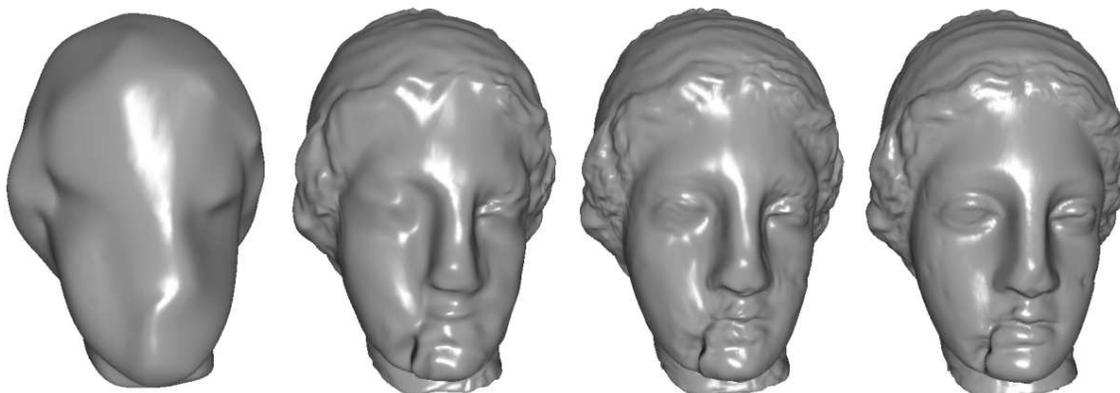


FIG. 3.7 : Résultat de la compression géométrique progressive de Khodakovsky et Guskov [KG00]. Les ondelettes de Loop [Loo87] sont utilisées pour la reconstruction, ce qui explique la continuité obtenue sur le niveau de base.

de la géométrie, seule la composante normale à la surface est exploitée pour coder cette dernière. Le codeur progressif requiert ainsi trois étapes :

1. un remaillage après simplification [GH97] afin d'obtenir une structure régulière emboîtée coïncidant avec la définition d'un maillage normal (voir [GVSS00]),
2. une transformée par ondelettes de Loop [Loo87] pour obtenir des données incitant au codage entropique,
3. un codage par arbre de zéros (*zerotree*) combiné à un codeur entropique à modèle adaptatif afin d'élaborer un format échelonnable sur la géométrie,
4. et un codage par conquête d'arêtes et exploitation des valences inspiré de Touma et Gotsman [TG98] afin de coder la connectivité et les positions du maillage de base.

Enfin, l'algorithme exploite une reconstruction adaptative sur le terminal par subdivision combinée au filtrage utilisé par la transformée en ondelettes, le tout piloté par un critère relié à la courbure mesurée localement sur la surface. La figure 3.7 illustre une reconstruction progressive de la surface *venus* codée par ondelettes de Loop [Loo87].

La scalabilité générique

Li et Kuo [LK98a] ont proposé une méthode de codage progressif permettant d'organiser linéairement les données dans le train binaire dans un ordre décroissant de l'innovation. Les positions et toutes autres données d'attributs sont quantifiées successivement après une prédiction linéaire basée sur le calcul d'une moyenne locale. Les bits les plus significatifs sont regroupés et codés en premier dans le train binaire afin de permettre une reconstruction progressive des modèles. La scalabilité obtenue sur la connectivité exploite la méthode définie par Schroeder [SZL92]. Cette dernière méthode supprime un sommet et retriangule la couronne du trou résultant, ce qui introduit un degré de liberté, d'autant plus grand que la valence du sommet était importante. Les auteurs utilisent alors un code de longueur variable pour indiquer le mode de triangulation, modulo la rotation.

Le format *MetaStream* [AdRL⁺99] encode un maillage avec une scalabilité sur la connectivité. Un maillage est représenté sous la forme d'un historique d'opérations élémentaires (deux types de séparation de sommet et l'opérateur de permutation d'arête). Le schéma est ainsi très général et accepte la famille des algorithmes de simplification fonctionnant par fusion d'arête [Hop96] ou fusion d'arête généralisée [GH97] [GH98] afin d'autoriser le codage de maillages multiples (ou *manifold*). Les attributs d'apparence sont compressés par

quantification, prédiction et codage entropique à modèle adaptatif. Ce format prend également en compte la compression des textures sous une forme échelonnée en exploitant le format JPEG progressif.

Nous venons de décrire les principales méthodes de codage de la connectivité. Chacune correspond au contexte applicatif correspondant ou à des contraintes matérielles. L'algorithme le plus performant en terme de codage sans perte de la connectivité correspond à la méthode de Touma et Gotsman [TG98], toutefois il peut s'avérer aussi très intéressant de pouvoir borner le coût de codage comme l'algorithme de Rossignac [Ros99]. Regardons maintenant les principes employés pour le codage des positions.

Les positions

D'une façon générale, les positions sont codées soit par quantification globale (figure 3.8) et codage entropique, soit par prédiction, quantification puis codage entropique. La figure 3.9 met en évidence le gain obtenu par la quantification sur les résidus différentiels sur un exemple simple. La plupart des auteurs [TG98] [Dee95] [THLR97] [Tau98] proposent de quantifier les données entre 8 et 16 bits en moyenne. Ahn et Ho [AH99] transforment les positions différentielles des sommets prélevées le long d'un arbre de sommets en coordonnées sphériques (r, θ, ϕ) . La norme du rayon est quantifiée de façon uniforme, puis les paires (θ, ϕ) correspondantes sont quantifiées via une partition de la sphère de rayon r en n^2 régions. A noter : un article [CLA97] sur la compression vectorielle (PRVQ) des positions, et une comparaison avec le codage différentiel scalaire.

Les normales

Le codage des normales peut être effectué par un découpage de la sphère unité en octants, puis chaque octant en 6 zones triangulaires pour Deering [Dee95] [Dee98]. Une table de correspondance de 2000 normales associée à chaque zone triangulaire permet de générer une normale sur le décodeur à partir d'un code de 17 bits. Ondet [Ond99] a récemment proposé dans MPEG4 une version améliorée, avec projection sur un cube centré. Tout vecteur unitaire est ainsi projeté sur une des faces du cube, avant d'être prédit par parcours sur les faces du cube, et quantifié (éventuellement sous une forme échelonnée).

Les couleurs

Les acteurs de la compression d'images fixes ou animées ont fourni des efforts de recherche considérables pour la compression des couleurs, et la fidélité des couleurs en infographie n'est pas une mince affaire [RP98]. L'essentiel de l'information étant véhiculée par la luminance, il en résulte des niveaux de quantification différents suivant les canaux de luminance/chrominance. Toutefois, les modèles 3D ne sont pas colorés de la même façon qu'une image est générée par un système d'acquisition. Même si l'on constate des corrélations entre la géométrie et les couleurs, la compression de ces données est effectuée en pratique indépendamment par des méthodes assez génériques en compression de données : prédiction, quantification, tables de correspondance, etc...

Maintenant munis d'une vision plus nette du panorama sur les travaux existants, nous pouvons situer et décrire les innovations mises en œuvre dans notre méthode de codage de maillages triangulaires.

3.5 Codage de maillages triangulaires

Cette section propose la description d'un codeur échelonné de maillages triangulaires, le codage représentant une étape complémentaire et indispensable aux techniques de simplification et d'approximation. La méthode présentée s'inscrit dans la lignée des travaux de Touma et Gotsman [TG98] et de Li et Kuo [LK98a]. Ces derniers ont mis au point en 1998 un algorithme original fonctionnant par conquête d'arêtes sur les mail-

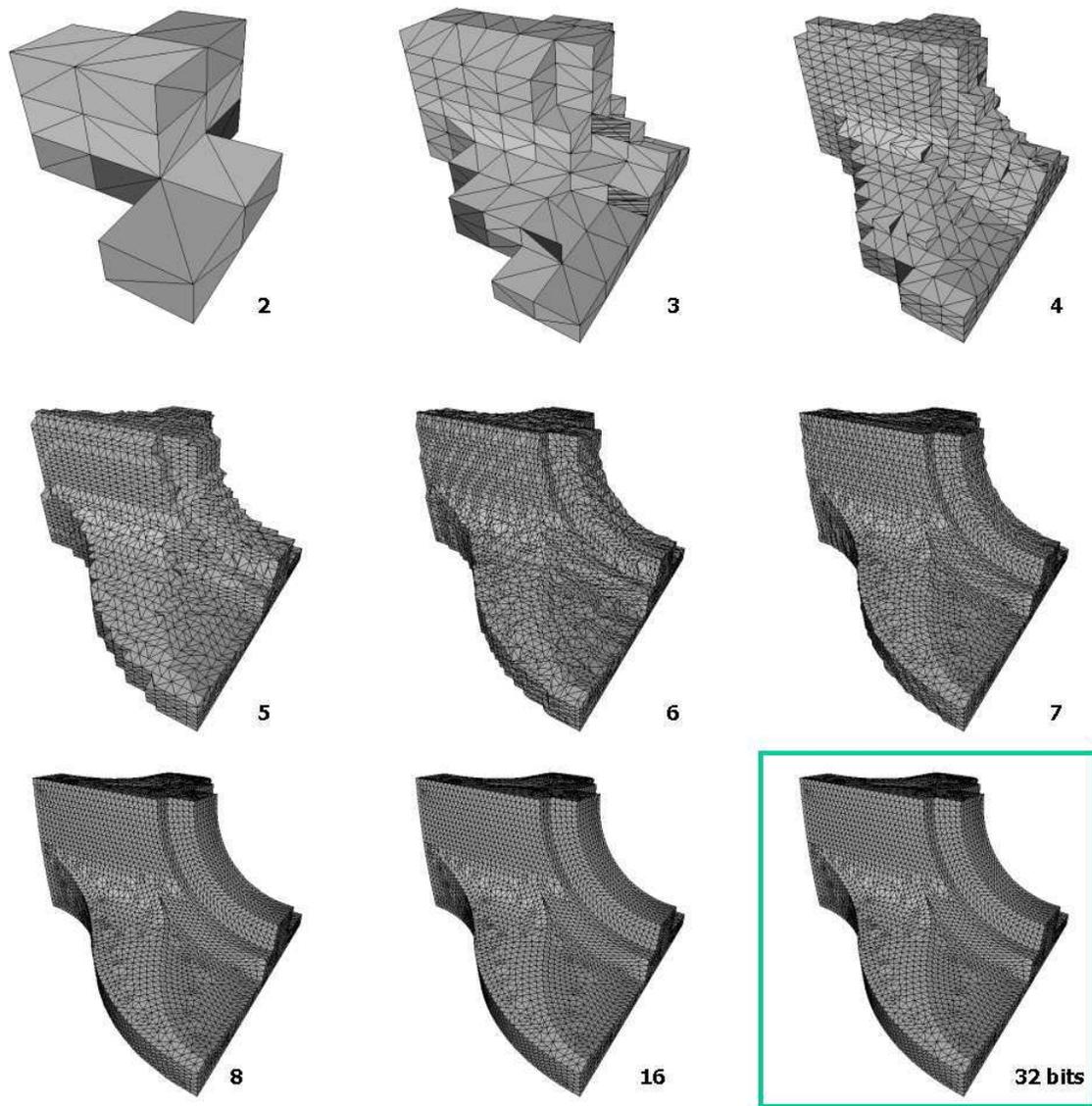


FIG. 3.8 : La quantification globale des positions. Les positions de la première figure sont codées sur 2 bits, correspondant ainsi à quatre positions différentes sur chaque coordonnée. Sur la plupart des objets, un niveau de quantification situé entre 10 et 12 bits coïncide avec la notion de codage sans perte visuelle.

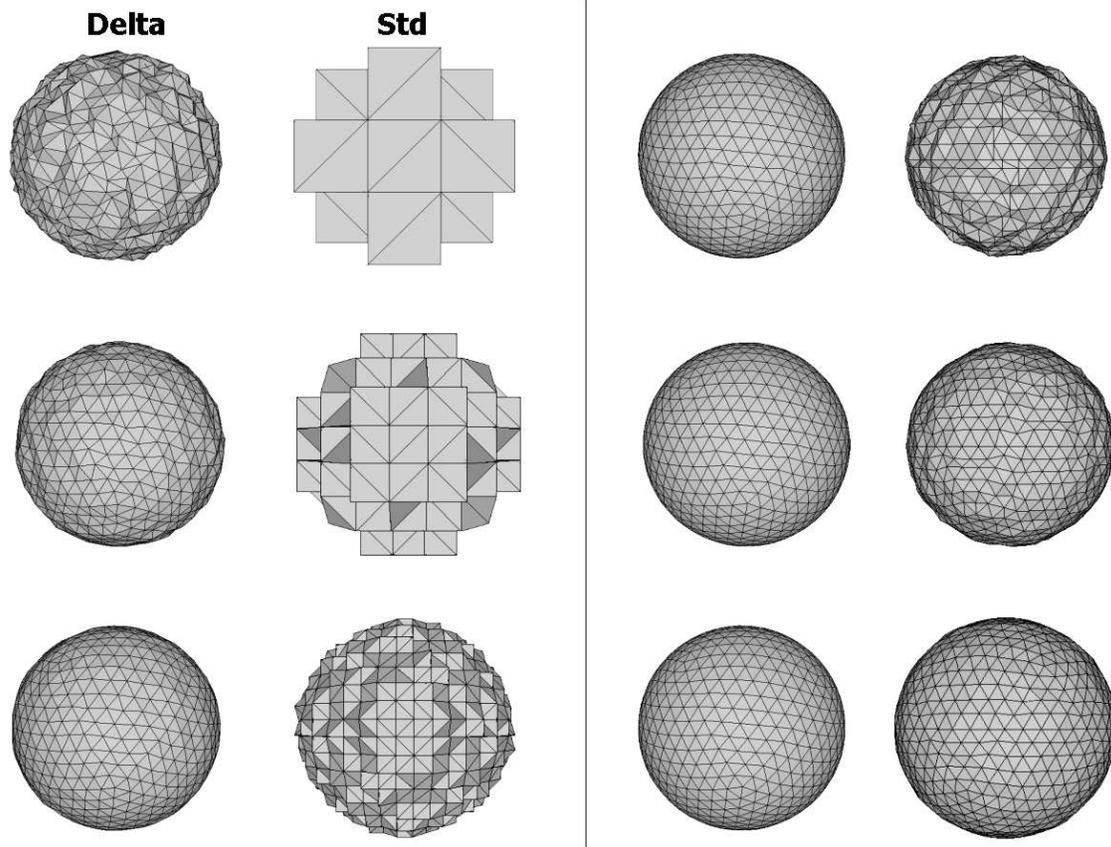


FIG. 3.9 : Quantification linéaire des résidus différentiels versus quantification globale. On constate qu'il est plus avantageux de quantifier les résidus différentiels sur cet exemple. En pratique, cela est vrai seulement lorsque l'excursion des valeurs résiduelles est inférieure à la taille de la boîte englobante du maillage.

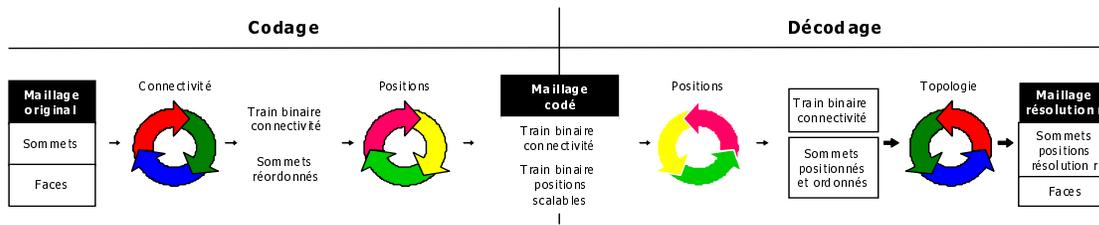


FIG. 3.10 : Les algorithmes de codage et de décodage travaillent indépendamment sur les deux types de données (géométrie et connectivité), ce qui garantit la robustesse de l’algorithme.

lages. Afin d’améliorer les performances de codage de cet algorithme et de gagner en robustesse, nous avons apporté diverses modifications portant essentiellement sur l’indépendance du codage connectivité / positions, la stratégie de conquête, la notion d’orientation et les codes générés en vue du décodage. Ces développements ont permis d’obtenir un gain moyen de plus de 35% en terme de compression sur la connectivité, de supprimer les limites inférieures constatées par les auteurs sur les méthodes précédentes (1 bit/face [THLR97], et 0.2 bit/face [TG98]) et d’accélérer sensiblement la vitesse de décodage. Enfin, une représentation compressée et échelonnable des positions est réalisée par un codage hybride exploitant l’ordre de description des sommets fourni par le codeur de la connectivité, une méthode de prédiction multiple et une gestion optimisée des accidents.

La section 3.5.1 présente le principe général du système codeur/décodeur. Les sections 3.5.2 et 3.5.3 décrivent respectivement la méthode de codage et de décodage. Des résultats commentés sont proposés en la section 3.5.4.

3.5.1 Principe général

Le codeur génère successivement deux trains binaires distincts : l’un décrivant la connectivité et l’autre la géométrie (ou les positions) sous une forme échelonnable (figure 3.10). La connectivité est codée par une stratégie de conquête basée arêtes sur les maillages, tandis que les positions sont codées sous une forme progressive en exploitant les corrélations locales inter-sommets et une prédiction multiple comprenant une gestion des accidents. Le codeur de la connectivité utilise une pile de listes d’arêtes et exploite l’information de valence des sommets, il s’appuie sur la notion d’orientation topologique définie autour des sommets. Des codes correspondant aux valences des sommets sont ainsi générés, puis compressés par codage entropique (voir définition dans la section 3.2.1).

3.5.2 Encodage

Dans cette section, nous explicitons le principe de compression de la connectivité et des positions des sommets d’un maillage. Les choix effectués nous ont permis de supprimer les redondances, de réduire la quantité d’information originale et de générer un train binaire adapté au décodage échelonnable des positions. Afin de situer la technique mise au point, on rappelle brièvement la redondance contenue dans le format VRML : une face triangulaire est décrite par trois références des sommets qu’elle relie, soit 96 bits/face si l’on code chaque indice sur 32 bits. Ainsi, les faces adjacentes à un sommet de valence n font n fois référence à ce dernier, or idéalement une référence suffit si l’on considère que toute redondance est superflue. Les positions sont stockées sous le format IEEE flottant, soit 96 bits par sommet. La précision correspondante est inutile et coûteuse en ce qui concerne les applications de visualisation en réseau. Toutefois, il en est autrement pour des applications

de calcul scientifique, ce qui justifie d'autant plus le développement d'un codeur échelonnable de la géométrie permettant une restitution complète de la précision.

Connectivité

Une technique couramment utilisée en codage correspond au changement de mode de représentation de l'information, on parle aussi de redistribution de l'énergie. Nous procédons de la sorte avec les valences des sommets, et commençons par calculer la valence moyenne d'un maillage triangulaire.

Soient : F le nombre de faces, S le nombre de sommets et A le nombre d'arêtes d'un maillage triangulaire. La valence moyenne d'un tel maillage se calcule à l'aide de la formule d'Euler : $F + S - A = 2$. Si l'on considère la relation d'incidence arête / face, chaque arête est incidente à deux faces et chaque face possède trois arêtes. On compte ainsi de deux manières différentes le nombre de paires incidentes arête / face, on en déduit la relation $2A = 3F$ que l'on ajoute à la relation d'Euler pour en déduire : $A = 3S - 6$. Par ailleurs, on exprime la somme des valences des sommets pour tout graphe par la relation : $\sum \text{valences} = 2A$. On en déduit : $\sum \text{valences} = 6S - 12$. La valence moyenne des sommets d'un maillage triangulaire s'écrit donc sous la forme : $6 - \epsilon$.

Les courbes de distribution des valences montrent un lien entre la régularité d'un maillage et la formation de pics sur ces premières (figures 3.11 et tableau 3.1). Ces courbes incitent au codage entropique des valences, à condition de fournir suffisamment d'information en plus au décodeur pour reconstruire sans pertes la connectivité du maillage à partir des valences. La connectivité est codée en premier car les taux de compression obtenus sur cette dernière sont en moyenne dix fois plus élevés que les résultats de la compression des positions. Un tel algorithme réordonnant les sommets, il convient donc de lui donner la priorité avant de lancer le codage des positions.

Valence	nœud	nefertiti	venus	sphère
1	0	0	0	0
2	0	0	0	0
3	0	0	7	0
4	0	20	69	0
5	0	138	183	12
6	3840	4195	249	630
7	0	56	131	0
8	0	16	62	0
9	0	2	7	0
10	0	0	3	0
11	0	2	0	0

TAB. 3.1 : Distribution des valences des maillages représentés par la figure 3.11. On remarque une dispersion autour de la valeur 6, qui correspond à la valence moyenne d'un maillage triangulaire.

Des notions

Avant de décrire le principe de l'algorithme, il convient de définir quelques notions utilisées par la suite :

- une *liste de conquête* : elle est définie par un ensemble de sommets successivement adjacents, formant une boucle en marchant sur les arêtes (figure 3.12). Elle est aussi appelée liste de reconstruction sur le

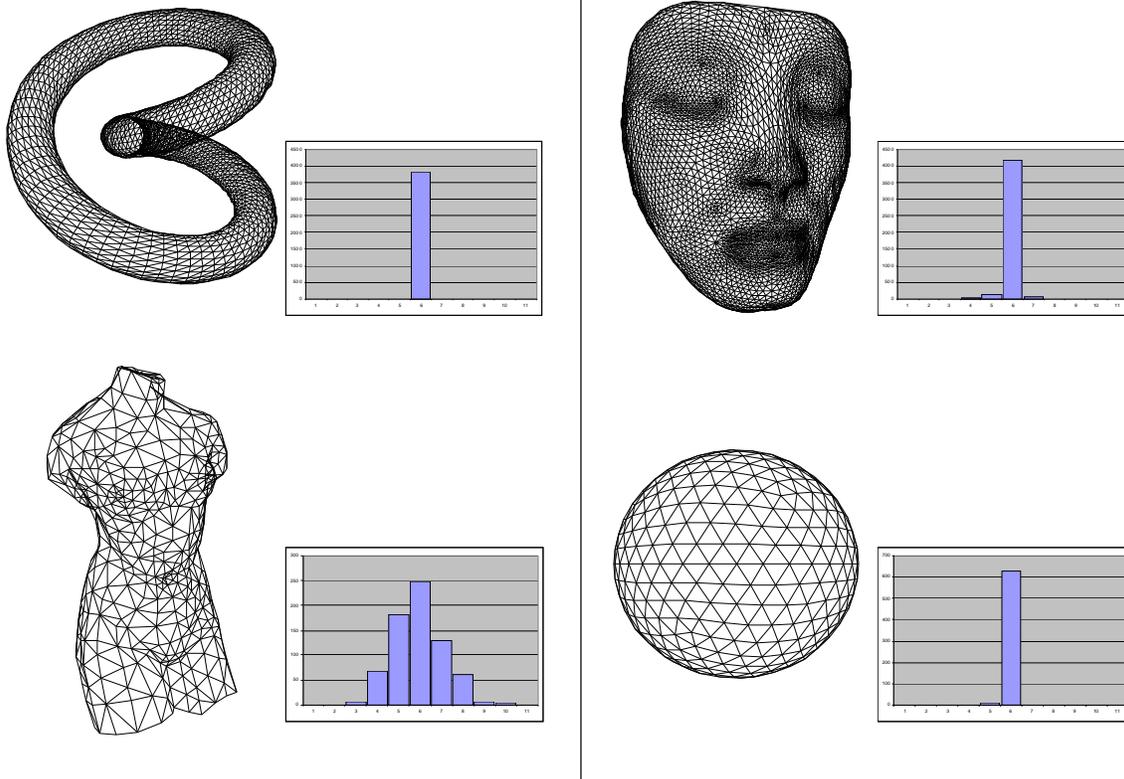


FIG. 3.11 : Plus un maillage est régulier, et plus la courbe de distribution des valences des sommets présente un pic centré sur la valeur moyenne 6, ce qui est propice au codage entropique de ces valeurs. Si on arrive à représenter un maillage seulement sous la forme d'une liste de valences et de quelques données supplémentaires, les gains en compression deviennent très performants.

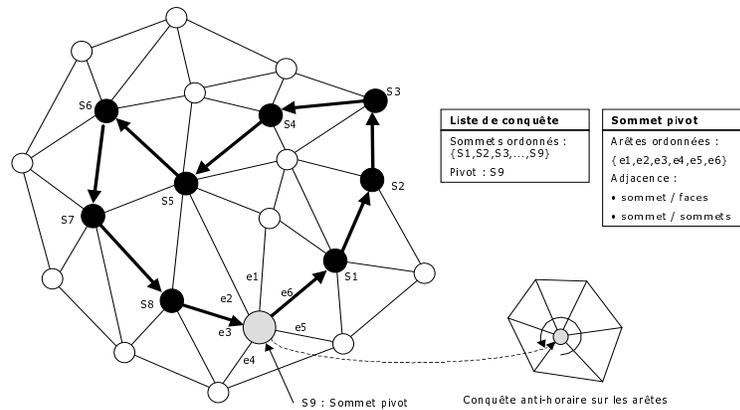


FIG. 3.12 : Une liste de conquête est formée par un ensemble ordonné de sommets successivement adjacents. Elle possède un sommet pivot, qui connaît l'orientation topologique de ses arêtes adjacentes dans le sens anti-horaire. Cette orientation autorise la conquête ordonnée des arêtes adjacentes au pivot.

décodeur,

- un *sommet pivot* associé à chaque liste de conquête organise la conquête de ses arêtes adjacentes suivant une orientation anti-horaire (figure 3.12),
- un *sommet fantôme* supprime virtuellement les bords du maillage afin d'en minimiser le coût de représentation,
- un *sommet complet* possède toutes ses arêtes adjacentes conquises,
- une liste de *conquête par expansion* (resp. *rétraction*) correspond à une conquête anti-horaire autour de pivots successifs d'une liste orientée anti-horaire (resp. horaire) (figure 3.13),
- un sommet pivot est considéré *contextuellement complet* lorsque le secteur angulaire orienté formé par les arêtes précédant et suivant le pivot dans la liste est vide (figure 3.13).

Conditions nécessaires

Le fonctionnement du codeur de la connectivité nécessite les conditions suivantes :

- Un sommet contribue toujours à la création d'une arête. Il n'y a donc pas de *sommet flottant*,
- Une arête contribue à la création d'au moins une face, et au plus de deux (le maillage est dit *2-manifold*) (voir figure 3.14),
- La liste des arêtes reliant les sommets adjacents d'un sommet doit former un cycle, cette condition n'étant plus vérifiée sur les bords ou les trous. Nous verrons comment cette contrainte est respectée par la suite.

En pratique la régularisation d'un maillage est possible au sens des contraintes que l'on vient d'énoncer par duplication de sommets sur les arêtes multiples, suppression des sommets isolés qui n'ont aucune contribution lors du rendu, etc... (voir [GTLH97]). Les étapes inverses de la régularisation utilisées au cours du codage peuvent être mises en œuvre après décodage pour retrouver l'intégralité du maillage initial.

Suppression virtuelle des bords

La conquête des arêtes autour d'un sommet pivot nécessite la présence d'une liste d'arêtes fermée reliant les sommets adjacents d'un pivot. Un sommet fantôme unique est alors ajouté au maillage en présence de bords (figure 3.15). Un tel sommet permet ainsi de régulariser simultanément tous les trous et les bords du maillage. La valence d'un tel sommet est équivalente au nombre de sommets appartenant aux bords du maillage, son

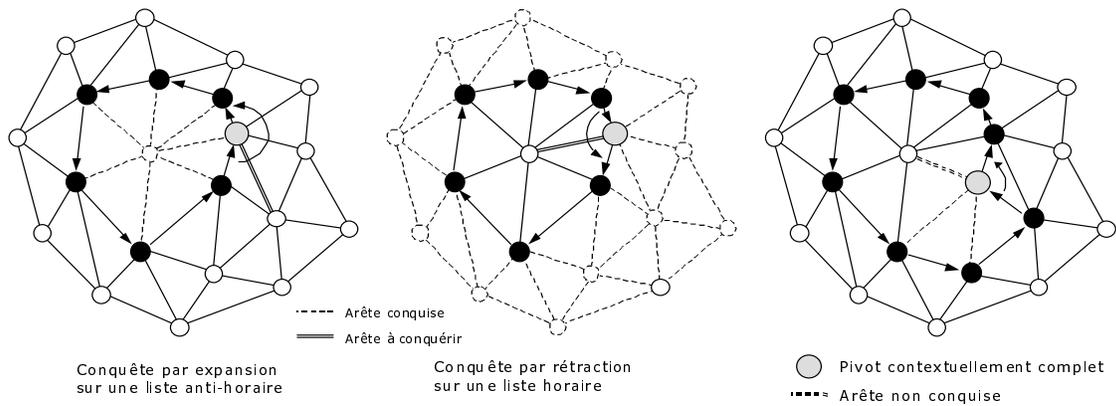


FIG. 3.13 : A gauche : la conquête anti-horaire des arêtes autour de sommets pivots successifs d'une liste anti-horaire aboutit à une expansion de la liste. Au centre : la même conquête sur une liste horaire aboutit à une rétraction de la liste. A droite : le sommet pivot est contextuellement complet lorsque le secteur angulaire orienté formé par les arêtes de la liste est vide. L'arête restante non conquise (en trait double pointillé) ne sera pas conquise par ce pivot.

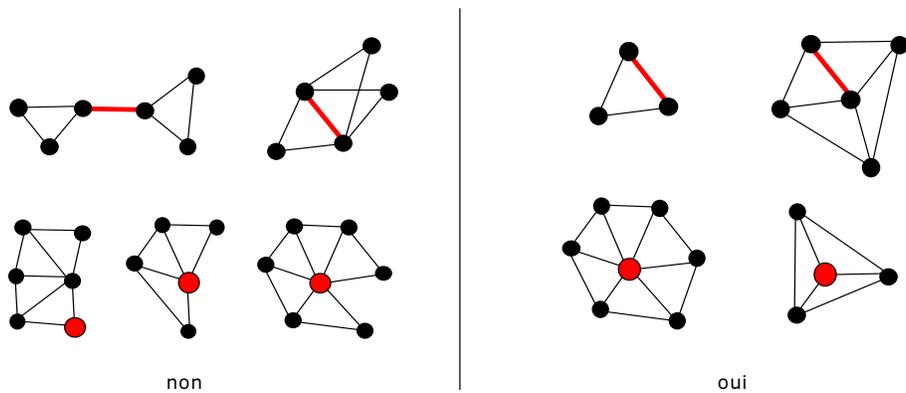


FIG. 3.14 : Conditions nécessaires au codage de la connectivité. Une arête contribue à la formation d'au moins une, et au plus deux faces, le maillage est dit *2-manifold*. La liste des arêtes reliant les sommets adjacents d'un sommet doit former un cycle, ce qui n'est plus vrai sur les bords ou les trous du maillage.

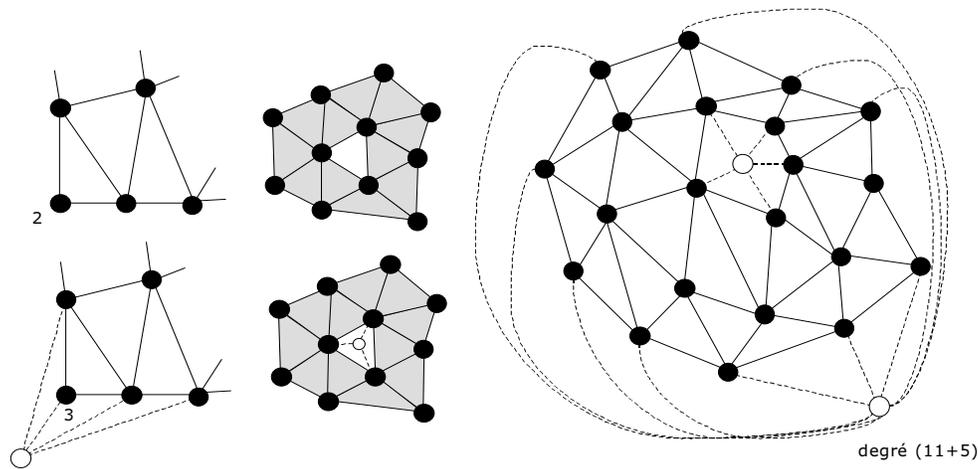


FIG. 3.15 : A gauche : la valence minimale d'un sommet vaut 2 (on fait l'hypothèse qu'un sommet contribue toujours à la construction d'une face), c'est le cas où un sommet se retrouve sur un coin du bord. Après ajout du sommet fantôme et de ses arêtes adjacentes, la valence minimale d'un sommet vaut 3 (au centre). A droite : un sommet fantôme est ajouté au maillage en présence de bords. Comme son nom l'indique, un tel sommet possède le don d'ubiquité, et régularise ainsi tous les sommets appartenant aux bords. D'un point de vue algorithmique, les arêtes et l'adjacence correspondante sont générées pour garantir la fermeture de la liste d'adjacence de ses sommets voisins, tandis qu'aucune face n'est générée. Notons qu'un sommet fantôme possède une valence équivalente à la taille des bords exprimée en nombre de sommets.

ordre de grandeur n'est donc pas comparable aux valences des sommets intérieurs.

Orientation topologique

Rappelons le contexte : la conquête s'effectue sur les arêtes de sommets pivots appartenant à une liste de conquête (figure 3.12). L'algorithme fonctionnant par conquête dans le sens anti-horaire des arêtes adjacentes à un sommet pivot, il convient de trier à l'avance la liste des arêtes adjacentes à chaque sommet. La figure 3.16 illustre le principe du calcul de l'orientation topologique, avec un exemple de la suppression des ambiguïtés survenant en présence de sommets de valence 3.

Nous considérons maintenant l'orientation topologique pré-calculée sur les sommets, et mesurons maintenant l'incidence des informations d'adjacence en terme de structure de données.

Structure de données

Au départ l'adjacence faces/sommets est construite sur le maillage, un sommet fantôme et les arêtes correspondantes sont ajoutés en présence de bords, puis l'adjacence sommets/arêtes est construite. Classiquement, le coût mémoire d'une telle structure de données est d'environ 4.3 fois celui du maillage original (voir tableau 3.2). Les listes de référence aux arêtes autour de chaque sommet sont ensuite triées suivant une orientation topologique anti-horaire (excepté le fantôme), puis une liste d'arêtes vide appelée liste active est générée et introduite dans une pile. La conquête peut alors commencer.

La conquête démarre

Une face germe est sélectionnée de façon arbitraire sur le maillage et ses trois sommets sont ajoutés à la liste active dans leur ordre de description dans la face germe. Simultanément à l'ajout des sommets dans la liste trois

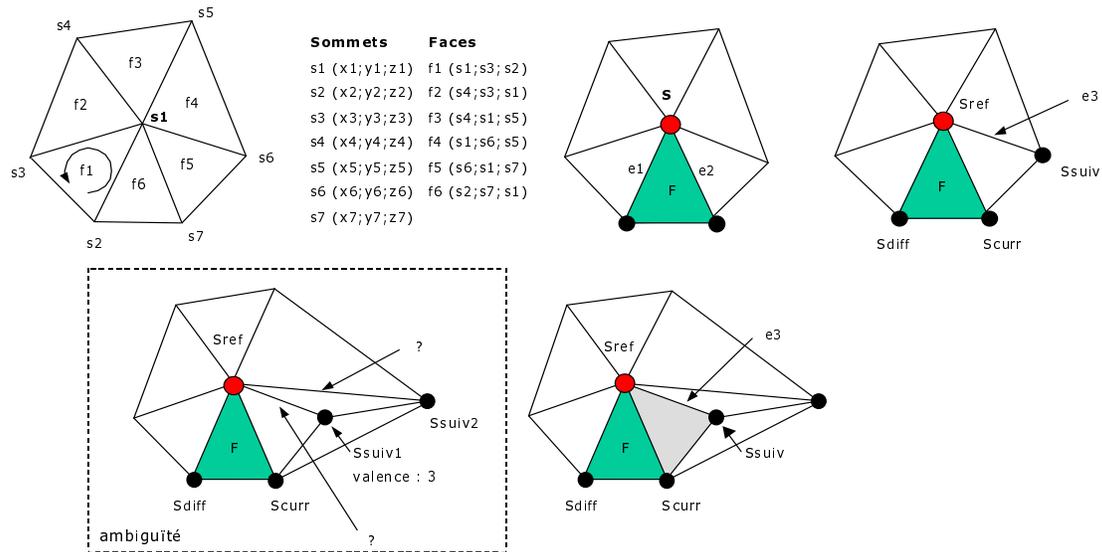


FIG. 3.16 : L'orientation topologique définie autour du sommet S est déduite de l'information d'orientation des faces et de l'hypothèse de multiplicité d'adjacence des arêtes (le maillage est considéré *2-manifold*). Observons la description des faces adjacentes au sommet s_1 dans le format VRML [vrml]. Pour chaque sommet S concerné, on commence par choisir arbitrairement une face F . Les faces sont définies par trois références aux sommets, et on fait l'hypothèse que l'ordre de description est anti-horaire et cohérent autour du sommet s_1 . Cet ordre va permettre le tri des arêtes adjacentes au sommet s_1 suivant une orientation anti-horaire. Au départ les sommets sont étiquetés comme suit : S_{ref} est le sommet concerné par le tri de sa liste d'arêtes adjacentes, S_{diff} et S_{curr} sont déduits de l'orientation définie sur la face F . Les arêtes e_1 et e_2 sont directement déduites de la construction de la face F , puis l'arête e_3 est déduite du sommet S_{suiv} , trouvé par relation d'adjacence entre les sommets (S_{suiv} est adjacent à S_{ref} et S_{curr} , et différent de S_{diff}). Il peut y avoir ambiguïté lorsque plusieurs sommets S_{suiv} répondent au test d'adjacence (en trait discontinu) ; ce cas survient en présence de sommets de valence 3. Cette ambiguïté est levée par un test sur les faces : on retient le sommet S_{suiv} qui forme une face avec S_{ref} et S_{curr} . En effet, si les sommets S_{ref} , S_{curr} et S_{suiv_2} formaient une face, le maillage ne serait plus *2-manifold*. A l'itération suivante les étiquettes de sommet se décalent suivant l'orientation anti-horaire calculée en cours de marche. Remarquons qu'aucune orientation ne peut être calculée autour d'un sommet fantôme puisqu'aucune face n'est connectée à ce dernier. En pratique ce sommet n'est jamais affecté comme pivot.

Maillage original		Maillage et adjacence pour le codage		
N sommets	F faces $F \simeq 2N$	N sommets	F faces $F \simeq 2N$	E arêtes $E \simeq N + F \simeq 3N$
position $\{x,y,z\}$ 3×32 bits	référence $\{s_i,s_j,s_k\}$ 3×32 bits	position $\{x,y,z\}$ 3×32 bits sommets adjacents 6×32 bits faces adjacentes 6×32 bits arêtes adjacentes 6×32 bits	référence $\{s_i,s_j,s_k\}$ 3×32 bits faces adjacentes 3×32 bits	référence $\{s_i,s_j\}$ 2×32 bits
coût moyen : 144 bits/face		coût moyen : 624 bits/face $4.3 \times$ coût original		

TAB. 3.2 : Coût mémoire pour l'encodage. Les hypothèses suivantes sont émises : l'adjacence quasi complète est nécessaire pour l'encodage, la valence moyenne d'un sommet vaut six, et chaque face possède au plus trois faces adjacentes. Les références sont représentées sous la forme de pointeurs stockés sur 32 bits, à l'instar des coordonnées de position.

codes correspondant à la valence de ces derniers sont générés. Le premier sommet est affecté comme pivot de la liste active. La liste active est ainsi orientée dans le sens anti-horaire, et prête pour la conquête sur le maillage. Il est à noter que la liste des valences ne contient pas d'élément inférieur à 3, c'est la valence minimale d'un maillage sans bords ni trous après fermeture grâce au sommet fantôme. La conquête sur les arêtes commence alors comme décrite figure 3.17.

Cette séquence illustre le fonctionnement de conquête par expansion, qui peut aboutir à un rebouclage de la liste, comme illustré sur la figure 3.18.

Séparation de liste

La liste de conquête étant initialisée suivant une orientation anti-horaire, et la conquête étant réalisée suivant la même orientation autour des pivots successifs, ceci garantit l'expansion de la liste. Si le maillage est irrégulier ou comporte des régions de densité variable, des rebouclages de liste peuvent intervenir. La liste active est alors séparée au point de rencontre en deux sous-listes dont l'une d'elle est orientée dans le sens horaire (voir figure 3.18). On parle alors de liste de conquête par rétraction, illustrée par les figures 3.19 et 3.20. La séparation de liste génère un code réservé, un bit de signe et un offset traduisant la position de la séparation dans la liste. Le bit de signe permet de réduire sensiblement l'excursion des valeurs d'offset lorsque la séparation intervient avant la position du pivot dans la liste. La séparation de liste déclenche ensuite une recherche des sommets complets constituant de parfaits candidats au pivot. Cette recherche s'effectue dans les deux directions à partir du sommet pivot, et la priorité est donnée au sommet rencontré pendant la recherche anti-horaire lorsque deux sommets complets se présentent. Le décodeur se comportera de la même manière afin de garantir la synchronisation codeur / décodeur, aucun code n'étant généré pour désigner le sommet complet affecté comme pivot. Notez qu'une recherche de sommets complets dans la liste peut s'avérer coûteuse, mais en pratique la recherche exhaustive n'intervient qu'en présence d'un code de séparation de liste, sensiblement moins probable qu'un code de valence. La recherche d'un sommet complet intervient également après la complétude d'un sommet pivot, toutefois il s'avère suffisant de limiter dans ce cas la recherche au voisinage immédiat du pivot courant puisque la conquête concerne seulement les arêtes adjacentes au sommet pivot.

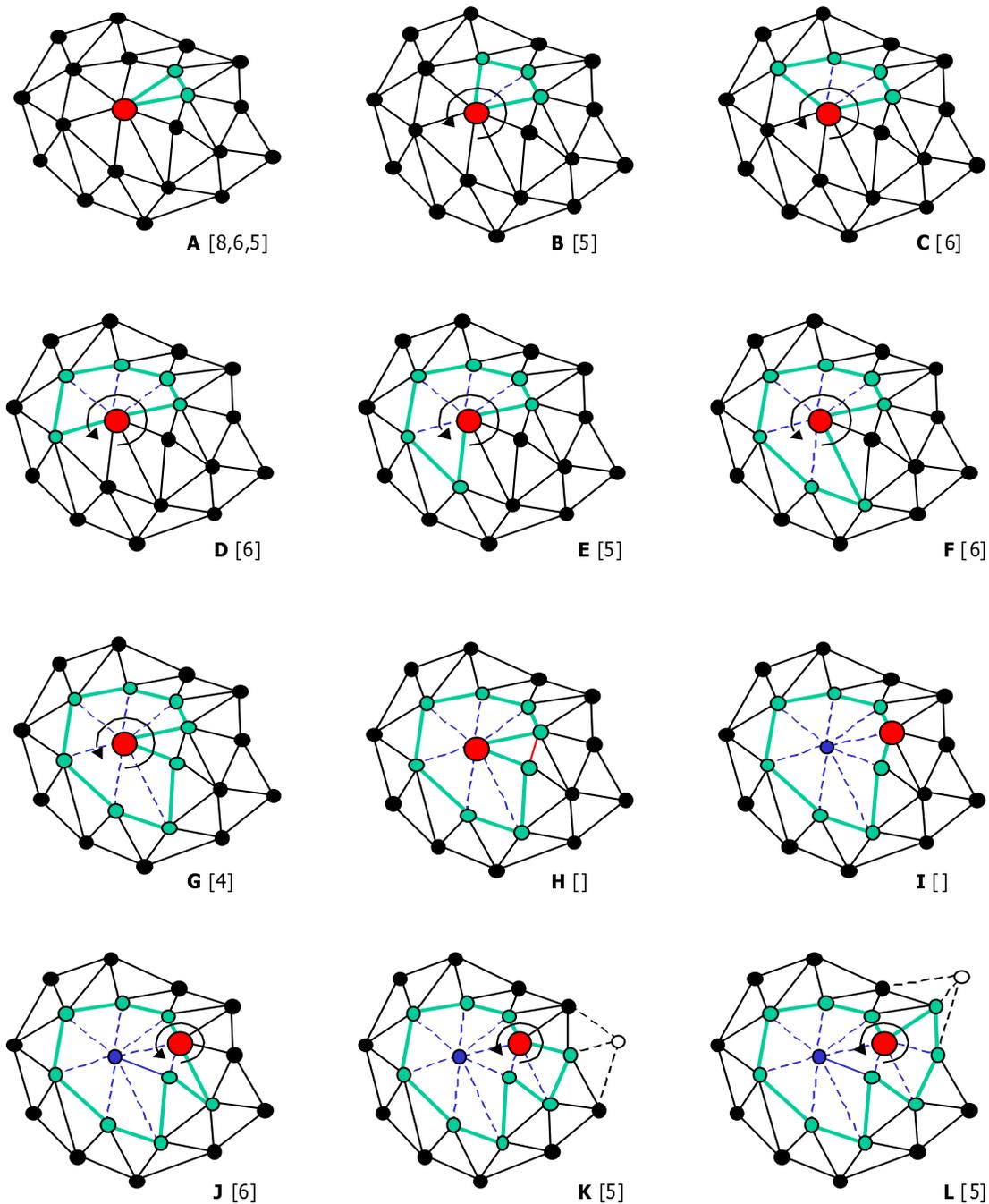


FIG. 3.17 : Conquête régulière par expansion. [A] Une face germe est sélectionnée sur le maillage, ses trois sommets sont ajoutés à la liste de conquête suivant une orientation anti-horaire. Trois codes $\{8,6,5\}$ correspondant aux valences respectives des sommets sont générés. Le premier sommet est affecté comme pivot. [B] La liste active commence sa conquête en marchant sur les arêtes libres du pivot dans l'ordre anti-horaire. Le nouveau sommet est inséré avant le pivot dans la liste et deux arêtes sont conquises : code $\{5\}$. [C] Code $\{6\}$. [D] Code $\{6\}$. [E] Code $\{5\}$. [F] Code $\{6\}$. [G] Code $\{4\}$. [H] Le pivot est complet, l'arête reliant les sommets précédant et suivant le pivot, est marquée sans aucun code. [I] Le pivot est supprimé de la liste, le sommet suivant est affecté comme pivot, sauf si un sommet déjà complet se trouve dans la liste. Notez que cette opération de mouvement du pivot permet la génération d'une face par *maillage* sur le décodeur, avec un coût de codage nul. [J] Code $\{6\}$. [K] Code $\{5\}$, fantôme compris. [L] Code $\{5\}$.

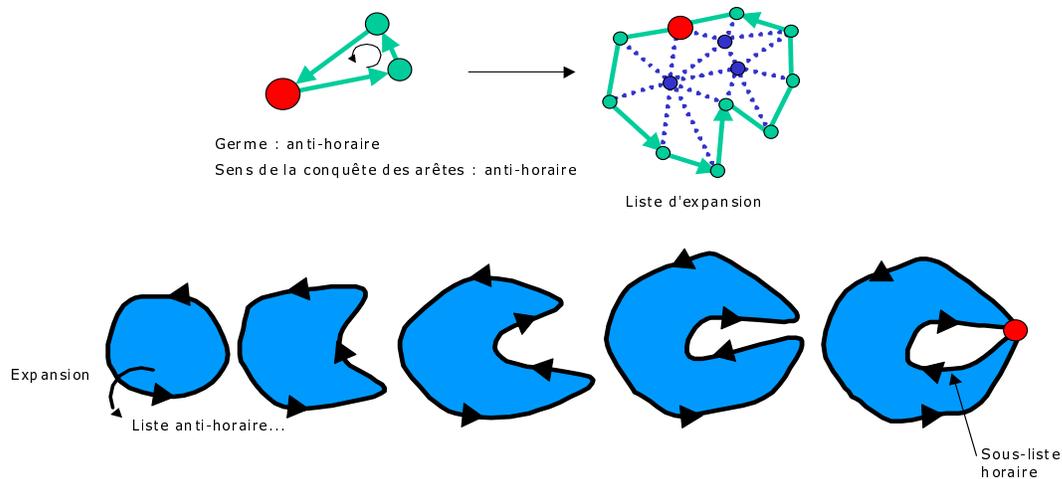


FIG. 3.18 : Conquête par expansion, et rebouclage de liste. Une liste germe anti-horaire s'étend par conquête des arêtes autour de sommets pivots successifs. Un rebouclage de liste, considéré comme un accident, survient lorsqu'une arête reliant deux sommets de la liste active est conquise. La liste active est alors séparée en deux listes. En pratique le nombre de rebouclage de listes est lié aux variations de densité géométrique sur le maillage, la liste de conquête progressant plus lentement dans les régions denses. Remarquons l'orientation horaire d'une sous-liste déduite d'une séparation d'une liste anti-horaire.

Codage des bords

La présence d'un sommet fantôme unique permet de coder les bords avec un coût sensiblement réduit, un code spécifique étant réservé à la conquête d'une arête reliée au sommet fantôme (voir figure 3.21). L'algorithme se comporte ainsi différemment en présence d'un sommet fantôme :

- la conquête d'une arête reliant le fantôme ne génère pas de face, contrairement aux sommets ordinaires,
- un sommet fantôme n'est jamais affecté comme pivot, le sommet suivant ou un sommet complet étant choisi à sa place,
- chaque liste de conquête peut contenir plusieurs occurrences de fantômes, chaque occurrence étant supprimée lorsque le sommet fantôme s'avère complet (figure 3.22),
- la valence du fantôme est codée à part, ce qui évite d'alimenter le codeur entropique avec une valence dont l'excursion dépasse les valeurs usuelles.

Fusion de listes

La conquête d'une arête adjacente à un sommet appartenant à une liste différente de la liste active engendre une fusion de listes, un code lui est alors réservé, suivi d'un index de la liste inactive dans la pile et d'un index du sommet traité dans cette liste. La disparité des codes associés aux fusions de liste nous autorise à les coder à part sans compression.

La conquête se termine

Au moins une arête est conquise à chaque itération de l'algorithme, assurant ainsi la progression de la conquête d'une liste. Comme un sommet pivot complet est systématiquement supprimé, cela assure la décroissance d'une liste active jusqu'à atteindre une taille nulle. La liste active est ensuite supprimée de la pile après la complétude de chaque sommet (la liste est vide), et la liste suivante dans la pile devient active. L'algorithme finit sa conquête lorsque la pile de listes est vide. La figure 3.23 illustre une conquête sur le maillage *nefertiti*, et l'annexe

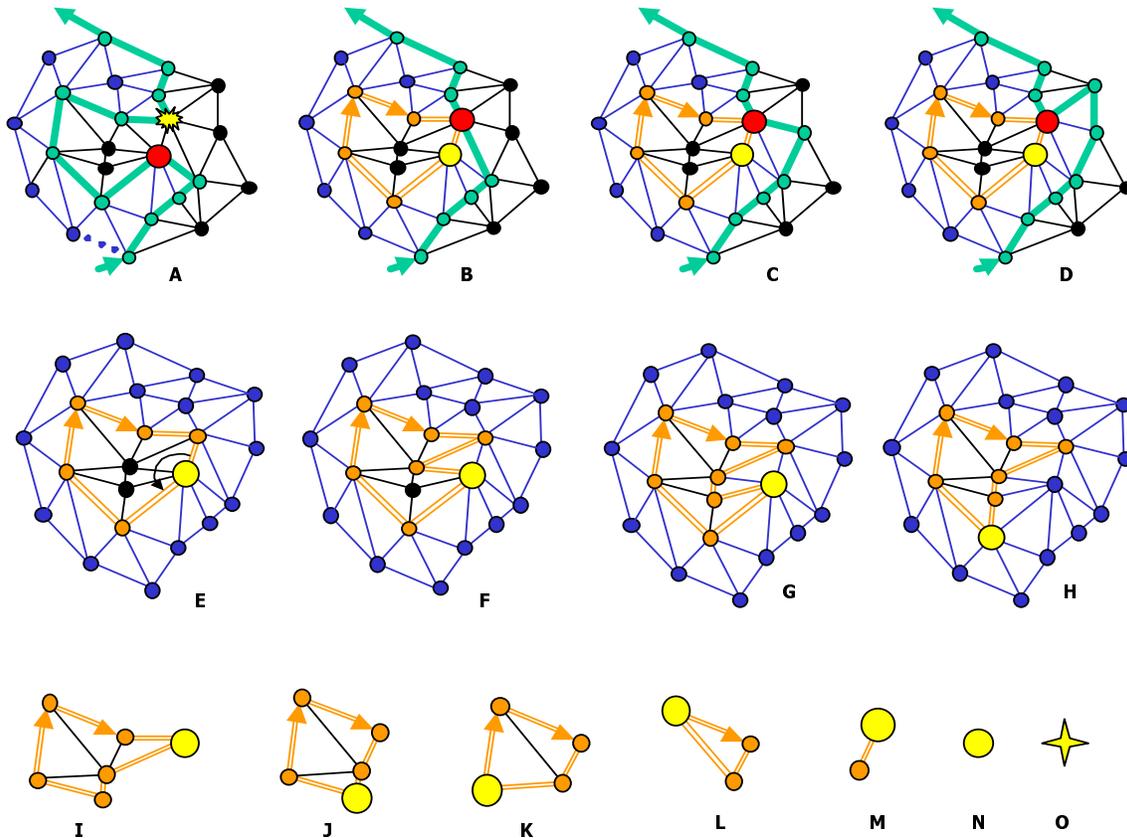


FIG. 3.19 : Séparation de liste et conquête par rétraction. [A] Supposons une telle configuration de conquête réalisée. La prochaine arête à conquérir suivant l'orientation anti-horaire fait référence à un sommet déjà transmis, ce qui va à l'encontre d'un système idéal qui ne génère qu'une référence par sommet. Il s'agit donc de transmettre au décodeur une information suffisante pour permettre le maillage du pivot avec le sommet déjà transmis. A priori tous les sommets du maillage peuvent être candidats, en pratique la corrélation locale étant élevée, un offset non signé exprimé en nombre d'arêtes et un bit de signe sont générés : codes $\{\text{split}, +, 5\}$. [B] La liste active est séparée en deux listes d'orientation inverse, un code réservé à la séparation est généré et la sous-liste est élevée dans la pile. Le nouveau sommet référencé est affecté comme pivot de la liste active, sauf si un sommet candidat complet se présente. Il en va de même pour la sous-liste qui possède aussi son pivot propre. [C] Code $\{5\}$, fantôme compris. [D] Code $\{5\}$. Le pivot n'est pas complet au sens strict, pourtant on le considère comme contextuellement complet puisque chaque liste doit conquérir sa région réservée pour garantir une bonne cohérence de l'algorithme. Un code réservé est généré à cet effet. [E] Supposons la conquête de la première liste terminée. La sous-liste est retirée de la pile et devient active. La conquête anti-horaire va se poursuivre autour du pivot, diminuant la liste. [F] Code $\{6\}$. [G] Code $\{4\}$. Le pivot est complet, l'arête correspondante est marquée, le pivot est supprimé et avance d'un rang. [H] Le pivot est complet. L'arête correspondante est marquée et le pivot n'est pas affecté au sommet suivant, ce qui créerait une séparation en deux listes et donc un code réservé, gage de surcoût. [I] Le sommet complet de la liste est préféré comme candidat au pivot. Cette modification de stratégie par rapport à la méthode originale [TG98] permet de terminer la conquête avec un coût nul. [J] [K] [L] [M] [N] Chaque sommet complet aboutit à la conquête d'une arête, encourageant ainsi la complétude des sommets voisins. [O] La liste active est vide, supprimée de la pile puis la liste suivante devient active.

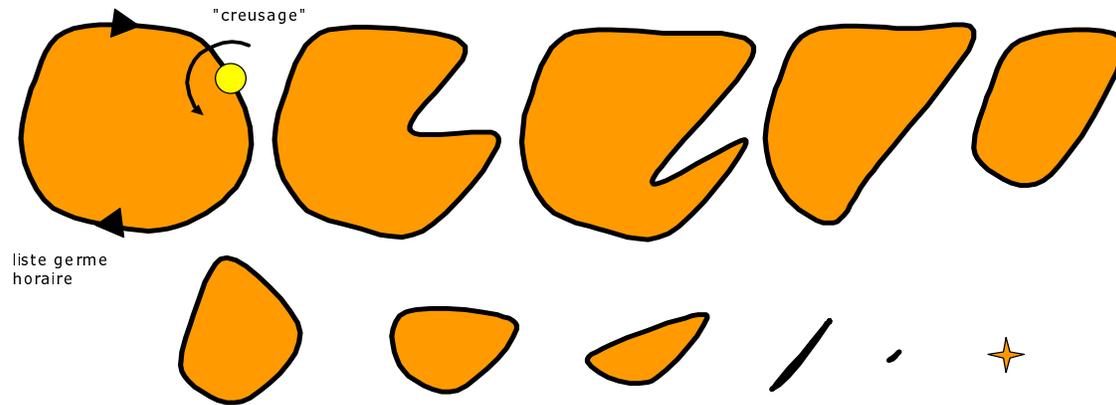


FIG. 3.20 : Une liste germe horaire aboutit à une conquête par rétraction puisque la conquête sur les arêtes s'effectue toujours dans le sens anti-horaire, avec l'arête précédant le sommet pivot comme référence (dans la liste de conquête active). La rétraction garantit la conquête intégrale de la région enfermée par la liste horaire, jusqu'à l'obtention d'une liste vide.

B décrit le pseudo-code du codage de la connectivité.

Codage entropique des valences

Après la conquête, les valences sont compressées par un codeur entropique à modèle adaptatif, tout comme les offsets non signés. Les bits de signe sont transmis séparément sans utilisation du codeur entropique car équiprobables. La valence d'un sommet n'étant jamais inférieure à 3, on réserve via un décalage unitaire les quatre premières valeurs pour les codes accidents (séparation, fusion de listes, sommet fantôme et sommet contextuellement complet). La section 3.5.4 illustre les résultats obtenus par la technique. Les valences des sommets étant codées dans l'ordre de parcours sur le graphe, cet ordre est fourni au codeur de la géométrie (i.e. des positions) qui fonctionne de manière indépendante.

Géométrie

Le codeur de connectivité agit sur l'ordre des sommets, tout en favorisant la proximité géométrique de deux sommets successifs. Après le codage de la connectivité, tous les sommets sont adjacents, à l'exception de sauts d'adjacence (figure 3.24). Ces accidents proviennent lors des sauts d'un bord à l'autre du maillage via le sommet fantôme, lors des changements de listes de conquête ou des changements de sommets pivots. Cette organisation incite au codage différentiel des positions, de plus nous montrons qu'une prédiction efficace peut être adaptée en observant la géométrie de plus près. En pratique la géométrie est prédite, puis les résidus résultants sont quantifiés et compressés par codage entropique.

Une valeur moyenne du taux de compression obtenu pour la connectivité est située aux alentours de 50. Un codage sans perte équivalent pour les positions est déraisonnable, puisque la précision originale est très élevée (32 bits, $3.4e+/-38$) et la redondance est faible. Nous proposons un codage échelonné en précision sur les positions, de façon à s'adapter aux contraintes dictées par l'application, les réseaux et les performances des terminaux. Une quantification grossière sur les valeurs obtenues après prédiction permet de réduire significativement l'espace mémoire occupé par une première version d'un maillage, et donc de garantir un bon niveau de compression et d'échelonnabilité sur les données.

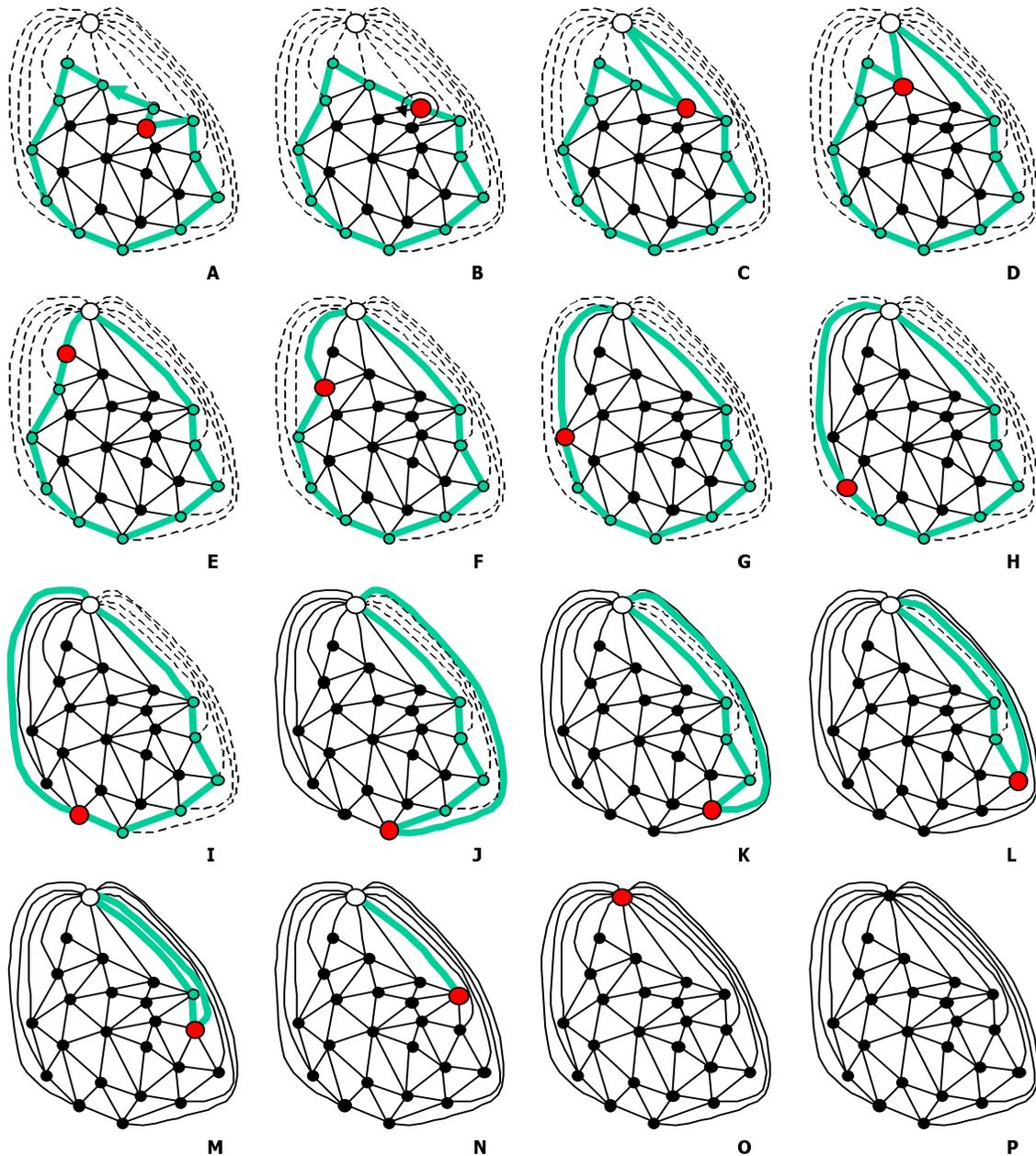


FIG. 3.21 : [A] Supposons une telle conquête réalisée. Le sommet pivot est complet. L'arête reliant les sommets avant et après le pivot est marquée, puis le pivot avance d'un rang dans la liste de conquête, le tout sans aucun code. [B] Le pivot poursuit sa conquête dans le sens anti-horaire, la prochaine arête à conquérir est reliée au sommet fantôme. Notons qu'aucune face n'a été créée entre le fantôme et les bords. [C] Un code réservé {fantôme} est généré. [D] Le pivot est complet, une arête est marquée, ce qui rend le futur pivot complet et diminue le nombre d'arêtes libres du fantôme. [E,F,G,H,I,J,K,L] Chaque marquage d'arêtes successif à l'avancée du sommet pivot a rendu le pivot suivant complet, donc aucun code n'est nécessaire. [M,N] Le pivot est complet, et toutes les arêtes sont déjà marquées, le pivot avance d'un rang. [O,P] Le sommet fantôme est affecté comme pivot. Il s'agit d'un cas particulier puisqu'il est complet. Ce dernier est donc supprimé de la liste et la conquête de la liste active est terminée. A l'inverse un fantôme non complet ne peut être affecté comme pivot puisqu'aucune orientation cohérente ne peut être définie autour de ce dernier : un fantôme unique peut régulariser simultanément plusieurs trous et bords.

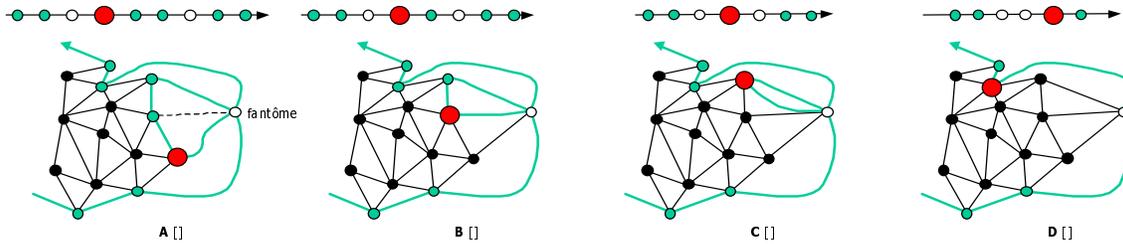


FIG. 3.22 : Une liste de conquête peut contenir plusieurs occurrences de fantômes. [A] Supposons l'état de conquête suivant. Chaque sommet du bord est relié au fantôme. Il en est de même pour chaque sommet de la liste de conquête. Le pivot est complet et avance d'un rang dans la liste en marquant une arête. [B] Le nouveau pivot est complet, avance d'un rang (on fait l'hypothèse qu'aucun sommet complet ne se trouve dans la liste). [C] Le pivot est encadré par deux occurrences du sommet fantôme dans la liste, et avance dans la liste jusqu'à trouver un sommet différent du fantôme (le fantôme ne peut être pivot). [D] plusieurs occurrences du fantôme coexistent dans la liste, ces dernières seront supprimées de la liste après le constat de la complétude du fantôme.

Prédiction multiple

En observant le graphe formé par les segments reliant les sommets ordonnés par le codage de la connectivité, on remarque principalement des zones oscillantes, des régions plus régulières (voir figure 3.25), et des accidents. Nous allons en déduire une méthode de prédiction adaptée.

Notons que les régions oscillantes peuvent également survenir sur des maillages très réguliers. La figure 3.26 en est une illustration sur une sphère, sur laquelle on visualise une phase de conquête régulière sur le premier hémisphère, et des régions oscillantes sur le second. Sur les régions oscillantes (figures 3.25 et 3.26) une prédiction suivant la méthode du parallélogramme (figure 3.27) est bien adaptée, tandis qu'une prédiction linéaire est préférée sur les seconds types de régions. Les ruptures d'adjacence sont codées comme accidents et la simple différence est conservée si aucune prédiction ne convient. La figure 3.27 explicite les quatre modes de prédiction utilisés. En pratique, moins de deux bits de prédiction par sommet sont utilisés puisque ceux-ci ne sont pas équiprobables (voir la seconde courbe de la figure 3.29), et sont donc compressés par codage entropique. Le mode de prédiction optimal minimise le résidu après prédiction. Les résidus sont exprimés en ratio de la moyenne mobile des longueurs d'arêtes pour s'affranchir des variations de dynamique géométrique d'une région à l'autre du maillage, cette moyenne étant calculée sur une série d'arêtes consécutives précédant le sommet à coder. Notons que chaque mode de prédiction se répercute sur les trois composantes des coordonnées d'un sommet.

Accidents

Un accident intervient (figure 3.27) lorsqu'aucun mode de prédiction ne ramène l'excursion du résidu en deçà de l'écart maximal mesuré entre deux sommets adjacents dans le réseau maillé initial. Un résidu d'une telle excursion remettrait en cause l'intérêt même du développement d'une méthode performante de prédiction, pourtant il ne s'agit pas de coder un résidu associé sous une forme non compressée puisque ceci aboutit à des coûts prohibitifs (i.e. 96 bits par accidents). Notons qu'un autre critère de détection pourrait être utilisé (seuil exprimé en ration de la moyenne mobile, distribution des longueurs d'arêtes) afin de traiter le cas des maillages présentant une très grande disparité de longueurs d'arêtes.

Le coût d'un accident est réduit par le biais d'un ratio entier K multiple d'un vecteur résidu différentiel r_1 , intégré au groupe des résidus après prédiction. Pour éviter un artefact visuel à la position $i + 1$, lié à l'amplifi-

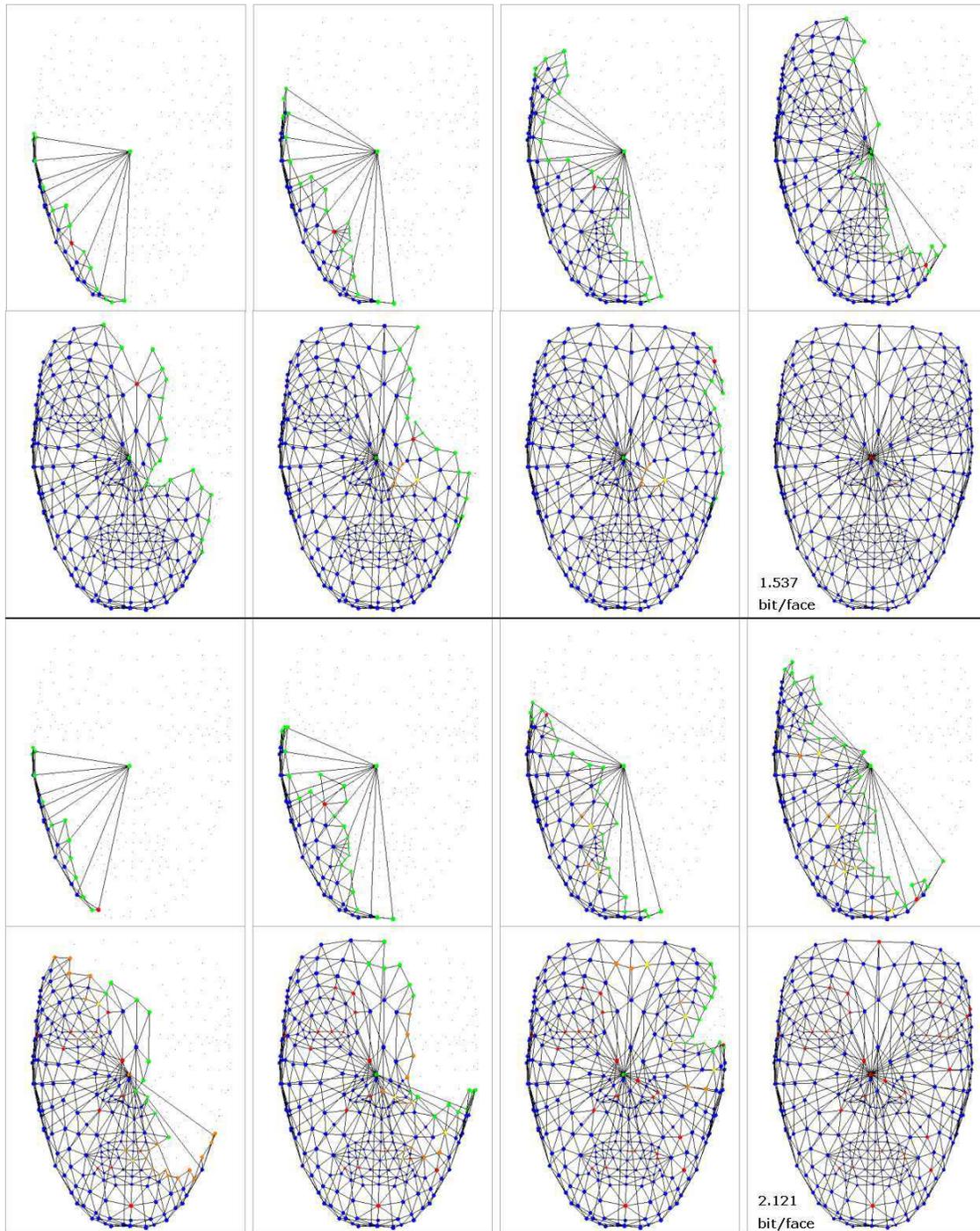


FIG. 3.23 : Codage de la connectivité du maillage *nefertiti* (562 faces, 299 sommets) par conquête d'arêtes. Les deux premières lignes illustrent la séquence de conquête avec notre technique tandis que les deux lignes suivantes illustrent la technique originale. Le sommet ajouté au centre correspond au sommet fantôme, le sommet germe étant choisi arbitrairement. Au cours du codage, la liste de conquête initiale progresse sur le maillage par conquête des arêtes adjacentes aux sommets pivots successifs. La technique présentée génère une seule séparation de liste et 302 codes de valences pour aboutir à 1.537 bit/face. La technique originale génère quant à elle 34 codes de séparation de liste et 344 codes de valences pour aboutir à 2.121 bit/face. Dans les deux cas le codage est effectué en 30 ms sur Pentium II 266 MHz (soit 18 733 faces/s).

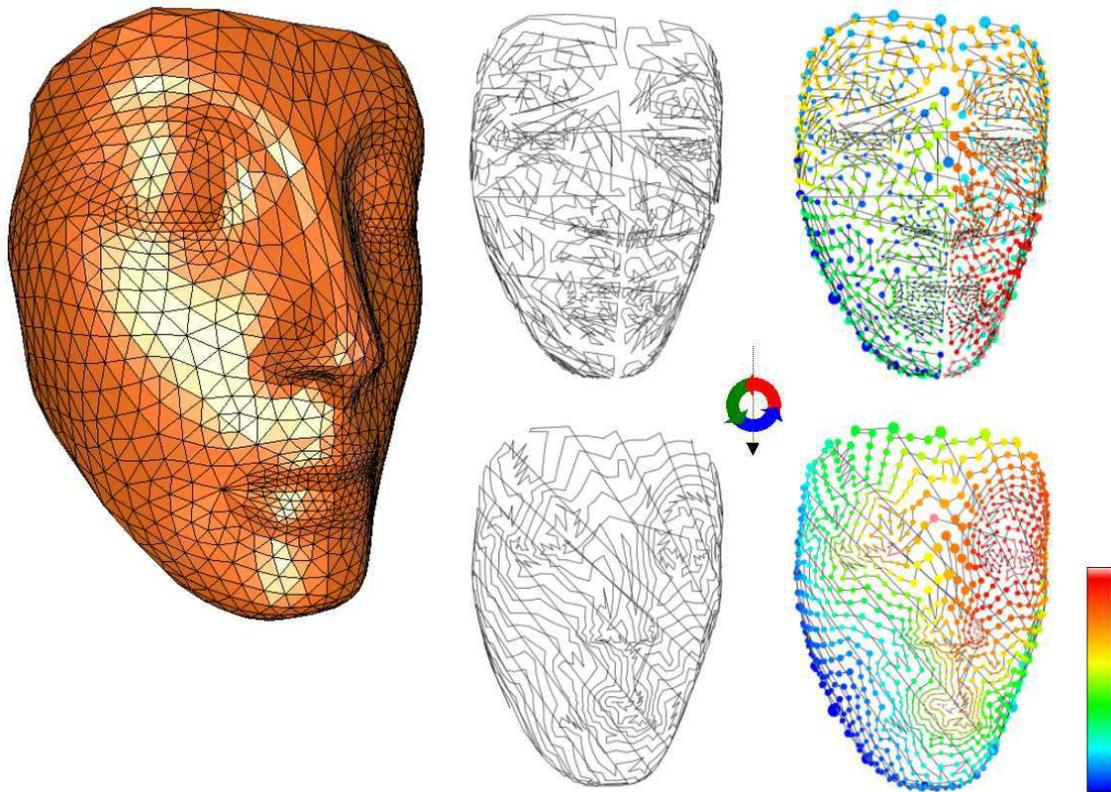


FIG. 3.24 : L'algorithme de codage de la connectivité réordonne les sommets tout en favorisant la proximité géométrique de deux sommets successifs. A gauche : le maillage original. A droite : un réseau de segments relie les sommets tels qu'ils sont stockés dans le mode représentation avant (en haut), puis après (en bas) le codage de la connectivité. On visualise ainsi l'effet de la conquête sur les arêtes adjacentes aux sommets pivots successifs : on obtient une meilleure cohérence après réordonnancement, ce qui incite au codage différentiel ou par prédiction. Sur cet exemple, les lignes reliant les sommets du bord au sommet fantôme ne sont pas représentées, et les grands segments obliques transitent par le sommet fantôme au cours de l'encodage.

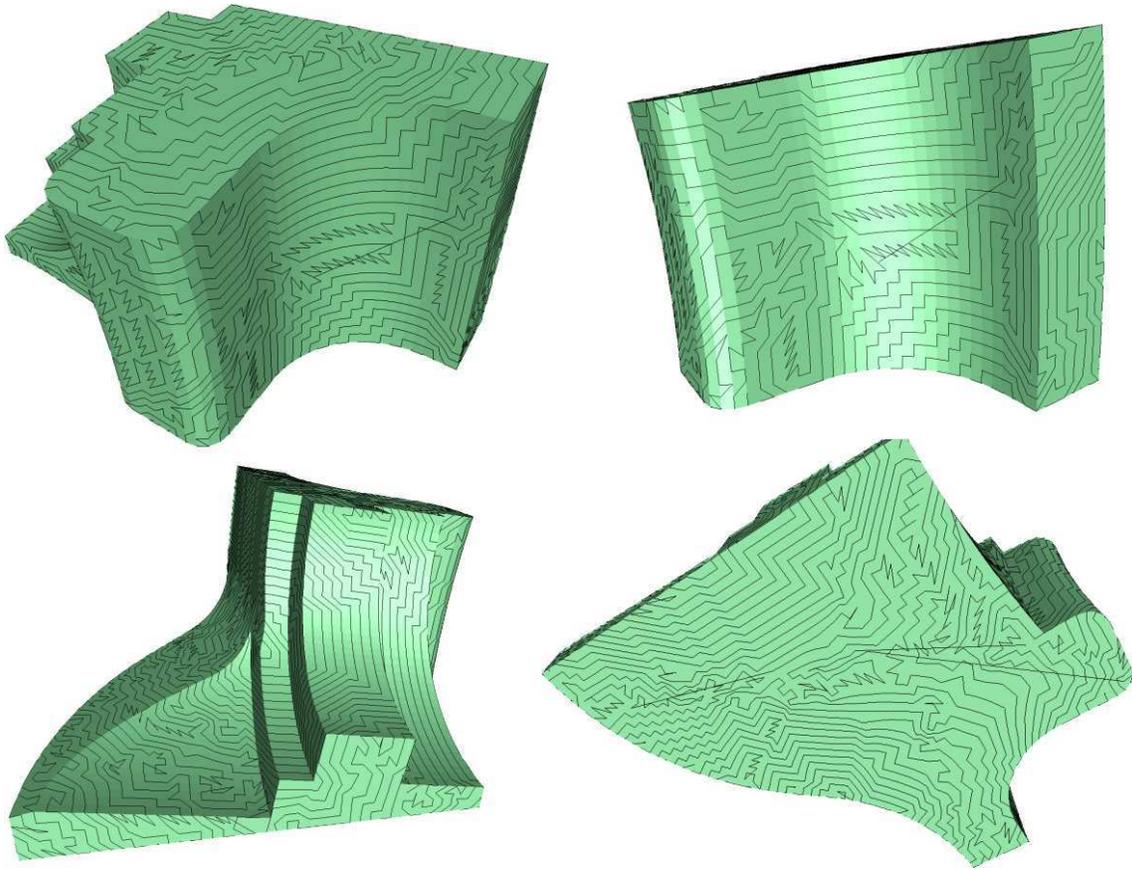


FIG. 3.25 : Maillage *fandisk*. Au cours du codage de la connectivité, les sommets sont réordonnés à partir d'un sommet pivot germe. En traçant un réseau de segments reliant consécutivement tous les sommets dont la position est à coder, on observe des régions régulières et d'autres plus oscillantes. Les longs segments reliant deux régions éloignées du maillage correspondent à des accidents.

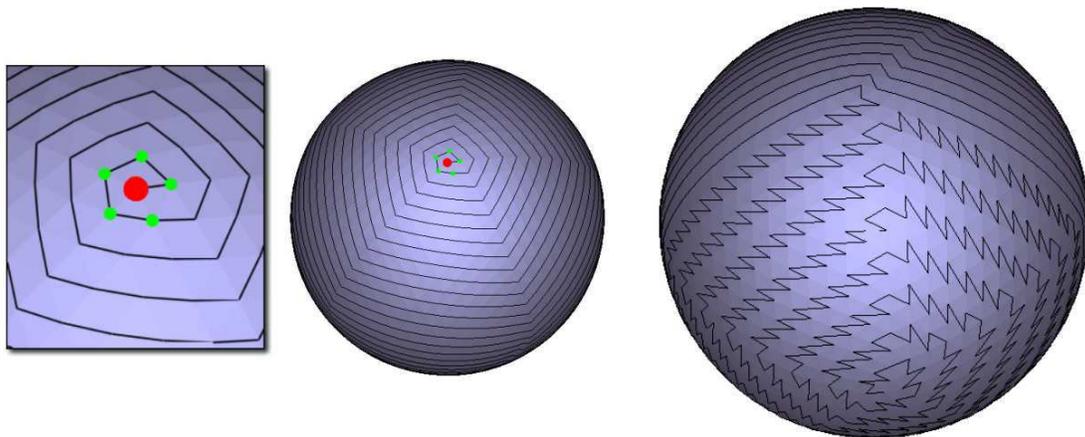


FIG. 3.26 : A gauche, le sommet pivot germe. On visualise au centre la conquête régulière par expansion sur un maillage parfaitement régulier.

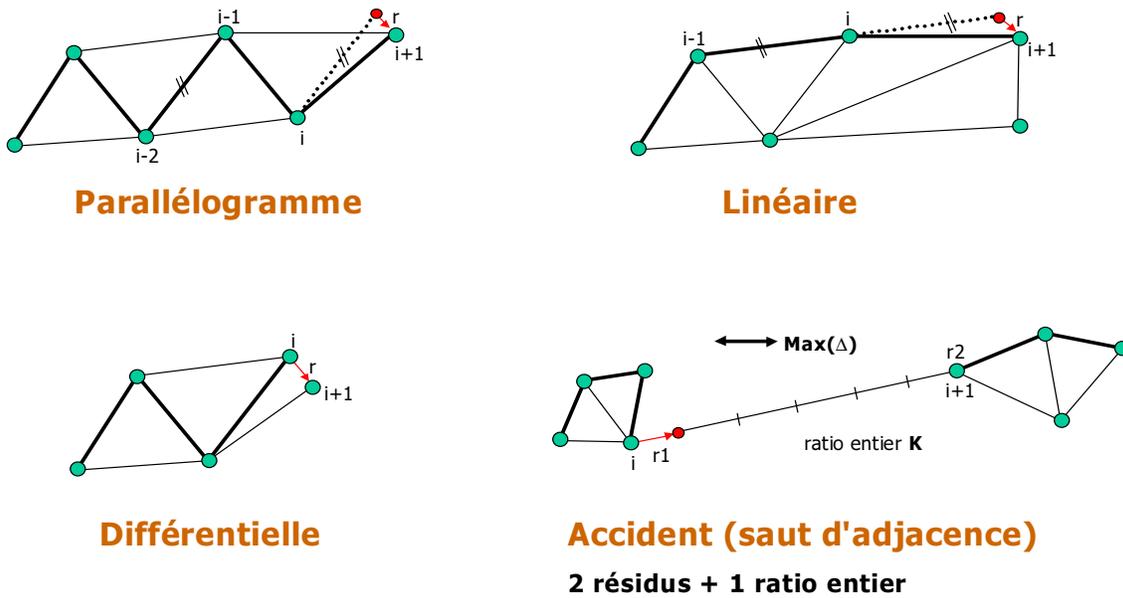


FIG. 3.27 : Quatre modes de prédiction sont employés : parallélogramme, linéaire, différentiel et accident. Considérons la position i codée, la position suivante $i + 1$ est à coder. Pour les trois premiers modes de prédiction, seul le résidu r entre la position prédite et la position réelle du sommet $i + 1$ est à coder. La prédiction parallélogramme nécessite la connaissance des trois antécédents i , $i - 1$ et $i - 2$ et reporte le segment $[i - 2; i - 1]$ à partir du sommet i . La prédiction linéaire nécessite la connaissance des deux antécédents i et $i - 1$ et reporte le segment $[i - 1; i]$ à partir du sommet i . Le mode différentiel ne prédit rien, tandis qu'un accident génère deux résidus successifs et un ratio K entier.

cation par le facteur K de l'erreur induite par la quantification du résidu r_1 , un second résidu différentiel r_2 est ensuite généré afin de borner l'erreur de reconstruction. Les ratios entiers K_1, \dots, K_n sont ensuite regroupés et compressés par un codeur entropique puisque leur distribution n'est pas uniforme. Les performances d'un tel codage s'expliquent par le gain sensible en excursion des résidus après prédiction. Un tel codage des accidents garantit notamment qu'un résidu ne peut excéder la longueur maximale des arêtes du maillage initial.

Quantification et codage entropique

La prédiction nécessite le codage de résidus signés. Les bits de signe, équiprobables, sont extraits et codés sans compression. Après une opération de normalisation, les valeurs absolues des résidus sont quantifiées de façon linéaire et indépendante suivant une précision P (exprimée en nombre de bits) déterminée à l'avance. La position du premier sommet est stockée sans compression, ainsi que le facteur d'échelle à appliquer lors du décodage sur chaque coordonnée. En pratique la distribution des valeurs après quantification est non uniforme et incite au codage entropique de ces dernières (voir première courbe de la figure 3.29).

Raffinements géométriques

A ce stade du codage les trains binaires générés permettent de reconstruire un maillage dont la fidélité au modèle original dépend du niveau de quantification paramétré (voir résultats en section 3.5.4). Pour permettre au décodeur de retrouver la précision originale progressivement, il s'agit de coder les raffinements géométriques des positions des sommets. Cette opération est réalisée indépendamment par sommet, et exploite une version apparentée au codage par plans de bits utilisé en compression d'image, mais adaptée à des nombres flottants. La technique s'inspire partiellement des travaux de Li et Kuo [LK98a].

Au départ, l'erreur maximale de reconstruction est mesurée pour tous les sommets et pour chaque coordonnée. Les erreurs sont ensuite normalisées dans l'intervalle $[0;1]$, puis une quantification successive est réalisée indépendamment par coordonnée de manière à générer des expansions binaires de longueur $(32-P)$ par sommet et par coordonnée. Le tableau 3.3 illustre une expansion binaire d'un ensemble de nombres compris dans l'intervalle $[0;1]$.

Nombre $[0;1]$	Expansion binaire : 32 bits	Ordre
0.001000000	000000000 1 0000011000101011100101	9
0.999899983	1 11111111111111111111111111111111	0
0.021000000	00000 1 01011000000110010011000101	5
0.101000004	000 1 1001110110111100110001011110	3
0.012000000	000000 1 1000100101000001010111011	6
0.005000000	0000000 1 010001111011011001111000	7
0.000710000	000000000 1 011101000100100000011	9
0.000200000	000000000000 1 1010001101111000111	11
0.039999999	0000 1 010001111011011001111000000	4
0.001000000	000000000 1 0000011000101011100101	9
0.000000300	00000000000000000000 1 0100001000	21
petit nombre	000000... 1	élevé

TAB. 3.3 : Expansion binaire de nombres normalisés dans l'intervalle unité. L'ordre du premier 1 rencontré dans le train binaire de chaque nombre est appelé ordre de grandeur. Les bits suivants correspondent à des bits de raffinement équiprobables. La taille des plans verticaux de raffinement augmente en même temps que la précision souhaitée. Il s'avère ainsi possible de reconstruire un nombre compris dans l'intervalle unité à partir de son ordre de grandeur et des bits de raffinements.

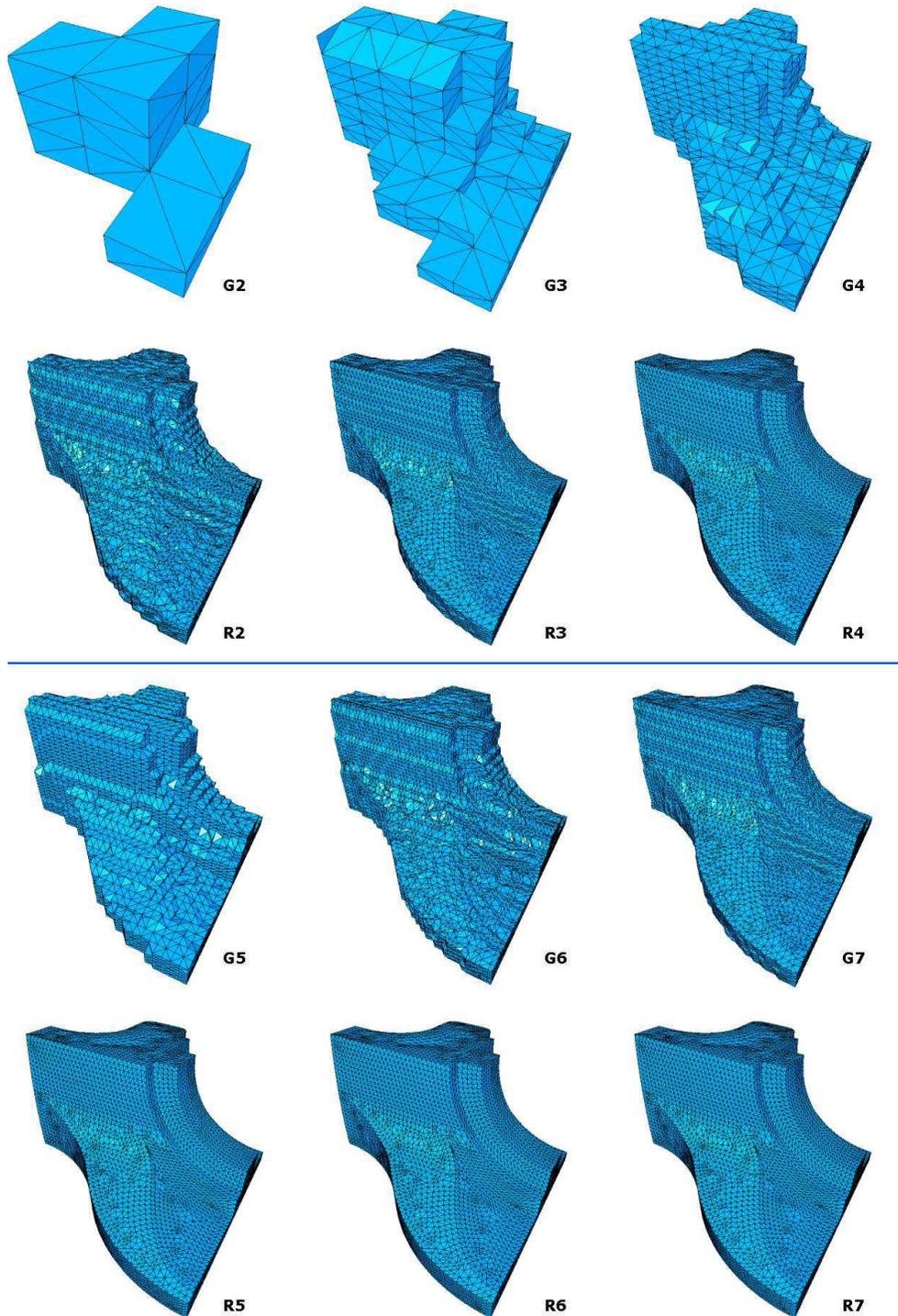


FIG. 3.28 : Comparaison qualitative entre la quantification globale (repérée par la lettre G) et la quantification des résidus après prédiction (lettre R) entre 2 et 7 bits. On constate un gain sensible dès le premier niveau de quantification par le mode de quantification des résidus, et un codage quasi sans perte visuelle dès le niveau 6, 11 bits étant nécessaires pour le mode de quantification globale.

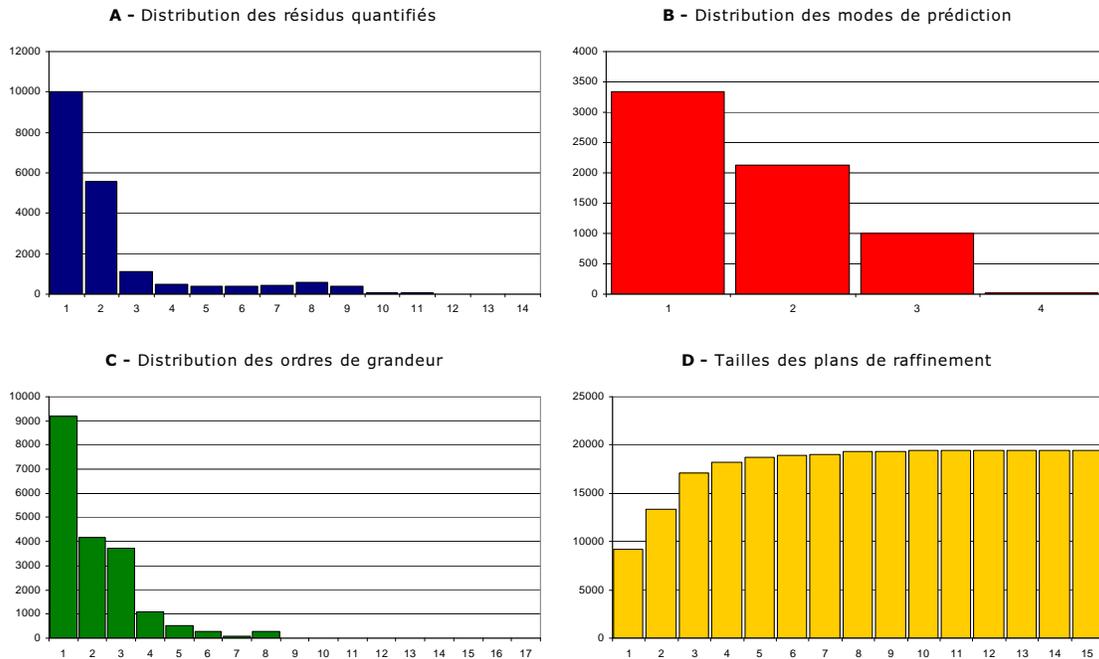


FIG. 3.29 : Maillage *fandisk*. [A][B][C] les distributions des valeurs absolues des résidus après quantification sur 4 bits, des modes de prédiction et des ordres de grandeur des expansions binaires sont non uniformes et donc compressées par codage entropique. [D] Les tailles des plans binaires successifs de raffinement augmentent progressivement avec le gain en précision. Les bits correspondants sont codés indépendamment et sans compression par ordre croissant dans l'expansion.

Il convient ensuite de coder les ordres de grandeur séparément via un codeur entropique puisque leur distribution est non uniforme (voir troisième courbe de la figure 3.29). Les bits de raffinement étant équiprobables, sont quant à eux stockés tels quels. Ils sont disposés par ordre de grandeur dans le train binaire de façon à organiser les raffinements en trains binaires de plus en plus importants (voir quatrième courbe de la figure 3.29). Cette organisation correspond bien à une représentation échelonnée dont la granularité n'est pas régulière (voir la scalabilité obtenue sur le maillage *fandisk* pour quatre niveaux de quantification figure 3.30). L'annexe B résume le pseudo-code du codage de la géométrie, tandis que la section suivante décrit la méthode de décodage des maillages.

3.5.3 Décodage

Cette section décrit les méthode de décodage de la connectivité et des positions. On rappelle que les données sont organisées de la manière suivante :

1. Connectivité
2. Géométrie : niveau de base des positions : P_0
3. Raffinement 0 en position
4. Raffinement 1 en position
5. Raffinement 2 en position
6. etc...

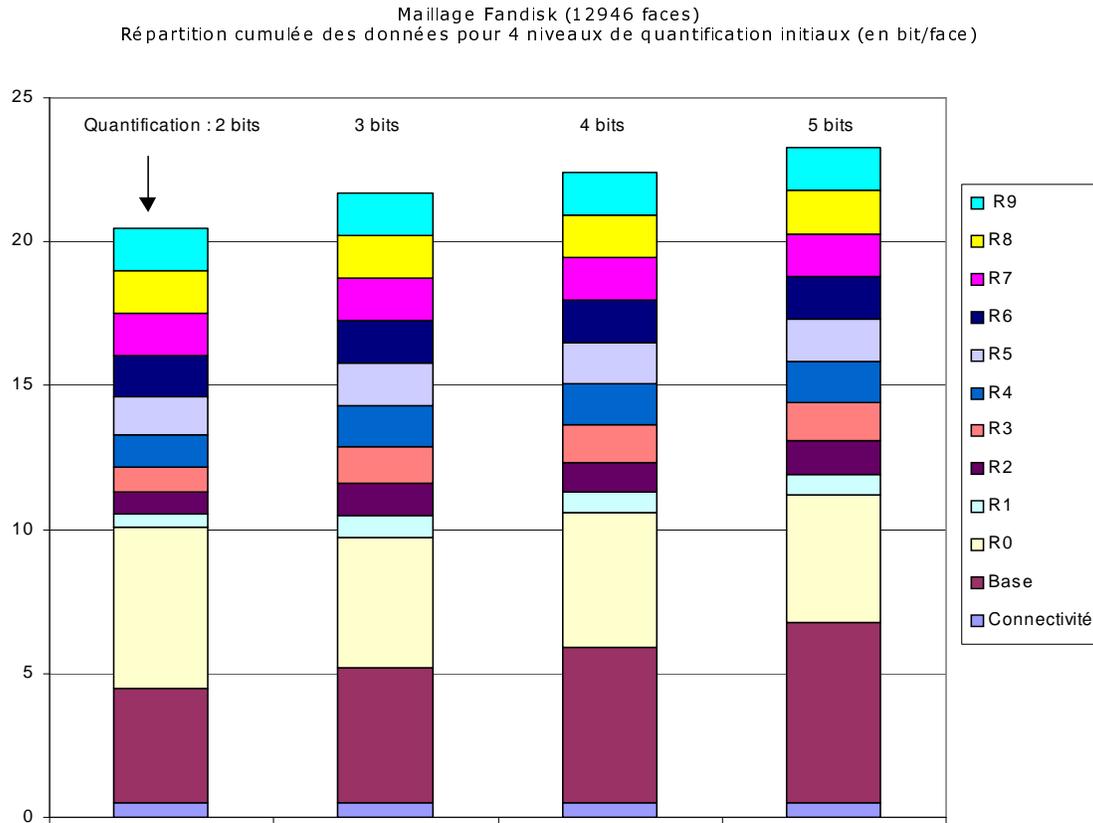


FIG. 3.30 : Maillage *fandisk*. De gauche    droite : 4 niveaux de quantification initiaux, respectivement 2, 3, 4 et 5 bits. La largeur des empilements illustre les quantit  s de donn  es contenues dans les trains binaires suivants : connectivit  , positions apr  s le niveau de quantification initial (Base), premier raffinement R_0 puis chaque plan de raffinement R_1, \dots, R_9 . Les r  sultats associ  s (section 3.5.4) sur ce maillage montrent les diff  rentes qualit  s de reconstruction obtenues apr  s transmission des deux premiers trains binaires, puis des couches binaires successives de raffinement. La taille du premier raffinement R_0 s'explique par la pr  sence dans le train binaire correspondant des ordres de grandeur des expansions binaires apr  s codage entropique. Cette information correspond    l'ensemble des chiffres 0 situ  s    gauche des premiers 1 rencontr  s dans l'expansion binaire (voir tableau pr  c  dent). Il est    noter que l'on pourrait obtenir une granularit   plus fine en transmettant simplement les plans verticaux de raffinement au prix d'un c  t de codage final plus   lev  .

Le décodeur est capable de décompresser les données de structure et d'attributs indépendamment. Toutefois, il commence par décompresser le niveau de base des positions P_0 , puis décode la connectivité de manière à présenter le plus tôt possible une version visuellement acceptable du maillage à décoder.

Connectivité

Le décodeur régénère les deux vecteurs de données à partir d'un décodeur entropique et des bits de signe codés sans compression. Son fichier d'en-tête comprend le nombre de sommets à décoder, la présence d'un fantôme et sa valence. Les faces sont reconstruites à partir de l'information de valence des sommets, des codes réservés et des codes d'offsets. Afin d'assurer au décodeur un comportement similaire au codeur, il doit prendre les mêmes initiatives que le codeur, y compris lorsqu'aucune information n'a été transmise à cet effet. On parle alors de synchronisation codeur / décodeur, ce qui représente une part non négligeable du travail de mise au point d'un *codec* (codeur / décodeur).

Initialisation

Après le décodage des positions, la phase d'initialisation comprend cinq étapes :

1. Un nouveau maillage composé uniquement de sommets est généré. Ces derniers sont ordonnés dans l'ordre de traitement du codeur des positions.
2. Une structure de données peu volumineuse en mémoire est utilisée, puisque seule l'information d'adjacence inter-sommets est requise, contrairement aux méthodes couramment utilisées qui construisent les mêmes structures que le codeur.
3. Un sommet fantôme est généré, on ne lui renseigne que sa valence, et il n'est pas directement ajouté au maillage.
4. Aucune liste d'arêtes n'est générée, il n'est donc pas nécessaire de réaliser de tri.
5. Une liste d'arêtes vide est générée et introduite dans la pile. Cette liste est appelée liste active. Les trois premiers codes sont lus dans le vecteur de codes des valences. Les trois premiers sommets du maillage reçoivent leur valence respective, et une face correspondante est construite (figure 3.31). Lorsqu'un sommet prend connaissance de sa valence, il construit une liste de références de ses sommets adjacents, toutes initialisées à des valeurs nulles. Ensuite lors de la création de la première face, chaque sommet est déclaré adjacent aux deux autres et les listes de références aux sommets commencent à se compléter. Le premier sommet est alors désigné comme pivot. La liste active est orientée dans le sens anti-horaire, et ainsi prête pour la reconstruction de la connectivité du maillage.

La reconstruction

La liste active commence alors une reconstruction autour de son sommet pivot en générant les faces et en renseignant les sommets de leur valence lorsqu'ils sont retirés du vecteur de codes. De plus, une ou plusieurs faces peuvent être générées sans qu'aucun code ne soit transmis. Après chaque retrait et analyse d'un code, l'algorithme peut prendre la décision de supprimer le sommet pivot de la liste active si ce dernier est complet et réitère ce processus tant qu'un sommet complet se trouve dans la liste de reconstruction.

Lorsqu'une valeur est retirée du vecteur de codes, cinq cas peuvent se produire :

- le code est supérieur ou égal à 4 : il s'agit d'une valence incrémentée d'une unité. Elle correspond à la valence du sommet S relié à la première arête libre du pivot suivant l'orientation anti-horaire. Une face est alors créée par *maillage* du pivot, de son prédécesseur et du sommet S . Cet ordre de description des sommets garantit une orientation cohérente de la face. Les adjacences entre sommets sont mises à jour. Le sommet est ajouté dans la liste avant la position du pivot et le nombre d'arête libre des sommets concernés est respectivement décrémenté (figure 3.32),

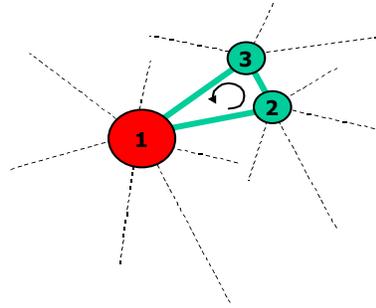


FIG. 3.31 : Décodage de la connectivité. Les trois premiers codes [8][6][5] permettent de créer une face, et d'informer les sommets de leur valence respective. Le premier sommet est nommé pivot. Les arêtes libres attendent une future connexion. A chaque *maillage* d'un sommet, son nombre d'arêtes vives est décrémenté. Sur cet exemple il reste 6 arêtes libres sur 8 en ce qui concerne le sommet pivot.

- code = 0 : il s'agit d'un code réservé à l'insertion d'une occurrence du fantôme dans la liste active. Le fantôme est inséré dans la liste et les adjacences sont mises à jour. Le nombre d'arêtes vives des sommets concernés est décrémenté, et aucune face n'est créée à cette occasion,
- code = 1 : il s'agit d'un code réservé à la séparation de la liste active en deux listes. L'offset signé permet de retrouver rapidement le sommet S à relier avec le pivot courant. Une face reliant le pivot, son prédécesseur et S est créée. Les adjacences correspondantes sont mises à jour et les listes sont séparées en deux listes de la même manière qu'au cours du codage. La nouvelle liste est élevée dans la pile, et deviendra active après le dépilement des listes la précédant. Aucune information n'est nécessaire pour changer de liste active puisque le décodeur est apte à déterminer si la liste active est vide. La nouvelle liste hérite du pivot de la liste active, ou de son premier sommet complet rencontré s'il existe (voir figure 3.34), et la liste courante se voit affecter le sommet S comme pivot,
- code = 2 : il s'agit d'un code réservé à l'information de sommet contextuellement complet. On force la liste active à supprimer son pivot même s'il n'est pas complet, à en changer et à générer une face (voir figure 3.34). De cette manière il n'est pas nécessaire de reconstruire une structure de données basée arêtes au décodeur puisque seul un nombre d'arêtes libres par sommet suffit. Le gain mémoire et la vitesse d'exécution sont significatifs,
- code = 3 : il s'agit d'un code réservé à la fusion de deux listes (ce cas se produit pour des surfaces de genre supérieur à zéro). Ce code est suivi d'un offset o_1 exprimé en nombre de listes dans la pile et d'un second offset o_2 dans la liste. Les deux listes sont fusionnées, la liste indexée par o_1 est supprimée de la pile puis la reconstruction continue normalement.

Après la décision relative à chaque retrait d'un code, le décodeur regarde si le sommet pivot est complet, c'est à dire si la liste de ses sommets adjacents est entièrement renseignée. Si c'est le cas une face reliant le sommet précédent, le sommet suivant et le pivot est créée (figure 3.33). Le pivot est ensuite supprimé de la liste, et les adjacences sont mises à jour. Le sommet suivant l'ancien pivot est affecté comme nouveau pivot, sauf si un sommet complet est détecté sur la liste, c'est en fait exactement la même loi appliquée au cours du codage. Cette opération est itérée tant que le nouveau pivot est complet. Ainsi une région entière peut être maillée avec un coût de codage nul (voir figure 3.34). On notera qu'une telle création de face met à jour l'adjacence de deux sommets, et réduit donc le nombre d'arêtes libres de 1 pour chacun, ce qui tend à rendre les sommets complets.

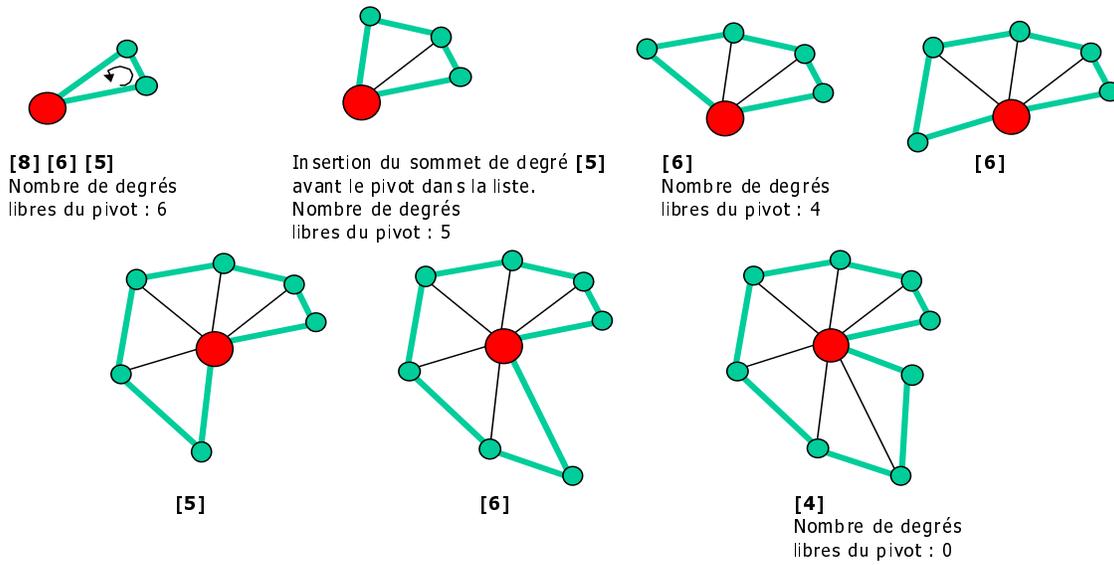


FIG. 3.32 : Les codes [8][6][5][5][6][6][5][6] permettent un maillage progressif des sommets de manière à reformer les faces originales du maillage.

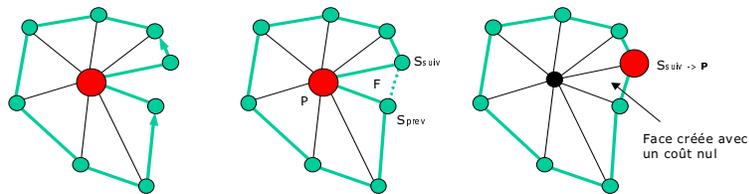


FIG. 3.33 : Aucun code. Le pivot est complet, la face F contenant P , S_{prev} et S_{suiv} est créée avec un coût nul. Le nouveau pivot est ensuite affecté au sommet suivant, ou au premier sommet complet rencontré dans la liste.

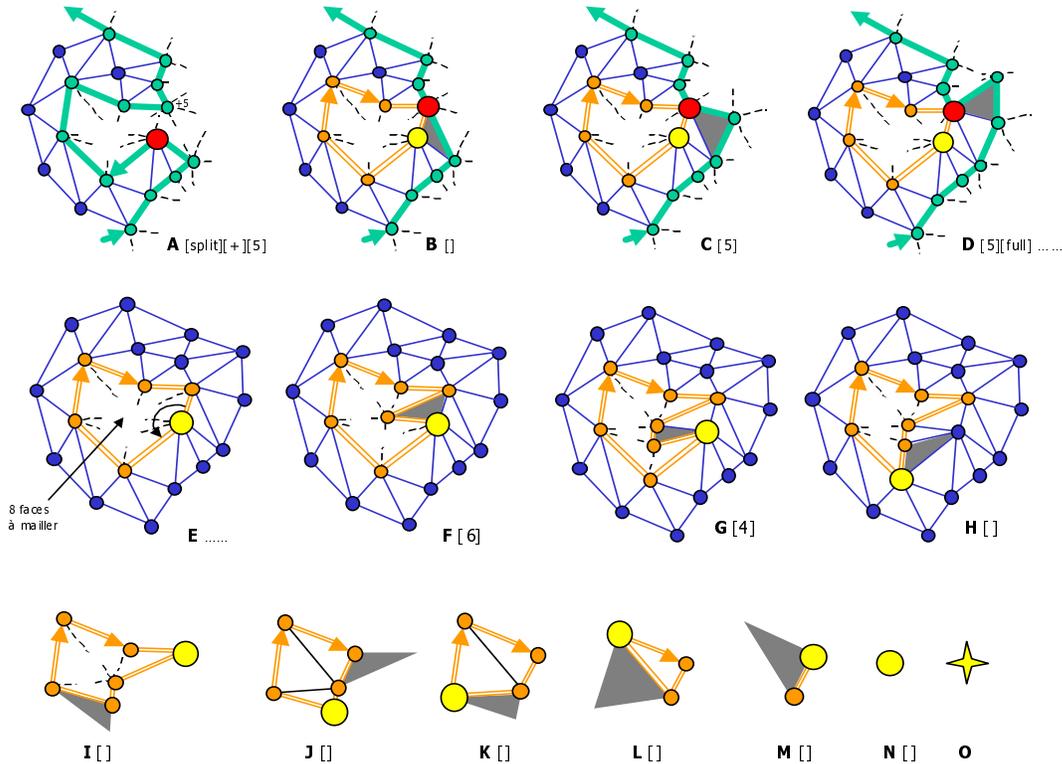


FIG. 3.34 : Reconstruction et séparation de liste. [A] Supposons une telle reconstruction réalisée. Les sommets affichés connaissent leur valence respective. A chaque code de valence, le décodeur retire un sommet du décodeur des positions. Les arêtes en trait discontinu représentent l'incertitude de connectivité sur les sommets tandis que les arêtes en pointillé symbolisent les arêtes reconstruites. A ce stade, il est impossible de prédire avec quels sommets seront connectées les trois arêtes libres du pivot, tous les sommets possédant des arêtes libres étant candidats à la connexion. Cette incertitude représente précisément l'information de *connectivité*. Le code {split,+,5} indique une séparation de liste à la position +5 sur la liste active. [B] Cette information permet de connecter le sommet pivot avec le sommet repéré +5 dans la liste, d'affecter le sommet cible comme pivot de la liste courante et l'ancien comme pivot de la nouvelle liste élevée dans la pile, puis de mailler une face par connexion anti-horaire idoine. Le pivot possède encore 3 arêtes libres. Rappelons qu'au décodeur ne réside aucune information d'arêtes, seul le nombre d'arêtes libres est affecté à chaque sommet. [C] le code {5} déclenche le maillage d'un nouveau sommet de valence 5 avec la liste active. [D] maillage d'un nouveau sommet de valence 5. Il reste une arête libre au pivot, pourtant un code spécifique le force à se comporter comme un pivot complet, ce qui déclenche la création d'une face par maillage anti-horaire du sommet précédent, suivant et le pivot. La reconstruction se poursuit ainsi jusqu'à la complétude de tous les pivots de la liste courante. [E] La liste générée par séparation devient active. Cette liste orientée dans le sens horaire aboutit à une reconstruction de sa région intérieure. La reconstruction s'effectue toujours dans le sens anti-horaire autour du pivot. [F] Le code {6} déclenche la création d'une face par maillage anti-horaire du sommet précédent le pivot, du nouveau sommet de valence 6 et du pivot. [G] Maillage d'un nouveau sommet de valence 4. [H] Le pivot est complet, ce qui déclenche le maillage d'une face et la réduction du nombre d'arêtes libres associé. [I] Le nouveau pivot est à nouveau complet. [J] Un sommet complet a été détecté dans la liste, il est affecté comme pivot, ce qui évite un code de séparation de liste, un bit de signe et un offset. [K][L][M][N][O] A chaque itération un sommet complet est détecté, maillant ainsi la région restante avec un coût nul.

Géométrie

Les données de position sont organisées de la manière suivante :

- premier sommet : $\{x, y, z\}$,
- facteur d'échelle global : $\{s_x, s_y, s_z\}$,
- précision de quantification exprimée en bits,
- largeur L de la moyenne mobile,
- modes de prédiction compressées par codage entropique,
- bits de signe des résidus,
- valeurs absolues des résidus après quantification et compression,
- drapeau binaire indiquant la présence de raffinements,
- facteur d'échelle des raffinements,
- bits de signe des ordres de grandeur,
- les ordres de grandeur après codage entropique,
- train binaire du raffinement 0,
- train binaire du raffinement 1,
- train binaire du raffinement 2,
- etc...

Les premières données sont directement lues dans le train binaire, tandis que les données codées par le codeur entropique sont décompressées via le décodeur correspondant. La précision de quantification permet d'initialiser le décodeur entropique, et les facteurs d'échelle pour chaque coordonnée permettent de replacer les sommets à leur position originale, modulo la perte de précision liée à la quantification.

Décodage d'une position

La moyenne mobile M des L longueurs d'arête précédentes est calculée. Le résidu est signé, mis à l'échelle, multiplié par M et appliqué au sommet prédit en fonction de la loi indiquée par le bit de prédiction. La figure 3.35 illustre le principe sur deux modes de prédiction : parallélogramme et accident.

Décodage des raffinements

En présence de bits de raffinements, les positions des sommets sont ajustées indépendamment. Un vecteur de trains binaires est généré, la longueur de chacun étant initialisée à 32 bits. Les valeurs de ces trains binaires sont initialisées en fonction des ordres de grandeurs et le premier niveau de raffinement est reconstruit via le facteur d'échelle. Chaque ordre de grandeur permet ainsi la génération d'un train binaire, et sans plus d'information le bit suivant l'ordre de grandeur est initialisé à 1, ce qui permet de favoriser le milieu de l'intervalle comprenant la valeur finale. Cependant les prochains bits de raffinements pouvant remettre en cause ce choix, tout en étant toujours suivis du chiffre 1 pour la même raison (voir exemple tableau 3.4).

3.5.4 Résultats

Cette section présente séparément les résultats de codage de la connectivité et de la géométrie afin de mettre en évidence les gains obtenus sur chaque type de données.

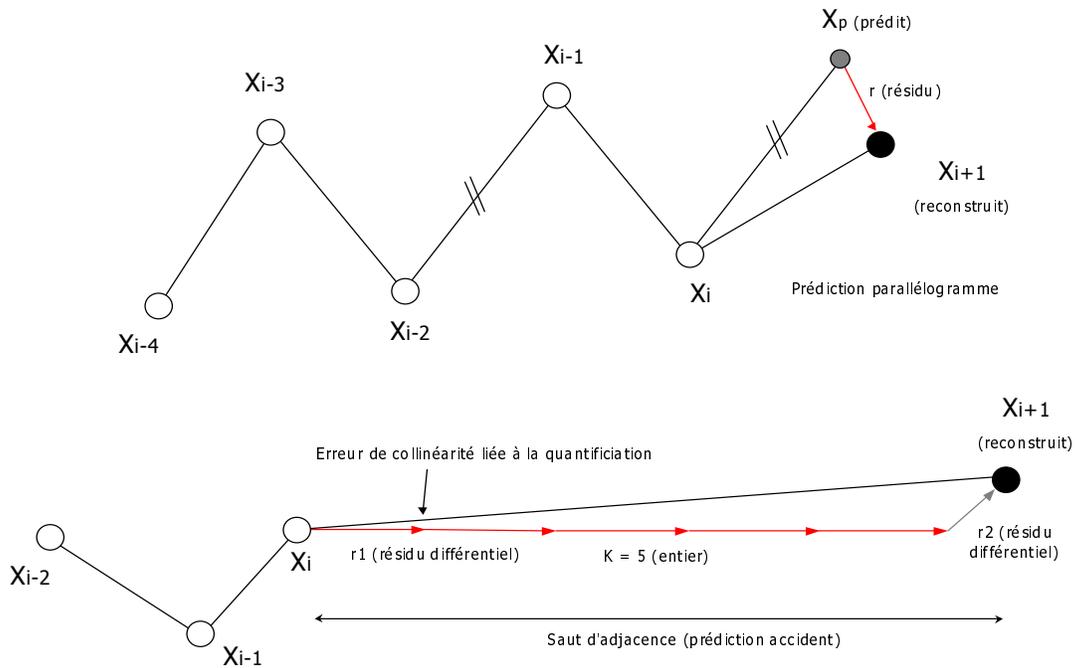


FIG. 3.35 : En haut : décodage d'une position avec la prédiction parallélogramme et un résidu r . Le décodeur calcule en permanence la moyenne mobile M des arêtes reconstruites sur une largeur L exprimée en nombre d'arêtes précédant le sommet courant X_i . Le résidu r est signé, mis à l'échelle, multiplié par M puis appliqué au sommet X_p calculé par prédiction parallélogramme à partir des sommets reconstruits X_{i-2} , X_{i-1} et X_i . En bas : décodage d'une position avec une prédiction de type accident, un ratio entier K et deux résidus successifs r_1 et r_2 . Le résidu r_1 est signé, mis à l'échelle et multiplié par le ratio entier K (5 sur cet exemple). On constate une erreur de colinéarité liée à l'erreur de quantification du résidu r_1 , tandis que le second résidu r_2 permet de rattraper l'amplification de l'erreur de quantification provoquée par le ratio K .

Ordre de grandeur	Signification	Train binaire	Valeur reconstruite
0	compris entre 0.5 et 1	110000000000...	0.75
1	compris entre 0.25 et 0.5	011000000000...	0.375
2	compris entre 0.125 et 0.25	001100000000...	0.1875
...	...	000110000000...	...
...	...	000011000000...	...
Ordre de grandeur	Raffinement	Train binaire	Valeur reconstruite
0	-	110000000000...	0.75
0	1	111000000000...	0.875
0	0	101000000000...	0.625
0	11	111100000000...	0.9375
0	10	110100000000...	0.8125
0	111	111110000000...	0.96875
0	110	111010000000...	0.90625
...
3	11	000111100000...	0.1171875
...

TAB. 3.4 : Décodage des raffinements : chaque ordre de grandeur permet de reconstruire un train binaire. Le bit 1 est ajouté directement après le bit de l'ordre de grandeur afin de favoriser le milieu de l'intervalle plutôt qu'une extrémité. Le bit réceptionné suivant peut remettre en cause ce choix.

Connectivité

Compression

Le tableau 3.5 rend compte des résultats obtenus par la méthode de conquête par arêtes, et la courbe 3.36 présente le gain obtenu. On constate notamment une réduction sensible du nombre de codes de séparation de liste, obtenue par l'encouragement des sommets complets comme candidats au pivot (voir courbe 3.37). Il en résulte une réduction des coûts de codage, et ce malgré la présence de nouveaux codes réservés aux sommets contextuellement complets. Les maillages correspondants sont présentés dans les figures 3.38, 3.39, 3.40, 3.41 et 3.42. La figure 3.43 montre l'influence de la face germe choisie sur le coût de codage du maillage *susie*. On constate une variation faible en fonction de la face choisie, toutefois le choix d'une face germe influence le coût de codage en cas de très faible complexité géométrique. Dans ce dernier cas il est préférable de choisir une face du bord si le maillage en comporte afin de limiter le nombre d'appels au code *sommet fantôme*.

Vitesse

La vitesse de codage s'exprime couramment en nombre de faces par seconde. Les vitesses du tableau 3.6 sont mesurées sur un processeur de type Pentium II 266 MHz, avec 256 Mo de mémoire. On note une vitesse de décodage sensiblement plus élevée que la vitesse de codage, les codes des sommets contextuellement complets ayant été définis à cet effet.

Mémoire

Les coûts mémoire présentés dans le tableau 3.7 sont théoriques, ils donnent une idée de l'ordre de grandeur des quantités de mémoire utilisées pour le maillage original, le codage et le décodage. On notera un coût sensiblement réduit lors du décodage (essentiellement lié au code des sommets contextuellement complets), ce qui rend envisageable la décompression sur un terminal de faible capacité.

Maillage	Faces	nombre de codes split		coût (bit/face)		Gain
		T & G	Notre méthode	T & G	Notre méthode	
Nefertiti irregular 1	3893	194	24	1.05677	0.754	28 %
Nefertiti irregular 2	15533	398	26	0.453	0.27	40 %
Nefertiti irregular 3	68780	810	27	0.245	0.149	40 %
Nefertiti regular 1	4952	161	10	0.811	0.578	28 %
Suzie	867	72	6	2.3	1.72	25 %
Fandisk	12946	247	5	0.676	0.521	22 %
Bunny	29783	354	30	1.4	1.02	27 %
Venus	1396	83	3	1.71	1.3	24 %
Rhombi regular base	44	2	0	4.227	2.568	72 %
Rhombi regular 1	176	4	0	1.477	0.96	35 %
Rhombi regular 2	704	8	1	0.511	0.399	22 %
Rhombi regular 3	2816	16	0	0.190	0.088	53 %
Rhombi regular 4	11264	32	2	0.069	0.034	50 %
Rhombi regular 5	45000	64	0	0.023	0.008	65 %
Rhombi regular 6	180000	128	2	0.01	0.00365	63 %

TAB. 3.5 : Résultats du codage de la connectivité. La modification de la stratégie de conquête réduit parfois de manière drastique le nombre de codes de séparation de liste (colonnes 3 et 4) par rapport à la technique originale de Touma et Gotsman [TG98] et la complexité algorithmique associée (séparation, élévation d'une nouvelle liste dans la pile, etc...). Un tel code générant également un bit de signe et un offset, le gain en terme de coût de codage est sensible (voir colonnes 5, 6 et 7 et courbe 3.36).

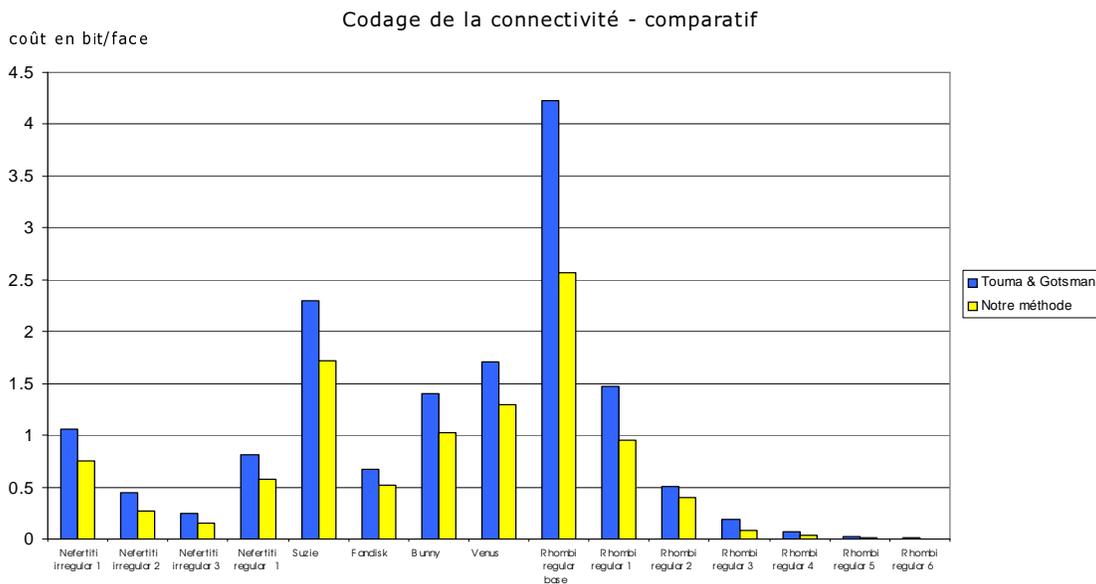


FIG. 3.36 : Courbe comparative des coûts de codage de la connectivité. Les coûts obtenus par la technique sont systématiquement inférieurs aux valeurs obtenues par la technique originale développée par Touma et Gotsman.

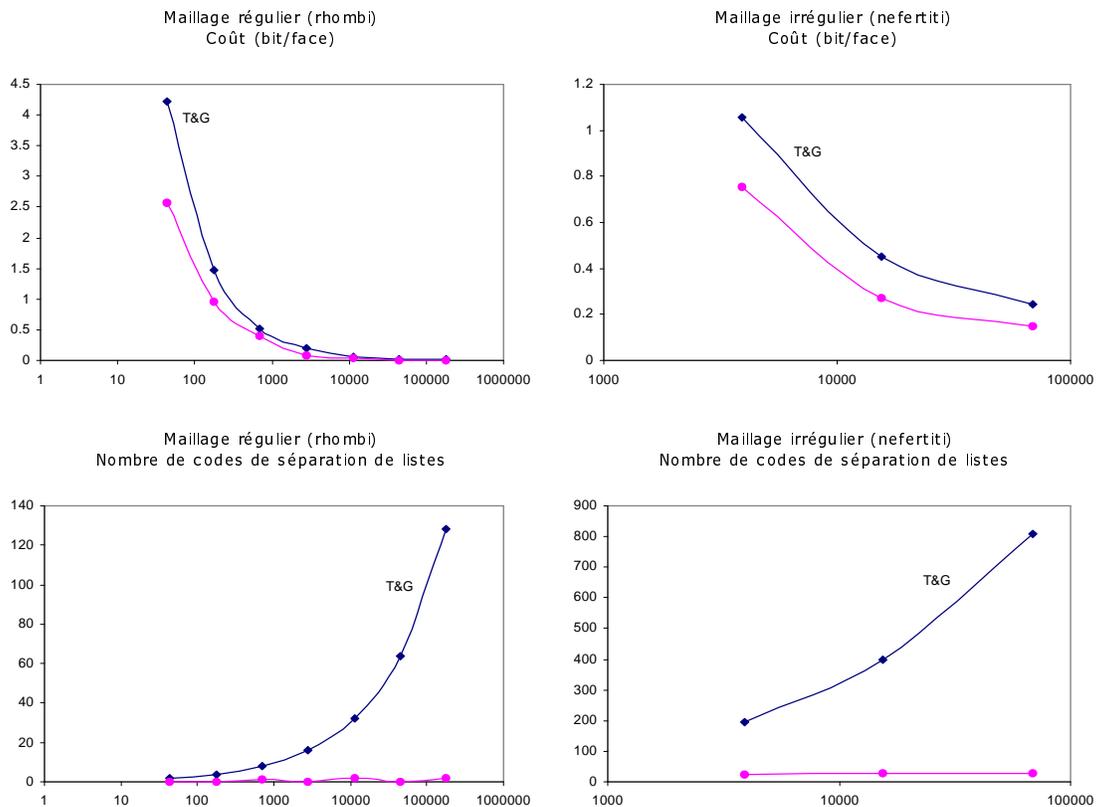


FIG. 3.37 : Première ligne : évolution du coût de codage en fonction du nombre de faces (échelle logarithmique). On constate une diminution sensible du coût, d'autant plus rapide sur un maillage régulier tel que le rhombicuboctaèdre (voir figure 3.42). Le nombre de codes de séparation de liste est réduit de manière drastique et n'augmente pas de manière significative avec le nombre de faces. Cette dernière caractéristique s'avère très intéressante et se distingue nettement du comportement de la méthode de Touma et Gotsman. En pratique cette amélioration s'explique par la suppression pure et simple des codes liés aux encoches formées par la conquête (les séparations de listes avec un offset de plus ou moins deux).

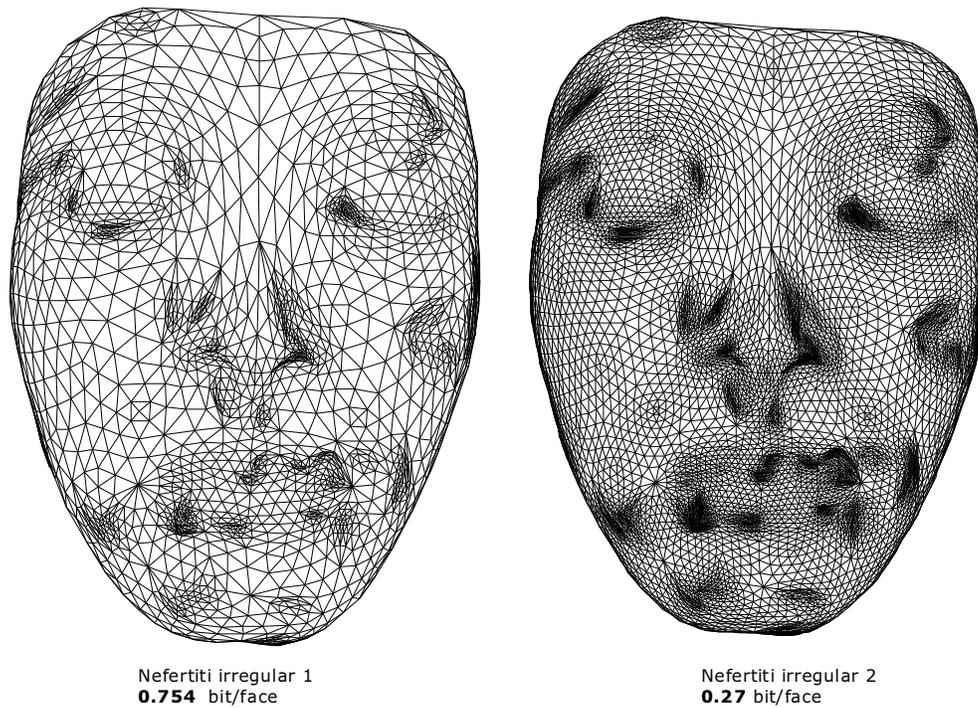
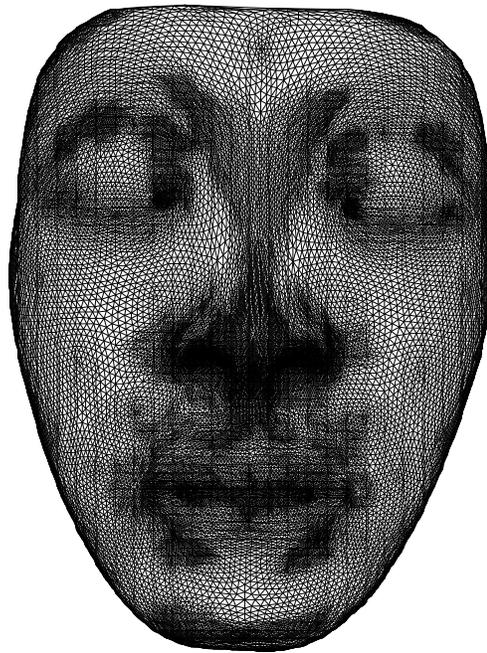


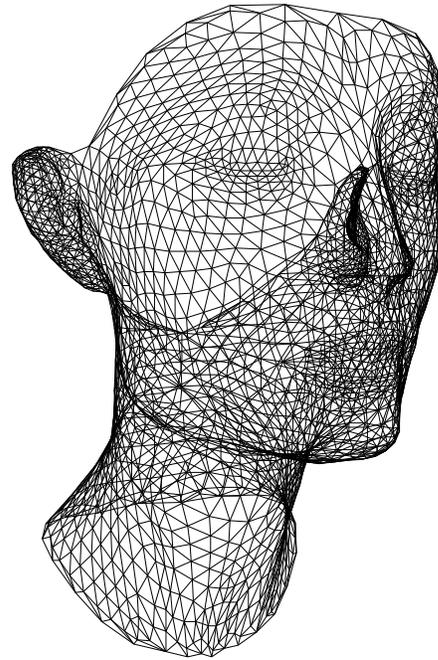
FIG. 3.38 : Codage de la connectivité sur deux maillages irréguliers (environ 4000 et 15000 faces), générés à partir d'une perturbation aléatoire de la densité.

Maillage	Faces	Vitesse codage (faces/s)	Vitesse décodage
Nefertiti irregular 1	3893	14321	29702
Nefertiti irregular 2	15533	7431	13733
Nefertiti irregular 3	68780	4112	9451
Nefertiti regular 1	4952	10854	24636
Suzie	867	16376	42121
Fandisk	12946	8500	16366
Bunny	29783	7800	15402
Venus	1396	17516	46533
Rhombi regular 1	176	18075	20111
Rhombi regular 2	704	17514	35200
Rhombi regular 3	2816	13344	31288
Rhombi regular 4	11264	9434	19387
Rhombi regular 5	45000	4837	10474
Rhombi regular 6	180000	2600	5124

TAB. 3.6 : Vitesse de codage / décodage de la connectivité.

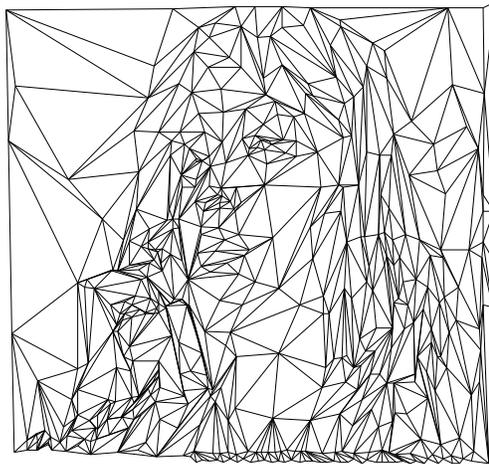


Nefertiti irregular 2
0.149 bit/face

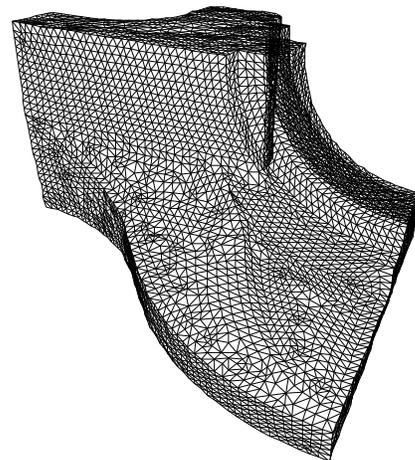


Nefertiti irregular 2
0.578 bit/face

FIG. 3.39 : A gauche : un maillage irrégulier de très haute complexité géométrique (environ 70 000 faces), dont la densité a été amplifiée sur les régions de forte courbure. A droite : un maillage plus régulier (5000 faces).



Suzie
1.72 bit/face



Fandisk
0.521 bit/face

FIG. 3.40 : A gauche : *susie*, un maillage hautement irrégulier (900 faces) utilisé pour le codage des images. A droite : le maillage *fandisk* (13 000 faces), représentatif d'une pièce mécanique.

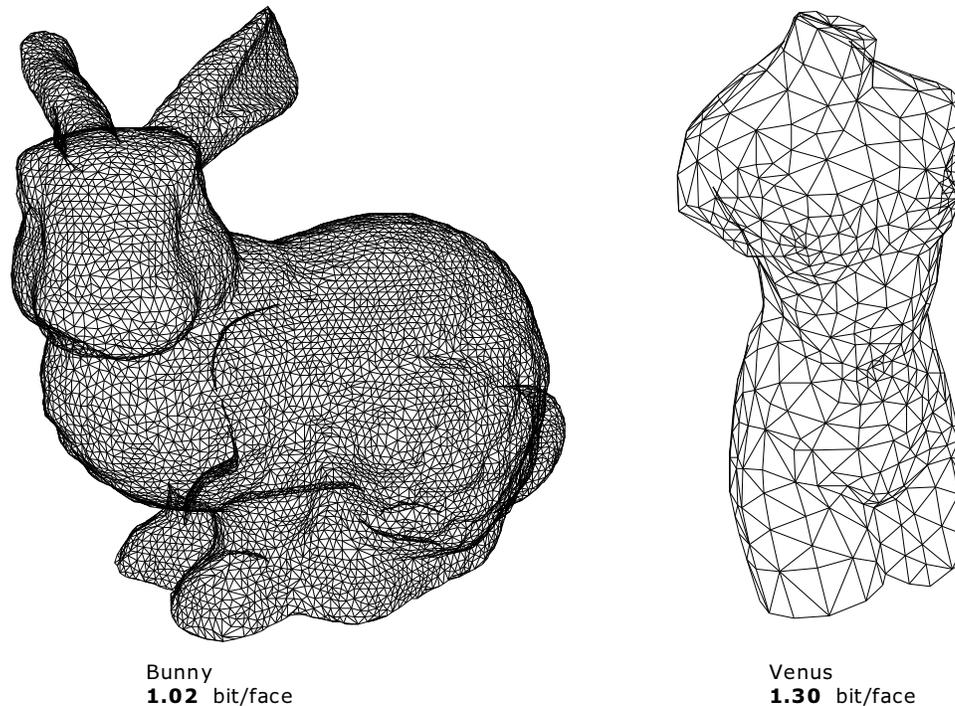


FIG. 3.41 : Deux maillages *bunny* et *venus* couramment utilisés comme références en compression de maillages triangulaires.

Géométrie

La reconstruction progressive du maillage *fandisk* est représentée figure 3.44, ceci à partir de maillages de base dont les résidus après prédiction ont été quantifiés respectivement avec 2, 3, 4 et 5 bits. Les coûts indiqués comprennent celui de la connectivité (0.53 bit/face) et des positions, le niveau de quantification devant être ainsi choisi suivant les contraintes dictées par l'application et la granularité de l'échelonnabilité souhaitée. Le volume de données du premier niveau de raffinement (noté R_0 sur la courbe de la figure 3.30) s'explique par la présence des ordres de grandeurs des résidus finaux et de leurs bits de signe. Ce premier raffinement est aussi le plus significatif visuellement. Les plans de bits de raffinement suivants coûtent en moyenne 1.5 bits/face et permettent une reconstruction finale sans perte visuelle (voir la qualité de reconstruction dès le niveau de quantification 6 figure 3.45). Ces raffinements sont classés par ordre décroissant de l'innovation puisqu'ils sont transmis par ordre de grandeur croissant dans l'expansion binaire. La figure 3.46 montre les résultats obtenus pour le maillage *nefertiti* dès le niveau de reconstruction de base pour les niveaux de quantification variant entre 2 et 5 bits. On constate quasiment aucune perte de la fidélité géométrique dès 5 bits de quantification lorsque l'option de lissage est activée, pour un coût de codage mesuré de 8.53 bits/face. Le changement du mode de prédiction et essentiellement un codage optimisé des accidents explique les gains obtenus. La figure 3.47 illustre le rendu progressif pour les mêmes niveaux de quantification. Le niveau de base du maillage *bunny* (figure 3.48) est codé avec 7.86 bits/face, dont 1.02 pour la connectivité. Sur cet exemple le niveau de quantification a été paramétré à 4 bits (n'oublions pas la présence supplémentaire de 4 modes de prédiction et des bits de signe).

D'une façon générale le coût de codage des positions a été sensiblement réduit (respectivement 13 et 10

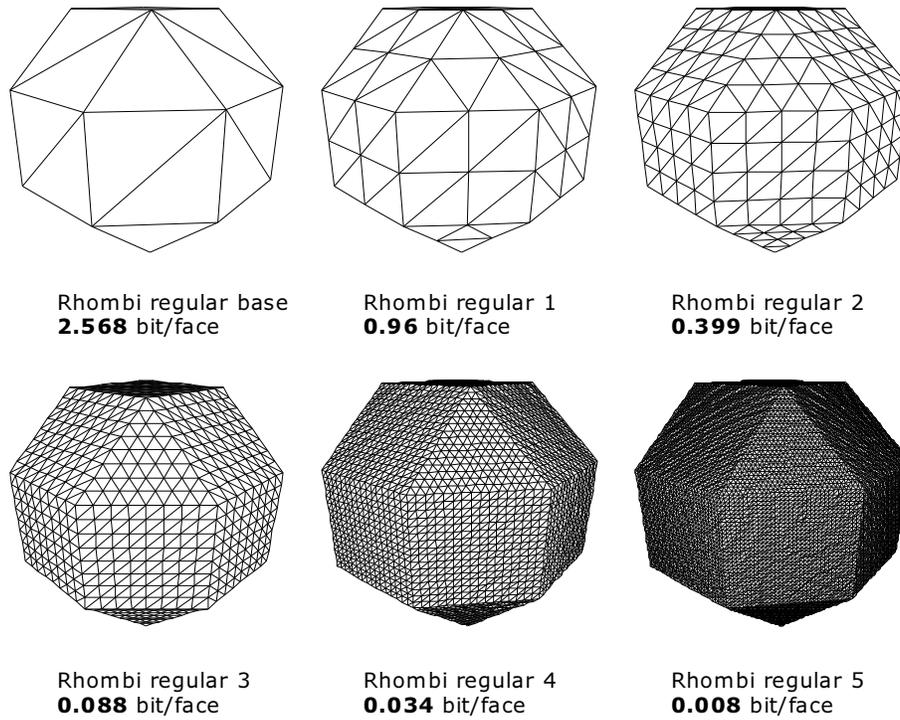


FIG. 3.42 : Un rhombicuboctaèdre parfaitement régulier suivant différents niveaux de complexité géométrique (44, 176, 704, 2816, 11264, et 45000 faces). On constate la suppression du plancher précédemment constaté à 0.2 bit/face.

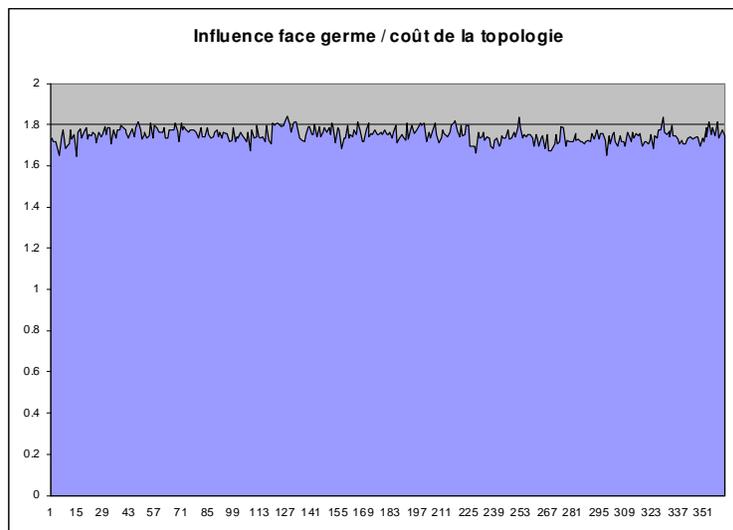


FIG. 3.43 : Influence du choix d'une face germe sur le coût de codage du maillage *susie*.

Maillage	Faces	Original (octets)	Codage	Décodage
Nefertiti irregular 1	3 893	70 320	305 540	117 528
Nefertiti irregular 2	15 533	279 840	1 213 460	466 728
Nefertiti irregular 3	68 780	1 238 604	5 369 164	2 065 092
Nefertiti regular 1	4 952	89 136	383 284	148 589
Suzie	867	16 020	70 800	27 252
Fandisk	12 946	233 052	1 009 972	388 452
Bunny	29 783	537 396	2 333 056	897 396
Venus	1 396	25 284	110 084	42 348
Rhombi regular base	44	816	3 616	1 392
Rhombi regular 1	176	3 192	13 912	5 352
Rhombi regular 2	704	12 696	55 096	21 192
Rhombi regular 3	2 816	50 712	219 832	84 552
Rhombi regular 4	11 264	202 776	878 776	337 992
Rhombi regular 5	45 000	810 000	3 510 000	1 350 000
Rhombi regular 6	180 000	3 240 000	14 040 000	5 400 000

TAB. 3.7 : Coûts mémoire du codage / décodage de la connectivité.

bits/face pour [THLR97] et [TG98]), et l'échelonnabilité obtenue sur ces données autorise une reconstruction progressive au cours de la transmission des plans de bits de raffinement correspondants.

3.6 Codage des textures

Cette section décrit la technique de codage échelonnable d'images fixes développée au sein du laboratoire [LLS98] [Lec99] [LLS99b] [DRLB00]. Des essais ont récemment permis de valider les performances du codage conjoint maillages triangulaires / DCT pour des textures plaquées sur un modèle 3D [LLS99a] [AL99b]. Cette méthode a donc fait l'objet d'une intégration afin d'obtenir un codage échelonnable des textures associées aux maillages tridimensionnels. Nous présentons ensuite une piste vers la visualisation locale adaptative de maillages texturés sur le terminal.

3.6.1 Représentation

La technique repose sur l'hypothèse suivante : une image $I(x, y)$, habituellement considérée comme une fonction sur un plan 2D (x, y) , peut également être regardée comme une surface $S = (x, y, I(x, y))$ dans l'espace tridimensionnel. La surface S correspond alors à l'élévation liée à l'intensité le long de la direction z de l'espace (x, y, z) . Une image est ainsi approximée par un maillage surfacique (voir [LSL97] [SL98]).

Echelonnabilité

La scalabilité, quant à elle, est obtenue par raffinement successif d'un maillage triangulaire de base. L'image est ainsi représentée par un maillage semi-régulier obtenu par subdivision adaptative du maillage de base (les faces sont subdivisées par bisection des arêtes), tandis qu'un triplet luminance / chrominance $\{L, u, v\}$ associé à chaque sommet permet de reconstruire une image suivant un modèle d'interpolation linéaire calculé sur les éléments finis constituant le maillage.

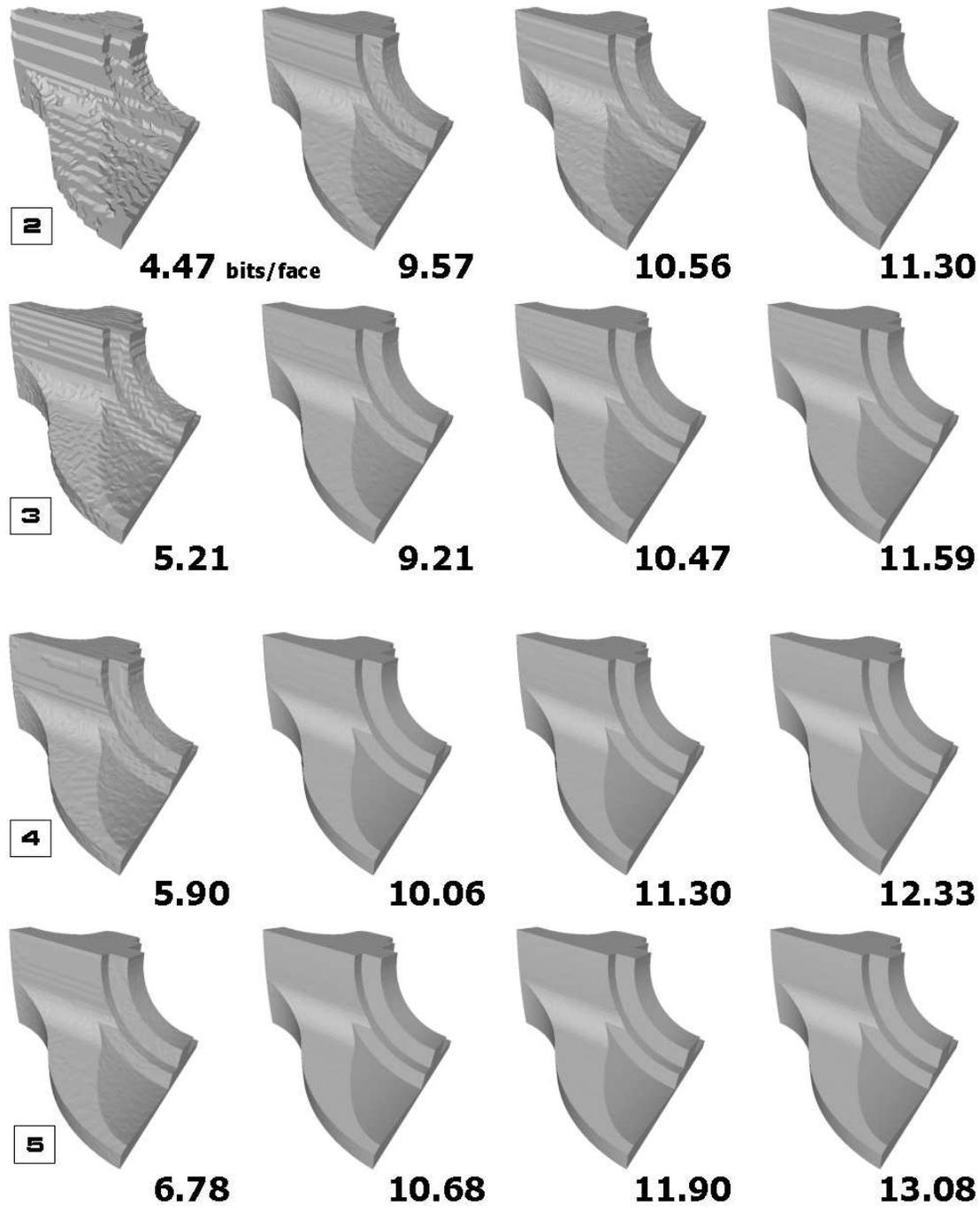


FIG. 3.44 : Reconstruction progressive du maillage *fandisk* pour différents niveaux de quantification initiaux. Respectivement et pour chaque ligne de haut en bas : quantification initialisée à 2, 3, 4 et 5 bits. Chaque colonne correspond respectivement à la reconstruction initiale (quantification des résidus de prédiction), au raffinement lié aux ordres de grandeurs des résidus finaux, puis aux 2 premiers plans binaires de raffinement successifs. Les coûts de codage indiqués comprennent la connectivité (0.53 bit/face) et les positions. On constate l'incidence du niveau de quantification de base sur la qualité finale après reconstruction : il est préférable d'*investir* quelques bits de plus lors de la quantification initiale (4 versus 2 bits) afin d'obtenir une meilleure reconstruction (voir les deux modèles pour un même coût de 11.30 bits / face).

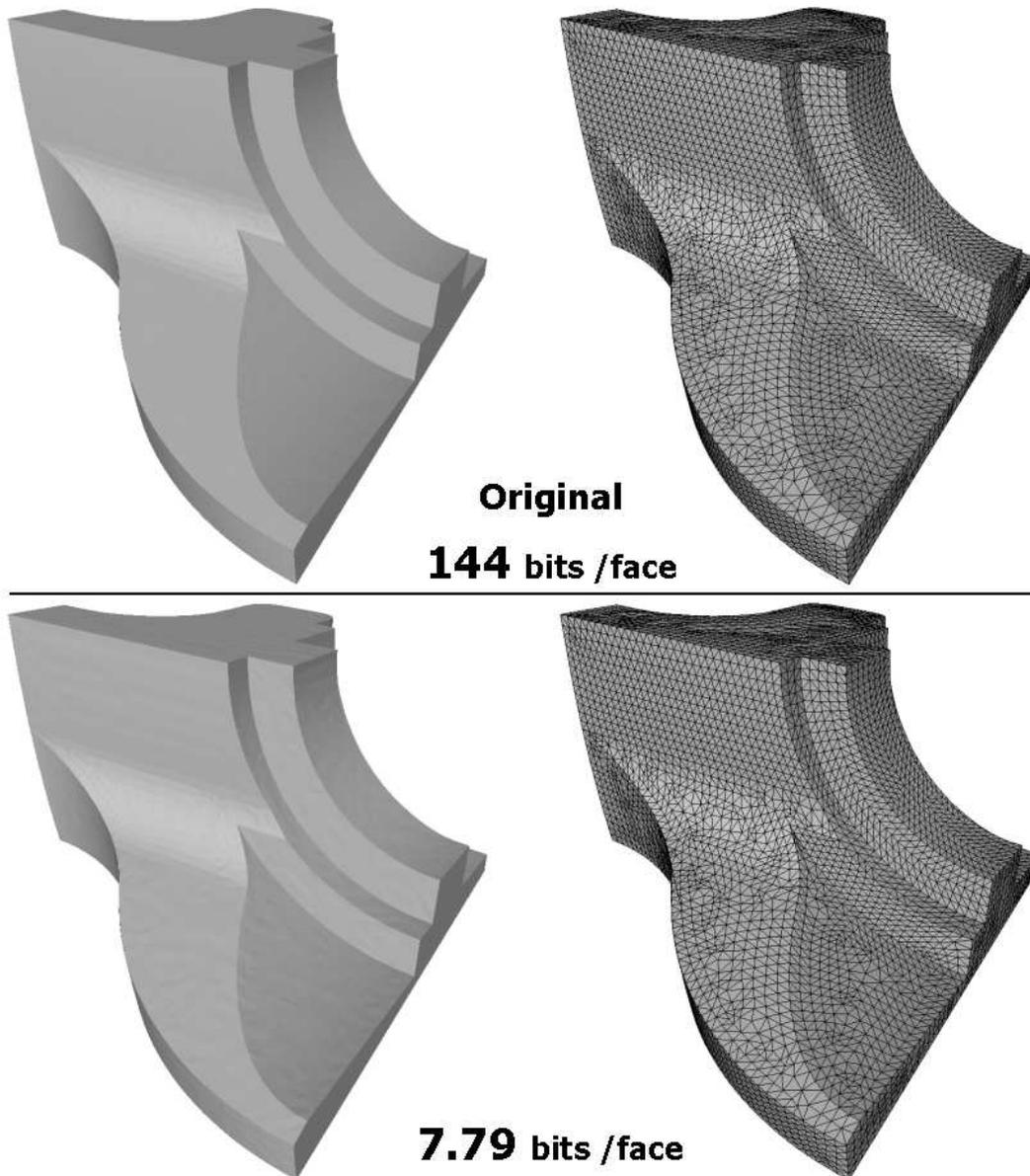


FIG. 3.45 : En haut : maillage *fandisk* original, pour un coût de codage théorique de 144 bits/face (et 395 bits/face pour le fichier VRML correspondant). En bas, le maillage reconstruit après quantification des résidus de prédiction paramétrée à 6 bits, pour un coût global de 7.79 bits/face, dont 0.53 pour la connectivité.

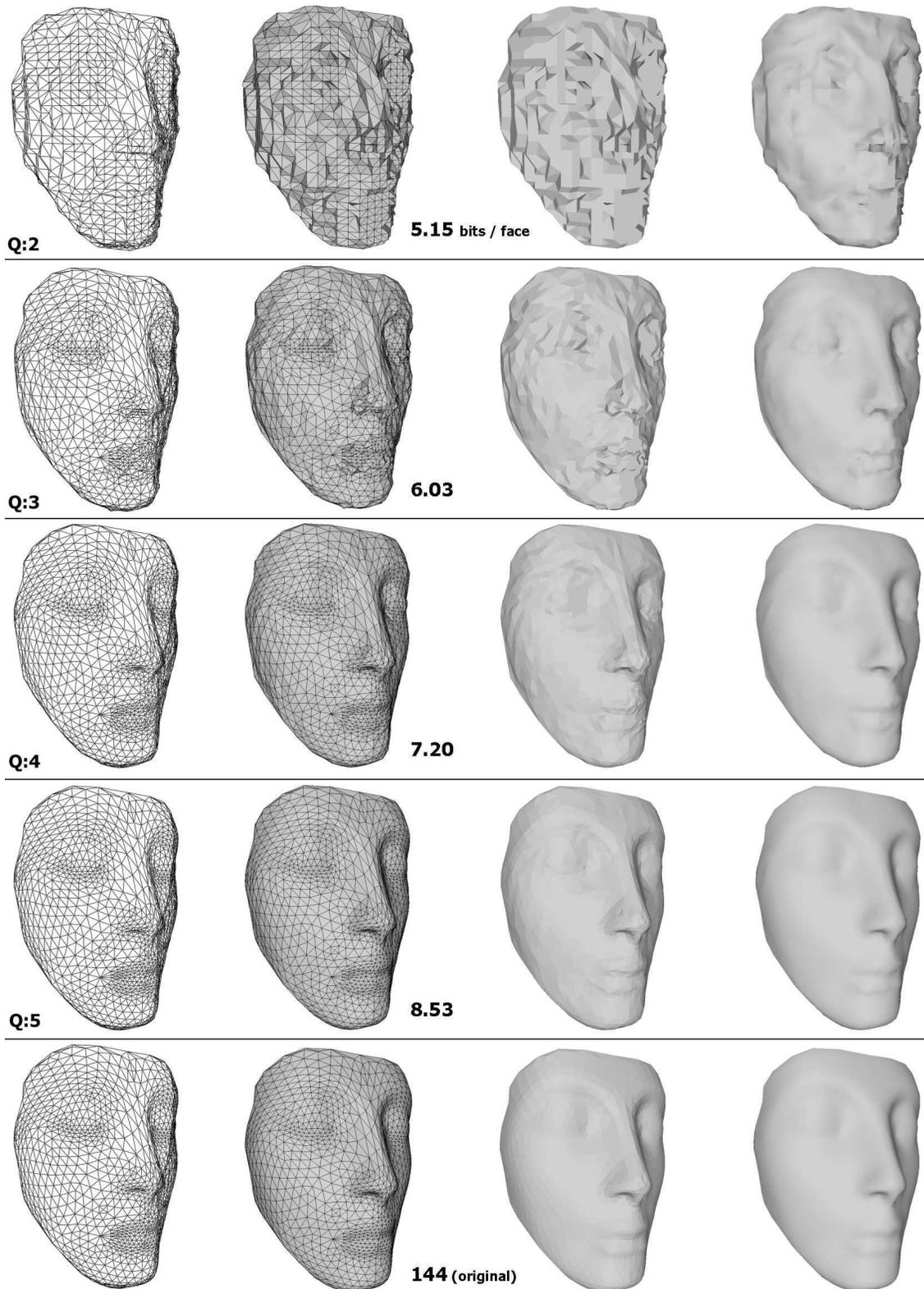


FIG. 3.46 : Maillage *nefertiti*. De haut en bas : résultats de reconstruction du niveau de base (sans raffinement) pour respectivement 2, 3, 4 et 5 bits de quantification, comparés au maillage original. De gauche à droite : le même maillage affiché en différents modes : maillage, faces + maillage superposé, faces et faces avec lissage. Ces images démontrent qu'en dessous d'un niveau de quantification de 4 bits, la fidélité au maillage est fortement dégradée. Les coûts de codage indiqués comprennent la connectivité (0.66 bit/face) et les positions.

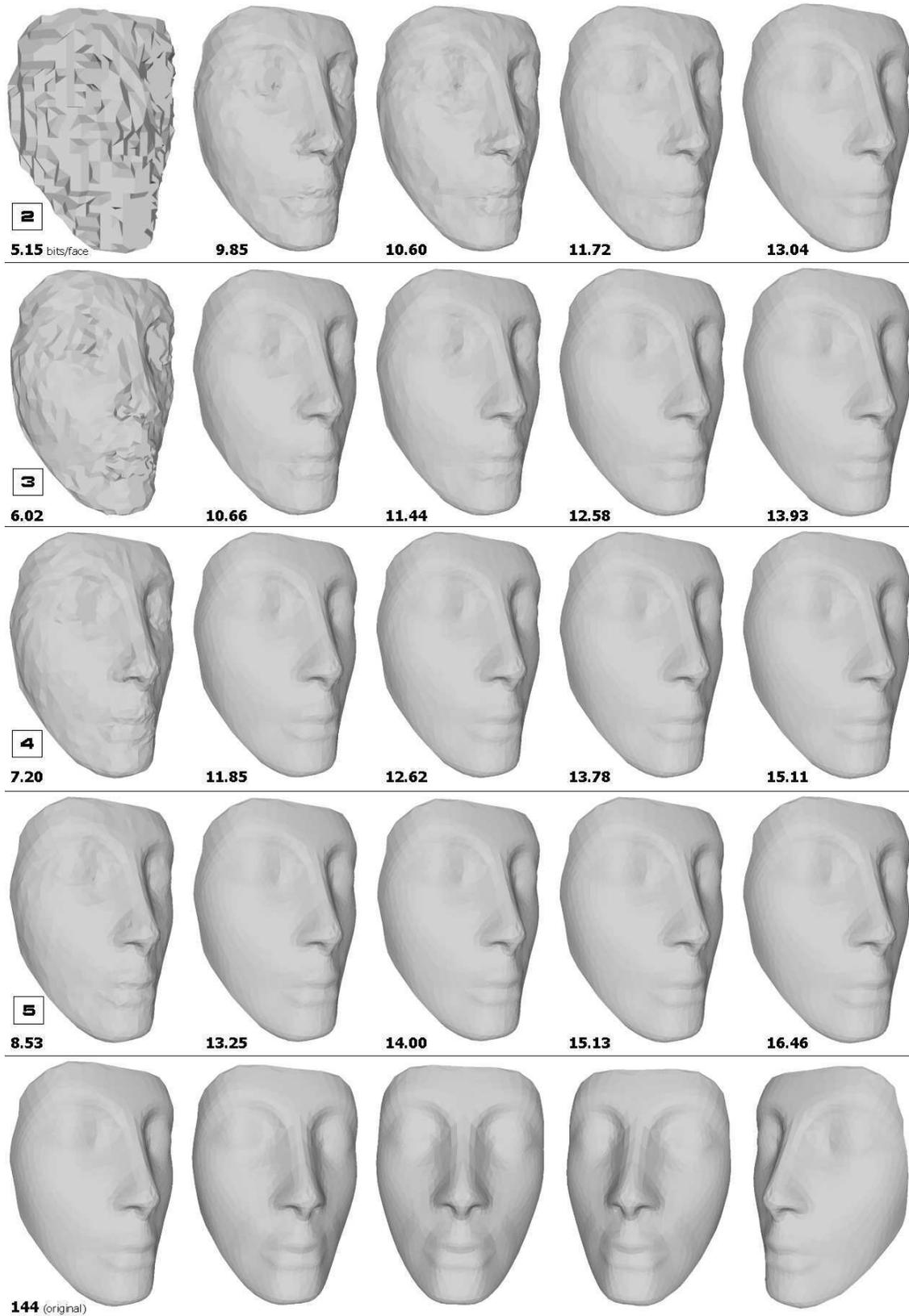


FIG. 3.47 : Maillage *nefertiti*. De haut en bas : résultats de reconstruction progressive pour respectivement 2, 3, 4 et 5 bits de quantification, comparés au maillage original. Les coûts de codage indiqués comprennent la connectivité et les positions.

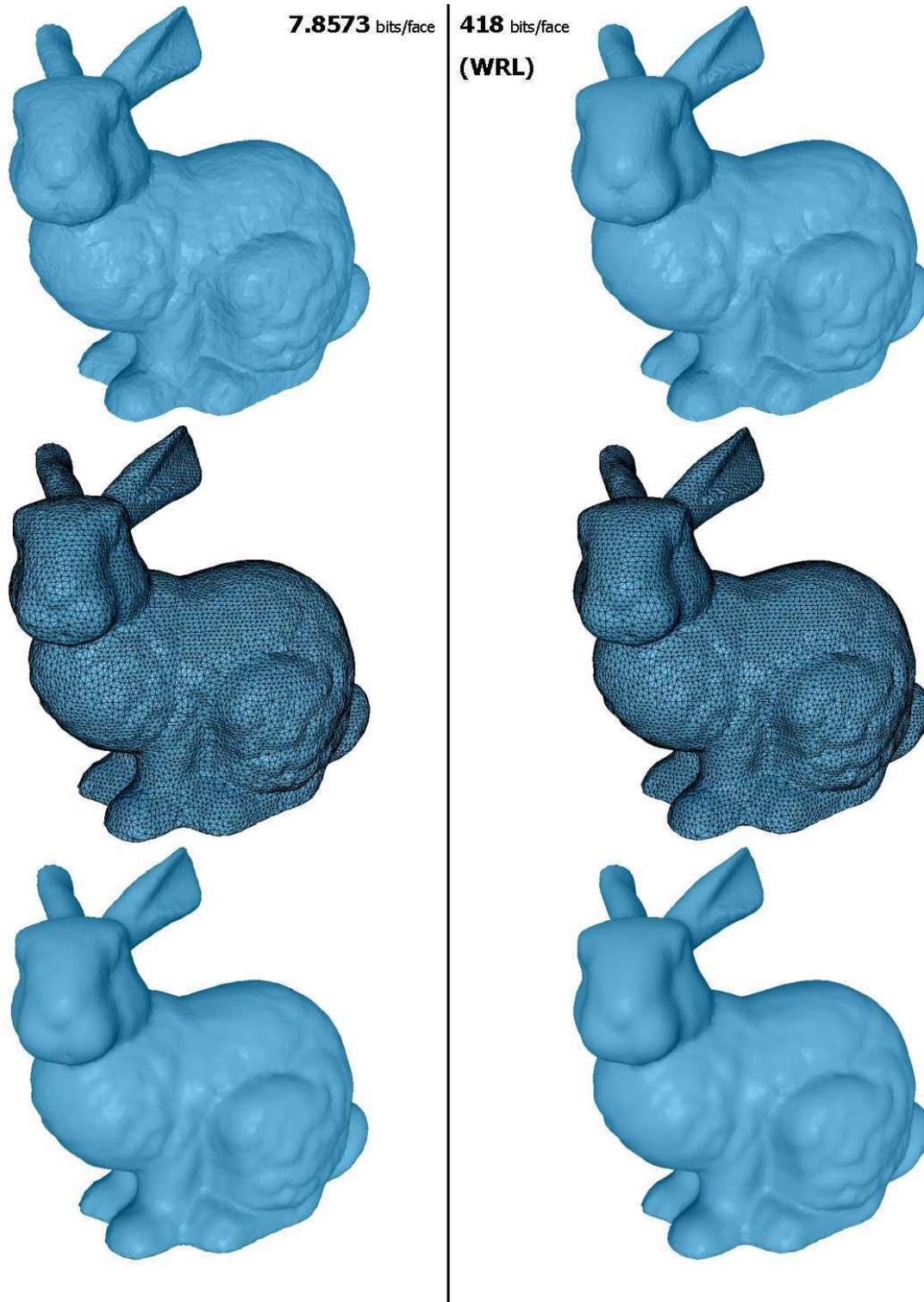


FIG. 3.48 : Maillage *bunny*. Colonne de gauche : le maillage de base reconstruit avec un coût de codage de 7.86 bits / face pour un niveau de quantification fixé à 4 bits, comparé au maillage original (colonne de droite). La connectivité occupe 1.02 bit / face.

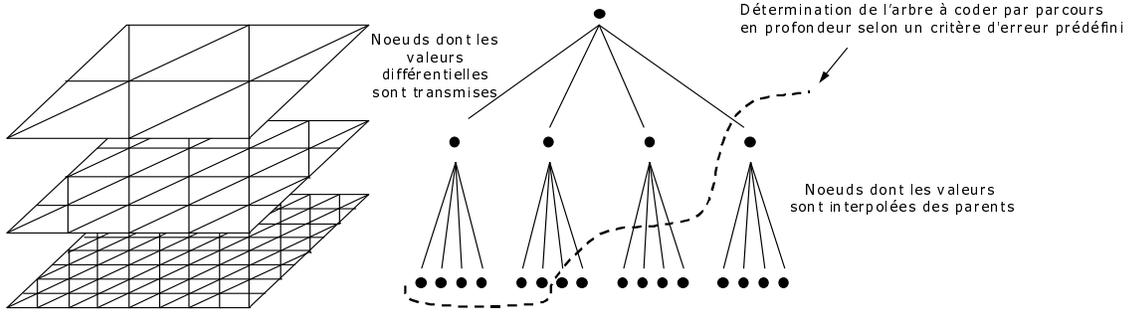


FIG. 3.49 : Principe du codage hiérarchique progressif d’une image : l’arbre est codé après détermination de sa profondeur, puis seules les valeurs différentielles de luminance / chrominance et position associées aux nœuds de l’arbre sont transmises. Les valeurs des nœuds associées aux branches non transmises de l’arbre sont alors interpolées.

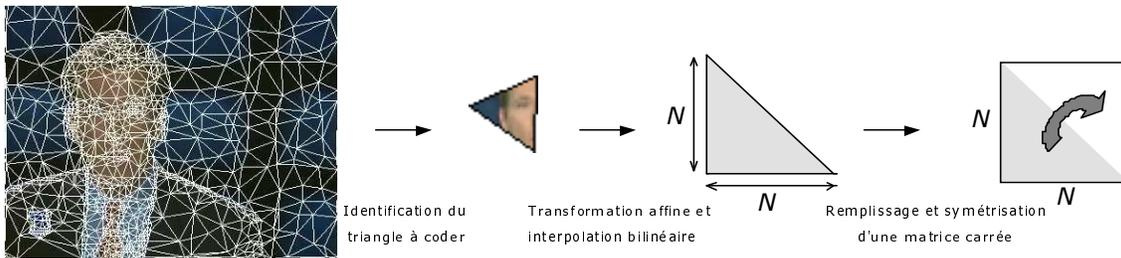


FIG. 3.50 : Procédé de transformation d’un triangle quelconque en une matrice carrée symétrique en trois étapes : sélection d’un triangle de forme quelconque dans la partition du maillage, transformation affine par déformation du triangle en un triangle rectangle isocèle dont les deux plus courts côtés sont de longueur N , et remplissage de la matrice par interpolation bilinéaire. La matrice est ensuite symétrisée afin de permettre le calcul de la DCT.

On obtient ainsi une hiérarchie de maillages triangulaires emboîtés, auquel un arbre quaternaire est associé (figure 3.49). A chacun de ses nœuds correspond un triangle du maillage prélevé dans la hiérarchie. Une erreur quadratique est évaluée sur les triangles afin de réaliser un parcours en profondeur dans l’arbre et de stopper ce parcours lorsque l’erreur est mesurée en deçà d’un seuil prédéfini.

Hybridation avec la transformée en cosinus discrète

Le codage par interpolation affine offre des résultats satisfaisants sur les régions homogènes peu texturées. Sa représentation continue évite notamment les artefacts traditionnels des découpages en blocs. Afin de coder efficacement les régions texturées, la technique est hybridée avec la transformée en cosinus discrète appliquée sur les triangles après un recalage dans le repère unitaire normalisé, une symétrisation et un échantillonnage régulier. En effet, une DCT ne peut se calculer directement sur un support triangulaire au même titre que les blocs d’une approche traditionnelle de type JPEG. Une matrice carrée symétrique est ainsi construite par transformation affine et interpolation bilinéaire à partir d’un triangle prélevé sur l’image (figure 3.50). Le calcul de la transformée DCT est ensuite appliqué sur la partie inférieure de la matrice (voir [LLS99a] et [AL99b]).

Le même arbre quaternaire est associé à la hiérarchie de maillages, puis l’erreur quadratique ϵ évaluée sur les triangles permet de choisir pour chacun des triangles la méthode la plus adaptée parmi trois modes de codage :

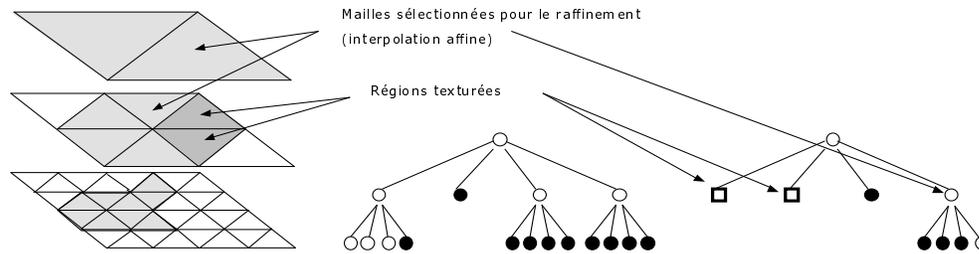


FIG. 3.51 : Principe du codage hybride : interpolation affine et codage de texture.

- si ε est inférieur à un premier seuil σ_1 : l'interpolation affine est suffisante pour représenter localement l'image,
- si ε est comprise entre σ_1 et σ_2 : quelques divisions supplémentaires du maillage sont alors suffisantes pour obtenir la qualité de représentation nécessaire suivant le procédé d'interpolation affine,
- si ε est supérieure à σ_2 : la région associée à la maille considérée présente une texture ne pouvant être codée efficacement par interpolation affine. Notre méthode de codage basée triangle est alors utilisée sur le résidu de l'image correspondant à la maille.

Enfin, le dernier niveau de la hiérarchie comprend une phase de permutation d'arête afin d'aligner les arêtes sur les contours de l'image et d'améliorer ainsi la qualité de l'approximation (voir figure 3.52). La section suivante décrit la méthode de compression utilisée.

3.6.2 Compression

La compression est obtenue sur les éléments finis par :

- codage différentiel des attributs de luminance / chrominance,
- quantification adaptative (la luminance, perceptuellement plus importante, se voit allouer un débit plus important que la chrominance),
- et codage entropique à modèle adaptatif.

Les coefficients résultant de la DCT appliquée sur les éléments sont quantifiés suivant une matrice de pondération reliée à des critères perceptuels, et les valeurs obtenues sont compressées par codage entropique à modèle adaptatif [LLS99a].

Enfin, Demaret et al ont récemment obtenu une échelonnabilité au sommet près par une adaptation aux maillages du *zerotree* [DRLB00]. La section suivante présente les résultats obtenus par la technique hybride.

3.6.3 Résultats

La figure 3.52 illustre le principe de codage hybride d'une texture associée à un maillage 3D, en utilisant un maillage de base régulier et une subdivision contrainte. La figure 3.53 illustre la progressivité obtenue sur la texture par la technique après placage sur le modèle 3D *head*, et la figure 3.54 illustre les derniers résultats obtenus par le codeur *zerotree* appliqué sur les maillages. Cette dernière technique présente l'avantage d'une granularité définie au sommet près.

La section suivante présente une piste vers la visualisation locale adaptative des textures sur le terminal.

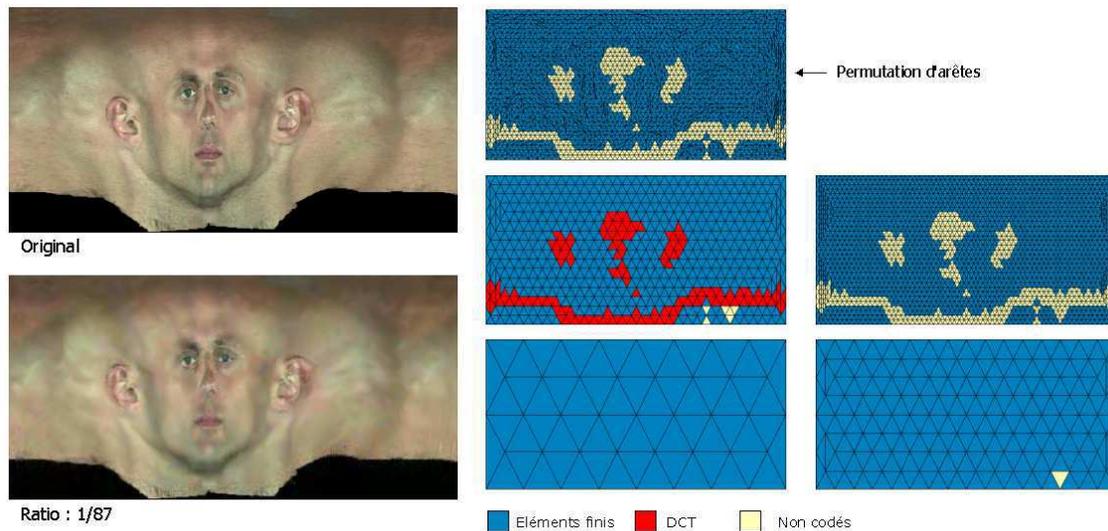


FIG. 3.52 : Une image est codée à partir d'une hiérarchie de maillages triangulaires dont les sommets sont dotés d'attributs de luminance / chrominance. Les régions composées de basses fréquences sont codées par éléments fins tandis que les hautes fréquences (les éléments texturés) sont codés par transformée DCT. Sur cet exemple, le maillage de base est initialisé par un échantillonnage régulier en quinconce sur l'image, et l'opérateur de permutation est appliqué au niveau le plus fin. A gauche : l'image originale et la version après compression avec un ratio de 1/87. Les régions fortement texturées sont reconstruites à partir de coefficients DCT, tandis que les régions lisses par morceaux sont reconstruites par interpolation linéaire des attributs photométriques et colorimétriques associés aux sommets.

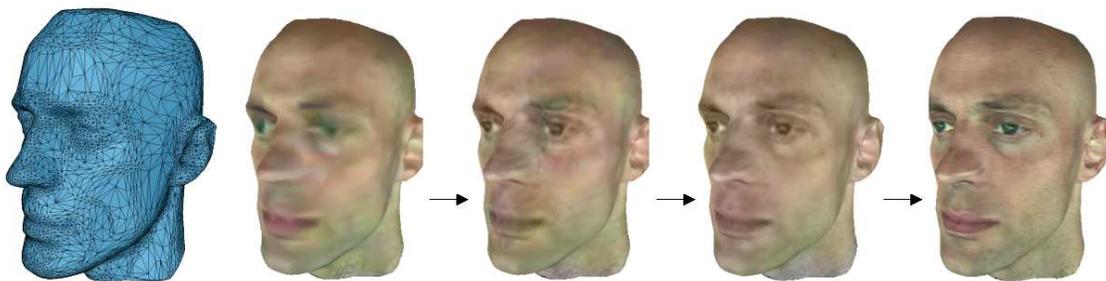


FIG. 3.53 : Un codage conjoint maillages triangulaires / éléments fins / DCT autorise la représentation échelonnée des textures.

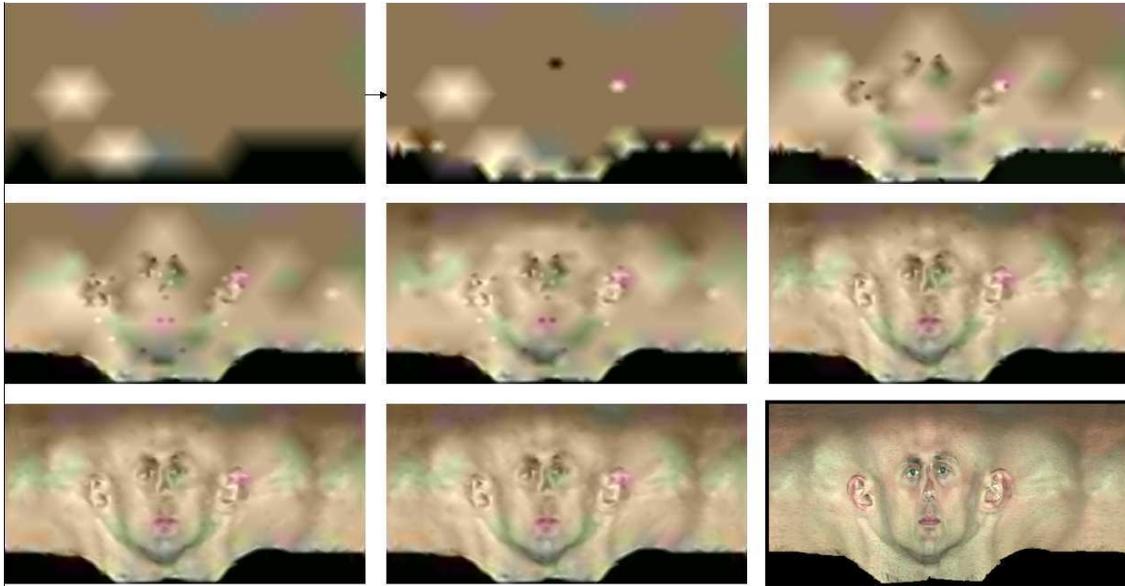


FIG. 3.54 : Progressivité obtenue par la technique du *zerotree* sur les maillages (l'image originale est encadrée). Le décodeur est capable de reconstruire une image à tout moment au cours de la transmission, avec une granularité définie au sommet près.

3.6.4 Une piste vers la visualisation locale adaptative

Un codage par région

Les formats échelonnables précédemment décrits (*zerotree* ou raffinements par seuillage dans la hiérarchie) autorisent la transmission progressive d'une image dans sa globalité (figure 3.53), mais il s'avère plus judicieux d'ajouter la notion de région reliée au contenu sémantique d'une texture pour une finalité de visualisation adaptative (voir figure 3.55). Chaque région correspond ensuite à un niveau de qualité et un train binaire indépendant, tandis qu'un fichier commun consigne la description géométrique des régions et les niveaux de qualité associés. Un tel fichier, transmis au début de la transmission, est amené à faire le lien entre la transmission progressive des régions et la visualisation adaptative sur le terminal. La notion de requête caméra s'avère alors nécessaire.

La notion de requête caméra

On définit ensuite la notion de requête caméra au cours de la visualisation sur le terminal, cette dernière étant mise en coïncidence avec la texture (voir figure 3.56). Les régions composées de triangles orientés dos à la caméra sont invisibles par convention avec les infographistes, et la distance au point de vue correspond au critère relié à la notion de niveau de qualité de l'image. Ce dernier point constitue une piste de recherche privilégiée.

3.7 Discussion et perspectives

Décrivons maintenant la valeur ajoutée de la méthode de codage de maillages triangulaires, son intégration dans un schéma global de diffusion de scènes et les évolutions souhaitables.

Valeur ajoutée

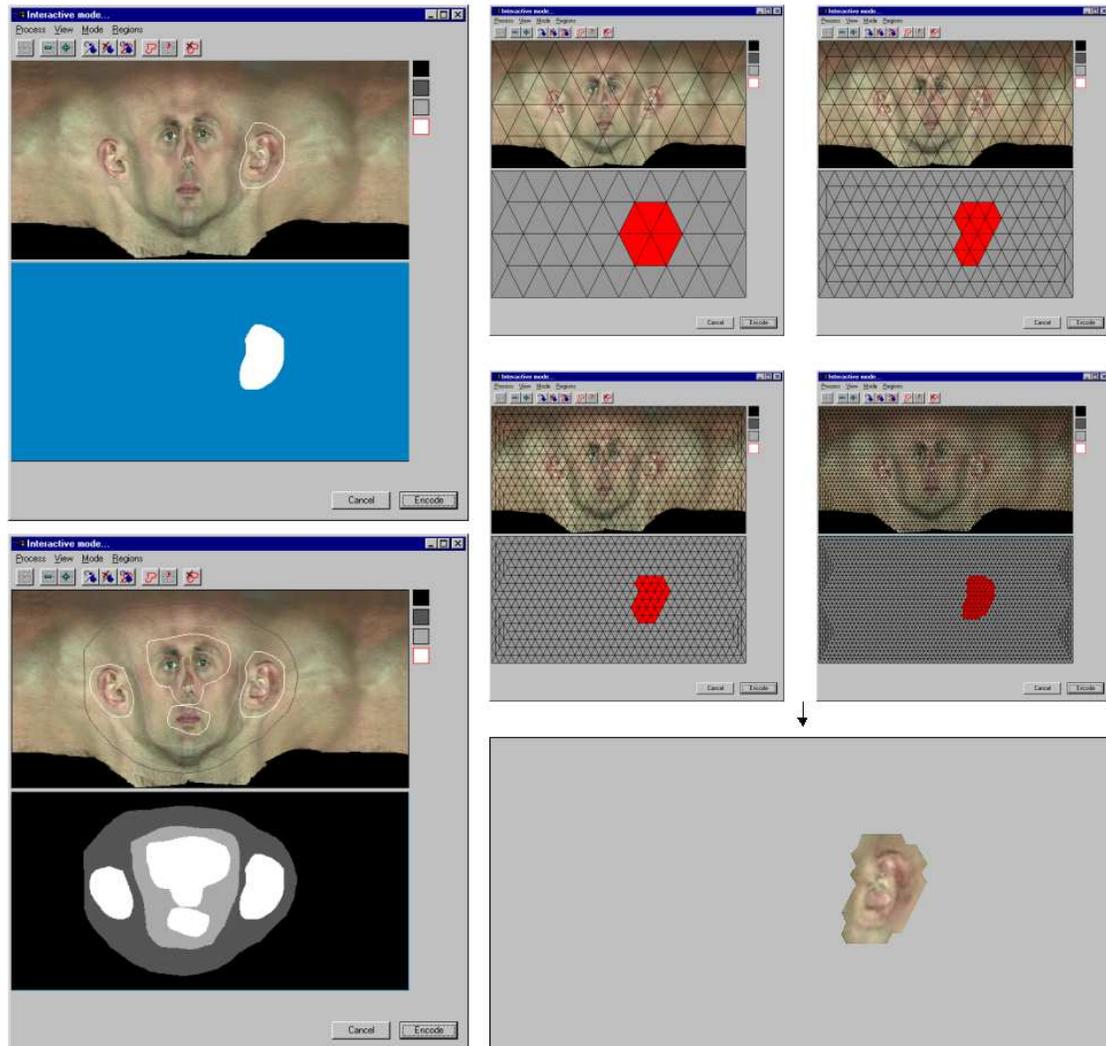


FIG. 3.55 : Codage par régions d'intérêt. En haut à gauche : une région d'intérêt sélectionnée par l'infographiste lors de la préparation d'une texture. A droite : les éléments à coder dans l'arborescence sont déterminés par marquage récursif ascendant des éléments de l'arbre suivant la région sélectionnée. Seuls les éléments marqués sont pris en compte lors du codage, les autres étant simplement interpolés lors de la reconstruction. En bas à gauche : l'utilisateur associe un niveau de qualité (représenté en niveau de gris) à chaque région. Le niveau correspond ensuite à une profondeur du parcours descendant lors du codage. Chaque région ainsi sélectionnée donne lieu à la création d'un train binaire, où seule l'information différentielle entre deux niveaux emboîtés est codée.

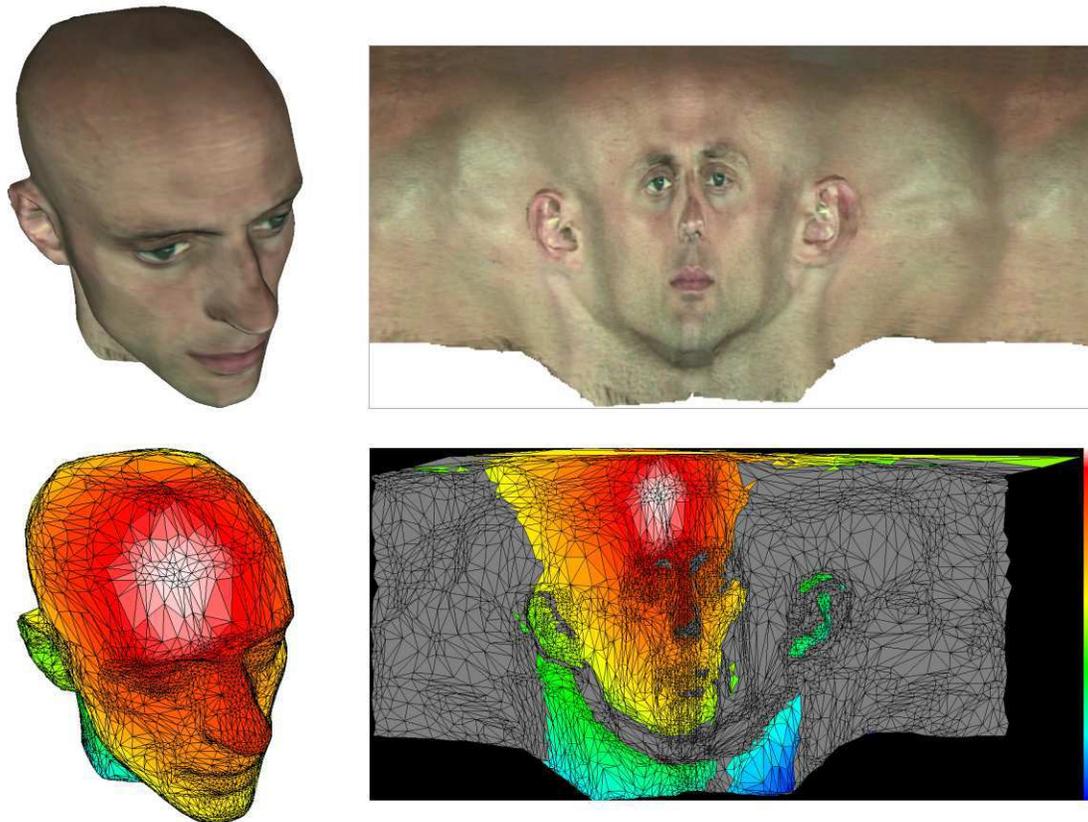


FIG. 3.56 : Une carte de requête caméra est construite en fonction du point de vue et permet de relier la notion de niveau de qualité à la distance au point de vue. Cette carte est mise en coincidence avec la hiérarchie de maillages lors de la reconstruction. Seules les régions orientées face à la caméra sont reconstruites par interpolation linéaire des attributs colorimétriques et photométriques affectés aux sommets, ou à partir des coefficients DCT sur les régions texturées. La distance à la caméra représentée en fausses couleurs est amenée à piloter l'algorithme de transmission progressive et de construction adaptative de la texture. La hiérarchie correspond alors à une structure de données hybride compacte susceptible de générer au vol une image lorsqu'une requête caméra lui est adressée.

Les bénéfices apportés par l'algorithme de codage comprennent :

- Un coût de codage amélioré d'environ 35% en moyenne sur la connectivité par rapport à la méthode originale de Touma et Gotsman, et la suppression du plancher précédemment constaté à 0.2 bit par face. Ce gain est essentiellement dû à une modification de stratégie (par encouragement des sommets complets comme candidats au pivot) lors de la conquête sur les arêtes, qui minimise de façon drastique le nombre de codes de séparations de liste (il suffit de regarder le tableau comparatif 3.5 pour s'en convaincre),
- L'indépendance complète du codage connectivité / positions, même si ces dernières sont codées en exploitant le réordonnement fourni par le codage des positions,
- L'accélération très sensible du décodage, par l'ajout de codes spécifiques aux sommets contextuellement complets (lorsque le pivot a terminé la conquête sur son territoire réservé), qui permettent d'éviter une structure de données basée arêtes et surtout une adjacence complète,
- Un gain lors du codage des positions par une gestion optimisée des accidents, les sauts d'adjacence étant représentés par un ensemble de ratios entiers compressés par codage entropique, ou noyés dans les résidus quantifiés après prédiction. Il en résulte un taux de compression d'environ 50 par rapport à un fichier VRML équivalent, soit 2% du temps de transmission équivalent, ce qui répond assez bien à la contrainte d'échelonnabilité sur les données pour afficher *tout de suite quelque chose*,
- Un codage échelonnable des positions par quantifications successives et représentation par plans de bits de tailles variables, à noter que la réduction du nombre de codes de sauts de liste apporte encore un gain sur les positions puisque le nombre de sauts en position d'une liste à l'autre est réduit.

Ces développements ont fait l'objet de deux publications [AL99a] [AL99b] et d'un dépôt de brevet [AL99a] en 1999.

Intégration

Un tel algorithme de compression est particulièrement adapté pour des objets dont on ne saurait tolérer une perte de la connectivité (bâtiments, grandes structures), mais où la précision originale de la géométrie n'est pas cruciale, notamment au début de la transmission. La même technique est également indispensable pour encoder les modèles de base des maillages semi-réguliers employés par les dernières techniques basées ondelettes (publications à paraître [GVSS00] [KG00]). Les auteurs ont notamment su rapidement intégrer la technique originale de Touma et Gotsman [TG98]. Gageons qu'ils sauront intégrer les améliorations apportées par nos développements.

Evolutions souhaitables

Même si les ratios de compression sont spectaculaires, notamment sur la connectivité, le talon d'Achille de cette méthode concerne principalement une contrainte : le maillage doit être *manifold*. Il apparaît pertinent de coupler l'algorithme à la méthode de régularisation de Guézic et al [GTLH97]. Ensuite, les performances sont étroitement liées au degré de régularité des données, cet algorithme n'est donc pas exploitable sur les *polygons soups*, littéralement un ensemble de polygones sans aucune organisation particulière.

Chapitre 4

Visualisation

Nous sommes maintenant sur le terminal, où les données compressées sont progressivement acheminées par le réseau. A chaque instant le défi consiste à reconstruire des images de qualité optimale à partir des données dont on dispose, le tout à une cadence interactive et en fonction du point de vue de l'utilisateur dans la scène.

La section 4.1 introduit le domaine concerné, puis un état de l'art des principaux travaux est exposé dans la section 4.2. La section 4.3 décrit ensuite une méthode de reconstruction adaptative procédant par subdivision de régions d'intérêt. Enfin, la section 4.4 présente une discussion et les perspectives d'évolution.

4.1 Introduction

Le chapitre introductif a présenté le principe de l'exploitation d'une scène 3D, et notamment le rendu sur l'écran d'un terminal. Ce chapitre concerne précisément les principes algorithmiques susceptibles d'assurer d'une part une cadence interactive des images lors de la navigation, et d'autre part une qualité perceptuelle maximale de rendu. Interprétons maintenant en termes plus intuitifs ces principes.

Lorsque nous figeons le temps sur le terminal, le maillage est représenté avec un certain niveau de résolution, ceci grâce à un format échelonnable idoine. On rappelle que les capacités d'affichage d'un accélérateur graphique s'expriment en nombre de polygones par seconde. Si l'on définit à trente images par seconde la cadence minimale afin d'assurer un rendu temps-réel, alors une simple division nous permet d'obtenir le *budget polygones* pour chaque image. La problématique peut également s'énoncer différemment suivant le but recherché : soit un maillage initial, il s'agit de développer toutes les méthodes susceptibles d'augmenter le réalisme de l'image sans pour autant fausser la représentation initiale définie par l'infographiste. Ce cas peut se produire lorsque la caméra de l'observateur est fixe, ou lors d'une pause durant la navigation, le budget polygones augmentant linéairement en fonction de la durée de cette dernière.

4.1.1 Un budget polygones par image

En fonction des capacités locales du terminal, et de la complexité de la scène, deux cas sont à distinguer :

1. La complexité de la scène excède les capacités de rendu : il convient alors d'ajuster la complexité des objets via une stratégie de simplification adaptative en fonction du point de vue de l'observateur (la

caméra virtuelle).

2. Les capacités de rendu excèdent la complexité de la scène, on dispose alors de marges permettant de raffiner la géométrie sur les régions visuellement pertinentes des objets. Il s'agit à nouveau d'une affaire de point de vue.

Une composante commune aux deux cas présentés concerne assurément la vitesse et la robustesse des algorithmes adaptatifs (ou dépendants du point de vue) de simplification et de raffinement. En effet, la structure de données doit autoriser un brassage extrêmement rapide des données afin de coïncider avec les cadences de rafraîchissement du rendu. On ne dispose en fait que d'un trentième de seconde entre deux images successives pour adapter la géométrie des objets et assurer le rendu. La navigation rapide dans une scène autorise toutefois une définition simplifiée de la géométrie, la vision humaine requérant un certain laps de temps pour discerner les détails.

Afin de comprendre les objectifs d'amélioration du rendu, la section 4.1.2 présente les critères liés au réalisme d'un objet.

4.1.2 Les critères de réalisme

Définir précisément les critères de réalisme d'une scène tridimensionnelle n'est pas chose aisée, il s'avère plus naturel de décrire les caractéristiques inverses qui introduisent une rupture du réalisme :

- L'aspect polygonal correspond certainement au défaut le plus flagrant lors de la visualisation d'une scène succinctement décrite, et plus particulièrement sur les surfaces courbes définies par les réseaux maillés. Cela provient de la phase de modélisation qui nécessite un compromis complexité / réalisme permanent. Les infographistes ont développé des algorithmes d'interpolation des normales (Gouraud, Phong [FvDFH96]) afin d'inhiber cet inconvénient, qui subsiste malgré tout parfois de manière flagrante sur les silhouettes puisqu'aucune interpolation géométrique n'est effectuée par ces méthodes. Enfin, il n'est pas nécessaire de définir une géométrie plus fine que la précision du périphérique graphique (un écran ou un casque de réalité virtuelle), et l'aspect polygonal diminue en fonction de la distance séparant l'objet de l'observateur,
- Les ombrages : la qualité du résultat produit par ombrage de Gouraud dépend largement de la taille des polygones rendus. Ce défaut de procédé provient de l'interpolation linéaire et du calcul des reflets restreint aux sommets, ce qui produit des dégradés monotones. Cette technique est suffisante pour simuler une illumination générale mais pose problème pour restituer des effets produits par un éclairage plus sophistiqué ou pour simuler l'interaction lumière / matière avec une surface courbe. Il faut ajouter à cela le fait que l'ombrage de Gouraud n'est pas correct au sens de la perspective puisque l'illumination varie linéairement sur toute la surface des polygones par incréments réguliers de pixel en pixel. Pour pallier cette faiblesse, les infographistes définissent la notion de cartes de normales plaquées sur les polygones, de cartes d'éclairage, etc... (voir seconde partie du livre de Michael Abrash [Abr00]). La subdivision des polygones reste toutefois la technique la plus rapide lorsque le rendu de l'ombrage est adressé par interpolation de Gouraud (ce principe étant systématiquement gravé dans le silicium des accélérateurs graphiques actuels),
- La fidélité aux couleurs : des couleurs trop saturées et incorrectement dégradées trahissent une image de synthèse,
- Les artefacts visuels lors des changements de résolution : une commutation trop brusque entre deux niveaux de détails introduit des sauts visuels à l'écran, auxquels l'œil est très sensible, notamment en l'absence de masquage. Les transitions progressives et imperceptibles constituent un gage de réalisme.

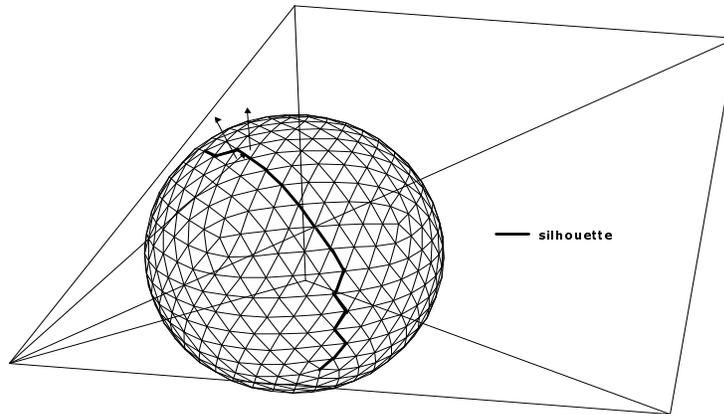


FIG. 4.1 : Disposant du point de vue, la détection de la silhouette est réalisée de manière directe : une arête appartient à la silhouette si elle partage deux faces dont les normales sont d'orientation mutuellement opposées par rapport à la caméra.

La liste précédente, loin d'être exhaustive, suffit à l'instauration d'une nouvelle stratégie. Au niveau d'une scène, on essaiera de définir précisément les objets dans le champ de vision, non masqués, et disposant d'une contribution surfacique importante après projection. L'objectif consiste à répartir l'information (i.e. les triangles) d'une part sur les régions situées en face de la caméra, et d'autre part sur la silhouette afin d'en supprimer l'aspect polygonal. On décompose ainsi la problématique en deux tâches : la *détection* des régions concernées par un opérateur de raffinement ou de simplification, et l'*application* précise de l'opérateur.

Avant d'exposer nos travaux, nous proposons une revue comparative des diverses innovations des acteurs du domaine.

4.2 Travaux précédents

L'état de l'art suivant est composé de trois familles d'algorithmes : la détection des silhouettes, le raffinement et la simplification adaptatives. En effet, l'enjeu consiste d'une part à détecter rapidement les silhouettes visibles et les régions situées dans le point de vue, et d'autre part à définir des algorithmes susceptibles de modifier la géométrie afin d'améliorer le rendu ou d'accélérer la vitesse d'affichage.

4.2.1 Détection des silhouettes

La silhouette joue un rôle prépondérant dans la perception visuelle [Koe84], puisqu'elle caractérise la forme globale d'un objet, et qu'elle correspond souvent *aux premiers coups de crayon d'une esquisse*. On ne peut ignorer le rôle privilégié qu'elle occupe dans les processus cognitifs, puisque l'on peut identifier une personne dans la pénombre ou le brouillard uniquement à partir de sa silhouette.

D'un point de vue plus technique, la silhouette se définit par rapport à une position et à une orientation de caméra. On dit que les arêtes d'un maillage appartiennent à la silhouette si elles sont adjacentes à deux triangles, dont les orientations des normales sont mutuellement opposées vis à vis de la caméra (figures 4.1 et 4.2). Toutefois, la figure 4.3 illustre la nécessité de distinguer les silhouettes au sens de cette définition et les silhouettes *visibles* [GC90].

Les acteurs du domaine s'intéressent à la silhouette pour plusieurs raisons :

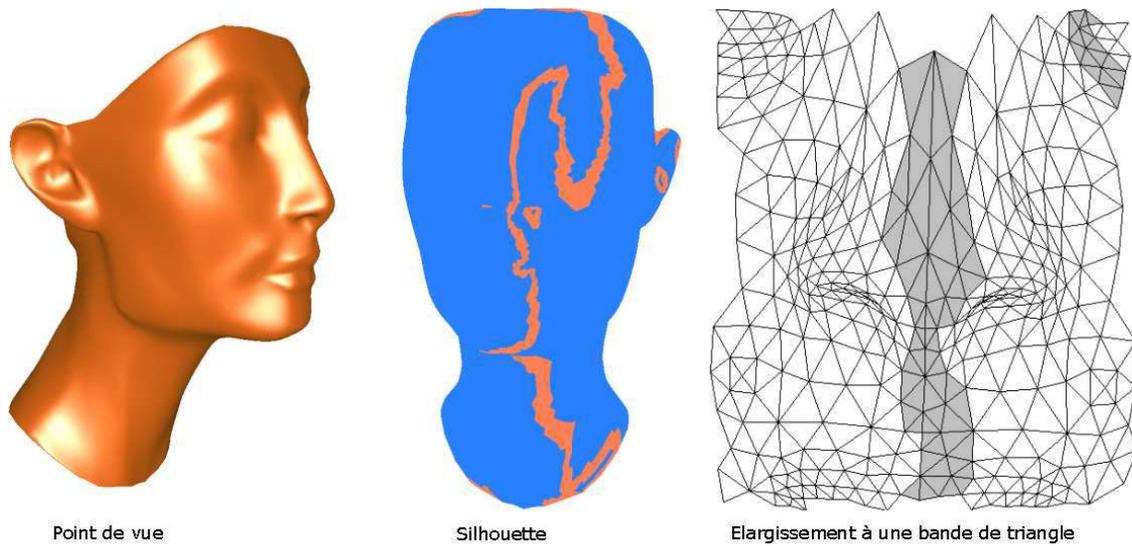


FIG. 4.2 : De gauche à droite : le point de vue du maillage *nefertiti*, la silhouette correspondante représentée en couleur claire au centre et la zone d'intérêt correspondante élargie à une bande de triangles (en gris sur l'image de droite).

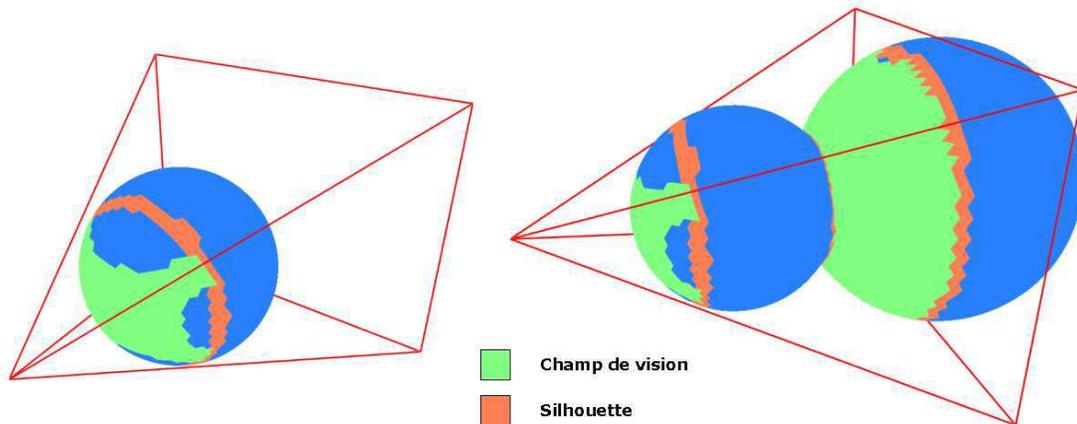


FIG. 4.3 : A gauche : les faces en dehors de la pyramide de vision sont invisibles, et les faces tournant le dos à la caméra ne sont pas affichées par convention avec les infographistes. A droite : la silhouette au sens de la définition donnée plus haut est à différencier des silhouettes visibles, la distinction nécessitant des calculs de masquage complexes.

- la vision par ordinateur : la silhouette est utilisée par des algorithmes de reconnaissance de formes lorsque ceux-ci sont couplés à une connaissance a priori des modèles,
- la suppression des lignes cachées : la silhouette sépare les régions visibles des régions invisibles,
- le rendu basé image [GGH⁺99] : correspond à une technique utilisée afin de réduire les coûts de calculs lors du rendu. La géométrie complexe est remplacée par un très petit groupe de polygones sur lesquels une texture précalculée correspondant au rendu de l’objet hors-ligne est plaquée. L’illusion est souvent très satisfaisante avec une exception sur les silhouettes de l’objet puisque la géométrie n’est pas raffinée en pratique par le principe. Gu et al [GGH⁺99] proposent alors de précalculer un ensemble de silhouettes projetées sur une sphère suivant différents points de vue et à diverses résolutions afin d’exploiter ces données sous la forme d’un *emporte-pièce*. Le contour de l’objet est ainsi finement détourné. Dans le même esprit une récente publication de Sander et al [SGG⁺00] décrit une technique de rendu hybride dépendant du point de vue. Un maillage est représenté sous la forme de trois données : une enveloppe convexe progressive emboîtée, un ensemble d’arêtes organisées en un arbre optimisé pour une recherche ultérieure et une carte des normales. Pour chaque point de vue, la silhouette est rapidement extraite de l’arbre d’arêtes afin de découper précisément la silhouette d’un objet succinctement décrit. L’efficacité d’une telle technique est spectaculaire, d’autant plus lorsque le crénelage des silhouettes est supprimé,
- le rendu artistique ou non réaliste [RC99] [DC90] : la silhouette est détectée et superposée à un rendu volontairement simplifié des ombrages afin d’aboutir à un aspect de type dessin,
- la détection rapide [BDG⁺99] [BE99] : Benichou et Elber ont proposé à *Pacific Graphics* en 1999 une solution de détection très performante, puisqu’après le précalcul d’une structure de données les silhouettes sont calculées en complexité sortie-dépendante (i.e. en fonction du nombre d’éléments appartenant à la silhouette). On suppose une multiplicité des arêtes considérée ne dépassant pas deux (le maillage est dit *2-manifold*), et la conformité du maillage. Chaque arête possédant deux faces adjacentes de normales différentes construit une géodésique de longueur minimale sur la sphère des normales (appelée *sphère Gaussienne*). Chaque géodésique est ensuite projetée sur un cube circonscrit à la sphère pour aboutir à un maximum de quatre arcs répartis sur les six faces du cube. Chaque requête de caméra se traduit en un plan orthogonal à la direction de vision passant par l’origine et intersectant le cube. Les silhouettes sont enfin déduites par détection des intersections entre les géodésiques et l’ensemble des segments formés par l’intersection du plan de vision avec le cube. En pratique, si n représente le nombre de polygones d’une sphère régulièrement maillée, alors \sqrt{n} correspond au nombre d’arêtes moyen appartenant à sa silhouette pour un point de vue donné. Cette mesure fournit un ordre de grandeur moyen des performances obtenues par Benichou et Elber.

Détecter la silhouette ne suffit pas, puisqu’il s’agit également de détecter les triangles orientés face à la caméra afin de raffiner l’interpolation de l’ombrage sur les régions d’intérêt correspondantes. Zhang et Hoff [ZH97] ont élaboré un algorithme de détection rapide des faces orientées face à la caméra, qui pourrait être exploité pour accélérer les calculs en masquant des régions entières du maillage, ces dernières étant invisibles car elles sont composées de faces orientées dos au point de vue. Une fois la détection effectuée, il nous reste à raffiner la géométrie sur les régions visuellement pertinentes.

4.2.2 Raffinement adaptatif

Cette technique raffine les régions d’intérêt d’un objet afin d’augmenter le réalisme lors de la visualisation.

Il faut toutefois distinguer deux objectifs, suivant que l’on dispose :

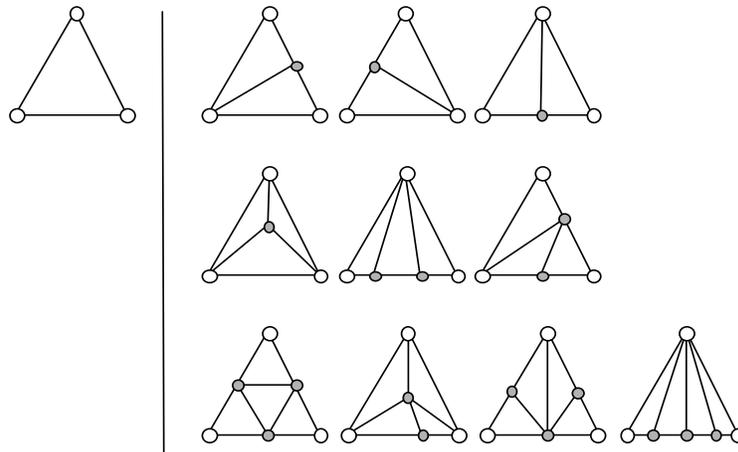


FIG. 4.4 : Les principaux opérateurs de subdivision. Un triangle original, et ses versions subdivisés respectivement par les opérateurs 1 vers 2, 1 vers 3 et 1 vers 4. La subdivision barycentrique des faces ou des arêtes correspond au mode de raffinement le plus répandu.

- d'un modèle géométriquement très simplifié et de données additionnelles, il convient alors de minimiser la distorsion géométrique sur les régions situées dans le champ de vision. Une analyse multirésolution et une organisation hiérarchique des données autorisent alors le raffinement adaptatif (section 4.2.3),
- ou de maillages pour des raisons liées à la modélisation. Lors de la création, l'infographiste imagine plutôt une surface courbe comme un réseau de patches B-Splines, la structure de maillage étant imposée pour la simulation ou par l'outil de modélisation. La subdivision des surfaces (section 4.2.4) combinée à un filtrage de la géométrie correspond alors au moyen privilégié dans le but de reconstruire une image, et notamment de supprimer l'aspect polygonal sur les régions d'intérêt [EH96].

L'analyse multirésolution et la subdivision de surfaces ont en commun des opérateurs de subdivision, illustrés figure 4.4.

Commençons par décrire le principe de l'analyse multirésolution.

4.2.3 L'analyse multirésolution

Le terme multirésolution reprend des idées provenant de la technique des *ondelettes*, qui décrit un signal à différents niveaux de résolution par le biais de coefficients capturant uniquement les différences entre deux niveaux successifs [EDD⁺95] [CPD⁺96] [SS95] [LDW97] [SGW98] [LSS⁺98] [GSS99] [KCVS98] [DGSS99]. Cette analyse des objets tridimensionnels est notamment intimement liée à la subdivision des maillages [KS98], puisqu'elle assure la transition entre les différents niveaux de détails. Cependant, nous partons ici du principe que l'ensemble des données est décompressé à l'aide du décodeur présenté dans le chapitre 3. Les données sont progressivement acheminées sur le terminal tandis que la géométrie des maillages augmente en résolution. L'échelonnabilité n'étant pas obtenue en terme de connectivité, il s'agit alors de procéder à un raffinement du maillage décodé à l'instant t ne disposant d'aucune information supplémentaire. La subdivision de surface à des fins de reconstruction s'avère alors particulièrement bien adaptée.

4.2.4 La subdivision de surfaces

Chaikin [Cha74] proposait en 1974 de générer une courbe par raffinement successif d'une chaîne polygonale définie par des points de contrôle, le raffinement consistant à insérer de nouveaux éléments (i.e. des som-

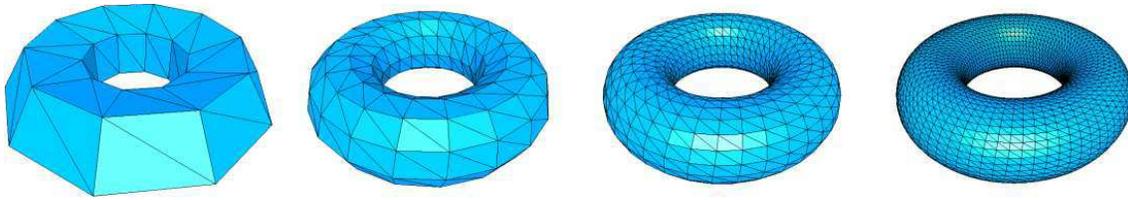


FIG. 4.5 : La surface est reconstruite par approximation (à distinguer de l'interpolation) à partir d'une surface de référence. Sur cet exemple la subdivision de Loop est globale.

ments et les arêtes associées). Catmull-Clark [CC78] et Doo-Sabin [DS78] ont étendu ce concept aux surfaces en 1978. Dans ce cas, une surface est définie par une infinité de subdivisions du maillage initial de contrôle. La subdivision consiste encore à insérer de nouveaux éléments (i.e. sommets, faces et arêtes correspondantes). Il faut s'assurer pour cela que le processus de subdivision converge effectivement vers une surface d'une part, et que la surface possède les propriétés différentielles voulues d'autre part. La définition de principes appliqués lors des opérations élémentaires de raffinements conditionne directement la qualité des surfaces obtenues.

La subdivision de surface est couramment utilisée pour l'approximation [ZS99] (figure 4.5) ou l'interpolation [HKD93] à des fins de reconstruction [HDD⁺94] [EH96] [STK99] des surfaces pour le rendu. L'interpolation laisse les positions des sommets originaux inchangées à l'inverse des techniques d'approximation, qui fonctionnent par filtrage successif des positions. Nous nous intéressons dans le cas présent uniquement aux techniques d'approximation fonctionnant sur les maillages triangulaires puisque la conservation des positions initiales ne constitue pas une contrainte pour les applications de visualisation envisagées.

Plus précisément, la subdivision définit les surfaces courbes comme la limite d'une séquence infinie de raffinements successifs [ZS99]. L'immense avantage de cette technique provient de sa capacité à générer la géométrie de façon robuste et semi-procédurale, en disposant d'une unique information : le modèle de base utilisé comme maillage de contrôle. Ce dernier point représente un critère très séduisant pour nos contraintes de codage. Les surfaces de subdivision sont aussi par nature échelonnables et sont capables de fournir la même qualité de rendu qu'une surface implicite ou paramétrique de type NURBS, tout en conservant une structure de données polygonale. Les techniques de subdivision de surfaces nécessitent la définition d'opérateurs élémentaires de subdivision (figure 4.4), le plus répandu étant certainement la subdivision régulière 1 vers 4 : chaque triangle est subdivisé par subdivision barycentrique des arêtes (figure 4.6). Les sommets sont ensuite déplacés en fonction d'une pondération des sommets situés dans un voisinage (le filtrage), et l'on peut définir un comportement spécifique sur les bords ou les singularités géométriques des maillages en modifiant simplement le jeu des coefficients de pondération (voir [HDD⁺94]). Un des algorithmes clés développant le procédé d'approximation par subdivision 1 vers 4 a été développé par Loop [Loo87] en 1987 (figure 4.7). Cette méthode correspond à la généralisation des Box-splines pour des maillages triangulaires quelconques et a été déclinée en une version adaptative par Hoppe [HDD⁺94] afin de respecter les singularités géométriques des maillages, la figure 4.8 illustrant les effets parfois indésirables du lissage systématique. Une autre approche appliquée aux maillages composés de quadrilatères a été développée par Kobbelt [Kob98], et l'on peut trouver dans la littérature de nombreux travaux autour de la reconstruction de surfaces à partir d'un maillage de référence [ZSS96] [HKD93] [Man98] [SZSS98] [Lev99] [Gus98] [JH99] [War95].

Une des préoccupations majeures en subdivision de maillages consiste à générer des surfaces ayant des propriétés différentielles bien spécifiques, correspondant à la C^k différentiabilité. L'objectif consiste essentiellement à garantir cette continuité sur l'intégralité de la surface après l'application d'une infinité de subdivisions,

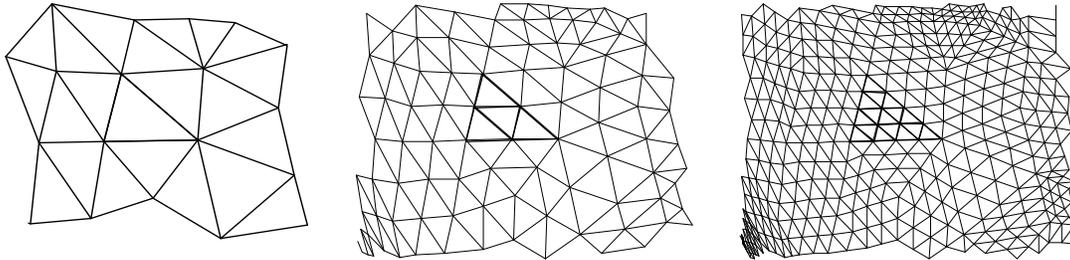


FIG. 4.6 : Subdivision régulière 1 vers 4 par bisection des arêtes. Le nombre d'éléments est multiplié par 4 à chaque itération. La réversibilité et la robustesse de l'opérateur autorisent une modification au vol de la résolution d'un maillage. Les nouveaux sommets générés par subdivision possèdent une valence 6, générant ainsi un maillage semi-régulier.

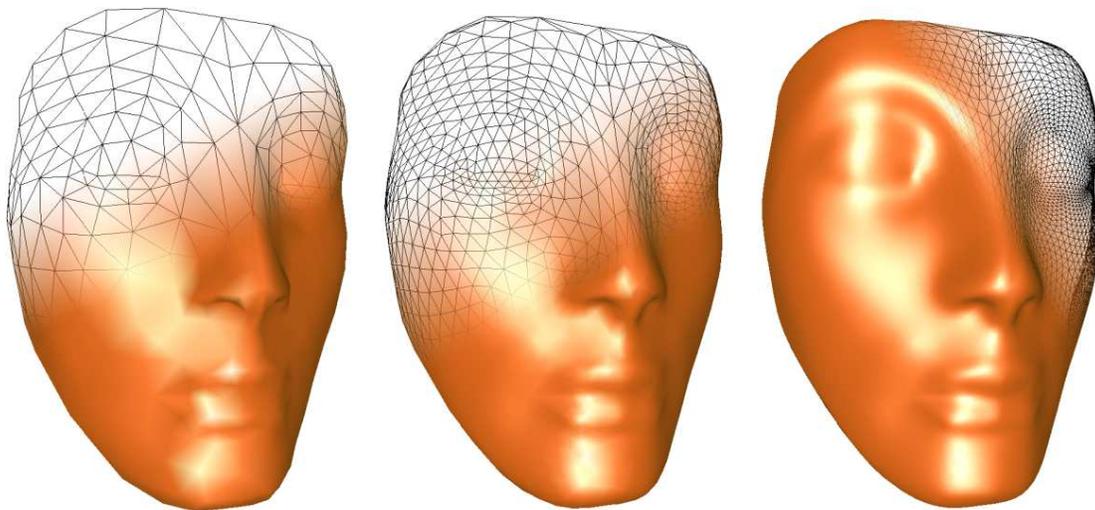


FIG. 4.7 : Subdivision de Loop sur le maillage *nefertiti*. L'amélioration visuelle est sensible sur des objets composés de régions courbes car il en résulte une meilleure approximation de la géométrie et surtout des normales, celles-ci étant le support du calcul de l'ombrage. On rappelle que les normales jouent un rôle prépondérant dans la perception (et parfois l'illusion) de la géométrie.

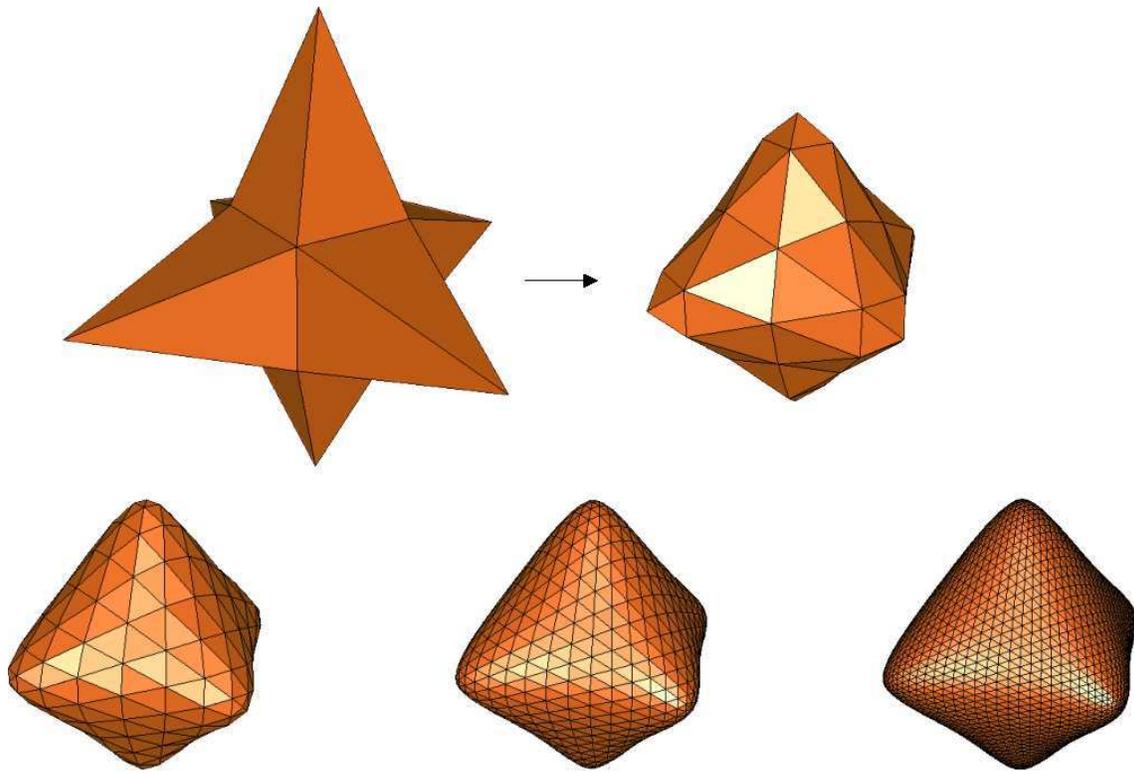


FIG. 4.8 : Subdivision de Loop [Loo87] sur le maillage *star*. On remarque un lissage des discontinuités et l'obtention d'une surface possédant certaines propriétés différentielles après une infinité de subdivision. Plus intuitivement, la notion de continuité est liée à une *douceur* d'aspect du modèle. Sur cet exemple la surface limite est C^2 continue en chaque point, excepté sur un nombre fini de points appelé sommets extraordinaires (i.e. avec une valence différente de 6 sur le maillage original).

et ce pour tous les types de maillages. De telles propriétés garantissent notamment la suppression de l'aspect polygonal sur les silhouettes, mais aussi un dégradé de l'ombrage visuellement satisfaisant en ce qui nous concerne.

En pratique, le schéma de subdivision s'écrit sous la forme d'une matrice stochastique dont l'analyse spectrale conditionne les propriétés différentielles de la surface. Plus précisément, on écrit les positions des sommets sous une forme matricielle, et on applique un ensemble d'itérations de subdivision afin d'obtenir un schéma auto-répliatif. Les valeurs propres de la matrice stochastique résultante caractérisent le comportement asymptotique du modèle sous-jacent, et l'ajout de contraintes sur les rapports entre les valeurs propres permettent l'obtention des propriétés différentielles souhaitées. Prautzsch et Umlauf en ont fait une étude détaillée dans [Pra98] [PU98a] [PU98b] [PU98c] [Uml95].

Si l'approximation des surfaces par subdivision est bien adaptée au raffinement adaptatif des maillages, il nous reste à étudier les algorithmes réciproques, concernant la simplification adaptative.

4.2.5 La simplification adaptative

Devant satisfaire des contraintes de robustesse et de vitesse, la simplification adaptative est à différencier de la simplification *off-line*. Il est en pratique inconcevable d'utiliser des métriques et des techniques d'optimisation afin d'améliorer la qualité d'approximation en temps-réel sur le terminal. En effet, modifier la résolution d'un objet de façon adaptative doit occuper une fraction de seconde entre deux images successives. Les acteurs du domaine s'orientent alors le plus souvent vers des structures hiérarchiques précalculées [Lue96] [ESV99] [XESV97] [Lue98]. Enfin, il est raisonnable d'admettre que si l'on dispose d'une structure de données échelonnée en connectivité rapide, robuste et fiable, celle-ci pourrait remplacer la simplification adaptative sur le terminal.

4.2.6 La piste choisie

La piste de recherche choisie concerne une technique de reconstruction adaptative et dépendant du point de vue d'un observateur. Les qualités requises d'une telle méthode sont :

- la vitesse de raffinement : afin de coïncider avec les contraintes de la navigation dans une scène,
- un raffinement localisé sur les silhouettes et les régions d'intérêt. Il s'agit donc de réaliser une cohabitation harmonieuse entre les différents niveaux de détail,
- un opérateur robuste et réversible car le point de vue d'un observateur s'avère hautement variable au cours de la visualisation.

La technique de reconstruction développée opère par approximation et subdivision successive, et combine une subdivision barycentrique 1 vers 3 avec l'opérateur de permutation d'arête. Le choix d'un tel opérateur s'inspire des travaux de Kobbelt [Kob99], qui exploite ce même opérateur de subdivision par vagues de deux itérations successives afin d'opérer de manière robuste la subdivision 1 vers 9 des faces par trisection des arêtes. La section suivante montre qu'en pratique ce même opérateur appliqué avec une granularité unitaire s'avère particulièrement bien adapté au cahier des charges précédemment défini.

4.3 Raffinement adaptatif de maillages triangulaires

La méthode élaborée dans cette section autorise la reconstruction de surfaces courbes par morceaux par raffinement adaptatif. En pratique la double adaptativité s'obtient par la prise en compte du point de vue d'un

observateur d'une part, et par la conservation des singularités géométriques telles que les coins, les arêtes vives ou encore les lignes de style explicitement définies par l'infographiste d'autre part. La section suivante présente le principe général de la technique.

4.3.1 Principe général

La technique utilise itérativement un algorithme générique de raffinement adaptatif, ce dernier pouvant être découpé en trois phases principales :

- détection des régions d'intérêt,
- subdivision hybride,
- filtrage adaptatif de la géométrie.

Comme pour le codage, la finalité consiste à maximiser le ratio qualité perceptuelle / quantité d'information sur le terminal. La quantité d'information disponible s'exprime alors en nombre de triangles et correspond à la contrainte de budget polygones définie en section 4.1. Hormis cette dernière contrainte, les degrés de liberté sont nombreux puisque l'on peut :

- raffiner le maillage initial par subdivision des triangles,
- permuter les arêtes initiales afin d'éviter la dégénérescence du maillage raffiné,
- et modifier la géométrie par filtrage des positions des sommets du maillage initial.

Des contraintes sont ajoutées dans le but de conserver les singularités géométriques comme les coins, les arêtes vives ou les lignes de style des maillages. On obtient ensuite un raffinement continu par une interpolation conjointe géométrie / normales, et une structure de données optimisée autorise la réversibilité du raffinement tout en fournissant la robustesse nécessaire à une visualisation interactive. Les sections suivantes détaillent les étapes de notre approche.

4.3.2 Détection des régions d'intérêt

Les régions d'intérêt peuvent soit être définies explicitement par l'utilisateur, soit déduites d'une phase de détection des régions jugées visuellement pertinentes. On définit tout d'abord le point de vue d'un observateur par le centre d'une caméra virtuelle (*l'œil* de l'observateur), et une pyramide de vision formée par l'œil et la fenêtre de visualisation. Il est à noter qu'une telle pyramide existe seulement dans le cas d'une projection perspective et d'une fenêtre de visualisation rectangulaire, ce qui coïncide avec la majorité des usages.

Les régions d'intérêt concernent :

- les faces situées à l'intérieur de la pyramide de vision,
- les faces orientées face à la caméra, les autres n'étant pas visibles par convention avec les infographistes. Ainsi la modélisation de l'intérieur d'une surface visible nécessite la création d'une seconde occurrence de chaque face d'orientation inverse et superposée sur la première,
- les silhouettes (voir définition section 4.2.1). On redéfinit la notion de faces appartenant à la silhouette par la bande de triangles adjacents au sous-ensemble des arêtes formant la silhouette du maillage pour le point de vue courant.

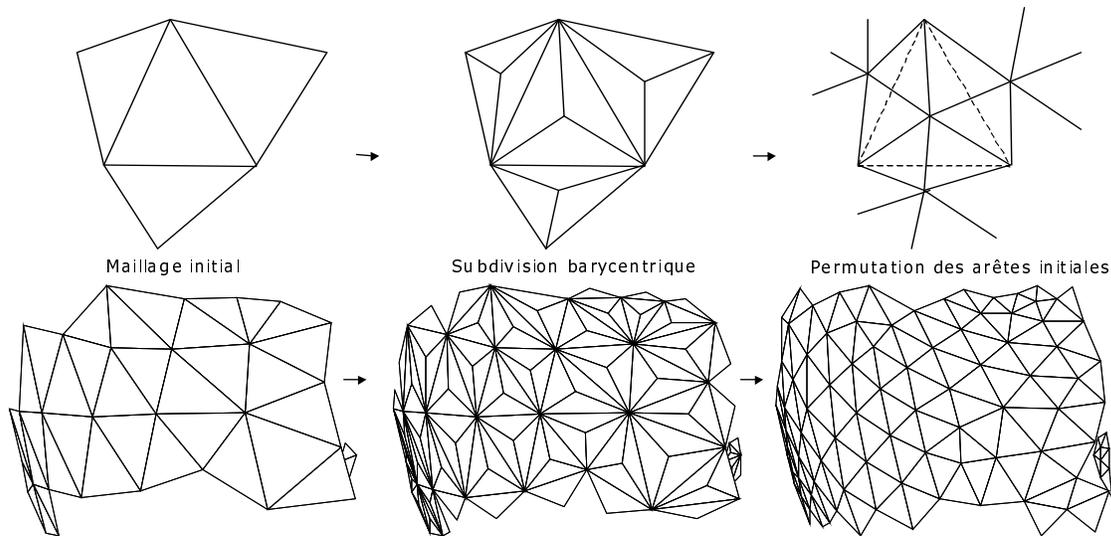


FIG. 4.9 : L'opérateur élémentaire de subdivision combine une subdivision barycentrique des triangles et la permutation des arêtes du maillage initial. Chaque triangle est ainsi découpé en trois triangles superposables par l'insertion d'un sommet en son barycentre, puis les arêtes du triangle original sont échangées entre les sommets initiaux et les nouveaux sommets générés par la subdivision.

4.3.3 Subdivision hybride

Le maillage est itérativement raffiné par subdivision hybride combinant subdivision barycentrique des triangles et permutation des arêtes du maillage original (figure 4.9). L'opérateur de subdivision barycentrique 1 vers 3 est choisi pour sa faible granularité, tandis que la permutation des arêtes du maillage original permet d'éviter une dégénérescence des triangles liée à l'opérateur élémentaire de subdivision (voir figure 4.10). Plus précisément, la permutation d'arête opère une redistribution des valences sur le maillage puisque chaque arête permutée réduit d'une unité la valence de ses deux sommets initiaux et augmente celle de ses deux nouveaux sommets (voir figure 4.11). Il en résulte une distribution plus équilibrée des valences et une meilleure qualité en forme (ou compacité) des triangles.

L'enjeu consiste à localiser le raffinement sur les silhouettes et les régions d'intérêt, tout en conservant une complexité géométrique réduite afin de satisfaire la contrainte de budget polygones. La figure 4.12 illustre les qualités de l'opérateur choisi permettant une cohabitation harmonieuse des niveaux de détail. La silhouette étant limitée à une bande de triangle, le raffinement peut s'avérer très localisé puisque l'opérateur est appliqué de manière itérative, en alternance avec une phase de détection de la silhouette devenant progressivement de plus en plus étroite au cours du raffinement (voir figure 4.13). Rappelons que l'effet souhaité correspond à la suppression de l'aspect polygonal des silhouettes. A chaque itération de subdivision, les sommets du maillage original sont positionnés afin d'obtenir des contraintes différentielles sur la région d'intérêt obtenue en théorie après une infinité de subdivisions. On parle aussi de C_k continuité, reliée à une notion de *douceur d'aspect*. La technique fonctionne ainsi par approximation, ou filtrage de la géométrie.

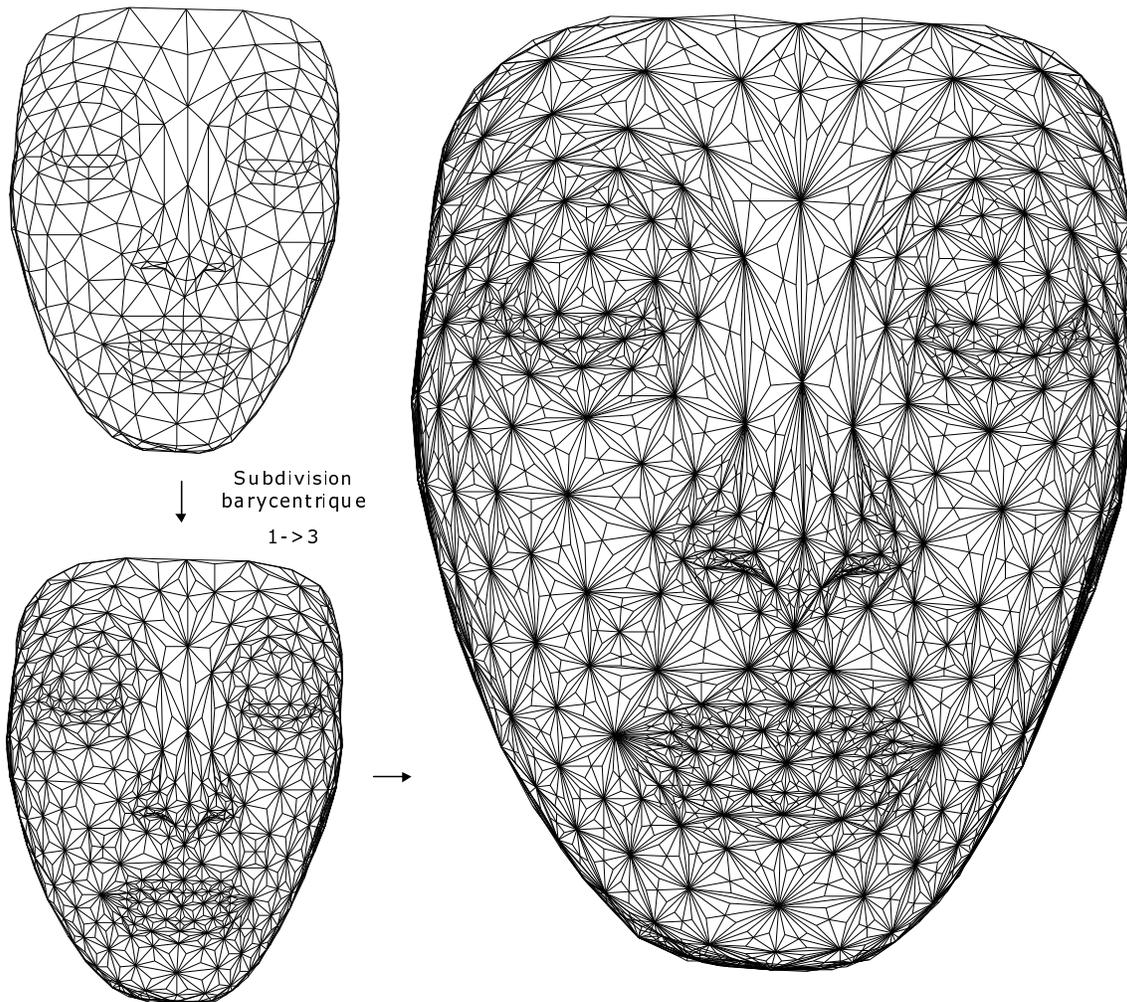


FIG. 4.10 : Le maillage initial, puis trois itérations successives de subdivision barycentrique des triangles. Chaque nouveau sommet inséré lors d'une étape de subdivision possède une valence égale à 3, et augmente d'une unité la valence de chaque sommet adjacent appartenant au maillage initial. Il en résulte la formation de triangles très peu compacts et une forte accumulation des valences sur les sommets initiaux.

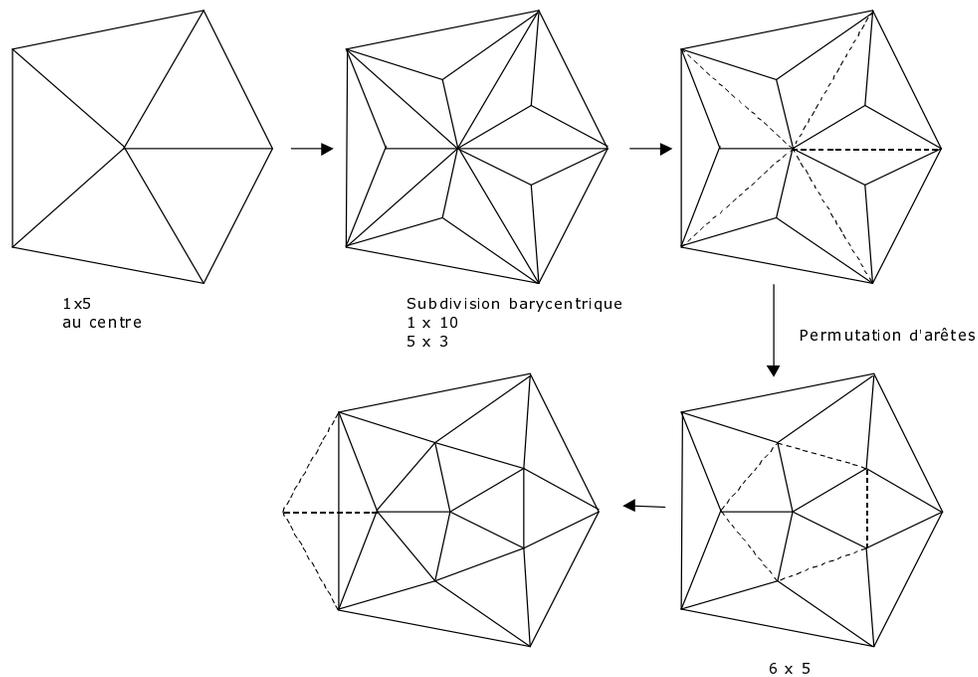


FIG. 4.11 : La permutation des arêtes du maillage initial opère une redistribution des valences sur le maillage. Sur cet exemple le sommet central possède une valence égale à 5. Après subdivision barycentrique, la valence du sommet initial est multipliée par 2 tandis que tous les nouveaux sommets insérés ont une valence égale à 3. Après permutation localisée des arêtes du maillage initial, 6 sommets possèdent la valence 5. On remarque sur la dernière figure que tous les nouveaux sommets auraient une valence 6 si la permutation d'arêtes n'était pas limitée à la région composée des faces adjacentes au sommet central.

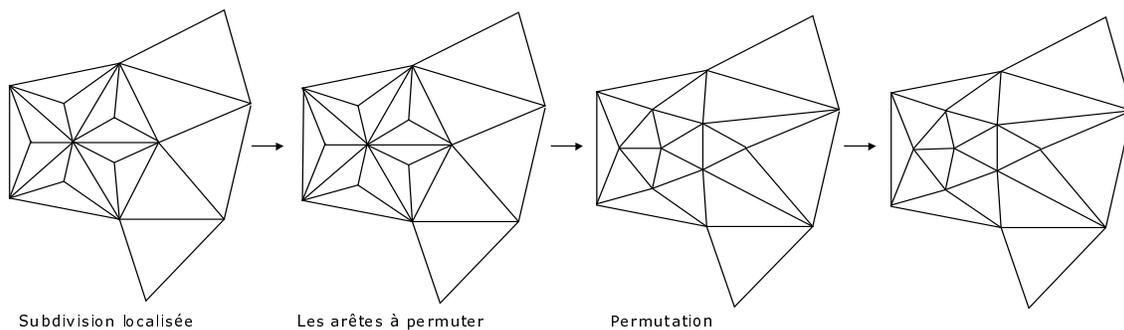


FIG. 4.12 : L'opérateur de subdivision choisi, autorise une cohabitation harmonieuse entre une région localement subdivisée et le maillage original. Le raffinement est en pratique très localisé sur la bande de triangles adjacents à l'ensemble des arêtes constituant la silhouette.

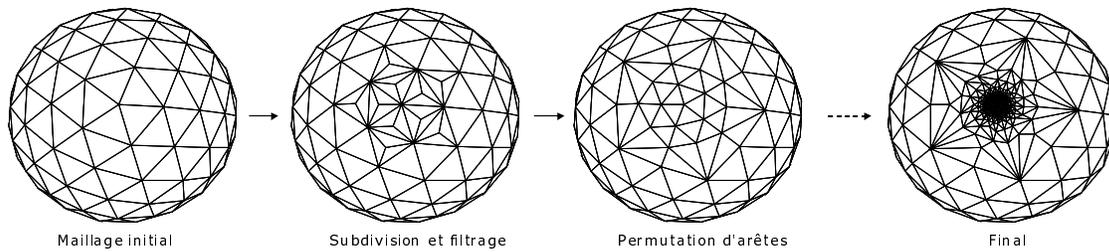


FIG. 4.13 : Le raffinement est appliqué localement et de manière itérative sur les régions d'intérêt. Sur cet exemple la région d'intérêt est progressivement réduite afin de simuler un raffinement localisé sur une zone appartenant à la silhouette. Plus intuitivement, il s'agit d'opérer le *branchement* entre une région très dense du maillage raffiné avec le maillage initial. La faible granularité de l'opérateur de subdivision (facteur 3) et l'opérateur de permutation d'arêtes sont à l'origine du raffinement progressif d'une région éparse vers une région dense.

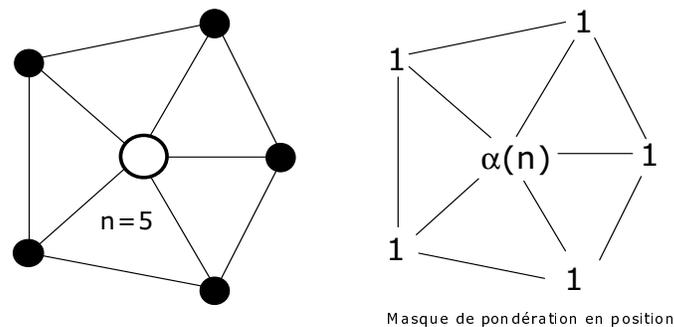


FIG. 4.14 : Masque de pondération en position pour un sommet de valence $n = 5$. L'enjeu consiste à déterminer les valeurs $\alpha(n)$ pour chaque valence afin d'obtenir des contraintes différentielles sur la surface après une infinité de subdivisions d'un maillage de contrôle irrégulier. Sur le terminal, le maillage de contrôle correspond au modèle décodé au cours de la transmission.

4.3.4 Filtrage de la géométrie

La nouvelle position d'un sommet du maillage initial est déduite d'une somme pondérée des positions de ses sommets adjacents et de lui-même. Le masque de pondération dépend alors de la valence du sommet (voir figure 4.14).

D'après [Loo87] [Pra98] [PU98a] [PU98b] [PU98c] [Um195] [Kob98], le calcul des coefficients de pondération $\alpha(n)$ pour chaque valence n s'obtient par l'analyse du comportement asymptotique d'une matrice de subdivision globale stochastique obtenue après numérotation des sommets. Cette matrice traduit ainsi la transformation de l'opérateur topologique de subdivision en un opérateur algébrique agissant seulement sur la géométrie au voisinage d'un sommet. Rappelons que les contraintes de différentiabilité sont requises sur la surface, indépendamment de la connectivité. Si l'on considère l'opérateur élémentaire de subdivision barycentrique après numérotation décrite par la figure 4.15, la matrice $A(5)$ permet de déterminer les positions des sommets après application locale de l'opérateur et du filtrage de la position d'un sommet V de valence 5 :

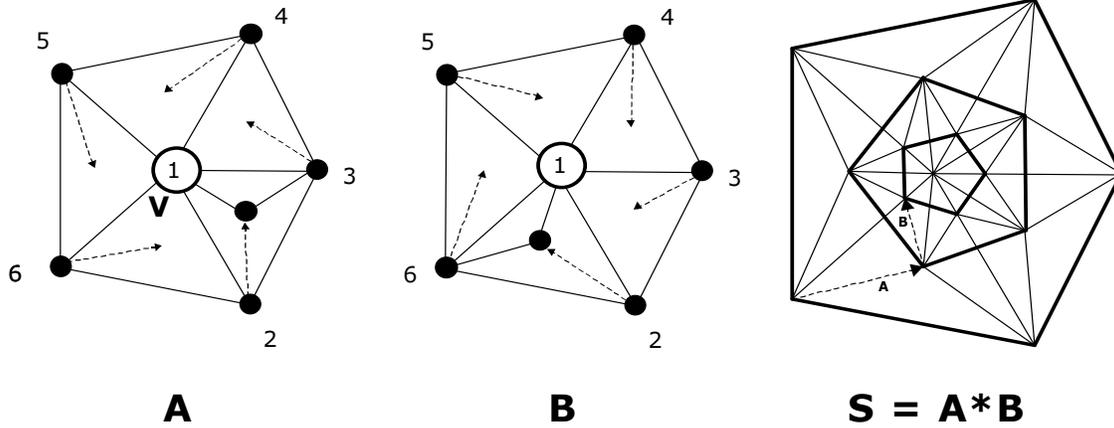


FIG. 4.15 : Matrice A : une itération de subdivision 1 vers 3 orientée dans le sens anti-horaire. Matrice B : une itération de subdivision 1 vers 3 orientée dans le sens horaire. La matrice globale $S = A * B$ permet d’obtenir un schéma auto-répliatif nécessaire avant toute analyse spectrale. Il est à noter que l’application de deux matrices de subdivision dans le sens horaire aurait nécessité l’application d’une matrice de permutation, inutile dans le cas présent.

$$A(5) = \begin{bmatrix} \frac{\alpha}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} \end{bmatrix}$$

L’analyse spectrale d’une matrice de subdivision nécessite la définition d’un schéma auto-répliatif, or la matrice $A(5)$ entraîne les sommets adjacents à V en rotation dans le sens anti-horaire (figure 4.15). Cette rotation est donc compensée par l’application successive d’une seconde matrice de subdivision $B(5)$:

$$B(5) = \begin{bmatrix} \frac{\alpha}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

L’expression matricielle $S(5)$ de l’opérateur de subdivision s’écrit donc sous la forme de deux itérations de subdivision barycentrique successives d’orientation inverse (voir figure 4.15). Les matrices $A(5)$, $B(5)$ et $S(5)$ sont ainsi liées par la relation : $S(5) = A(5) \cdot B(5)$.

Les valeurs propres multiples de la matrice stochastique $S(5)$ sont ensuite calculées sous une forme symbolique pour chaque valence. On obtient ainsi pour la valence 5 :

$$\lambda(5) = \left\{ 1, \frac{(2\alpha-5)^2}{9(\alpha+5)^2}, \frac{1}{6} + \frac{1}{18}\sqrt{5}, \frac{1}{6} - \frac{1}{18}\sqrt{5}, \frac{1}{6} + \frac{1}{18}\sqrt{5}, \frac{1}{6} - \frac{1}{18}\sqrt{5} \right\}$$

L'enjeu consiste alors à calculer le coefficient de filtrage $\alpha(5)$. Or d'après Prautzsch, Umlauf [PU98c] et Kobbelt [Kob98] [Kob99] les contraintes différentielles de C_2 -continuité sont obtenues par la résolution de la relation $\lambda_1^2 = \lambda_3$ après ordonnancement décroissant des valeurs propres multiples. Plus précisément, la résolution d'un tel système garantie une surface C_2 continue sur les sommets ordinaires (de valence 6 sur les maillages triangulaires), et C_1 continue ailleurs. Après résolution de l'équation $\frac{(2\alpha-5)^2}{9(\alpha+5)^2} = \left(\frac{1}{6} + \frac{1}{18}\sqrt{5}\right)^2$, on obtient $\alpha(5) = 8.305869973$. Les coefficients $\alpha(n)$ sont ensuite calculés pour les valences les plus courantes (voir tableau 4.1). L'annexe D consigne le détail des calculs pour chaque valence.

Valence n	$\alpha(n)$
3	2.4
4	5
5	8.305869973
6	12
7	15.88395686
8	19.84452688
9	23.82127931
10	27.78391446

TAB. 4.1 : Coefficients de filtrage de la géométrie calculés pour les valences les plus répandues.

La figure 4.16 compare deux maillages obtenus avec ou sans filtrage des positions, tandis que la figure 4.17 compare le lissage obtenu avec la technique de reconstruction de référence sur les maillages irréguliers [Loo87]. La technique de raffinement combinée au filtrage des positions s'avère bien adaptée aux surfaces courbes : l'aspect polygonal des silhouettes peut ainsi être supprimé et les ombrages sont mieux interpolés sur les régions orientées face à la caméra. Toutefois, si l'on considère le maillage de la figure 4.18, il peut s'avérer judicieux d'ajouter des contraintes sur la surface si l'on juge que certaines régions ne doivent pas être modifiées.

4.3.5 Ajout de contraintes

Le groupe des singularités géométriques comprend notamment les arêtes vives, les lignes de style et les coins. La définition de contraintes sur les faces, les sommets ou les coins du maillage respectivement à ne pas subdiviser, déplacer ou permuter permet de conserver ces éléments au cours du raffinement. Les lignes de style sont dessinées sur les surfaces par les infographistes afin d'être traitées spécifiquement par les algorithmes de simplification et de rendu. Parfois une telle ligne ne correspond pas systématiquement à la définition d'une arête vive (i.e. formant un pli supérieur à un seuil prédéfini), et possède pourtant une valeur esthétique très importante. Il s'avère donc utile de la conserver au cours du raffinement au même titre qu'une arête vive, le graphiste devant alors fournir une étiquette spécifique pour l'ensemble des arêtes représentant les lignes de style.

La figure 4.19 illustre le comportement adaptatif de l'algorithme de raffinement après classification des sommets en trois classes suivant leur nombre d'arêtes vives adjacentes. Il est ainsi naturel d'interdire :

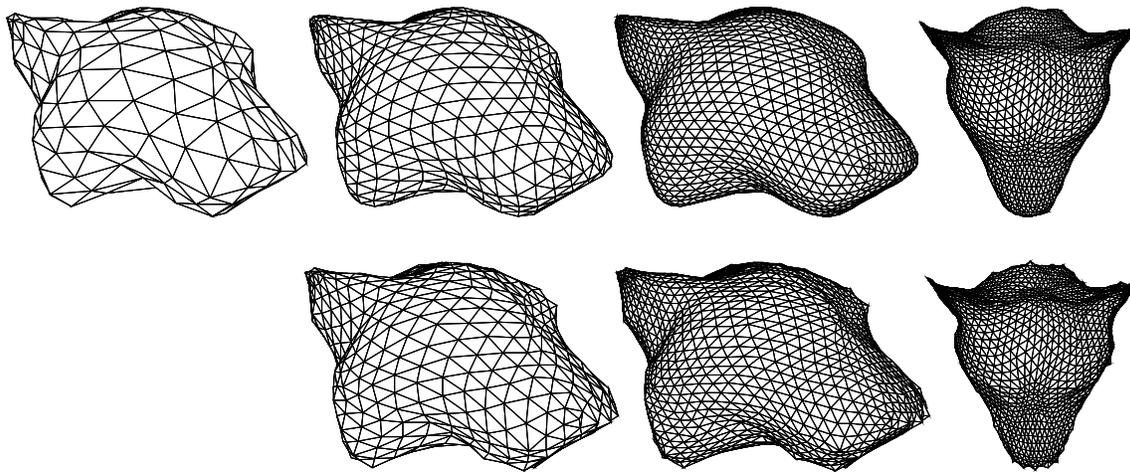


FIG. 4.16 : En haut, et respectivement de gauche à droite : maillage initial, première itération de subdivision hybride avec filtrage des positions, seconde itération avec filtrage, et détail sur la silhouette d'un point de vue différent. En bas : même descriptif mais sans filtrage des positions. On constate dans ce cas les artefacts liés à la permutation d'arêtes.

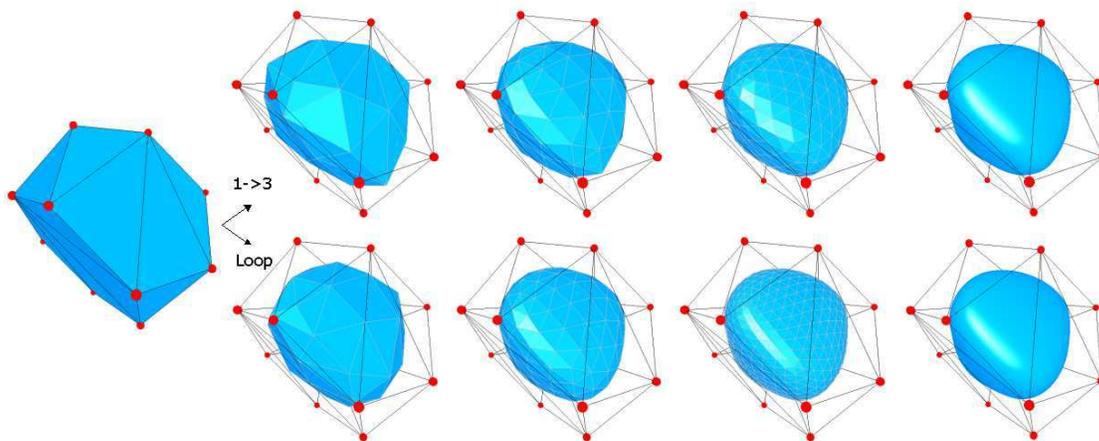


FIG. 4.17 : Le modèle de base (20 faces) correspond à un ensemble de sommets de contrôle et d'une connectivité définie sur les sommets. Sur cet exemple, la technique de subdivision est comparée avec la subdivision de Loop. Dans les deux cas le modèle converge vers une surface C_2 continue sur les sommets de valence 6 (les sommets ordinaires) et C_1 continue ailleurs (les sommets extraordinaires). Chaque nouveau sommet inséré possédant la valence 6, on peut en conclure que la surface finale est C_1 continue sur un nombre fini de points. Après 6 itérations de subdivision globale, l'algorithme 1 vers 4 atteint 81920 faces via 6 itérations en 7.711 s sur un Pentium II 266 MHz (soit 10623 faces/s) tandis que la méthode 1 vers 3 atteint 14580 faces en 2.018 s (soit 7224 faces/s). Cette différence de vitesse s'explique par la permutation d'arêtes et la mise à jour des adjacences dans le cas 1 vers 3.

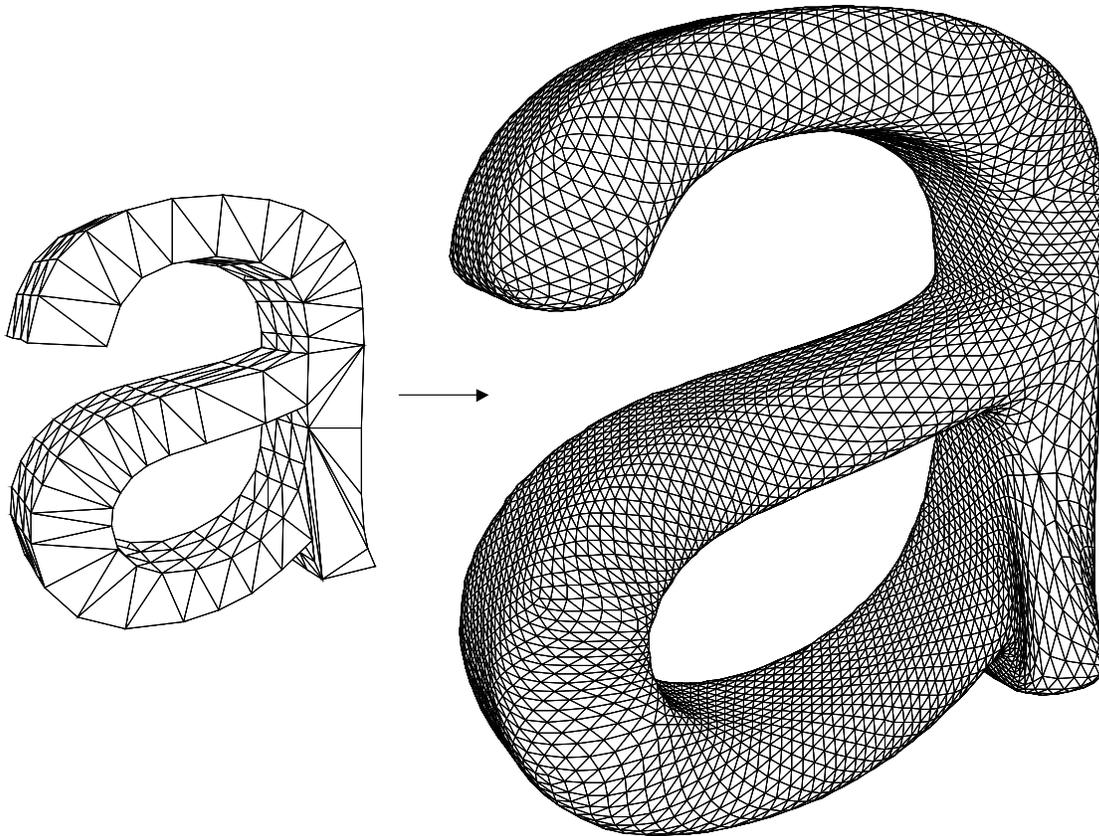


FIG. 4.18 : La subdivision non contrainte d'un maillage comportant des singularités géométriques peut entraîner un lissage indésirable. Sur cet exemple, les coins et les arêtes vives sont lissés par la combinaison du filtrage de la géométrie et de la permutation d'arêtes.

- la permutation des arêtes formant une arête vive ou une ligne de style,
- le déplacement des sommets formant des coins,
- et la subdivision (inutile) des faces situées sur des régions quasi planes.

La figure 4.19 montre l'efficacité de l'absence de permutation des arêtes vives. Toutefois, la dégénérescence soulignée par la figure 4.10 resurgit localement le long des arêtes vives régulières (voir figure 4.20). L'enjeu consiste alors à approximer par lissage anisotrope (i.e. dans la direction de plus faible courbure) des courbes tridimensionnelles formées par les arêtes vives régulières. Comme pour les surfaces, l'outil de base de l'approximation des courbes consiste en un opérateur de subdivision combiné au filtrage des positions des sommets. Les arêtes vives subissent ainsi une subdivision barycentrique (figure 4.21) tandis que les positions des sommets sont filtrées dans la direction de l'arête. De la même manière que Hoppe et al [HDD⁺94], les positions des sommets (nouveaux ou initiaux) sont déterminées par le biais des masques de filtrage présentés par la figure 4.22. Le processus de subdivision des arêtes vives converge ainsi vers les B-splines cubiques unimodales à partir de sommets réguliers au sens de la définition fournie par Hoppe [HDD⁺94] (voir figure 4.23). On constate toutefois que la densité des sommets sur une arête vive approximée augmente plus rapidement que la densité du maillage. Cette augmentation est liée au fait qu'une bissection des arêtes vives est réalisée, tandis que la subdivision hybride effectue en réalité une trisection toutes les deux itérations de subdivision (voir figure 4.24), en même temps qu'une subdivision régulière un vers neuf des faces. Une augmentation uniforme de la densité géométrique nécessiterait ainsi l'application d'une projection des sommets toutes les deux itérations sur la courbe approximée de l'arête vive régulière. Le choix de la bissection des arêtes est toutefois conservé afin d'obtenir une granularité unitaire de raffinement, tandis que la trisection des arêtes est appliquée une fois sur deux sur les bords du maillage tel que le préconise Kobbelt [Kob99].

4.3.6 Continuité du raffinement

Le raffinement doit s'effectuer sans artefacts visuels ni modifications brusques de la géométrie des surfaces. Idéalement, la subdivision hybride combinée au filtrage devrait s'effectuer de manière visuellement continue. Or l'opérateur comprend une phase de permutation d'arêtes, ce qui rend délicat l'application d'une interpolation géométrique. Cependant, la conservation de cet opérateur est souhaitable puisqu'il présente des qualités intéressantes en terme de granularité et de cohabitation des niveaux de détail (voir section 4.3.3).

Afin de résoudre le problème de la continuité de l'opérateur de permutation d'arête, les trois pistes suivantes ont été explorées. Chacune présente des caractéristiques variables en terme de complexité algorithmique :

1. l'interpolation conjointe géométrie / normale : la subdivision barycentrique d'une face combinée au filtrage des positions peut être directement interpolée par migration progressive d'un sommet source choisi arbitrairement sur cette face, vers le centre de la face. Cette interpolation s'effectue simultanément avec la migration des sommets originaux vers leur position respective après filtrage, et une interpolation des normales (voir figure 4.25).

La permutation d'arête est plus délicate puisque la migration des sommets le long des arêtes entraîne soit une dégénérescence des triangles, soit la génération de quadrilatères. Nous axons donc l'interpolation uniquement sur l'illusion provoquée par les normales (l'effet de relief). Les normales cibles sont cette fois calculées après permutation d'arête, cette dernière opération étant réalisée en une itération (voir figure 4.26).

Cependant, l'effet de relief (ou effet *bump*) ainsi exploité ne fonctionne pas sur les silhouettes du maillage puisque la géométrie modifiée de manière brusque masque ou laisse apparaître le fond de la scène suivant le sens de la permutation. Nous pourrions en rester là si nous considérons que l'effet serait per-

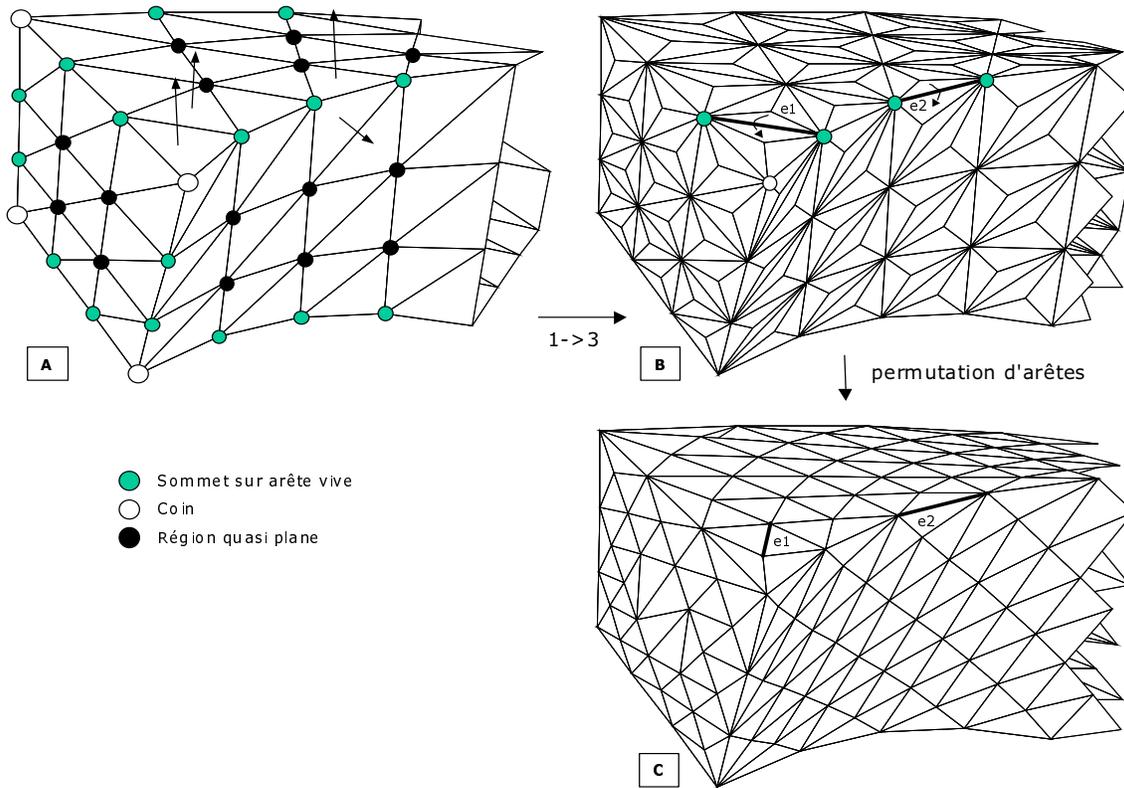


FIG. 4.19 : [A] le maillage initial. Une classification des sommets est opérée suivant leur nombre d'arêtes vives adjacentes. Un sommet appartient à une arête vive s'il possède exactement deux arêtes vives adjacentes, un coin possède au moins trois arêtes vives adjacentes et un sommet lisse réside sur une région quasi plane en l'absence d'arête vive. Les sommets de dards possèdent exactement une arête vive adjacente et sont classés avec les sommets lisses en l'état actuel des développements. [B] la subdivision barycentrique s'opère simultanément avec un filtrage systématique des sommets, excepté sur les coins ou les arêtes vives. [C] La permutation d'arête n'est pas systématique. Remarquons que l'arête vive e_2 n'est pas permuée, à l'inverse de l'arête e_1 non vive formée par deux sommets possédant pourtant deux arêtes vives adjacentes.

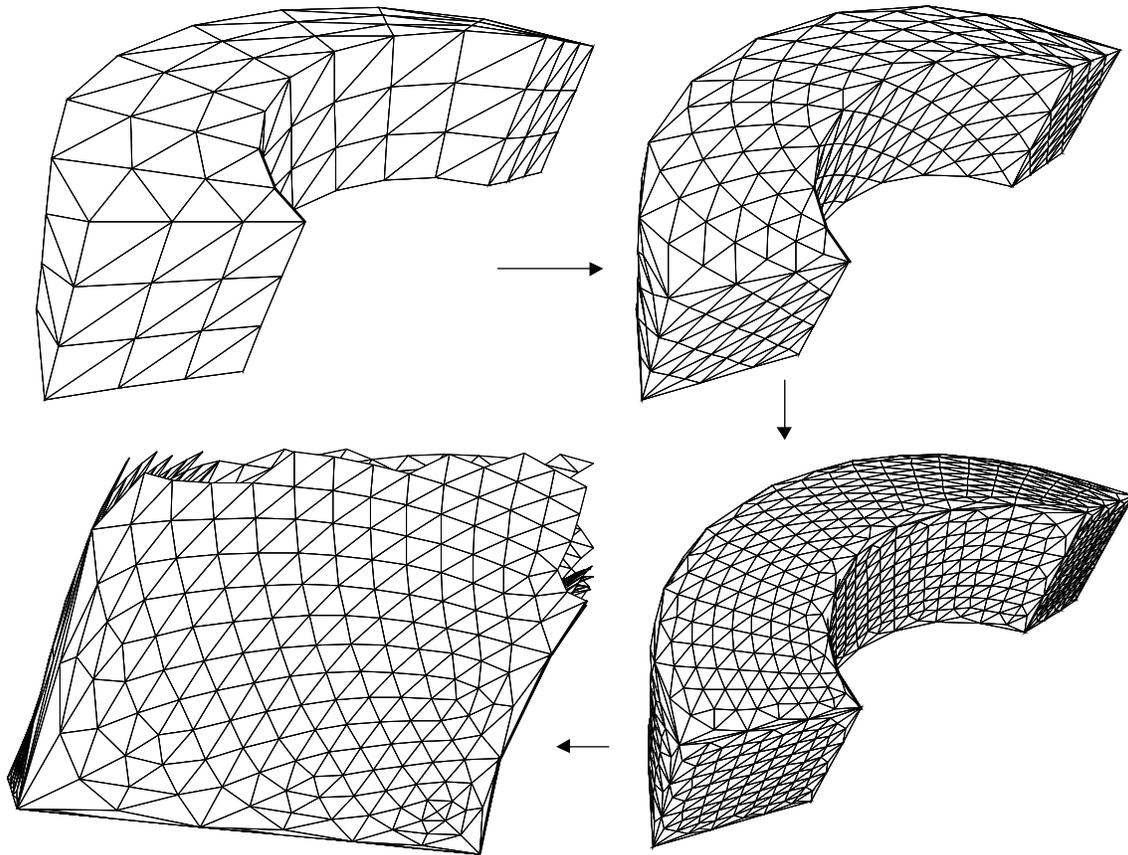


FIG. 4.20 : Sur cet exemple les positions des sommets situés sur les arêtes vives et les coins ne sont pas filtrés, et les arêtes vives (en gras) ne sont pas permutées. L'absence de permutation d'arête localisée sur les arêtes vives aboutit à une dégénérescence des éléments composant les triangles adjacents à une arête vive. On remarque également un aspect polygonal des arêtes après plusieurs itérations de subdivision globale du maillage. L'enjeu consiste maintenant à reconstruire les arêtes vives régulières par subdivision et filtrage anisotrope.

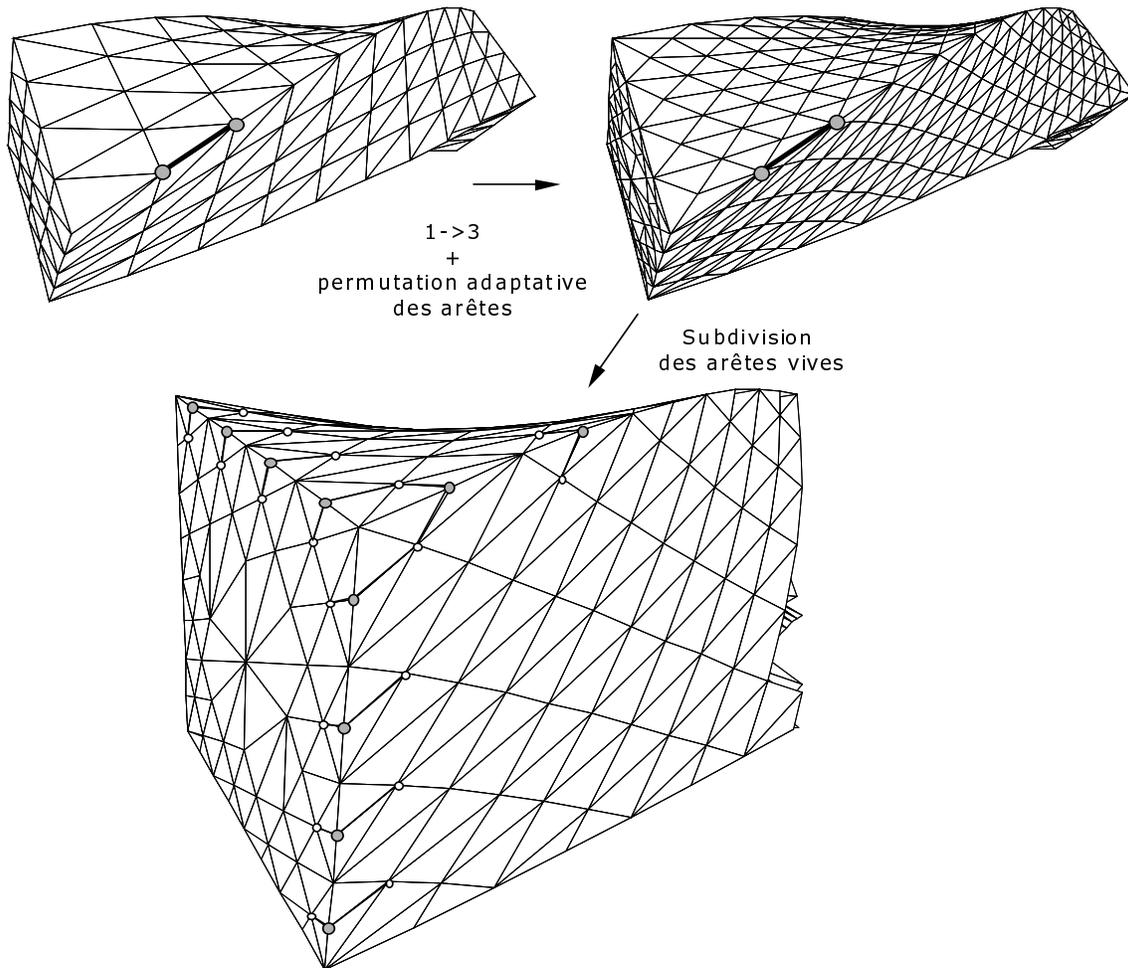


FIG. 4.21 : la subdivision des courbes et des surfaces constitue l'outil de prédilection pour la reconstruction. Le lissage adaptatif des maillages est ainsi obtenu par la subdivision barycentrique des arêtes vives, associée à un filtrage anisotrope des positions des sommets la composant. La méthode s'hybride ainsi avec la technique de subdivision régulière un vers quatre dans sa version adaptative [HDD⁺94].

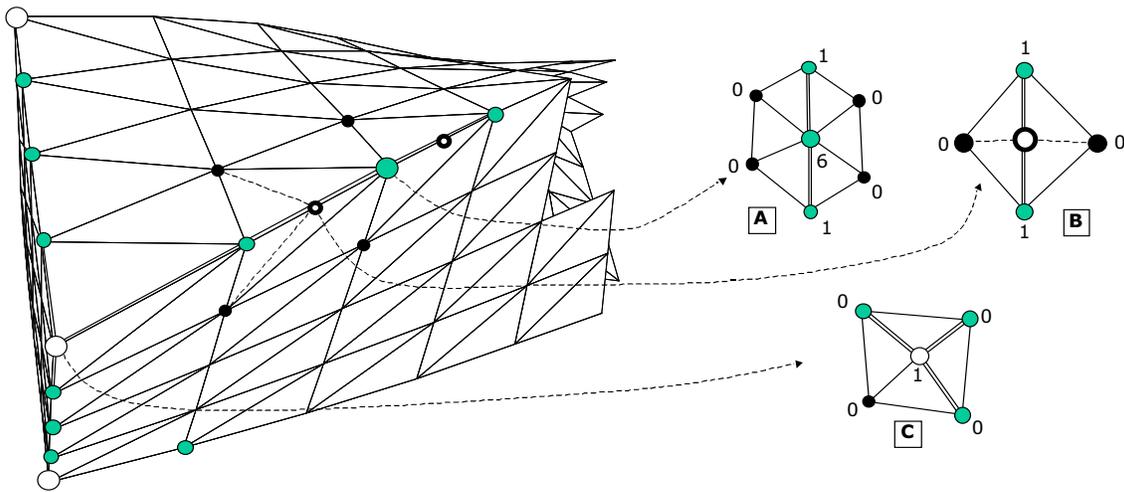


FIG. 4.22 : A gauche : le maillage initial. La subdivision barycentrique de l'arête vive régulière est définie par le masque de pondération *B* appliqué avant le filtrage des sommets initiaux. Le filtrage du sommet sur l'arête est donné par le masque de pondération *A*, qui annule l'influence des sommets situés de part et d'autre d'une arête vive. Ainsi la courbe formée par les arêtes vives initiales est lissée par filtrage de ses points de contrôle représentés par les sommets situés sur l'arête vive. Le masque de filtrage d'un coin *C* illustre l'absence de déplacement par annulation de toute influence ; ainsi les coins sont conservés au cours de la reconstruction.

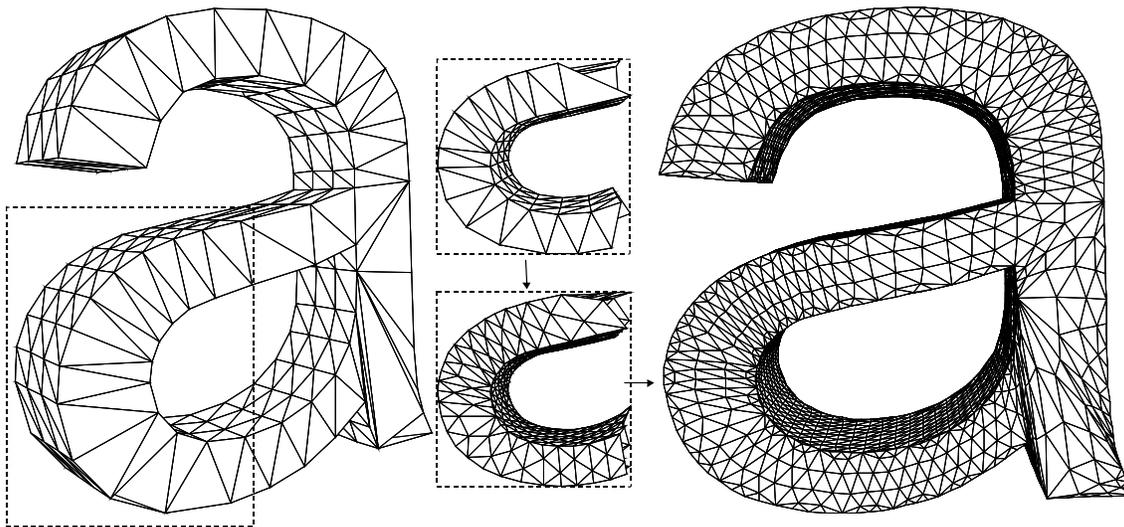


FIG. 4.23 : Sur cet exemple, deux itérations successives de subdivision adaptative sont appliquées sur un maillage comportant des coins, des arêtes vives et des régions courbes. On constate à droite la conservation des arêtes vives tout en obtenant un lissage des arêtes.

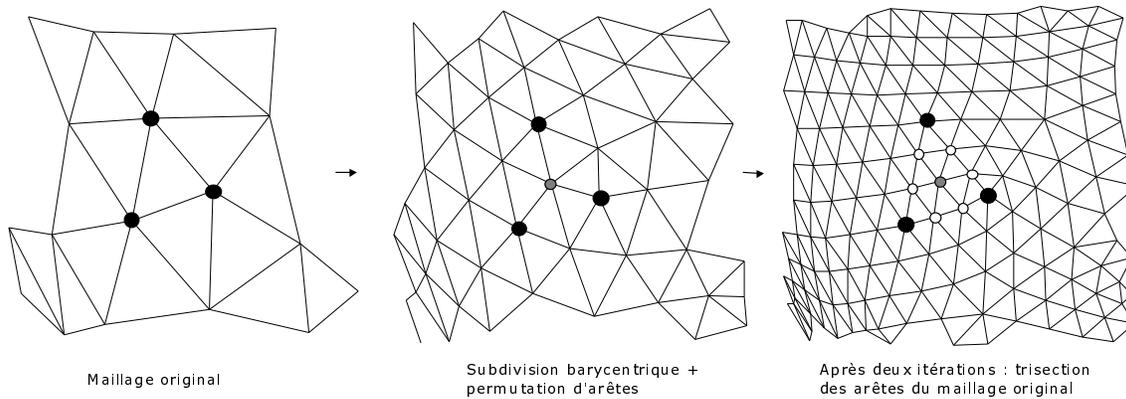


FIG. 4.24 : Deux itérations successives de subdivision barycentrique et permutation d'arêtes correspondent à une subdivision un vers neuf régulière, associée à une trisection des arêtes du maillage original. En pratique, les arêtes vives sont subdivisées à chaque itération par bissection afin de conserver une granularité unitaire de raffinement.

ceptible seulement sur les arêtes de forte courbure de la silhouette, a priori considérées vives et donc non permutées (voir section 4.3.5). Les pistes suivantes approfondissent le sujet.

2. la permutation d'arête cachée : les deux premières lignes de la figure 4.27 illustrent le principe de l'interpolation conjointe géométrie / normales lors de la subdivision barycentrique. Une telle interpolation génère au départ pour chaque face originale deux faces dégénérées (voir figure 4.25), une permutation d'arête entre deux faces dégénérées adjacentes pouvant s'opérer de manière parfaitement imperceptible. Les deux dernières lignes de la figure 4.27 illustrent une partie du principe qui n'est en réalité pas applicable sur la totalité de la surface, la dégénérescence de toutes les faces étant requise afin de masquer de manière exhaustive les permutations d'arête. Cette piste s'avère donc exploitable à condition de permuter seulement l'ensemble des arêtes adjacentes à deux faces dégénérées (voir figure 4.28), puis d'exploiter l'effet de relief de la piste précédente.
3. les faces temporaires : la permutation d'arête peut être interpolée par migration progressive d'une copie des sommets composant l'arête (voir première ligne de la figure 4.29). Cette opération génère des quadrilatères et n'est pas applicable si l'on souhaite conserver une structure de données composée de triangles. Il est toutefois possible de trianguler ces quadrilatères en générant deux faces temporaires se substituant aux faces originales en fin d'interpolation. Cette dernière piste s'avère efficace malgré le coût mémoire des sommets et des faces temporaires nécessaires à l'interpolation.

Nous venons de décrire trois pistes classées par ordre de complexité croissante. Retenons qu'il s'avère indispensable dans les trois cas d'interpoler les normales cibles après permutation de telle sorte que l'on ne transite jamais par une configuration de normales calculées après la subdivision simple 1 vers 3 (les sommets de degré trois générant des artefacts visuels). Une fois cette considération prise en compte, examinons une solution tirant partie des qualités de chacune des méthodes. La seconde piste s'avère très intéressante lorsque le masquage de la permutation est possible, tandis que la première exploite l'effet de relief fourni par les normales et s'applique parfaitement sur les régions de faible courbure et hors silhouette. Enfin, la dernière s'avère coûteuse en mémoire puisque la structure de données s'enrichit de nouveaux éléments temporaires lors de la permutation des arêtes, mais garantit la continuité de manière systématique. La solution proposée consiste donc naturellement à faire coïncider les arêtes de la silhouette avec les arêtes adjacentes à deux faces dégénérées de manière à masquer leur permutation. La silhouette correspond ainsi à une région privilégiée puisque la moindre modification de la géométrie s'avère très visible sur cette région. La piste des faces temporaires s'avère ensuite

bien adaptée aux régions de forte courbure n'appartenant pas à la silhouette, tandis que la piste d'interpolation conjointe s'avère finalement bien adaptée aux régions restantes.

Au cours d'une visualisation interactive sur un terminal, le point de vue est amené à varier très rapidement. L'enjeu consiste alors à construire une structure de données adaptée à un raffinement rapide, robuste et réversible. La section suivante concerne l'implantation logicielle efficace de ce raffinement.

4.3.7 Optimisation et robustesse

L'optimisation concerne la vitesse d'exécution et le coût mémoire, tandis que la structure de données influence sensiblement les performances et la robustesse d'un algorithme. La robustesse est obtenue dans le cas présent en simplifiant à l'extrême le nombre de types de données manipulées, et en limitant le nombre de prédicats mis en œuvre [Abr00]. L'enjeu consiste à construire un moteur de raffinement susceptible de brasser plusieurs milliers de faces par seconde au cours de la navigation dans une scène virtuelle aussi bien en création qu'en suppression, après avoir opéré une phase de détection des régions d'intérêt.

L'annexe C décrit les algorithmes sous la forme de pseudo-codes, puis les sections suivantes détaillent la détection des régions d'intérêt et la structure de données employée.

Détection des régions d'intérêt

Afin d'accélérer les temps de calcul de la phase de détection, les éléments appartenant aux régions d'intérêt sont déterminés de manière dichotomique en trois vagues successives classées par ordre décroissant de complexité des prédicats mis en œuvre :

1. la pyramide de vision : calculée à partir des coins de la fenêtre de visualisation et du centre de la caméra. Seules les faces adjacentes aux sommets situés à l'intérieur de cette pyramide sont concernées par le raffinement adaptatif,
2. l'orientation des faces : seules les régions composées d'éléments orientés face à la caméra sont raffinées,
3. la silhouette : sur les faces retenues lors des deux phases précédentes, seules les faces adjacentes aux sommets composant les arêtes de la silhouette sont retenues pour un raffinement localisé.

En pratique l'algorithme raffine le maillage par vagues successives en utilisant itérativement l'opérateur de subdivision. Les premières itérations sont appliquées sur les faces contenues par la pyramide de vision et orientées face à la caméra (dans le champ de vision), tandis que les suivantes ne concernent que les régions localisées autour de la silhouette. Au cours de la navigation, la caméra et les objets sont amenés à être transformés par des homothéties, des translations et des rotations, le tout pouvant être appliqué sur des groupes d'objets assemblés lors de la modélisation. La détection nécessite quant à elle la définition d'un repère absolu pour la caméra et les coordonnées des sommets des maillages définis dans ce repère. Afin d'éviter le calcul des coordonnées de l'ensemble des sommets de chaque maillage, un équivalent exprimé en transformation de caméra est exprimé par la technique des quaternions [Wag98]. A titre d'exemple, entraîner un maillage en rotation autour de son centre de gravité se traduit en une rotation inverse de la caméra autour du centre de gravité de l'objet. La caméra étant définie par un point unique (son centre), le temps de calcul s'en trouve réduit de manière drastique (le rapport équivaut au nombre de sommets du maillage).

Absence de détection par héritage

La détection des faces situées dans le champ est effectuée de manière exhaustive uniquement lors de la première itération de détection car les faces issues de la subdivision de ces régions sont considérées à nouveau

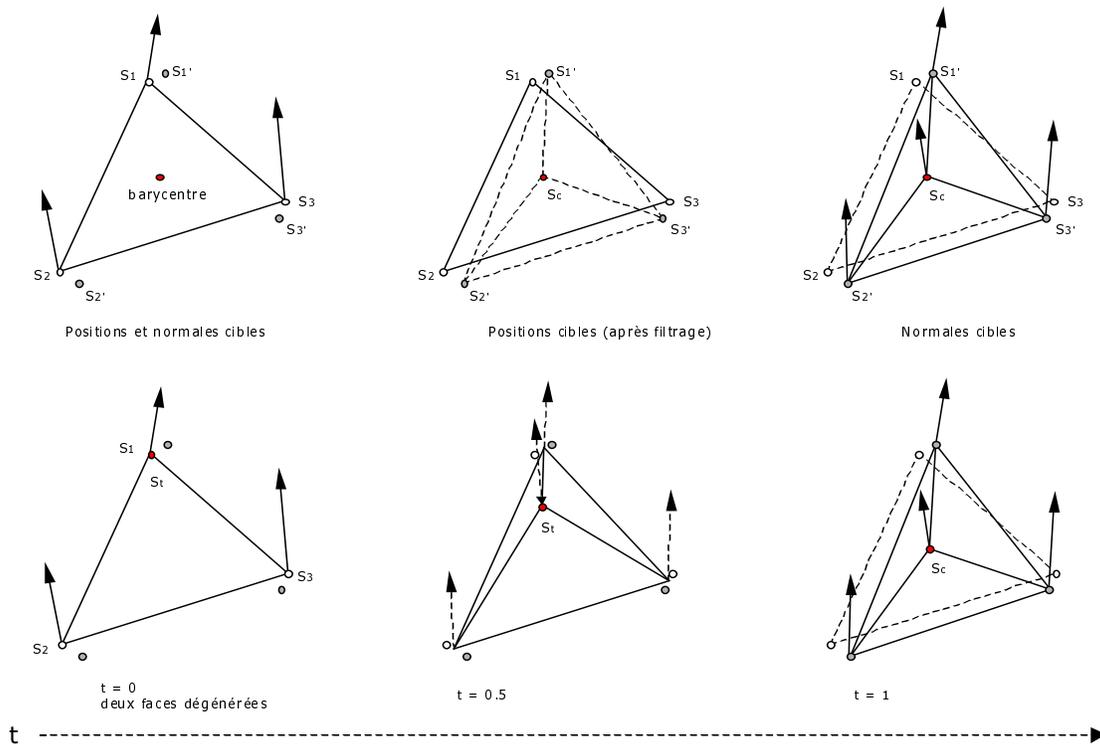


FIG. 4.25 : Interpolation conjointe géométrie / normales. Première ligne : la subdivision barycentrique se décompose en plusieurs étapes : insertion d'un sommet S_c au centre de la face formée par S_1, S_2 et S_3 , modification de la face (le sommet S_c remplace S_1), création de deux faces $\{S_1, S_c, S_2\}$ et $\{S_1, S_3, S_c\}$, et filtrage des positions ($\{S_1, S_2, S_3\}$ deviennent $\{S'_1, S'_2, S'_3\}$). Le calcul des positions finales donne également lieu à un calcul de normales cibles. Seconde ligne : à l'instant $t = 0$ les nouveaux éléments (faces et sommets) sont générés tandis que le sommet S_c (dénote S_t temporairement le temps de l'interpolation) est superposé sur S_1 . Les deux nouvelles faces créées sont donc dégénérées car de surface nulle. La continuité est ensuite obtenue par interpolation linéaire conjointe géométrie / normale entre les valeurs initiales et les valeurs cibles. L'interpolation des normales peut s'effectuer sous une forme linéaire ou suivant tout autre principe plus sophistiqué comme l'interpolation le long d'une géodésique sur la sphère unité. Notons qu'une telle interpolation ne résout toutefois pas la continuité visuelle de la permutation d'arêtes.

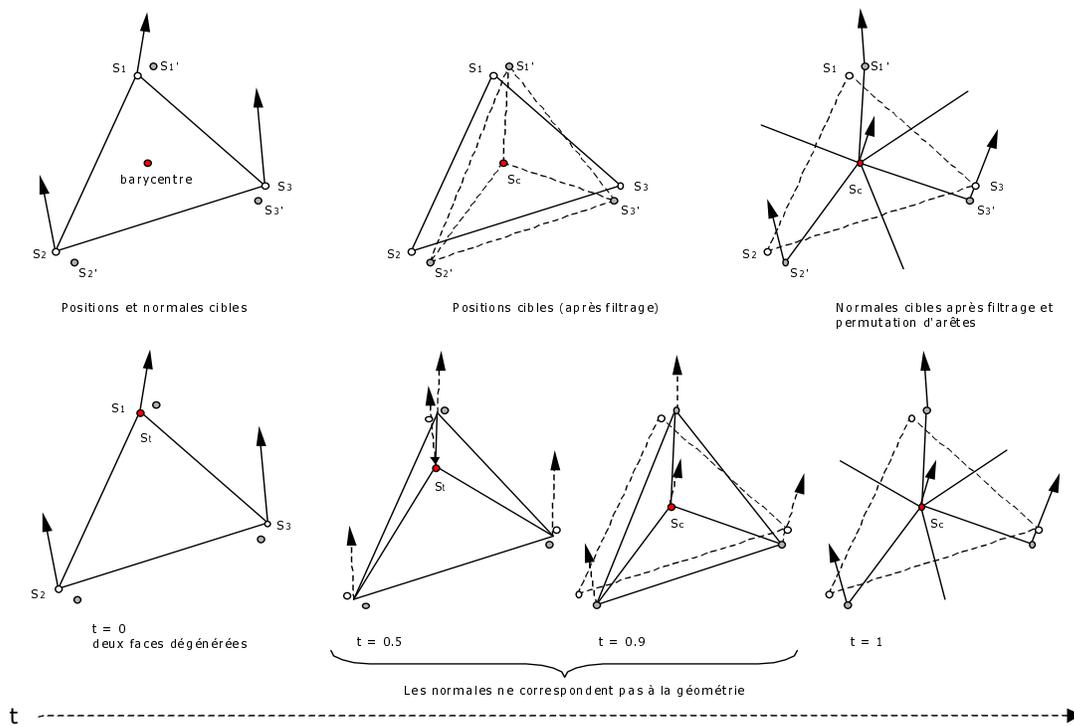


FIG. 4.26 : Les nouvelles positions et la permutation d'arêtes donnent lieu à un nouveau calcul de normales cibles. Lors de l'interpolation, les normales ne sont volontairement pas en accord avec la géométrie afin de rendre imperceptible la permutation opérée lors de la dernière étape de l'interpolation (entre $t = 0.9$ et $t = 1$ sur cet exemple).

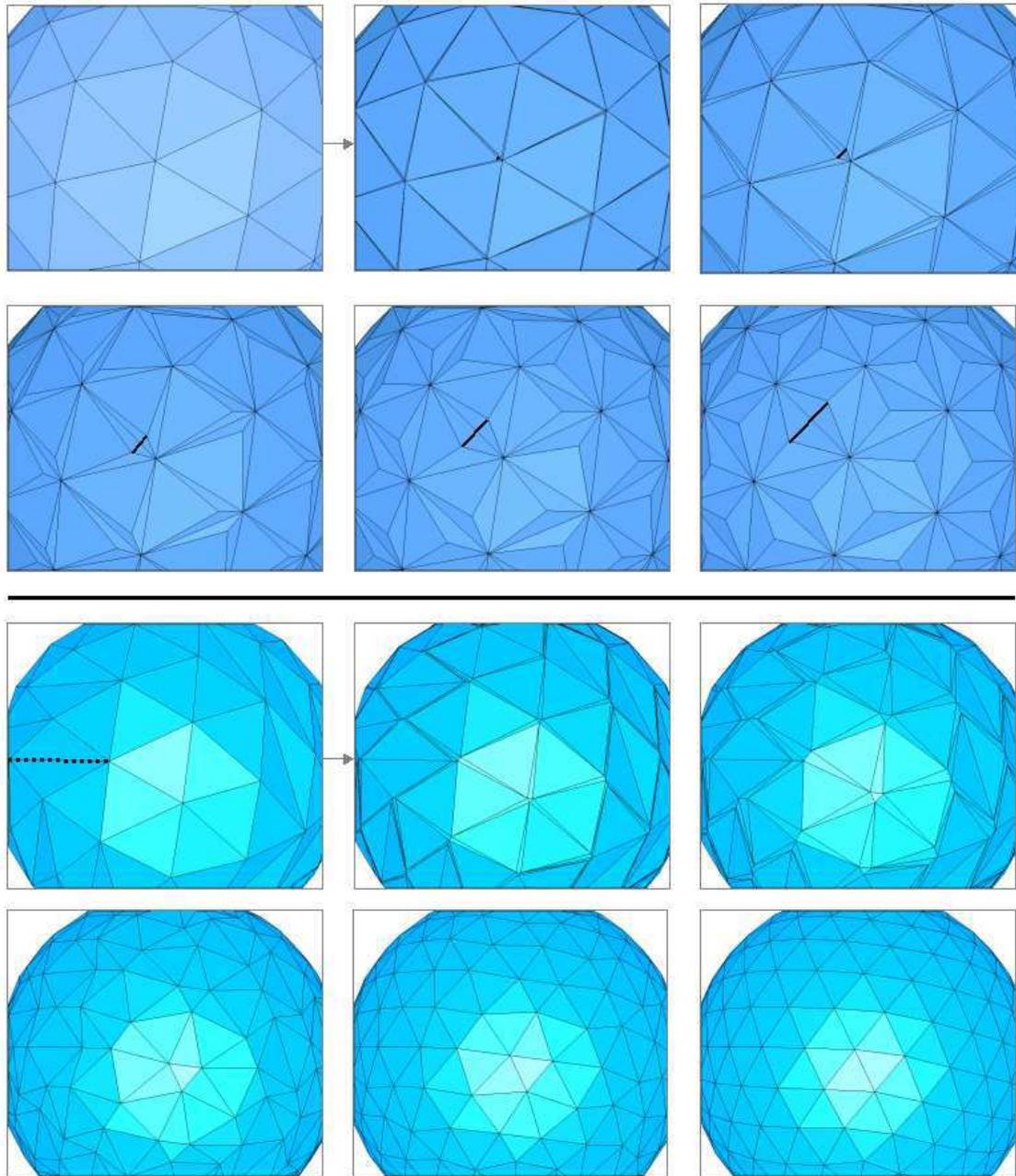


FIG. 4.27 : Les deux premières lignes illustrent le principe de l'interpolation géométrique au cours du raffinement géométrique par subdivision barycentrique un vers trois des faces originales. Pour chaque face originale, deux faces dégénérées (i.e. de surface de nulle) sont créées. En gras : une arête adjacente à deux faces adjacentes et dégénérées, mise en évidence après permutation. La permutation d'une telle arête est imperceptible lorsqu'elle est effectuée en position dégénérée. La piste consiste alors à permuer initialement et systématiquement les arêtes du maillage original afin de *masquer* cette opération (troisième et quatrième ligne). En pratique, cette opération est impossible puisqu'elle nécessiterait la présence de deux faces dégénérées adjacentes à chaque arête originale, soit un maillage entièrement dégénéré. Cette constatation explique la présence de permutations visibles dès la première itération d'interpolation (en pointillé), et renforce le choix d'une application partielle de la technique, combinée à une interpolation conjointe géométrie / normales.

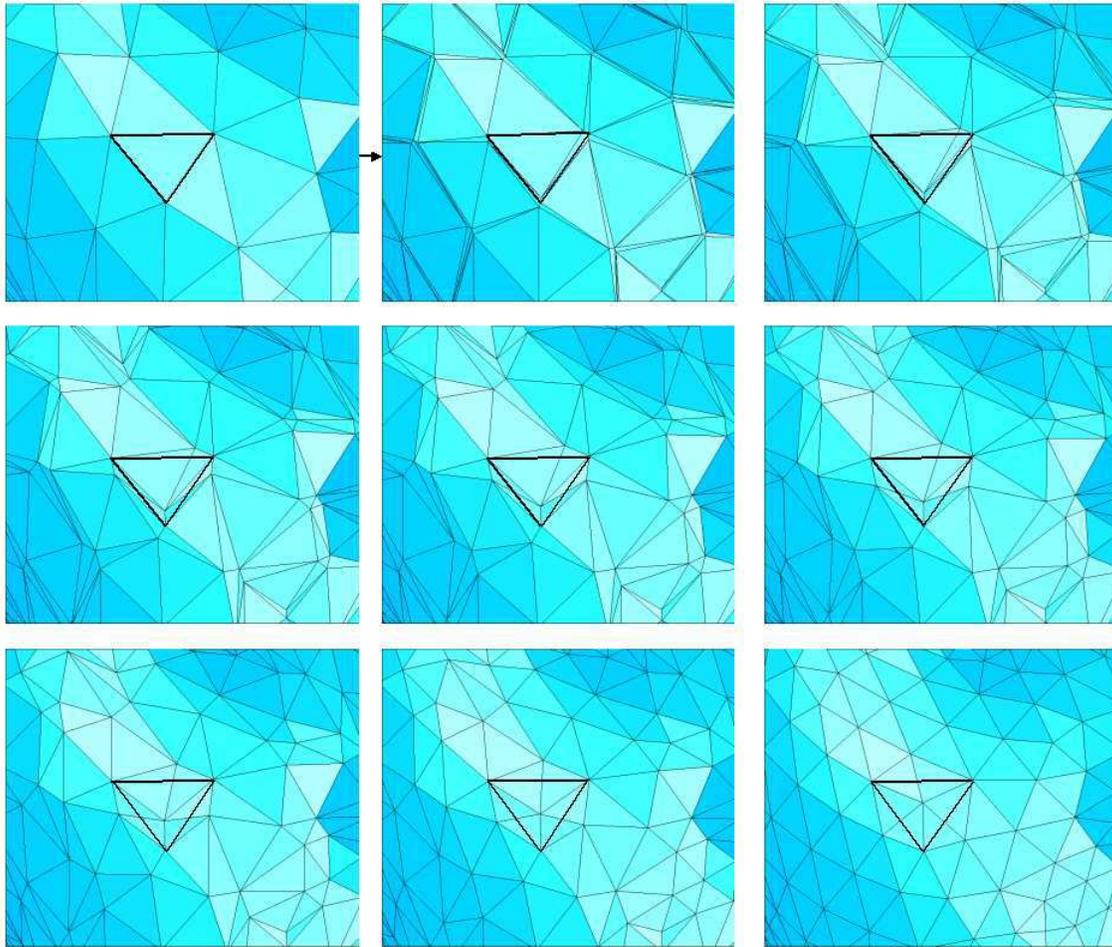


FIG. 4.28 : La progressivité de l'opérateur de subdivision utilisé est obtenue de manière correcte sur les arêtes adjacentes à deux faces dégénérées. La face mise en évidence est subdivisée en son centre, tandis que les arêtes du maillage original sont systématiquement permutes avant l'interpolation.

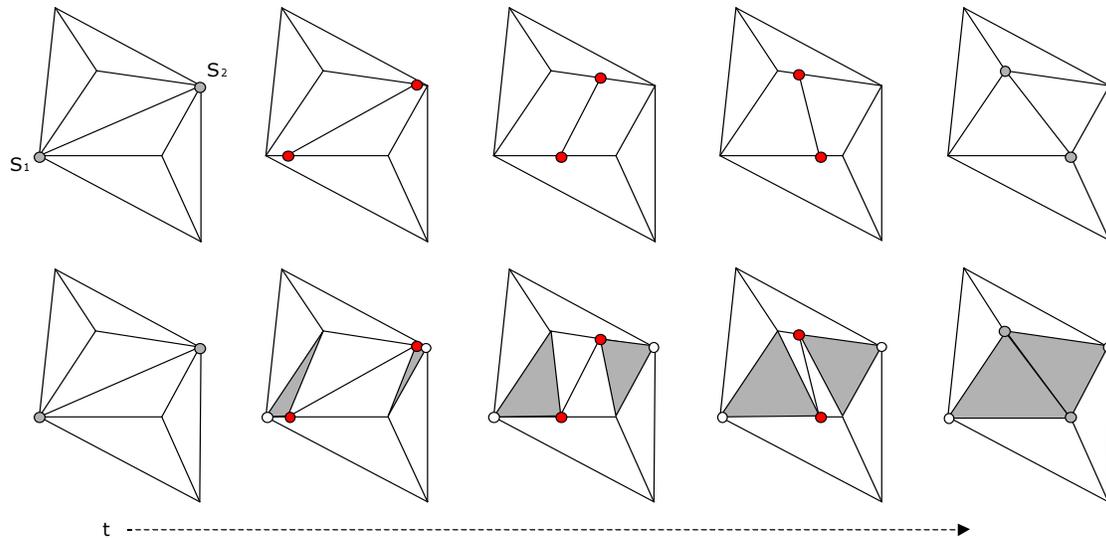


FIG. 4.29 : Première ligne : la permutation d'arêtes peut être interpolée par une transformation des triangles en quadrilatères due au déplacement progressif le long des arêtes des sommets S_1 et S_2 . En pratique, il est nécessaire de générer une copie des sommets afin d'éviter la dégénérescence des triangles au cours de l'interpolation. Seconde ligne : les quadrilatères sont triangulés par deux faces additionnelles se substituant aux faces originales à la dernière étape de l'interpolation.

dans le champ de vision par héritage des propriétés de leurs faces mères. Le niveau de raffinement est fixé par un nombre d'itérations, ce dernier pouvant toutefois être lié à une grandeur géométrique mesurée après projection à l'écran ou à un critère de courbure. Le raffinement est notamment superflu sur les régions quasi planes.

Accélération par filtrage et héritage

La première détection de la silhouette ne s'applique que sur le sous-ensemble germe des faces situées dans le champ de vision puisqu'une arête située sur la silhouette partage exactement deux faces dont une est orientée face à la caméra. La région étant élargie à une bande de triangles adjacents aux sommets composant les arêtes de la silhouette, le support de la silhouette n'est pas entièrement situé dans le champ de vision. Pour les raffinements successifs, seules les faces issues de la détection initiale sont retenues comme candidates à la nouvelle détection de silhouette par héritage des propriétés lors de la subdivision. On constate en pratique un rétrécissement de la bande définissant la silhouette, ce dernier étant lié aux raffinements géométriques successifs (figure 4.36).

La détection étant effectuée, examinons maintenant une structure de données adaptée à la visualisation interactive.

Structure de données

Les régions d'intérêt comme les silhouettes sont itérativement raffinées pour un point de vue à l'instant t avant d'être restaurées à leur complexité initiale à l'instant $t + 1$ lorsque le point de vue a été modifié par un mouvement de caméra et / ou d'objet. Deux solutions se présentent alors :

1. une copie du maillage est conservée en permanence afin de restaurer le modèle en une itération. Cette solution s'avère coûteuse en mémoire et présente l'inconvénient de ne pas autoriser la continuité visuelle

de l'opération inverse du raffinement,

2. une structure de données additionnelle stockant l'information de raffinement est générée afin de restaurer le modèle en une séquence de simplifications élémentaires. Cette solution a été choisie car elle présente l'avantage d'autoriser une réversibilité visuellement continue pour un faible coût, puisque seule l'information de raffinement localisé aux régions d'intérêt est générée.

Un historique réversible de raffinement

Un historique composé d'actions élémentaires est généré au cours du raffinement. Il est ensuite aisé d'appliquer à rebours la séquence d'actions composant l'historique. En pratique, chaque élément de type face ou sommet supporte sa propre histoire de raffinement, et chaque action est labellisée par son type et son instant d'application dans le repère temporel du raffinement. Une action labellisée par l'instant t ne concerne pas l'application effective du raffinement mais consigne plutôt l'information nécessaire au rétablissement de l'élément dans sa position à l'instant $t - 1$. Ainsi l'application à rebours d'une action de type *création de face* consiste à supprimer cette face de la mémoire. Nous commençons à percevoir l'intérêt de l'approche en terme d'optimisation mémoire puisque toute information inutile se libère lors de l'application à rebours d'un historique.

La liste suivante consigne les opérations élémentaires appliquées sur les éléments du maillage au cours du raffinement, et les quatre actions élémentaires associées :

- Création d'une face : lors de la subdivision barycentrique des faces ou des arêtes vives. En pratique une face subdivisée est conservée en mémoire tandis que deux nouvelles faces sont créées (figure 4.30). Il en est de même pour une face adjacente à une arête vive non permutée et subdivisée,
- Création d'un sommet : au centre d'une face (figure 4.30) ou au milieu d'une arête vive lors de la subdivision,
- Déplacement d'un sommet : lors du filtrage des positions des sommets combiné à l'opérateur de subdivision (figure 4.30),
- Permutation d'arête : successivement à une subdivision barycentrique. Pour des raisons de robustesse et de coût mémoire la structure ne comprend pas d'arête, une permutation étant équivalente à un échange de sommets entre deux faces adjacentes. La permutation correspond ainsi à deux actions élémentaires d'affectation de nouveaux sommets sur des faces (figure 4.31).

Nous venons de décrire une structure réversible par la réalisation de quatre actions élémentaires. Ces actions sont définies avec la compacité mémoire suivante :

- 8 bits pour caractériser l'action (2 bits idéalement),
- 8 bits pour indexer une action dans le temps, soit 256 actions de raffinement (15 représente toutefois le niveau maximal appliqué lors des essais),
- 8 bits pour indexer l'un des trois sommets d'une face (2 bits idéalement),
- 32 bits pour les références aux sommets (un code à longueur variable par index étant envisageable en connaissant les limites autorisées du raffinement exprimées en nombre de sommets),
- 3 x 32 bits pour les déplacements en position des sommets.

Une optimisation mémoire supplémentaire pourrait être développée en opérant une quantification des déplacements élémentaires et une représentation plus compacte des paramètres des actions.

Enfin, l'optimisation en vitesse est axée sur deux idées principales :

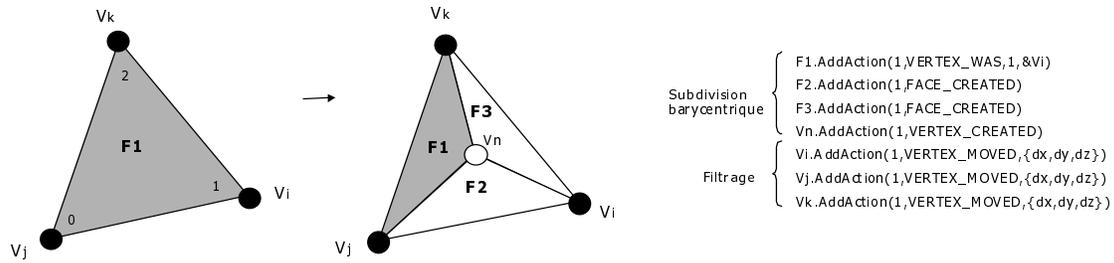


FIG. 4.30 : La subdivision barycentrique d’une face génère 4 actions élémentaires sur les faces et les sommets. La première action (labellisée par l’instant 1) concerne la face F_1 et signifie que son sommet indexé 1 pointait sur le sommet V_i avant la subdivision. De cette manière, il sera possible de restaurer cette face dans sa position initiale. Les faces F_2, F_3 et le sommet V_n sont ajoutés au maillage courant et se voient simultanément affectés d’une action de type création. Ainsi l’application antichronologique de la première tranche d’historique labellisée 1 supprimera ces éléments de la mémoire. Le filtrage des positions des sommets originaux se caractérise par l’ajout d’une action de type mouvement sur ces éléments.

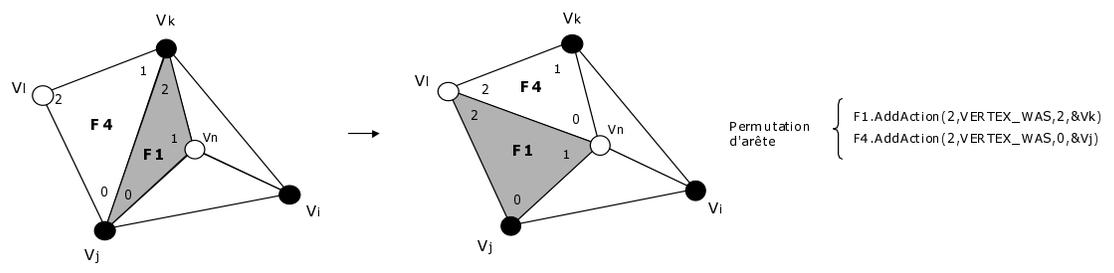


FIG. 4.31 : La permutation d’arête adjacente à deux faces se caractérise simplement par l’ajout de deux actions élémentaires de type *affectation*. Sur cet exemple, le sommet indexé 0 de la face F_4 pointait sur V_j , tandis que le second sommet de la face F_1 pointait sur V_k .

- Le pré-calcul des coefficients de filtrage des positions des sommets. Partant du principe que le nombre de valences distinctes est borné sur un maillage triangulaire et que la division de nombres flottants est sensiblement plus coûteuse qu'un produit, les coefficients $\alpha/(n + \alpha)$ et $1/(n + \alpha)$ sont pré-calculés dans une table (typ. entre les valences 3 et 11).
- L'allocation mémoire par blocs ou granulaire : La phase de détection des régions d'intérêt comptabilise le nombre d'éléments (faces, sommets et actions) à allouer, permettant ainsi l'allocation par blocs et non pas granulaire au cours de l'application des opérateurs de subdivision.

Ces deux propositions permettent d'augmenter la vitesse de raffinement d'un facteur 5, pour atteindre un ordre moyen de 5000 faces par seconde (détection comprise) sur un Pentium II 266 MHz. Bien entendu une optimisation plus approfondie agissant sur l'organisation des données et une implantation spécifique tirant partie du processeur (architecture superscalaire, instructions MMX, etc...) permettrait certainement d'augmenter ces performances [Abr00].

4.3.8 Résultats

Le raffinement adaptatif a pour but principal d'embellir les images générées à partir de modèles succinctement décrits pour des raisons de coûts de codage, de transmission et de capacités du terminal. Ces résultats mettent ainsi en évidence le gain en qualité obtenu à partir de maillages de base, que nous appellerons ici abusivement les *maillages de contrôle*. L'intérêt se situe également du côté de l'optimisation en vitesse d'affichage, en mémoire et même en temps de chargement d'un modèle tridimensionnel. Dans la pratique, on constate que pour un point de vue donné, et en moyenne, seule une fraction de la surface totale d'un objet est visuellement pertinente. Enfin, nous avons pu également constater qu'il est parfois sensiblement plus rapide de générer une géométrie par subdivision récursive plutôt que de décoder un train binaire ou plus simplement de lire dans un fichier les données au format brut. Les temps de calcul exprimés dans cette section sont mesurés sur un PII 266 MHz avec 256 Mo de mémoire, les coûts de construction des listes d'affichage pour la bibliothèque graphique *OpenGL* [SA98] étant exclus.

On présente successivement les résultats du raffinement global, le raffinement localisé sur les silhouettes, limité aux régions d'intérêt puis la version adaptative de l'algorithme. Enfin, la reconstruction de maillages en cours de transmission est également illustrée afin de replacer la technique dans son contexte final.

Comparaison avec la subdivision de Loop

La planche 4.32 associée aux courbes de la figure 4.33 illustre trois itérations successives de subdivision globale 1 vers 4 de Loop [Loo87] comparée avec la subdivision 1 vers 3 avec permutation d'arêtes. L'intérêt de la technique provient de la granularité sensiblement réduite puisque son facteur multiplicatif réduit autorise un raffinement progressif.

Raffinement global

La planche 4.34 illustre le fait que si la méthode de subdivision globale non adaptative s'avère bien adaptée aux objets composés de surfaces courbes, elle provoque néanmoins un effet lissage des singularités géométriques telles que les coins et les arêtes vives.

Raffinement dépendant du point de vue

Afin d'optimiser la qualité du rendu tout en limitant la complexité géométrique, il s'avère judicieux de ne raffiner que les régions perceptuellement pertinentes, où la silhouette joue un rôle prépondérant. La figure 4.35 illustre le gain en qualité obtenu par un raffinement localisé sur la silhouette et les régions orientées face à la caméra. La figure 4.36 décompose la détection et la subdivision de la silhouette sur le maillage *nefertiti*. La figure 4.37 met en évidence l'intérêt de la technique en terme d'économie de codage et de calcul lors du rendu sur le maillage *mannequin*. Sur cet exemple, la silhouette est raffinée jusqu'à l'obtention d'arêtes dont la projection n'excède pas la résolution de l'écran, cette heuristique permettant d'éviter l'aspect polygonal sur cette dernière. Les régions composées de faces orientées dos à la caméra ne sont pas raffinées. Cette économie de calcul est encore renforcée par l'application aux régions d'intérêt explicitement définies par l'utilisateur ou directement déduites du point de vue courant (figure 4.38).

Reconstruction de surfaces courbes par morceaux

La reconstruction de surfaces lisses par morceaux est illustrée par la figure 4.39. Les arêtes vives sont approximées par subdivision barycentrique des arêtes et lissage anisotrope dans la direction de faible courbure, tandis que les régions lisses sont approximées par subdivision barycentrique des faces, permutation d'arêtes et filtrage isotrope.

Reconstruction en cours de décodage

La reconstruction des surfaces est applicable en cours de décodage progressif des maillages à condition de répercuter les raffinements de la géométrie avant subdivision. La figure 4.40 illustre les qualités de reconstruction obtenues à partir des niveaux de base pour une très forte quantification des résidus après prédiction (voir chapitre 3 pour plus de précision). Ces résultats montrent l'intérêt de la technique de reconstruction du point de vue de la contrainte réseau. Le maillage *mannequin* peut ainsi être reconstruit avec respectivement 1327 et 6704 bits pour la connectivité et la géométrie (résidus quantifiés avec 4 bits), soit 8 kbit au total ou 9.57 bit/face. La distorsion géométrique liée à la quantification des résidus après prédiction des positions est pratiquement imperceptible après raffinement. A titre comparatif le fichier *mannequin.wrl* occupe 283 kbit, soit un taux de compression de 35. La figure 4.41 présente un exemple d'application mettant en œuvre deux briques technologiques développées au cours de cette thèse : le codage échelonné en position, associé à la reconstruction par subdivision.

La section suivante présente une discussion des méthodes présentées, ainsi que les perspectives de développement.

4.4 Discussion et perspectives

Discussion

La piste des méthodes de subdivision adaptative sur les silhouettes et les régions situées dans la pyramide de vision a été privilégiée dans la mesure où ces algorithmes répondent assez bien aux contraintes de réversibilité, de robustesse, de vitesse et surtout de codage pour une exploitation dans le contexte client-serveur. L'opérateur utilisé s'avère bien adapté à la cohabitation des régions très raffinées avec les régions du maillage de contrôle initial. On constate en pratique qu'un raffinement très localisé sur les silhouettes et les régions d'intérêt suffit à améliorer considérablement la qualité des images produites tout en respectant un budget exprimé en nombre de polygones sur un terminal. Des contraintes ajoutées (les lignes de style) ou automatiquement détectées (les

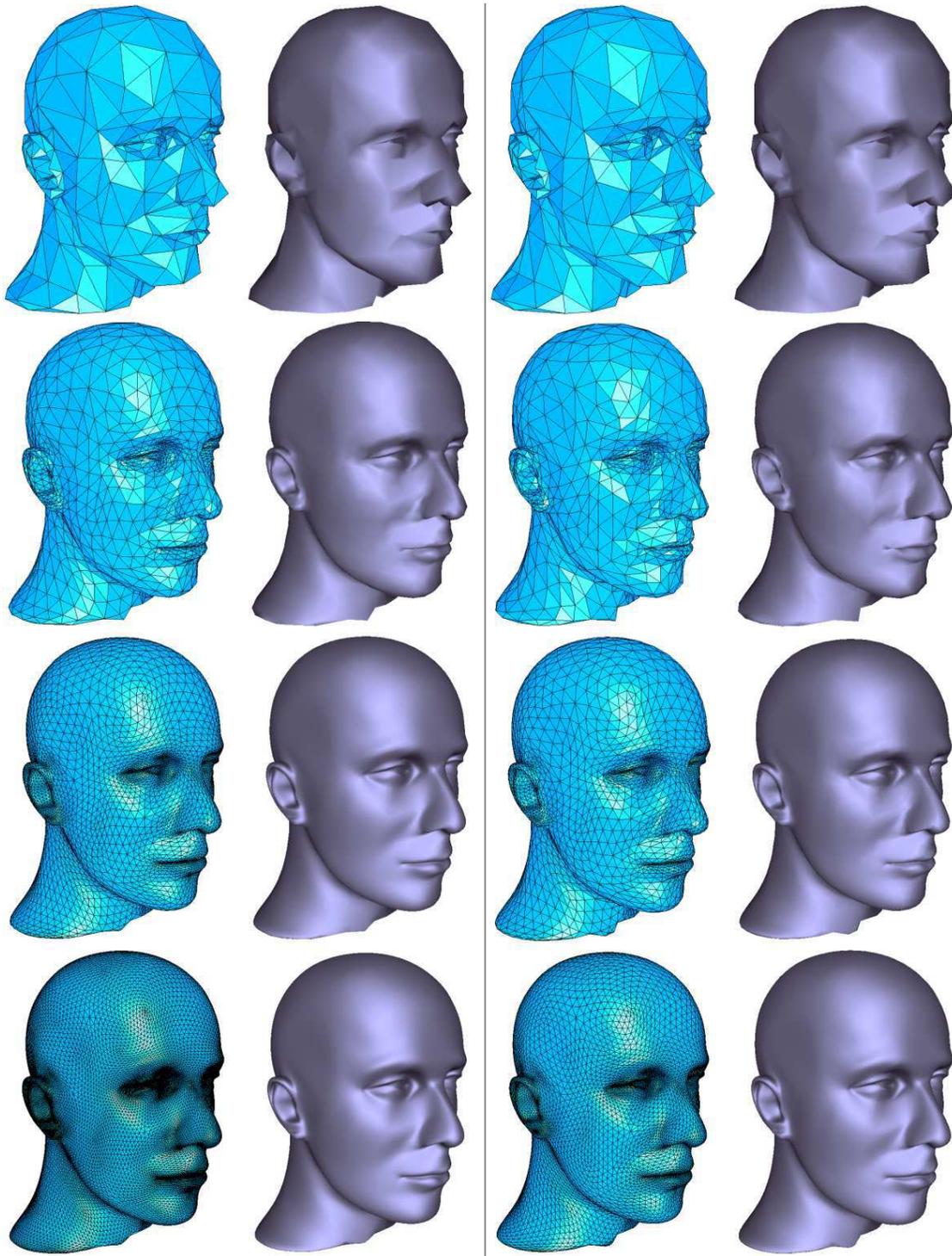


FIG. 4.32 : A gauche : le maillage initial est successivement raffiné par la technique de subdivision de Charles Loop [Loo87]. Le nombre de triangles est multiplié par 4 à chaque itération pour aboutir à la progression suivante : 838, 3352, 13408 et 53632 faces. A droite : l'opérateur de subdivision 1 vers 3 combiné à l'opérateur de permutation d'arêtes : 838, 2514, 7542 et 22626 faces. Une telle subdivision est intrinsèquement échelonnée et possède une granularité avantageuse, notamment pour l'adaptation à la puissance d'un terminal graphique.

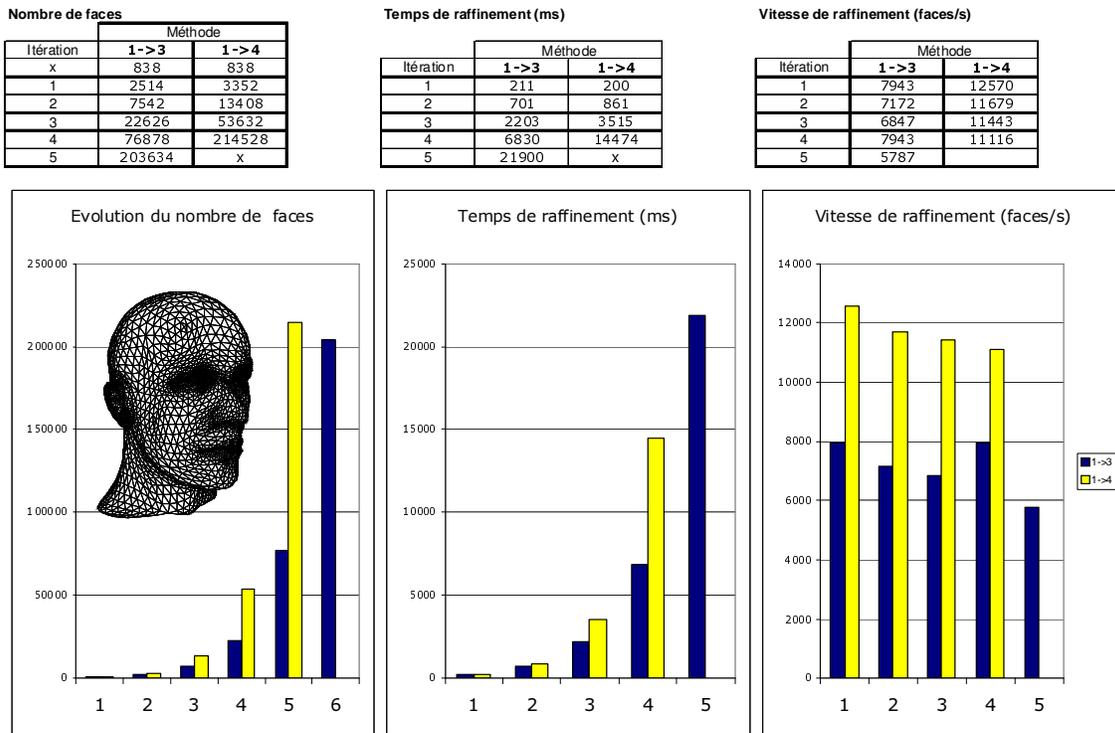


FIG. 4.33 : Statistiques liées à la figure 4.32. Les temps de calcul sont mesurés sur un PII 266 MHz. Le maillage *mannequin* est raffiné par subdivision globale 1 vers 3 ou 1 vers 4. On constate une granularité plus fine avec le raffinement 1 vers 3, mais une vitesse de raffinement inférieure principalement liée à l’opérateur de permutation d’arêtes nécessitant une mise à jour locale de l’adjacence. La vitesse ne correspond toutefois pas au critère prioritaire dans le cas présent puisque le raffinement 1 vers 3 permet d’obtenir une qualité de rendu équivalente dès la quatrième itération de subdivision globale, soit après 6.8 s contre 14.474 s pour la subdivision de Loop.

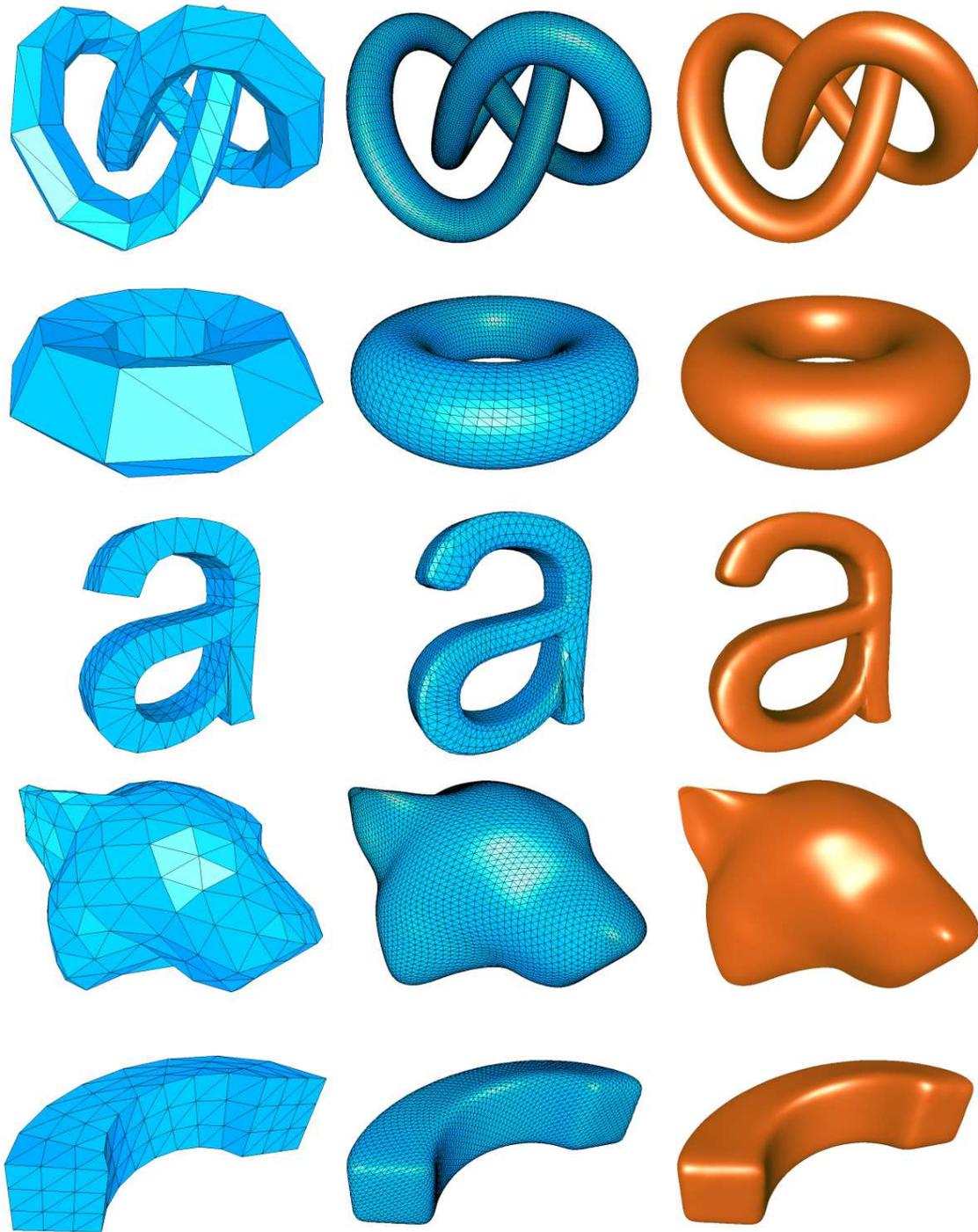


FIG. 4.34 : La qualité du rendu est sensiblement améliorée par la technique de subdivision globale sur les objets composés de surfaces courbes. Toutefois les singularités géométriques telles que les coins et les arêtes vives sont dégradées par le filtrage des positions et la permutation systématique des arêtes.

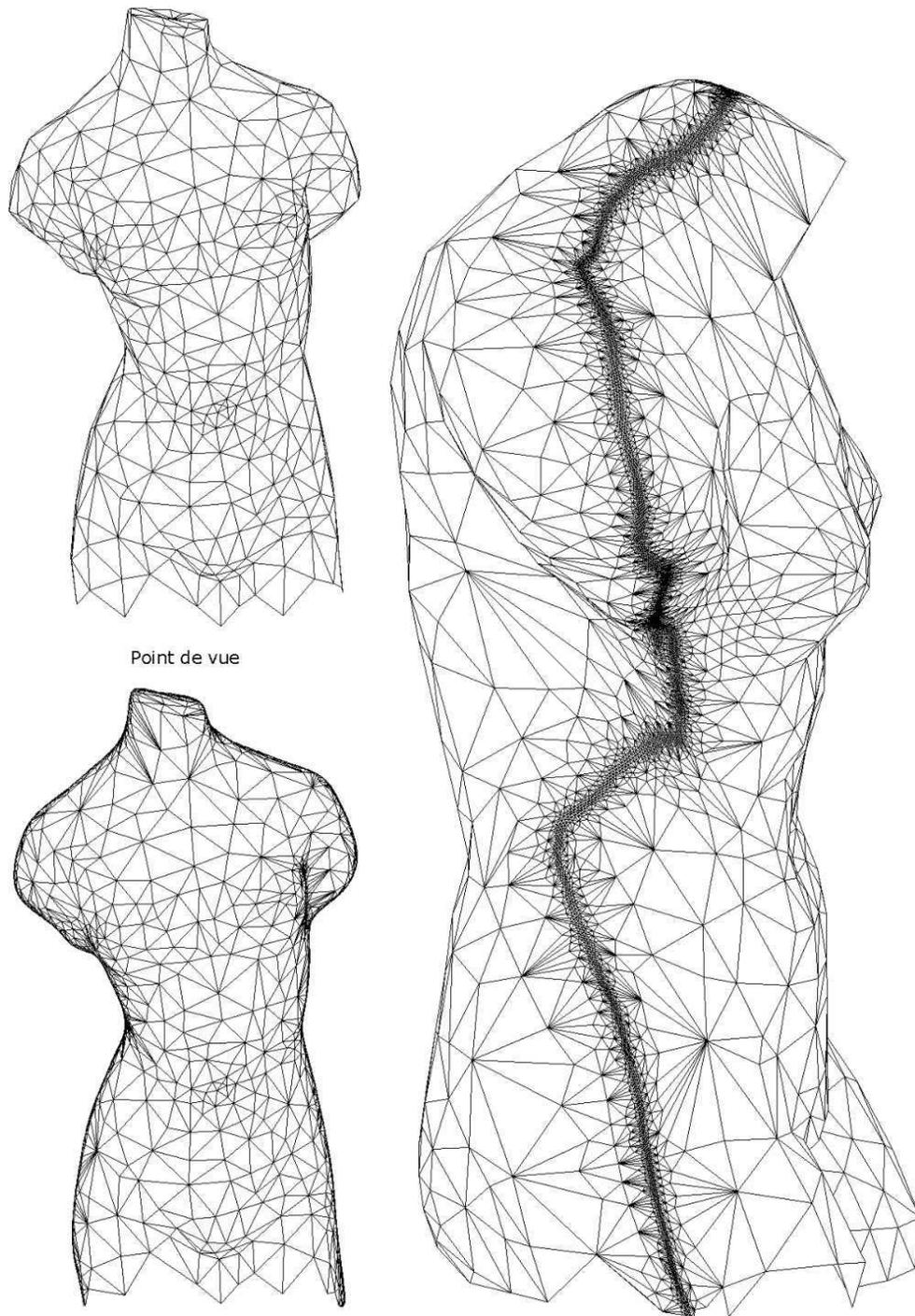


FIG. 4.35 : Maillage *venus*. 6 itérations de subdivision sont successivement appliquées sur la région composée des faces adjacentes aux arêtes de la silhouette. A gauche : l'aspect polygonal de la silhouette est supprimé par le raffinement localisé, et l'opérateur choisi autorise une cohabitation harmonieuse des niveaux de résolution sur le maillage. La complexité géométrique augmente de 1396 à 21544 triangles en 3,14 s. La détection de la silhouette est effectuée après chaque subdivision, les faces candidates à la silhouette étant limitées à la zone de silhouette de la précédente détection afin d'accélérer les temps de calcul.

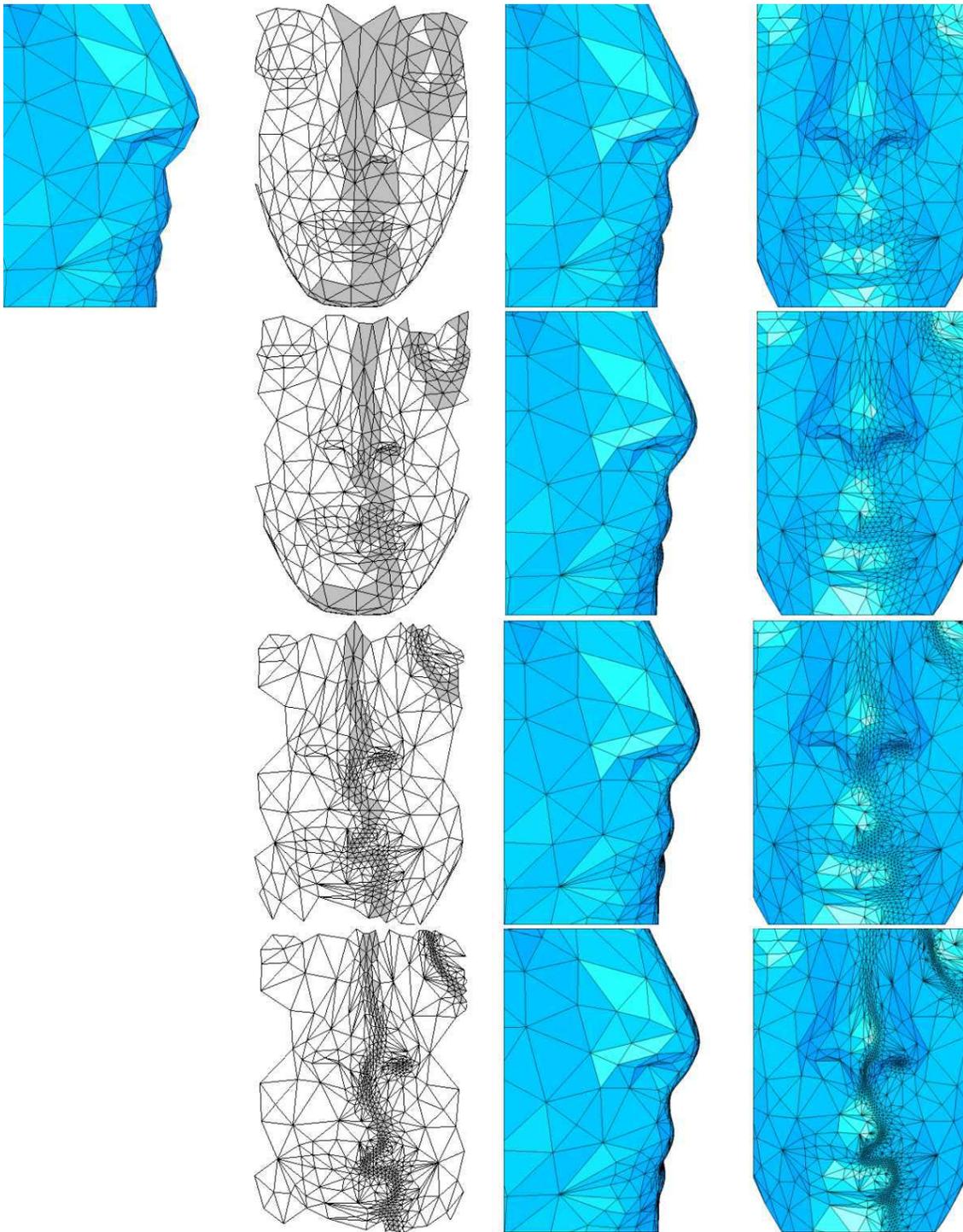


FIG. 4.36 : Maillage *nefertiti*. 4 itérations de raffinement sont itérativement appliquées sur la silhouette du modèle. Colonne 2 : la bande formée par les triangles adjacents aux sommets des arêtes de la silhouette se rétrécit à chaque raffinement par subdivision. Colonne 3 : raffinement localisé résultant de la détection précédente de la silhouette. Colonne 4 : la densité géométrique augmente progressivement sur la région d'intérêt.

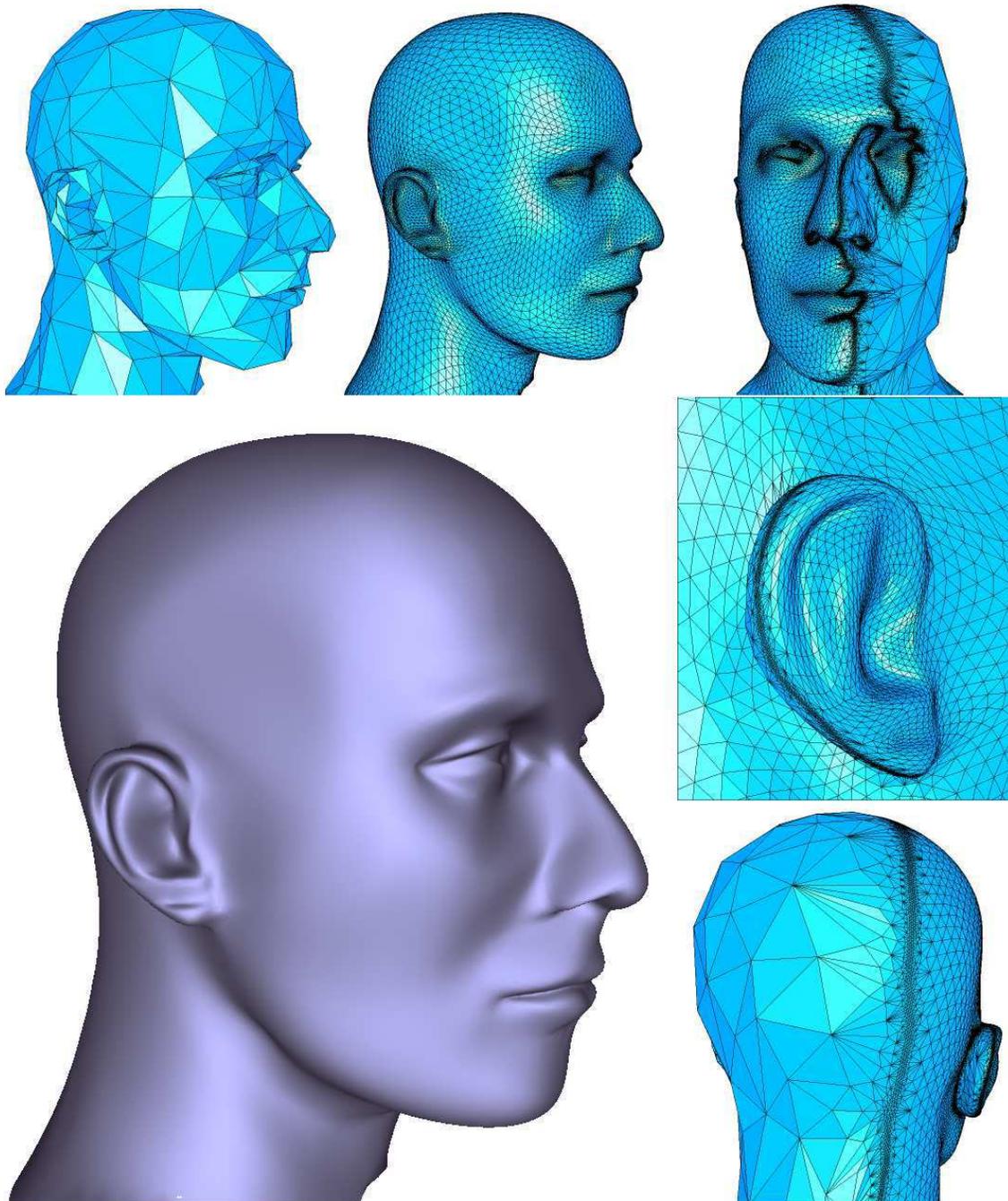


FIG. 4.37 : Maillage *mannequin*. En haut et de gauche à droite : le maillage original, sa version raffinée sur la silhouette et les régions orientées face à la caméra pour le point de vue courant et le maillage raffiné tracé sous un point de vue différent afin de mettre en évidence la densité obtenue sur la silhouette. Le nombre de faces varie de 838 à 43294 faces en 6,4 s lorsque trois itérations de subdivision sont appliquées sur les régions face à la caméra. En bas à gauche : le résultat final après lissage de Gouraud : l'aspect polygonal des silhouettes a totalement disparu, il en est de même en cas de zoom sur une silhouette puisque le raffinement est obtenu jusqu'à l'obtention d'une longueur d'arête projetée inférieure à la résolution de l'écran. En moyenne trois itérations additionnelles sont appliquées sur la silhouette pour cet exemple, totalisant ainsi 6 itérations de subdivision très localisées. A titre de comparaison, 6 itérations de subdivision globale équivalentes aboutissent à un maillage comportant environ 600 000 faces, pour une qualité d'image équivalente. A droite : le détail d'une silhouette localisée sur le lobe de l'oreille et le maillage raffiné vu de dos.

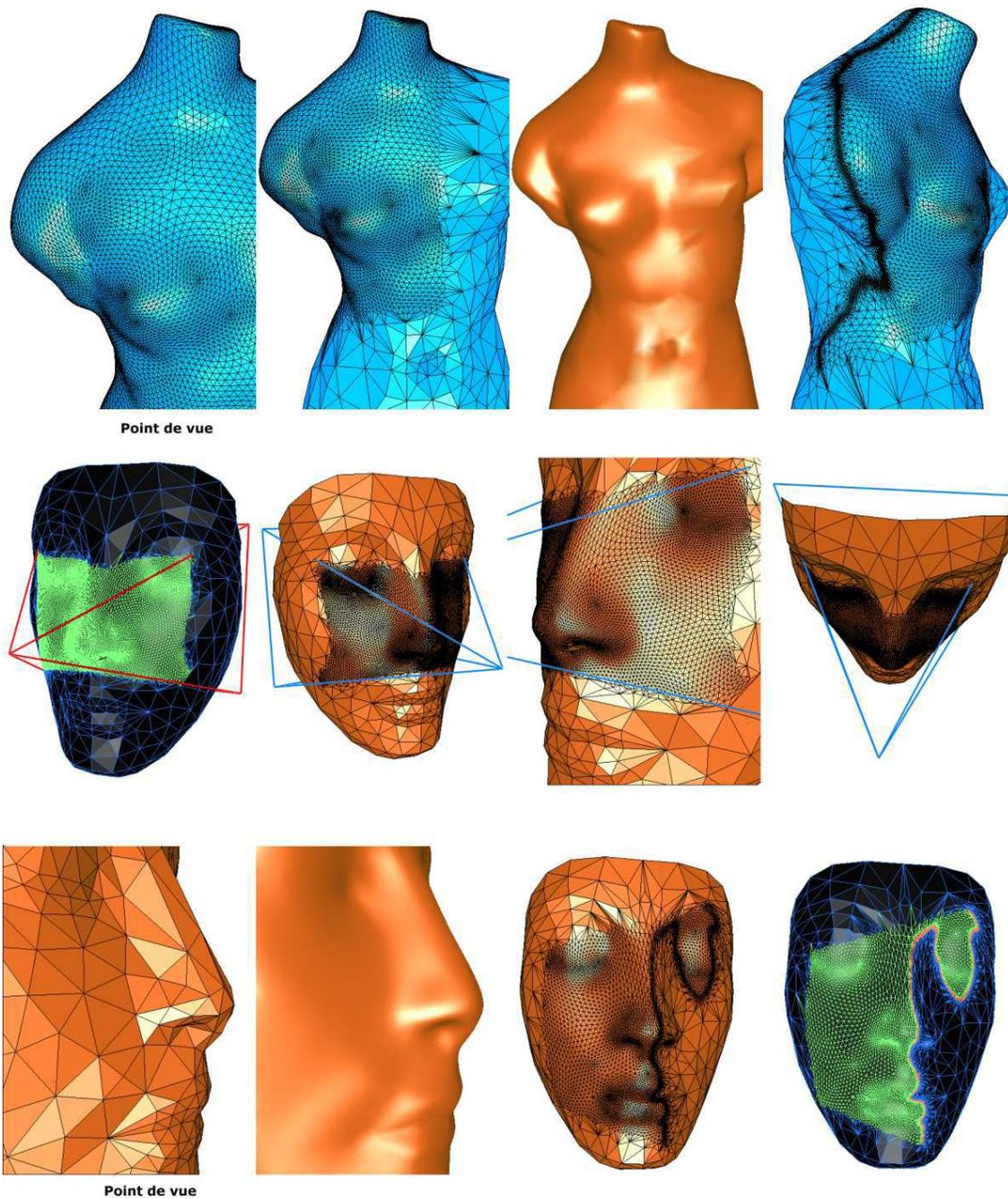


FIG. 4.38 : Ligne 1 : le maillage *venus* est raffiné sur la silhouette et les régions orientées face à la caméra. Le sous-ensemble d'intérêt correspond aux faces incluses dans la pyramide de vision définie par les coins de la fenêtre de visualisation et la caméra virtuelle. Ligne 2 : la région d'intérêt est explicitement définie par l'utilisateur sous la forme d'une pyramide. Notez que tout autre mode de sélection pourrait être utilisé. Ligne 3 : le maillage *nefertiti* avant puis après raffinement adaptatif, appliqué sur la région d'intérêt (mise en évidence sur la dernière colonne). La forme de la zone d'intérêt est liée à la pyramide de vision déduite du mode de projection perspective utilisé par le point de vue (première colonne).

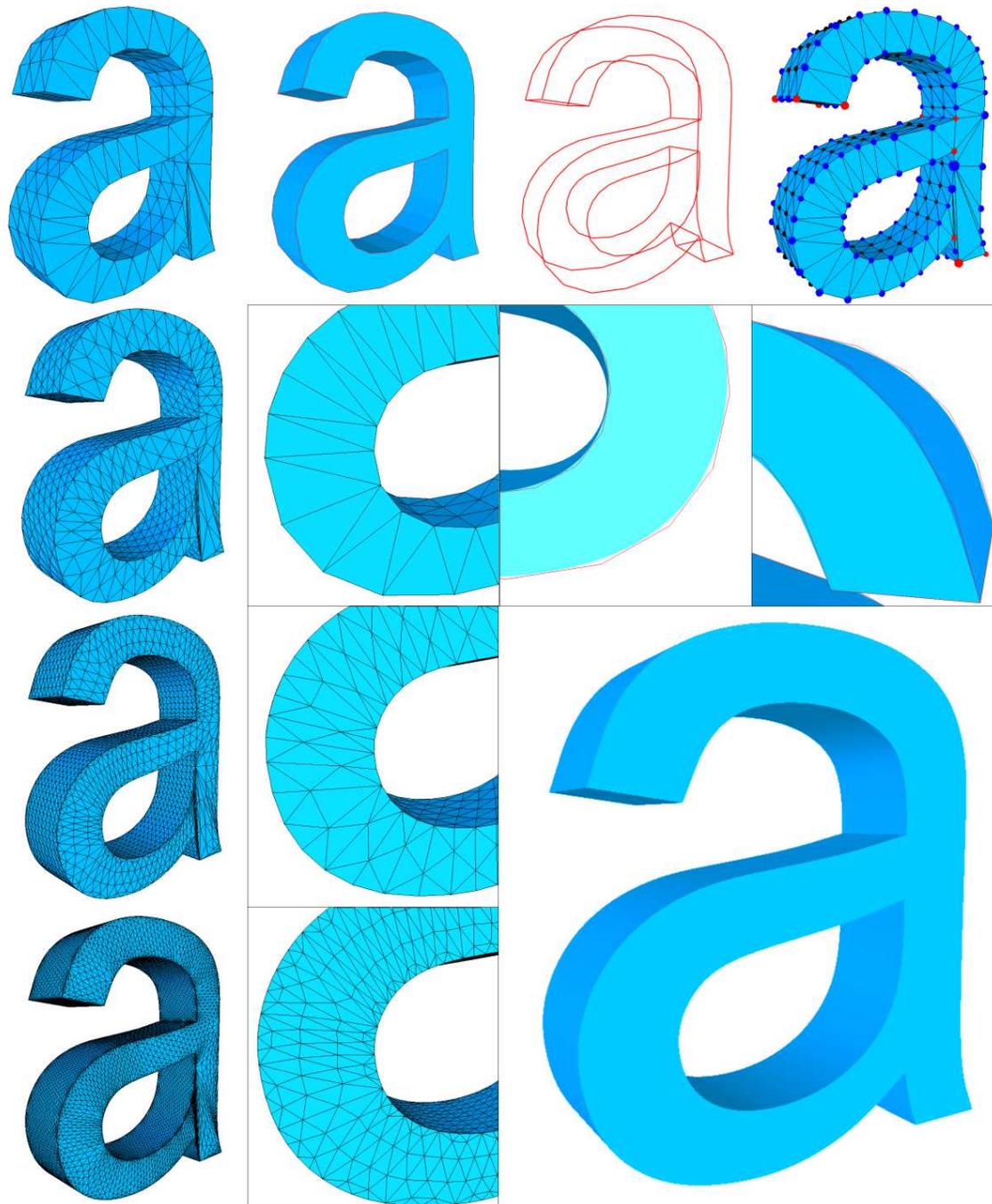


FIG. 4.39 : Le maillage initial comporte des arêtes vives régulières, des coins et des régions lisses. L'algorithme adaptatif autorise la reconstruction de la surface et des singularités géométriques. Ligne 1, et respectivement de gauche à droite : maillage initial, détection des arêtes vives, aspect polygonal des arêtes, et classification des sommets suivant leur nombre d'arêtes vives adjacentes (0 : sommet lisse, ≥ 3 : coin, 2 : sommet situé sur une arête vive régulière). Colonne 1, et respectivement verticalement : trois itérations successives de raffinement adaptatif. Colonne 2 : détail sur l'approximation de l'arête vive de la boucle du caractère *a* par filtrage anisotrope (l'aspect polygonal de l'arête est supprimé dans une direction et sa densité augmente plus rapidement que les régions lisses). Ligne 2, colonnes 3 et 4 : superposition de l'arête vive initiale et de la surface après approximation. En bas à droite : la surface finale après reconstruction.

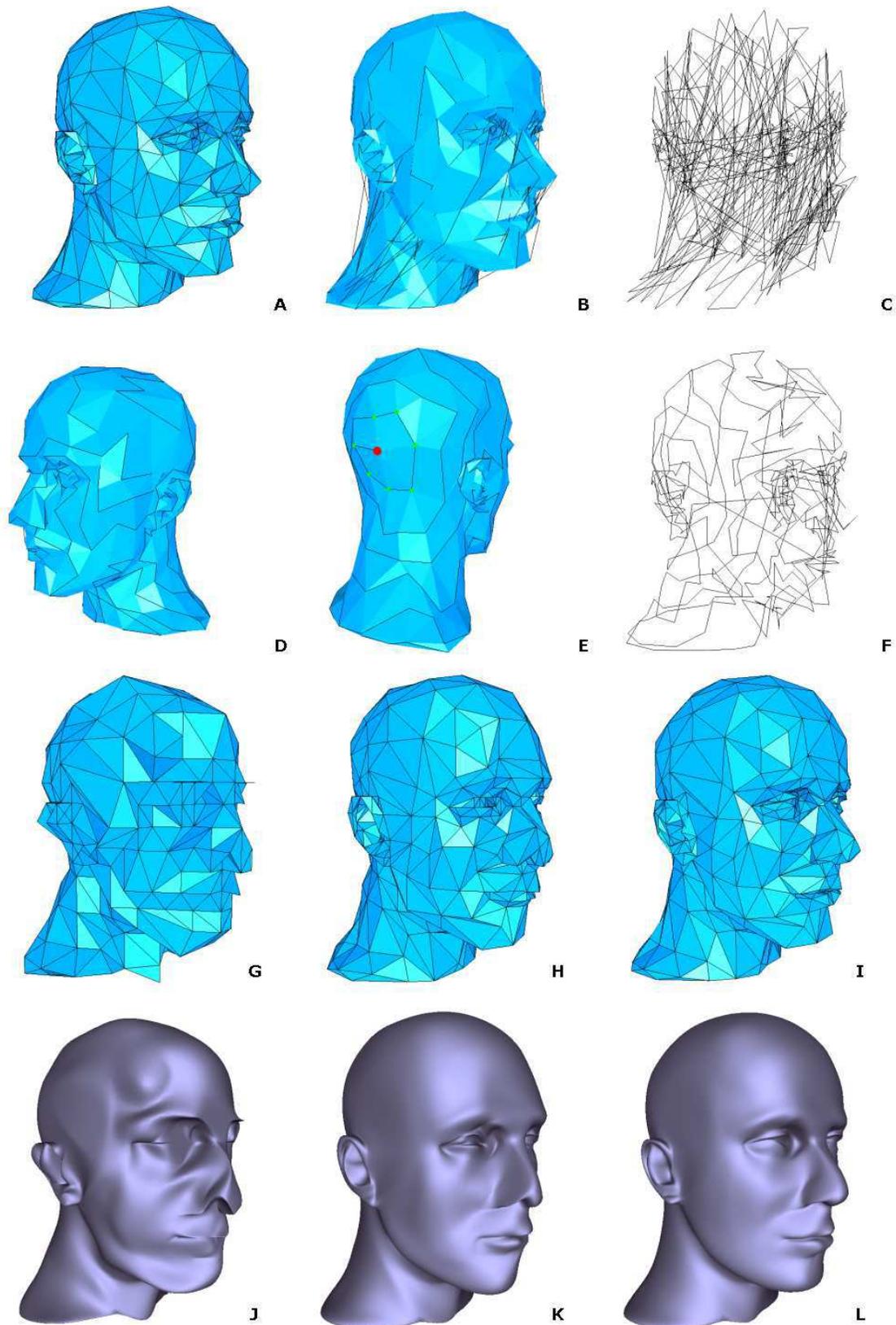


FIG. 4.40 : [A] Le maillage original (854 faces). [B,C] L'ordre initial sur les sommets [D,E,F] L'ordre sur les sommets après codage de la connectivité. Le pivot germe est représenté sur l'image E. On note à droite la cohérence géométrique obtenue. [G,H,I] Les niveaux de base après quantification de la géométrie avec respectivement 2, 3 et 4 bits. [J,K,L] Les résultats obtenus après trois itérations de raffinement global.

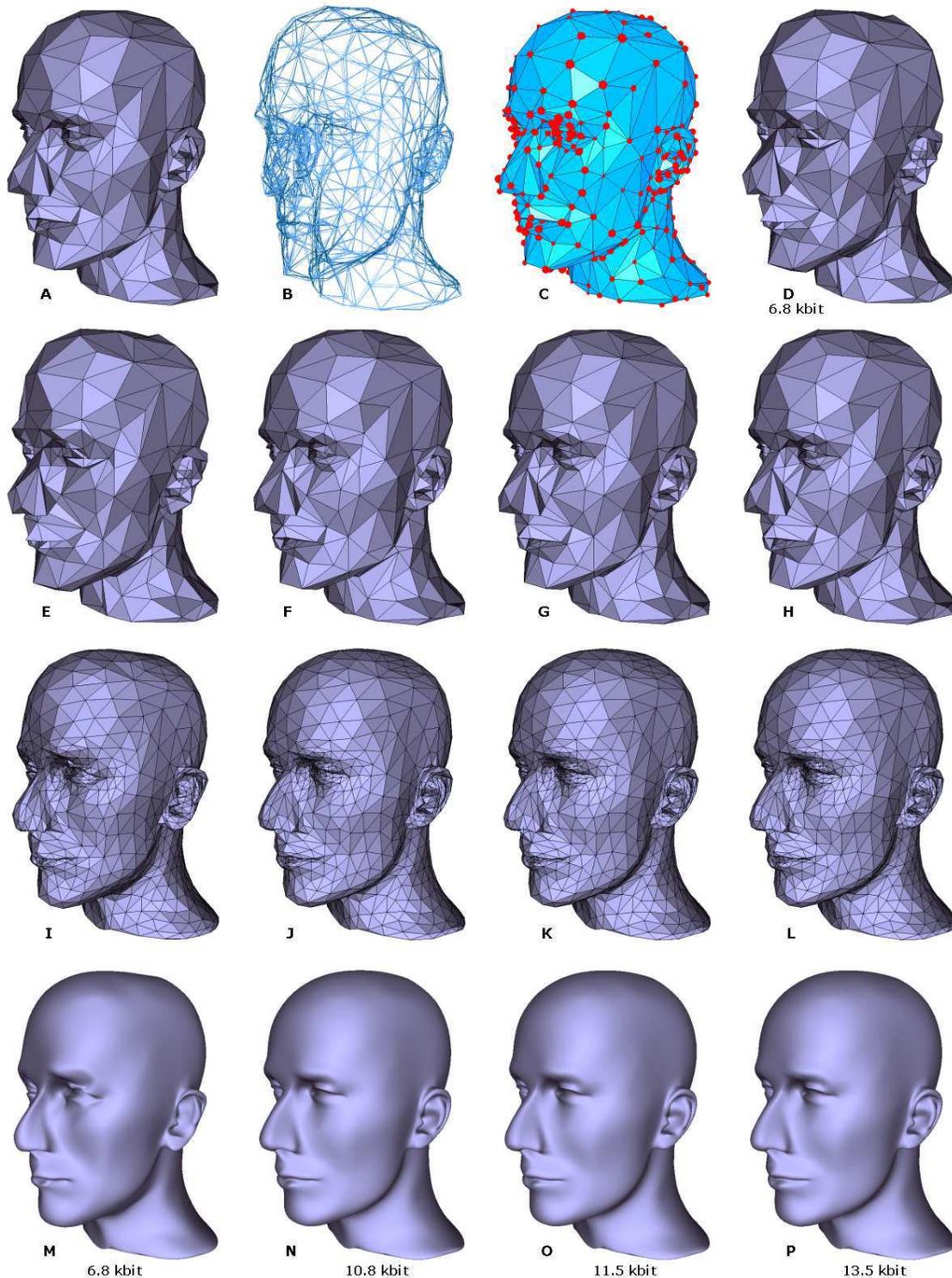


FIG. 4.41 : [A] Le maillage *mannequin* avant codage. [B] La géométrie du maillage a été progressivement codée avec un niveau de quantification initialisé à 3 bits (le niveau de base). Les raffinements par plans de bits sont superposés en mode maille. [C] Les sphères d'erreur correspondent à la quantification du niveau de base. Le niveau de base après décodage : 6.8 kbit au total, soit un ratio de 41.6 par rapport au fichier vrmf correspondant. [D] Le niveau de base [E,F,G,H] Raffinement progressif de la géométrie par intégration des ordres de grandeur et des plans de bits : respectivement 6.8, 10.8, 11.5 et 13.5 kbit. [I,J,K,L] après une itération de raffinement global. [M,N,O,P] après trois itérations de reconstruction globale et lissage de Gouraud activé.

arêtes vives et les coins) permettent de préserver les singularités au cours de la subdivision. Nous avons notamment montré qu'une arête vive régulière est lissée uniquement dans sa propre direction afin d'approximer une courbe spline tridimensionnelle, préservant ainsi les arêtes tout en restaurant leur courbure. La méthode s'avère ainsi bien adaptée à la reconstruction de surfaces courbes définies par morceaux.

Grâce à cette approche, la consigne : *déporter l'intelligence sur les terminaux* s'avère pleinement respectée. L'adaptation au terminal est directement applicable dans la mesure où le niveau de raffinement est paramétrable par un nombre d'itérations de subdivision sur les régions d'intérêt, par un critère d'erreur comme la longueur d'arête projetée ou encore plus simplement par une complexité exprimée en nombre de triangles. On a pu également constater que le format échelonnable autorise l'application du principe de l'amplification géométrique en cours de transmission puisque le raffinement s'avère parfaitement réversible. Enfin, la continuité visuelle des transitions est assurée lors des raffinements : les variations brusques de géométrie et d'ombrage sont notamment évitées par une interpolation conjointe géométrie / normales.

La technique a fait l'objet d'un dépôt de brevet [AL00] et une soumission est actuellement en préparation.

Perspectives

Les perspectives de travail concernent la prise en compte des *dards* lors de la reconstruction, ainsi que l'étude de la continuité au voisinage des singularités géométriques [Sch96] (voir les sommets extraordinaires positionnés sur les arêtes vives figure 4.42). La trisection des arêtes préconisée à l'origine sur les bords par Kobbelt [Kob99] est également à tester dans le contexte adaptatif sur les arêtes vives et les dards. Actuellement les singularités géométriques sont repérées par détection automatique des plis adjacents aux sommets, cette opération nécessitant un seuil délicat à paramétrer. Un étiquetage des sommets ou des arêtes lors d'une phase d'optimisation avant le codage serait bien plus judicieux [HDD⁺94] [Hop94]. Si l'on poursuit ce raisonnement plus avant, la piste privilégiée concerne un algorithme d'optimisation globale agissant sur :

1. Les positions des sommets du maillage de contrôle : la notion de distance étant définie entre le modèle à approximer et le maillage après reconstruction adaptative via subdivision du maillage de contrôle,
2. La connectivité du maillage : permutation ou fusion d'arête,
3. L'étiquetage des sommets en 4 catégories : lisse, dard, pli, ou coin,
4. La *dureté* des arêtes : dans la technique de Hoppe et al [HDD⁺94], les arêtes sont classées en deux catégories : lisses ou vives. Or DeRose et al [DKT98] ont récemment défini très astucieusement la notion d'arête semi-vive par le biais d'une caractérisation entière sur les arêtes du maillage de contrôle, le tout en conservant un comportement binaire de l'algorithme de subdivision (soit une arête est vive et filtrée de manière anisotrope, soit celle-ci est lissée dans toutes les directions). Chaque arête du maillage de contrôle se voit affecter un nombre entier exprimé en nombre d'itérations avant lissage. De cette manière une arête caractérisée par la dureté ∞ ne sera jamais lissée tandis qu'une arête caractérisée 2 sera lissée à la troisième itération de subdivision. La technique permet ainsi d'obtenir des arêtes perçues comme vives de loin, mais plus lisses de près. On rejoint ainsi la notion physique d'échelle illustrée en figure 4.43. La notion de ligne de style pourrait également être définie de cette manière, et cette caractérisation entière se présente sous un jour très séduisant pour un codage entropique puisque la distribution de l'information de dureté des arêtes s'avère non uniforme.

Encoder le maillage de contrôle résultant, nécessiterait alors la prise en compte du type des sommets [Sch96] et de la dureté des arêtes au sens de DeRose et al [DKT98]. Dans les modes de représentation échelonnable traditionnels, les raffinements contiennent essentiellement une information de même nature : plans de bits, coefficients d'ondelettes, etc... Dans le cas évoqué en perspective, l'échelonnabilité résiderait en une

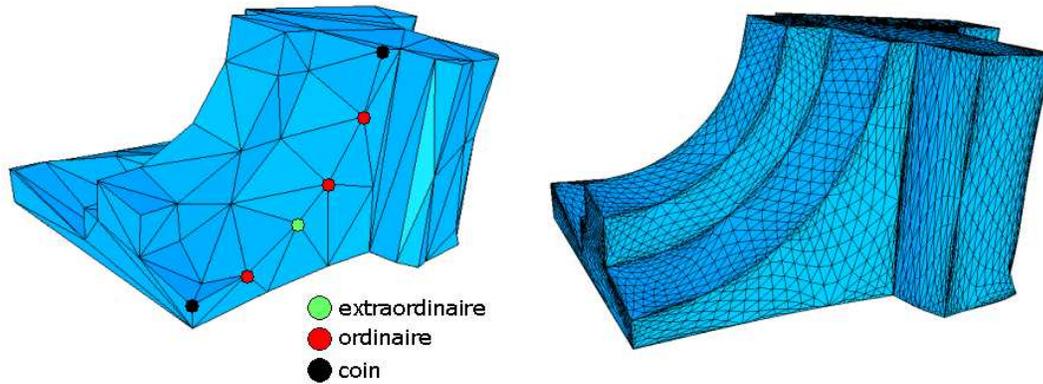


FIG. 4.42 : Les sommets sont classés par type suivant le nombre d'arêtes vives adjacentes (lisse, dard, pli, ou coin) puis la méthode de reconstruction garantit une C_2 continuité sur les sommets ordinaires de l'arête vive, et une C_1 continuité ailleurs. Un sommet ordinaire possède deux arêtes vives adjacentes, et exactement deux arêtes lisses de part et d'autre du pli formé par l'arête (voir [HDD⁺94]).

succession de raffinements mixant plans de bits pour la géométrie, étiquetage du type des sommets et caractérisation de la dureté des arêtes. Enfin, il pourrait être également judicieux de contrôler la déviation géométrique au cours de la phase d'optimisation du maillage.

Le chapitre suivant présente une conclusion générale et les perspectives d'évolution des travaux présentés dans ce document.

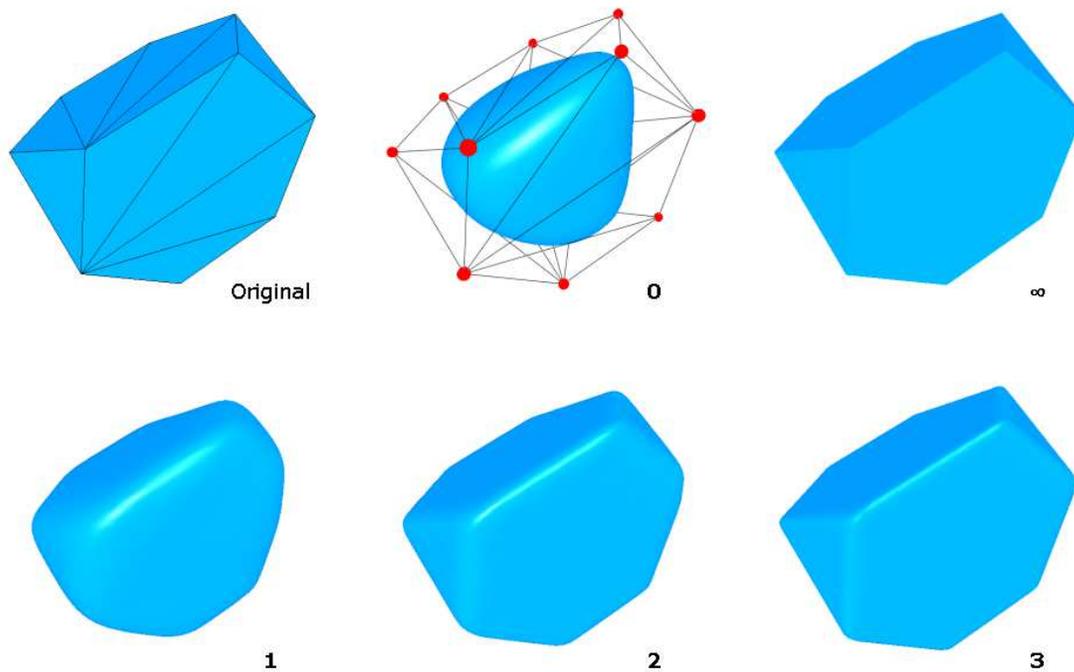


FIG. 4.43 : DeRose et al [DKT98] définissent la notion d'arête semi-vive par le biais d'une caractérisation entière. Une arête est caractérisée une fois pour toutes sur le maillage de contrôle par sa *dureté* exprimée en nombre d'itérations κ avant filtrage. Sur cet exemple, les arêtes vives subissent globalement une subdivision barycentrique sur les κ premières itérations tandis que le maillage est raffiné par subdivision barycentrique des faces et permutation d'arêtes ensuite. Les auteurs [DKT98] proposent également la définition d'un κ scalaire par interpolation linéaire de la géométrie reconstruite entre les deux itérations encadrant cette valeur. L'optimisation avant codage pourrait ainsi concerner le κ propre à chaque arête du maillage de base, quelques valeurs s'avérant suffisantes pour reconstruire fidèlement la géométrie (6 itérations de subdivision correspondant à une multiplication du nombre de faces par 729 dans le cas présent), typ. $\{\infty, 0, 0.5, 1, 2, 3, 4, 5\}$.

Chapitre 5

Conclusion et perspectives

Cette section propose un récapitulatif des travaux, la liste des travaux en perspective, et une liste non exhaustive des prochains challenges à relever.

5.1 Récapitulatif

Ce mémoire a décrit plusieurs procédés algorithmiques participant à la mise en œuvre d'un moteur de réalité virtuelle distribuée en réseau. Le problème de la représentation des données liées aux scènes tridimensionnelles nous a rapidement amenés à travailler essentiellement sur les maillages triangulaires, puisqu'ils constituent aujourd'hui la structure de données reflétant le meilleur compromis entre le matériel, la modélisation et le traitement. Le problème se posait dans les termes suivants : comment réduire le volume de données occupé par les maillages triangulaires ? Le choix s'est ensuite naturellement orienté vers les approches géométriques, plutôt que les approches *tout image*. En effet, les données 3D autorisent une visualisation interactive très souple après transmission au terminal.

Récapitulons maintenant les algorithmes de l'étude :

Approximation basée volume

L'originalité de la méthode d'approximation présentée repose sur l'exploitation d'une métrique basée sur le volume entre le modèle et le maillage original. La pertinence de cette métrique s'est rapidement démontrée sur les maillages tridimensionnels, notamment sur les singularités géométriques comme les coins ou les arêtes vives. Elle évite notamment le paramétrage et la détection des singularités nécessaires à l'optimisation L^2 . De plus, elle peut être utilisée seule (comme un post-traitement) ou conjointement avec toute méthode de simplification de maillage. Nous avons toutefois développé une méthode de simplification adaptée au processus d'optimisation.

Simplification par fusion d'arêtes

Afin d'obtenir une méthode robuste, rapide et facile à paramétrer, la simplification s'effectue au moyen de l'opérateur de fusion d'arêtes, et d'une double queue de priorité basée sur deux critères organisés en une hiérarchie. Afin d'adapter cette méthode à l'algorithme d'optimisation précédemment décrit, et notamment de préserver la compacité des triangles, le critère prépondérant est relié à un terme de courbure, tandis que le second concerne un ordre de grandeur géométrique. La simplification est ensuite couplée à un processus de repositionnement des sommets après fusion, basé sur un dénombrement des arêtes vives incidentes aux sommets. Ce dernier processus accélère la convergence et améliore singulièrement la qualité de l'approximation puisque

la pertinence des solutions initiales est renforcée. On réserve toutefois le rôle de cet algorithme de simplification à la suppression de la redondance sur les maillages plutôt qu'à une réduction drastique (jusqu'au triangle unique par exemple). L'intérêt de la méthode est liée à la facilité de paramétrage (on *privilégie* un critère par rapport à l'autre, ce qui est plus intuitif qu'un réglage et même accessible à un utilisateur novice), et à la vitesse d'exécution obtenue lorsque des structures de données optimisées sont employées.

Codage par conquête sur les arêtes

Il restait de nombreux degrés de liberté aux méthodes basées sur la conquête et l'exploitation des valences. La modification de la stratégie et l'ajout de codes spécifiques ont abouti à un gain sensible en coût de codage de la connectivité, en volume mémoire lors du décodage, et en robustesse par l'obtention d'une indépendance connectivité / positions. Enfin, si les données géométriques (les positions) sont notoirement difficiles à compresser, la méthode de prédiction multiple ainsi que la gestion optimisée des accidents donnent finalement lieu à une représentation plus compacte que les approches précédentes. De plus, la représentation des positions est échelonnable, la scalabilité étant obtenue par une quantification successive et la représentation par plans de bits de tailles variables, où l'innovation est rangée par ordre décroissant. Ce dernier point répond notamment au cahier des charges d'un codeur échelonnable efficace.

Codage des textures

Le codage des textures est adressé par l'intégration d'une technique développée au sein du laboratoire. L'hybridation éléments finis / DCT sur une hiérarchie de maillages triangulaires s'avère bien adaptée à la transmission progressive et à la visualisation adaptative.

Visualisation par raffinement dépendant du point de vue

Du côté du terminal, la piste choisie entre dans le cadre des surfaces de subdivision, couplée à une détection optimisée des régions d'intérêt. Plus précisément, les silhouettes, les tâches spéculaires et plus simplement les régions situées dans le champ de vision sont concernées par l'interpolation des maillages.

Nous disposons maintenant de quatre outils efficaces pour la suppression de la redondance, la compression échelonnable et la visualisation. La section 5.2 décrit les travaux en perspective, engagés dans la course au *moteur idéal* de réalité virtuelle distribuée en réseau.

5.2 Perspectives

La liste suivante consigne les perspectives de recherche envisagées :

- Simplification et approximation : une récente publication de Borouchaki [Bor99] démontre la possibilité de concilier le *contrôle* de la déviation géométrique, de la régularité et de la qualité en forme (compacité) des éléments avec une vitesse d'exécution très satisfaisante. Il semble aujourd'hui opportun d'appliquer notre technique de minimisation de volume après une telle phase de simplification, d'autant plus que la phase d'approximation requiert le contrôle de la compacité des triangles. La prise en compte des trous et bords est également envisagée,
- Codage de maillages triangulaires : la connectivité est codée sans perte, tandis que l'échelonnabilité est obtenue sur la géométrie. La technique s'avère ainsi particulièrement bien adaptée au codage du maillage de base d'un modèle semi-régulier utilisé par les techniques d'ondelettes [GVSS00] [KG00] [KSS00]. Il n'apparaît pas nécessaire de chercher à réduire davantage le coût de codage de la connectivité car la géométrie occupe en moyenne un volume 8 fois supérieur pour obtenir un résultat sans perte visuelle. Le défi consiste donc à mesurer la véritable entropie des positions d'un maillage 3D, et à en déduire

un schéma de prédiction et de codage adapté. Enfin, la robustesse du codeur de connectivité pourrait être sensiblement amélioré afin de lever la contrainte de multiplicité des arêtes (le maillage doit être *2-manifold*, et même d'accepter en entrée un maillage sans aucune organisation particulière,

- Codage et transmission progressive des textures : le choix d'une commutation pixel / maillage pour la représentation des images s'avère pertinent pour plusieurs raisons :
 - le maillage autorise un *échantillonnage irrégulier* en concentrant les nœuds sur les régions informatives et en alignant les arêtes sur les contours de l'image,
 - une *hiérarchie* peut être aisément adaptée par subdivision afin de construire un format échelonné,
 - le maillage triangulaire constitue un matériau satisfaisant pour l'élaboration d'un *interpolateur* rapide et robuste. L'interpolateur linéaire utilisé garantit notamment une surface C^0 continue,
 - une *transformée DCT* peut être appliquée sur les triangles après recalage afin de coder les régions texturées.

En pratique, le bénéfice de l'échantillonnage irrégulier apporté par le maillage est pleinement obtenu seulement en présence d'un maillage de base irrégulier. Or, la structure de ce dernier (connectivité / géométrie) s'avère suffisamment coûteuse pour confirmer le choix d'un maillage de base régulier, dont la structure n'est plus à coder (hormis la disposition et la résolution). Seul le raffinement adaptatif dans la hiérarchie fournit ensuite une allocation variable du débit, avec une limitation toutefois liée à la contrainte de régularité imposée à la subdivision 1 vers 4 toujours pour des raisons de coût (seule l'information différentielle en luminance / chrominance étant codée). Le débit ainsi économisé est ensuite affecté au codage des raffinements dans la hiérarchie.

Si l'on conserve la contrainte de régularité, quelques degrés de liberté subsistent en ce qui concerne :

- l'opérateur de permutation d'arête : actuellement seuls deux états sont codés sur une arête du dernier niveau de la hiérarchie. Un troisième état permettrait l'application d'un interpolateur bilinéaire sur le quadrilatère formé par les triangles adjacents à l'arête, cette dernière n'intervenant pas dans le calcul,
- le choix d'un interpolateur : un interpolateur linéaire est actuellement appliqué sur les éléments triangulaires, tandis qu'une version bilinéaire pourrait aisément s'appliquer sur le même maillage transformé en un maillage quadrilatère par subdivision barycentrique des faces. Notons toutefois qu'un procédé d'interpolation ou d'approximation de type box-splines, Loop ou Butterfly est également directement applicable sur le maillage triangulaire (voir [Sch99]).
- Visualisation : l'efficacité des techniques basées sur la subdivision de surfaces n'est plus à démontrer pour le raffinement des maillages. Nous avons pu constater qu'un raffinement très localisé sur les silhouettes et les régions d'intérêt autorise un embellissement sensible des images tout en respectant les capacités graphiques d'un terminal. De plus, les singularités géométriques peuvent être conservées, voir même reconstruites dans le cas des arêtes vives. La piste privilégiée consiste à affiner la définition des singularités en ajoutant la notion d'arête semi-vive proposée par DeRose et al [DKT98], et à entrelacer la reconstruction adaptative lors du codage avec une optimisation des positions et la caractérisation de la dureté des arêtes. Enfin, si le raffinement par subdivision de surfaces constitue une alternative intéressante aux surfaces paramétriques ou implicites, le rendement de la technique n'est toutefois pas optimal en terme de volume mémoire et de calcul. En effet, si nous partons du principe que seul le maillage de contrôle est nécessaire au rendu d'une image, et que la subdivision est appliquée pour chaque point de vue au cours de la navigation, il faudrait idéalement ne générer aucune information supplémentaire en

mémoire, puis minimiser le nombre de calculs nécessaire au raffinement. Or, dans sa version actuelle, le volume de données décrivant le maillage et son adjacence augmente sensiblement en mémoire soit en fonction d'un nombre d'itérations de raffinement, soit en fonction d'un critère de longueur d'arête projetée sur la silhouette. Ce dernier critère semble naturel si la subdivision est stoppée lorsque la longueur d'arête projetée est inférieure à la résolution du périphérique de sortie. L'interpolation linéaire calculée sur les triangles est ensuite employée lors du rendu des triangles. Cette dernière étape s'avère non indispensable sur les silhouettes puisque la densité des sommets projetés suffit à *remplir* l'espace de l'écran. Le schéma exploite donc successivement deux techniques : une approximation par subdivision sur les données tridimensionnelles (le maillage), puis une interpolation linéaire sur les données bidimensionnelles après projection (les couleurs calculées aux sommets). Le pixel constituant la finalité graphique d'un modèle 3D, il s'avère intéressant de chercher le moyen de minimiser les calculs sur les données 3D afin d'essayer de raffiner la géométrie dans l'espace projeté d'une part, et de prendre explicitement en compte la résolution de la sortie graphique d'autre part. Ce dernier point constitue toutefois un véritable challenge, aussi nous nous autorisons à poursuivre ce raisonnement dans la section 5.3.

5.3 Les prochains challenges

Il reste de nombreux problèmes à résoudre avant d'aboutir à une représentation efficace des scènes 3D, la liste suivante relevant les principales fonctionnalités à développer :

- L'échelonnabilité générique : la diversité des contenus liés aux scènes en réalité virtuelle évolue à la même vitesse que l'imagination des infographistes, ces derniers étant plutôt prolifiques. Aujourd'hui, les scènes statiques requièrent le codage échelonnable de la connectivité, de la géométrie, des normales, des couleurs, des matériaux, des coordonnées de texture, des textures associées, etc... Les scènes dynamiques requièrent en sus le codage des animations, des interactions, et des collisions. Dans un futur proche, il sera nécessaire de coder des températures, des viscosités, des rugosités, des coefficients de transparence, et tous les ingrédients susceptibles de créer la sensation d'immersion. Pourtant, nous pensons avoir montré que chaque type de données nécessite un traitement spécifique, puisque la distorsion est étroitement liée à leur nature. Une question s'impose alors : est-il envisageable de trouver une méthode échelonnable générique ?
- La robustesse : la modélisation est souvent très découplée des contraintes de codage, ce qui aboutit parfois à des maillages comportant des irrégularités topologiques : plusieurs occurrences de faces ou de sommets exactement superposées, des maillages *non-manifold*, des arêtes flottantes, des auto-intersections, ou tout simplement un mode d'organisation désigné par l'anglicisme révélateur : les *polygon soups*,
- Une nouvelle structure de données : aujourd'hui le triangle règne en maître puisque c'est une structure proche du matériel, tous les procédés décrits dans ce mémoire étant fortement liés à cet élément. Les surfaces définies par des points de contrôle sont un exemple de représentation différente, tandis que les surfaces de subdivision se situent à la frontière entre les deux modes de représentation. Il n'est pas déraisonnable de penser que l'avenir nous réserve une nouvelle structure de données constituant peut-être une chance d'unifier la modélisation, la transmission et la simulation,
- Un rendu entrelacé avec le raffinement des surfaces : les techniques de subdivision de surfaces [ZS99] employées pour le rendu interactif procèdent habituellement en deux étapes distinctes : le raffinement dans le domaine 3D, puis le rendu basé triangles et une interpolation linéaire dans le plan de l'écran. Pourtant, et quelle que soit la complexité du modèle et le point de vue sur le terminal, le but consiste à afficher un nombre fini de pixels compris dans la fenêtre de visualisation. Si l'on considère l'hypothèse qu'une densité géométrique (mesurée après projection) supérieure à la résolution de l'écran constitue

une redondance d'information et de calcul, est il alors possible de guider le processus de raffinement des maillages directement depuis l'espace de l'écran ? Il pourrait par exemple s'agir d'autoriser la cohabitation d'une structure de données 3D avec sa version duale projetée. Ainsi le dégradé monotone et perspectivement incorrect (voir [Abr00]) de Gouraud pourrait être remplacé par un ombrage exact sur les régions courbes et/ou à fort gradient d'ombrage, tandis que ce dernier pourrait être conservé sur les régions restantes afin d'optimiser la vitesse de remplissage des éléments.

Une collaboration fructueuse et des problèmes similaires

Nous avons constaté dès le départ des familles de problèmes assez similaires en 1D, 2D ou 3D. Ces problèmes sont essentiellement inhérents à la limitation du débit, et concernent :

- l'*effet de bloc*, bien connu des codeurs d'images, qui trouve son équivalent dans l'*aspect polygonal* d'un objet 3D simplifié, et dans l'effet de pré-écho en audio (1D),
- les *saccades visuelles* constatées au cours de la visualisation d'une vidéo faiblement cadencée peuvent être comparées avec la notion de *singularité géométrique* sur un objet 3D,
- les *contours extérieurs* d'un objet vidéo ou les *contours* d'une image correspondent respectivement à la *silhouette*, et aux *arêtes vives* d'un objet 3D.

Enfin, les travaux consignés dans ce mémoire démontrent une collaboration efficace avec les personnes travaillant en 2D. La technique de codage d'images et de vidéos par maillages développée au sein du laboratoire constitue une source d'idées fructueuse pour la 3D (approximation basée volume, prédiction, réflexes de codage, raffinement adaptatif par subdivision). Cette technique s'avère par ailleurs directement adaptable au codage des textures. En retour, les techniques développées en 3D sont utilisables pour le codage de la connectivité, la simplification, et le raffinement dépendant d'une région d'intérêt d'un maillage irrégulier approximant une image.

Ce mémoire matérialise donc le premier jalon d'une maquette de service utilisant la conjonction de techniques touchant aux domaines des images synthétiques et naturelles. Il est maintenant permis de rêver à une future génération de services télécommunicants intégrant les concepts exposés...

Bibliographie

- [Abr00] Michael Abrash.
Programmation graphique c/c++ et assembleur.
Sybex, 2000.
Comprend l'intégralité du livre : Zen of Code Optimisation. Préfacé par John Carmak.
- [AdRL⁺99] Vadim Abadjev, Miguel del Rosario, Alexei Lebedev, Alexander Migdal, and Victor Pashaver.
Metastream.
VRML 99, 1999.
- [AH99] Jeong-Hwan Ahn and Yo-Sung Ho.
An efficient geometry compression method for 3D objects in the spherical coordinate system.
Proceedings of International Conference on Image Processing, 1999.
- [Alg95] Maria-Elena Algorri.
Génération et simplification de maillages pour la reconstruction de surfaces à partir de points non structurés.
PhD thesis, Ecole Nationale Supérieure des Télécommunications, 1995.
- [Ame99] Nina Amenta.
The crust algorithm for 3D surface reconstruction.
Proceedings of Fifteenth Annual ACM Symposium on Computational Geometry, june 1999.
- [AS96] Maria-Elena Algorri and Francis Schmitt.
Mesh simplification.
Computer Graphics Forum, 15(3), 1996.
- [BCPZ99] C. Bajaj, S. Cutchin, V. Pascucci, and G. Zhuang.
Error resilient streaming of compressed vrml.
submitted, 1999.
- [BDG⁺99] G. Barequet, C. A. Duncan, M.T. Goodrich, S. Kumar, and M. Pop.
Efficient perspective-accurate silhouette computation.
ACM SCG 99, 1999.
- [BE99] F. Benichou and G. Elber.
Output sensitive extraction of silhouette from polygonal geometry.
Pacific Graphics 99 conference proceedings, pages 60–69, 1999.
- [bif] BIFS, MPEG4 BInary Format for Scene description.
http://drogo.cselt.stet.it/leonardo/icjfiles/mpeg-4_si/5-BIFS_paper/5-BIFS_paper.htm.

- [Bor99] Houman Borouchaki.
Simplification des maillages de surfaces basée sur la distance de hausdorff.
C.R. Acad. Sci. Paris - Analyse numérique, 329(1) :641–646, 1999.
Académie des sciences. Editions scientifiques et médicales Elsevier SAS.
- [BPZ99] C. Bajaj, V. Pascucci, and G. Zhuang.
Single resolution compression of arbitrary triangular meshes with properties.
Proceedings of Data Compression Conference, pages 247–256, 1999.
- [BYG96] Reuven Bar-Yehuda and Craig Gotsman.
Time/space tradeoffs for polygon mesh rendering.
ACM Transactions on Graphics, 15(2) :141–152, 1996.
- [CC78] E. Catmull and J. Clark.
Recursively generated b-spline surfaces on arbitrary topological meshes.
Computer Aided Design, 10(6) :350–355, 1978.
- [Cha74] G. Chaikin.
An algorithm for high speed curve generation.
Computer Graphics and Image processing, 3 :346–349, 1974.
- [Che92] Xin Chen.
Modélisation Géométrique par Vision Artificielle.
PhD thesis, Ecole Nationale Supérieure des Télécommunications, 1992.
- [CKS98] Swen Campagna, Leif Kobbelt, and Hans-Peter Seidel.
Enhancing digital documents by including 3D-models.
SIGGRAPH 98 Conference Proceedings, 1998.
- [CLA97] Jin Soo Choi, Myoung Ho Lee, and Chieteuk Ahn.
Results of core experiment z3 : Geometry compression using PRVQ.
Technical Report MPEG97/m2475, ISO/IEC JTC1/SC29/WG11, july 1997.
- [COM98] Jonathan Cohen, Marc Olano, and Dinesh Manocha.
Appearance-preserving simplification.
SIGGRAPH 98 Conference Proceedings, 1998.
- [CPD⁺96] Andrew Certain, Jovan Popovic, Tony DeRose, Tom Duchamp, David Salesin, and Werner Stuetzle.
Interactive multiresolution surface viewing.
SIGGRAPH 96 Conference Proceedings, 1996.
- [CRS98] P. Cignoni, C. Rocchini, and R. Scopigno.
Metro : measuring error on simplified surfaces.
Computer Graphics Forum, 1998.
- [CVM⁺96] Jonathan Cohen, Amitabh Varshney, Dinesh Manocha, Greg Turk, Hans Weber, Pankaj Agarwal, Frederick Brooks, and William Wright.
Simplification envelopes.
SIGGRAPH 96 Conference Proceedings, pages 119–128, 1996.

- [DC90] D. Dooley and M. Cohen.
Automatic illustration of 3D geometric models : Lines.
Proceedings of the 1990 Symposium on Interactive 3D Graphics, pages 77–82, 1990.
- [Dee95] Michael Deering.
Geometry compression.
SIGGRAPH 95 Conference Proceedings, 1995.
- [Dee98] Michael Deering.
Hardware geometry compression : specification from java 3D.
Technical report, Sun Microsystems, 1998.
- [Dev98] Olivier Devillers.
Improved incremental randomized delaunay triangulation.
Proceedings of 14th Annual ACM Symposium on Computational Geometry, 1998.
- [DG99a] Olivier Devillers and Pierre-Marie Gandoïn.
Compression géométrique pour une transmission progressive.
AFIG 99 - Association Française d Informatique Graphique, nov 1999.
- [DG99b] Olivier Devillers and Pierre-Marie Gandoïn.
Compression géométrique pour une transmission progressive.
Technical Report 3766, INRIA Sophia Antipolis, France, sep 1999.
<http://www-sop.inria.fr/prisme/fiches/Compression/index.html.fr>.
- [DGSS99] Ingrid Daubechies, Igor Guskov, Peter Schröder, and Wim Sweldens.
Wavelets on irregular point sets.
Phil. Trans. R. Soc. Lond. A, 1999.
- [DKT98] T. DeRose, M. Kass, and T. Trunong.
Subdivision surfaces in character animation.
SIGGRAPH 98 Conference Proceedings, pages 85–94, 1998.
- [DRLB00] Laurent Demaret, Guillaume Robert, Nathalie Laurent, and Alexandre Buisson.
Scalable Image Coder mixing DCT and Triangular Meshes.
In *To appear in proceedings of ICIP 2000*, 2000.
- [DS78] D. Doo and M. Sabin.
Behaviour of recursive division surfaces near extraordinary points.
Computer Aided Design, 10(6) :356–360, 1978.
- [EDD⁺95] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle.
Multiresolution analysis of arbitrary meshes.
SIGGRAPH 95 Conference Proceedings, pages 173–182, 1995.
- [eFC00] Jean-Daniel Boissonnat et Frédéric Cazals.
Smooth Surface Reconstruction via Natural Neighbour Interpolation of Distance Functions.
ACM Symposium on Computational Geometry, 2000.
- [EH96] Matthias Eck and Hugues Hoppe.
Automatic reconstruction of b-spline surfaces of arbitrary topologicak type.
SIGGRAPH 96 Conference Proceedings, pages 325–334, 1996.

- [EM99] Carl Erikson and Dinesh Manocha.
Gaps : General and automatic polygonal simplification.
Proceedings of Symposium on Interactive 3D Graphics 99, 1999.
- [Eri96] Carl Erikson.
Polygonal simplification : An overview.
Technical Report TR96-016, UNC Chapel Hill, 1996.
- [ESV98] Jihad El-Sana and Amitabh Varshney.
Topology simplification for polygonal virtual environments.
IEEE Transactions on Visualization and Computer Graphics, 4(2), 1998.
- [ESV99] Jihad El-Sana and Amitabh Varshney.
Generalized view-dependent simplification.
Eurographics 99, 18(3), 1999.
- [Faq00] Frequently asked questions for the group comp.compression.research, 2000.
- [Fau93] Olivier Faugeras.
Computer Vision : a geometric approach.
MIT Press, 1993.
- [FB96] Pascal J. Frey and Houman Borouchaki.
Critères géométriques pour l'évaluation des triangulations de surfaces.
Technical report, INRIA, july 1996.
- [FvDFH96] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hugues.
Computer Graphics, principles and practice.
Addison Wesley, 2 edition, 1996.
- [GB97] Paul-Louis George and Houman Borouchaki.
Triangulation de Delaunay et maillage, applications aux éléments finis.
Hermes, 1997.
- [GBTS99] André Guézic, Frank Bossen, Gabriel Taubin, and Claudio Silva.
Efficient compression of non-manifold polygonal meshes.
IEEE Visualization 99, 1999.
- [GC90] G.Elber and E. Cohen.
Hidden curve removal for free form surfaces.
SIGGRAPH 90 Conference Proceedings, pages 95–104, 1990.
- [GGH⁺99] Xianfeng Gu, Steven J. Gortler, Hugues Hoppe, Leonard McMillan, Benedict J. Brown, and Abraham D. Stone.
Silhouette mapping.
Technical Report TR-1-99, Harvard University, 1999.
- [GH97] Michel Garland and Paul Heckbert.
Surface simplification using quadric error metrics.
SIGGRAPH 97 Conference Proceedings, pages 209–216, 1997.
- [GH98] Michel Garland and Paul Heckbert.
Simplifying surfaces with color and texture using quadric error metrics.
IEEE Visualization 98 Conference Proceedings, 1998.

- [GHJ⁺97] Tran S. Gieng, Bernd Hamann, Kenneth I. Joy, Gregory L. Schussman, and Issac J. Trotts.
Smooth hierarchical surface triangulation.
Proceedings of IEEE Visualization 97, pages 379–386, 1997.
- [Gio99] Patrick Gioia.
Reducing the number of wavelet coefficients by geometric partitioning.
Computational Geometry - Theory and Applications, 14 :25–48, 1999.
- [GK98] Stefan Gumhold and Reinhard Klein.
Compression of discrete multiresolution models.
SIGGRAPH 98 Conference Proceedings, 1998.
- [GS98] Stefan Gumhold and Wolfgang Strasser.
Real time compression of triangle mesh connectivity.
SIGGRAPH 98 Conference Proceedings, 1998.
- [GSS99] Igor Guskov, Wim Sweldens, and Peter Schröder.
Multiresolution signal processing for meshes.
SIGGRAPH 99 Conference Proceedings, 1999.
- [GTLH97] André Guéziec, Gabriel Taubin, Francis Lazarus, and William Horn.
Cutting and stitching : efficient conversion of a manifold polygonal surface to a manifold.
Technical Report 20935, IBM T.J. Watson Research Center, 1997.
- [GTLH99] André Guéziec, Gabriel Taubin, Francis Lazarus, and William Horn.
Simplicial maps for progressive transmission of polygonal surfaces.
VRML 1998, pages 25–31, 1999.
- [Gué97] André Guéziec.
Surface simplification inside a tolerance volume.
Technical Report RC 20440 (90191), IBM Watson Research Center, 1997.
- [Gus98] I. Guskov.
Multivariate subdivision schemes and divided differences.
Technical report, Pinceton University, 1998.
- [GVSS00] Igor Guskov, Kiril Vidimce, Wim Sweldens, and Peter Schröder.
Normal Meshes.
A paraître. <http://www.multires.caltech.edu/pubs/pubs.htm>, 2000.
- [HDD⁺92] Hoppe, DeRose, Duchamp, McDonald, and Stuetzle.
Surface reconstruction from unorganized points.
SIGGRAPH 92 Conference Proceedings, pages 71–76, 1992.
- [HDD⁺94] Hugues Hoppe, Tony DeRose, Tom Duchamp, Mark Halstead, Hubert Jin, John Mc Donald, Jean Schweitzer, and Werner Stuetzle.
Piecewise smooth surface reconstruction.
SIGGRAPH 94 Conference Proceedings, 1994.
- [HG97] Paul S. Heckbert and Michael Garland.
Survey of polygonal surface simplification algorithms.
Technical report, School of Computer Science, Carnegie Mellon University, 1997.

- [HKD93] Mark Halstead, Michael Kass, and Tony DeRose.
Efficient, fair interpolation using catmull-clark surfaces.
SIGGRAPH 93 Conference Proceedings, 1993.
- [Hop93] Hugues Hoppe.
Mesh optimization.
SIGGRAPH 93 Conference Proceedings, 1993.
- [Hop94] Hugues Hoppe.
Surface Reconstruction from Unorganized Points.
PhD thesis, University of Washington, 1994.
- [Hop96] Hugues Hoppe.
Progressive meshes.
SIGGRAPH 96 Conference Proceedings, pages 99–108, 1996.
- [Hop97] Hugues Hoppe.
View-dependant refinement of progressive meshes.
SIGGRAPH 97 Conference Proceedings, 1997.
- [Hop98] Hugues Hoppe.
Efficient implementation of progressive meshes.
Computers & Graphics, 1998.
- [Hop99a] Hugues Hoppe.
New quadric metric for simplifying meshes with appearance attributes.
VISUALIZATION 99 Conference Proceedings, 1999.
<http://research.microsoft.com/~hoppe/>.
- [Hop99b] Hugues Hoppe.
Optimization of mesh locality for transparent vertex caching.
SIGGRAPH 99 Conference Proceedings, 1999.
- [HRH⁺97] Paul S. Heckbert, Jarek Rossignac, Hugues Hoppe, William Schröder, Marc Soucy, and Amitabh Varshney.
Multiresolution surface modeling.
Technical report, SIGGRAPH 97, 1997.
Course notes.
- [Huf52] D.A. Huffman.
A method for the construction of minimum-redundancy codes.
Proc. Inst. Electr. Radio Eng., 30(6) :1098–1101, sept 1952.
- [IS99] Martin Isenburg and Jack Snoeyink.
Mesh collapse compression.
Symposium on Computational Geometry 1999, 1999.
- [JH99] Stephen Junkins and Allen Hux.
Subdividing reality. employing subdivision surfaces for real-time scalable 3d.
Technical report, Intel Architecture Labs, Intel corporation, 1999.

- [KBGT98] Mike Krus, Patrick Bourdot, Françoise Guisnel, and Guillaume Thibault.
Levels of detail and polygonal simplification.
Technical report, EDF/DER - IMA/TIEM/CAO, 1998.
<http://info.acm.org/crossroads/xrds3-4/levdet.html>.
- [KCS98] Leif Kobbelt, Swen Campagna, and Hans-Peter Seidel.
A general framework for mesh decimation.
SIGGRAPH 98 Conference Proceedings, 1998.
- [KCVS98] Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel.
Interactive multi-resolution modeling on arbitrary meshes.
SIGGRAPH 98 Conference Proceedings, 1998.
- [KG00] Andrei Khodakovsky and Igor Guskov.
Normal Mesh Compression.
A paraître. <http://www.multires.caltech.edu/pubs/pubs.htm>, 2000.
- [KGBT96] Mike Krus, Françoise Guisnel, Patrick Bourdot, and Guillaume Thibault.
Niveaux de détails et simplification polygonale.
AFIG 96 - Association Française d'Informatique Graphique, 1996.
- [Kle98] Reinhard Klein.
Multiresolution representations for surfaces meshes based on the vertex decimation method.
Computers & Graphics, 1998.
- [KLS96] R. Klein, G. Liebich, and W. Strasser.
Mesh reduction with error control.
ACM Visualization 96, 1996.
- [Kob98] Leif Kobbelt.
A subdivision scheme for smooth interpolation of quad-mesh data.
EUROGRAPHICS 98, 1998.
- [Kob99] Leif Kobbelt.
The simplest subdivision scheme for smoothing triangle meshes.
in preparation, 1999.
- [Koe84] J.J. Koenderink.
What does the occluding contour tell us about solid shape.
Perception, 13 :321–30, 1984.
- [KPhyJgC99] Yang-Soo Kim, Dong-Gyu Park, ho-young Jung, and Hwan-gue Cho.
An improved tin compression using delaunay triangulation.
PACIFIC GRAPHICS 99 Conference Proceeding, 1999.
- [KS98] Leif Kobbelt and Peter Schröder.
A multiresolution framework for variational subdivision.
ACM Transactions on graphics, 17(4) :209–237, 1998.
- [KSS00] Andrei Khodakovsky, Peter Schröder, and Wim Sweldens.
Progressive Geometry Compression.
A paraître. <http://www.multires.caltech.edu/pubs/pubs.htm>, 2000.
Démo disponible sur Internet : <http://www.multires.caltech.edu/software/pgc/>.

- [Lav99] Philippe Lavoie.
An introduction to NURBS.
Technical report, Ottawa university, 1999.
<http://yukon.genie.uottawa.ca/~lavoie/software/nurbs/>.
- [LC87] Lorensen and Cline.
Marching cube : A high resolution 3D surface construction algorithm.
Computer Graphics, 21(4),163-169, july 1987.
- [LDW97] Michael Lounsbery, Tony DeRose, and Joe Warren.
Multiresolution analysis for surfaces of arbitrary topological type.
ACM Transactions on Graphics, 16(1) :34–73, 1997.
- [Lec99] Patrick Lechat.
Représentation et codage de séquences vidéo par maillages 2D déformables.
PhD thesis, Université de Rennes 1, 1999.
- [Lev99] Adi Levin.
Interpolating nets of curves by smooth subdivision surfaces.
SIGGRAPH 99 Conference Proceedings, 1999.
- [LK98a] Jiankun Li and C.-C. Jay Kuo.
Progressive coding of 3D graphic models.
IEEE Computer Graphics, 86 :1052–1063, june 1998.
- [LK98b] Jiankun Li and C.C. Jay Kuo.
A dual graph approach to 3D triangular mesh compression.
ISO/IEC JTC1/SC29/WG11 - Coding of Moving Pictures and Audio - MPEG98/3195, 1998.
- [LLS98] Patrick Lechat, Nathalie Laurent, and Henri Sanson.
Représentation d’images et estimation de mouvement basées maillage. application à un codeur tout-maillage.
In *Journées COMpression et REprésentation des Signaux Audiovisuels (CORESA)*, pages 7–15, Lannion, 9-10 Juin 1998.
- [LLS99a] Patrick Lechat, Nathalie Laurent, and Henri Sanson.
Codage d’images basé maillage : approche conjointe interpolation affine et codage de textures.
In *CORESA*, 1999.
- [LLS99b] Patrick Lechat, Nathalie Laurent, and Henri Sanson.
Scalable image coding with fine granularity based on hierarchical mesh.
In *Proc. of VCIP’99*, San Jose - USA, Jan 1999.
- [Loo87] Charles Loop.
Smooth surface subdivision based on triangles.
PhD thesis, University of Utah, departement of Mathematics, 1987.
Master’s thesis.
- [LSL97] Patrick Lechat, Henri Sanson, and Lilian Labelle.
Image approximation by minimization of a geometric distance applied to a 3D finite elements based model.
Proc. of ICIP 97, Santa Barbara CA-USA, 2 :724–728, 1997.

- [LSS⁺98] Aaron W.F. Lee, Wim Sweldens, Peter Schröder, Lawrence Cowsar, and David Dobkin.
Maps : multiresolution adaptive parametrization of surfaces.
SIGGRAPH 98 Conference Proceedings, 1998.
- [LT97] Kok-Lim Low and Tiow-Seng Tan.
Model simplification using vertex clustering.
ACM Symposium on Interactive 3D Graphics, 1997.
- [LT98] Peter Lindstrom and Greg Turk.
Fast and memory efficient polygonal simplification.
IEEE Visualization Conference Proceedings, 1998.
- [Lue96] David Luebke.
Hierarchical structures for dynamical polygonal.
Technical Report TR96-006, University of North Carolina at Chapel Hill, 1996.
- [Lue97] David Luebke.
A survey of polygonal simplification algorithms.
Technical Report TR97-045, University of North Carolina at Chapel Hill, 1997.
- [Lue98] David Luebke.
Simplification culling of static and dynamic scene graphs.
Technical Report TR98-013, University of North Carolina at Chapel Hill, 1998.
- [Mac95] David MacKay.
A short course in information theory - outline.
DJCM, 1995.
- [Man98] Stephen Mann.
An improved parametric side-vertex triangle mesh interpolant.
Graphics Interface 98, pages 35–42, 1998.
- [McR98] Tom McReynolds.
Advanced graphics programming techniques using OpenGL.
Technical report, SIGGRAPH 98, april 1998.
Course notes. Demonstrates sophisticated and novel Computer Graphics programming techniques, implemented using OpenGL.
- [met00] Metacreation. <http://www.metacreation.com/>, 2000.
- [MNW95] A. Moffat, R. M. Neal, and I. H. Witten.
Arithmetic coding revisited.
Preliminary pres. at IEEE Data Compression Conference, 1995.
- [mpea] MPEG home page. <http://drogo.csel.stet.it/mpeg/>.
- [mpeb] MPEG4 home page. <http://drogo.csel.stet.it/mpeg/standards/mpeg-4/mpeg-4.htm>.
- [NG96] Mark Nelson and Jean-Loup Gailly.
The data compression book.
MT, 1996.
- [nur] The Nurbs++ package. <http://yukon.genie.uottawa.ca/~lavoie/software/nurbs/>.
Non-Uniform Rational B-Splines (NURBS) curves and surface are parametric functions which can represent any type of curves or surfaces. This C++ library hides the basic mathematics of NURBS.

- [Ond99] Olivier Oudet.
Réseaux maillés tridimensionnels. simplification, codage et compression.
Technical report, ENST Paris, june 1999.
Mémoire d'option réalisé sous la direction de Francis Schmitt.
- [ope] OpenGL, site officiel. <http://www.opengl.org/>.
- [PH97] Jovan Popovic and Hugues Hoppe.
Progressive simplicial complexes.
SIGGRAPH 97 Conference Proceedings, 1997.
- [PML99] S. Pateux, G. Marquant, and C. Labit.
Codage d'images fixes par maillages.
In *Coresa*, june 1999.
- [Pra98] Hartmut Prautzsch.
Smoothness of subdivision surfaces at extraordinary points.
Adv. in : Comp. Math, 9 :377–390, 1998.
- [PU98a] Hartmut Prautzsch and Georg Umlauf.
A G^1 and a G^2 subdivision scheme for triangular nets.
Submitted to : Int. J. Shape Modelling, 1998.
- [PU98b] Hartmut Prautzsch and Georg Umlauf.
A G^2 subdivision algorithm.
Proceedings of the Workshop In Geometric Modeling, 1998.
- [PU98c] Hartmut Prautzsch and Georg Umlauf.
Improved triangular subdivision schemes.
Proceedings of the CGI 98, Hannover, 1998, pages 626–632, 1998.
- [RB92] Jarek Rossignac and Paul Borrel.
Multi-resolution 3D approximations for rendering complex scenes.
Technical Report 17697, IBM, February 1992.
Technical report, Yorktown Heights, NY 10598, February 1992. IBM Research Report RC 17697 (77951).
- [RC99] Ramesh Raskar and Michael Cohen.
Image precision silhouette edges.
Symposium on Interactive 3D Graphics 1999, Atlanta, 1999.
- [Ris83] Jorma Rissanen.
A universal data compression system.
IEEE Transactions on Information Theory, 29 :656–1983, sept 1983.
- [RL79] J. Rissanen and G.G. Langdon.
Arithmetic coding.
IBM J. Res. Dev., 23(2), march 1979.
- [Ros99] Jarek Rossignac.
Edgebreaker : Connectivity compression for triangle meshes.
IEEE Transactions on Visualization and Computer Graphics, pages 47–61, 1999.

- [RP98] Gilles Rougeron and Bernard Péroche.
Color fidelity in computer graphics : a survey.
Computer Graphics Forum, 17 :3–15, 1998.
- [SA98] Mark Segal and Kurt Akeley.
The OpenGL Graphics System.
Silicon Graphics, John Leech, march 1998.
- [Sah00] Subhasis Saha.
Image compression - from dct to wavelets : A review.
Technical report, University of California, 2000.
<http://www.acm.org/crossroads/xrds6-3/sahaimgcoding.html>.
- [SBD86] F. Schmitt, B.A. Barsky, and W. Du.
An adaptive subdivision method for surface fitting from sampled data.
SIGGRAPH 86 Conference Proceedings, 20(4) :179–188, 1986.
- [Sch96] Jean E. Schweitzer.
Analysis and Application of Subdivision Surfaces.
PhD thesis, University of Washington, 1996.
- [Sch99] Peter Schröder.
Opportunities for subdivision-based multiresolution modeling.
PACIFIC GRAPHICS 99 Conference Proceedings, pages 104–105, 1999.
- [SGG⁺00] Pedro V. Sander, Xianfeng Gu, Steven J. Gortier, Hugues Hoppe, and John Snyder.
Silhouette clipping.
To appear in Siggraph 2000 Conference proceedings, 2000.
- [SGW98] Oliver G. Staadt, Markus H. Groos, and Roger Weber.
Multiresolution compression and reconstruction.
EUROGRAPHICS 98, 1998.
- [Sha48] C.E. Shannon.
A mathematical theory of communication.
Bell systems technical journal, 27 :379–423, 623–656, 1948.
- [SL98] Henri Sanson and Patrick Lechat.
Finite-element-based image approximation using a geometric distance.
Rapport de recherche interne CNET / DIH / HDM, januar 1998.
- [Soh99] C. Sohler.
Fast reconstruction of delaunay triangulation.
Proceedings of 11th Canadian Conference on Computational Geometry, 1999.
<http://www.cs.ubc.ca/conferences/CCCCG/>.
- [SR99] Andrzej Szymczak and Jarek Rossignac.
Grow & fold : Compression of tetrahedral meshes.
Fifth Symposium on Solid Modeling, 1999.
- [SS95] Peter Schröder and Wim Sweldens.
Spherical wavelets : Efficiently representing functions on the sphere.
Computer Graphics Proceedings (SIGGRAPH 95), pages 161–172, 1995.

- [SS97] Dieter Schmalstieg and Gernot Schaufler.
Smooth levels of detail.
Technical Report TR-186-2-97-03, Institute of Computer Graphics, Vienna University of Technology, 97.
- [STK99] Hiromasa Suzuki, Shinko Takeuchi, and Takashi Kanai.
Subdivision surface fitting to a range of points.
PACIFIC GRAPHICS 99 Conference Proceedings, 1999.
- [SZL92] William Schröder, Jonathan Zarge, and William Lorensen.
Decimation of triangle meshes.
SIGGRAPH 92 Conference Proceedings, pages 65–70, 1992.
- [SZSS98] Thomas W. Sedelberg, Jianmin Zheng, David Sewell, and Malcolm Sabin.
Non-uniform recursive subdivision surfaces.
SIGGRAPH 98 Conference Proceedings, 1998.
- [Tau98] Gabriel Taubin.
Snhc verification model 9.0 [3D mesh encoding].
Technical Report JTC1/SC29/WG11 MPEG98/M3809, ISO/IEC, 1998.
- [TG98] Costa Touma and Craig Gotsman.
Triangle mesh compression.
INTERFACE 98 Conference Proceedings, 1998.
- [TGHL98] Gabriel Taubin, André Guézic, William Horn, and Francis Lazarus.
Progressive forest split compression.
SIGGRAPH 98 Conference Proceedings, 1998.
- [THLR97] Gabriel Taubin, William Horn, Francis Lazarus, and Jarek Rossignac.
Geometry coding and VRML.
Technical Report 20925, IBM T.J. Watson Research Center, july 1997.
- [TRD⁺98] Gabriel Taubin, Jarek Rossignac, Michael Deering, Hugues Hoppe, Peter Schröder, and Hans-Peter Seidel.
3D geometry compression.
Technical report, SIGGRAPH 98, july 1998.
Course notes.
- [Tur92] Greg Turk.
Re-tiling polygonal surfaces.
SIGGRAPH 92 Conference Proceedings, 26 :55–64, 1992.
- [TVC96] M.J. Tsai, J. Villasenor, and F. Chen.
Stack-run image coding.
IEEE Transactions on Circuits and Systems for Video Technology, 6 :519–521, 1996.
- [Uml95] Georg Umlauf.
Analysing the characteristic map of triangular subdivision schemes.
To appear : Constr. Approx, 1995.
- [vhu] Virtual Humans.
<http://web3d.about.com/compute/web3d/msubvhum.htm>.

- [vrma] VRML, site officiel. <http://www.vrml.org/>.
- [vrmb] VRML 97, spécifications. <http://www.vrml.org/technicalinfo/specifications/vrml97/>.
- [VTK⁺99] Sébastien Valette, Fabien Thibon, Yun-Sang Kim, Ho-Youl Jung, Isabelle Magnin, and Rémy Prost.
Décomposition en ondelettes de maillages triangulaires 3D irrégulièrement subdivisés. application à la compression.
GRETSI 99, Vannes, pages 941–944, 1999.
- [Wag98] Michael G. Wagner.
Advanced animation techniques in VRML 97.
Technical report, Arizona State University, 1998.
<http://stop.at/michael.wagner> - Implémentation de Gaspard Breton.
- [War95] Joe Warren.
Subdivision methods for geometric design.
Technical report, Rice University - Department of Computer Science, 1995.
- [Wel84] Terry Welch.
A technique for high-performance data compression.
Computer, 17,6 :8–19, june 1984.
- [Wit97] Douglas Withers.
A rapid entropy-coding algorithm.
Dr Dobb's Journal, april 1997.
- [WMW86] G. Wyvill, C. McPheeters, and B. Wyvill.
Data structures for soft objects.
The Visual computer, pages 227–234, 1986.
- [WNC87] I. H. Witten, R. M. Neal, and J. G. Cleary.
Arithmetic coding for data compression.
Communications of the ACM, 30(6), june 1987.
- [wot] WOTSIT, the programmer's file format collection. <http://www.wotsit.org/>.
- [XESV97] Julie C. Xia, Jihad El-Sana, and Amitabh Varshney.
Adaptive real-time level of detail based rendering for polygonal models.
VISUALIZATION 97, 1997.
- [XHM99] Xinyu Xiang, Martin Held, and Joseph D. B. Mitchell.
Fast and effective stripification of polygonal surface models.
Proc. ACM Symposium on Interactive 3D Graphics, 1999.
- [YS00] Yucel Yemez and Francis Schmitt.
Modélisation progressive multiniveaux d'objets 3D par particules.
RFIA 2000, 2000.
- [ZH97] Hansong Zhang and Kenneth E. Hoff.
Fast backface culling using normal masks.
Symposium on Interactive 3D Graphics, 1997.

- [ZL78] Jacob Ziv and Abraham Lempel.
Compression of individual sequences via variable-rate coding.
IEEE Trans. Inf. Theory, 24(5) :530–536, sept 1978.
- [ZS99] Denis Zorin and Peter Schröder.
Subdivision for modeling and animation.
SIGGRAPH 99 course notes, 1999.
- [ZSS96] Denis Zorin, Peter Schröder, and Wim Sweldens.
Interpolating subdivision for meshes with arbitrary topology.
SIGGRAPH 96 Conference Proceedings, 1996.
- [ZT91] O.C. Zienkiewicz and R.L. Taylor.
La méthode des éléments finis, formulation de base et problèmes linéaires.
afnor technique, 4 edition, 1991.

Publications de l’auteur

- [AL99a] Pierre Alliez and Nathalie Laurent.
Compression et représentation échelonnable de maillages triangulaires.
AFIG’99 - Journées de l’Association Française d’Informatique Graphique, Reims, nov 1999.
- [AL99b] Pierre Alliez and Nathalie Laurent.
Mesh coding : application to texture and scalable 3D scene coding.
Annals of Telecommunications, 1999.
Parution prévue courant 2000.
- [ALS98a] Pierre Alliez, Nathalie Laurent, and Henri Sanson.
Approximation de maillages 3D utilisant une métrique basée volume.
AFIG’98 - Journées de l’Association Française d’Informatique Graphique, Dunkerque, pages 77–87, dec 1998.
- [ALS98b] Pierre Alliez, Nathalie Laurent, and Henri Sanson.
Quelques pistes vers un codage progressif de scènes 3D dans un contexte télécommunicant.
JGA’98 - Journées de Géométrie Algorithmique, Sophia Antipolis, 1998.
Communication orale.
- [ALS99] Pierre Alliez, Nathalie Laurent, and Henri Sanson.
Codage scalable de scènes 3D.
CORESA’99 Conference Proceedings, Sophia-Antipolis, juin 1999.
- [ALSS99] Pierre Alliez, Nathalie Laurent, Henri Sanson, and Francis Schmitt.
Mesh approximation using a volume-based metric.
PACIFIC GRAPHICS’99 Conference Proceedings, Seoul, Korea, oct 1999.

Brevets d’invention

- [AL99a] Pierre Alliez and Nathalie Laurent.
Procédé de codage d’un maillage par conquête sur les arêtes, privilégiant un sommet complet en tant que pivot, et procédé de décodage correspondant, juin 1999.
Brevet d’invention.

- [AL99b] Pierre Alliez and Nathalie Laurent.
Procédé de simplification d'un maillage source, tenant compte de la courbure locale et de la dynamique géométrique locale, et applications correspondantes, jan 1999.
Brevet d'invention.
- [AL00] Pierre Alliez and Nathalie Laurent.
Procédé de raffinement continu et dépendant du point de vue de maillages triangulaires, mai 2000.
Brevet d'invention, dépôt en cours.
- [ALS98] Pierre Alliez, Nathalie Laurent, and Henri Sanson.
Procédé de codage d'un maillage source par minimisation de volume et applications correspondantes, nov 1998.
Brevet d'invention.
- [LA00] Nathalie Laurent and Pierre Alliez.
Procédé de transmission progressive et de visualisation adaptative d'une image, mai 2000.
Brevet d'invention, dépôt en cours.

Publications INRIA

- [ADS97] Pierre Alliez, Olivier Devillers, and Jack Snoeyink.
Removing degeneracies by perturbing the problem or the world.
Technical Report 3316, INRIA Sophia Antipolis, projet Prisme, dec 1997.
Elaboré pendant le stage du DEA ARAVIS (Automatique Robotique Algorithmique Vision Image et Signal) de l'École Supérieure en Science Informatique.
- [ADS98] Pierre Alliez, Olivier Devillers, and Jack Snoeyink.
Removing degeneracies by perturbing the problem or the world.
Canadian Conference on Computational Geometry. Montréal, Québec, Canada, 1998.
<http://cgm.cs.mcgill.ca/~ccc98/>.
- [ADS00] Pierre Alliez, Olivier Devillers, and Jack Snoeyink.
Removing degeneracies by perturbing the problem or perturbing the world.
Reliable Computing, 6(1) :61–79, feb 2000.
<http://www.wkap.nl/journalhome.htm/1385-3139>. We describe two problem specific approaches to remove geometric degeneracies that we call perturbing the problem and perturbing the world. Using as our primary examples 2D and 3D Delaunay triangulation with Euclidean and polygonal metrics, we show that these approaches lead to relatively simple and efficient perturbations of the points that do not depend on a fixed ordering or index. Thus, they produce canonical output, which is important for producing test suites and verifiers for randomized or dynamic geometric algorithms.

Annexe A

Simplification

Le pseudo-code suivant résume la séquence algorithmique mise en œuvre pour la simplification de maillages.

Paramètres :

- F' : complexité géométrique souhaitée après simplification
- s : seuil initial de courbure
- c : valeur angulaire définissant le pli formé par une arête vive
- step : taux de décroissance du terme de courbure

M : maillage comportant F faces

1. Lister les arêtes du maillage M dans la liste L, cette liste stockant également la courbure et la longueur associée à l'arête courante.
2. Construire une adjacence arête / arêtes afin d'accélérer la mise à jour ultérieure lors d'une opération élémentaire de simplification par fusion d'arête.
3. Construire une double queue de priorité dont le critère prioritaire concerne la courbure et le critère secondaire la longueur d'arête.
4. Retirer itérativement l'arête (vi,vj) de moindre coût de la pile (i.e. l'arête de plus faible longueur parmi toutes les arêtes dont la courbure est mesurée en deçà du seuil de courbure s).
5. Contracter l'arête (vi,vj). Dénombrer les nombres d'arêtes vives adjacentes aux sommets vi et vj suivant la valeur c. Si ces nombres sont identiques le sommet v résultant de la fusion est positionné au milieu de l'arête, sinon il est placé sur le sommet de plus grand nombre d'arêtes vives.
6. Mettre à jour dans la pile les arêtes modifiées par cette contraction. La modification se traduit soit en une mise à jour longueur / courbure, soit en une suppression d'arêtes de la pile.
7. Réduire le seuil initial s de courbure afin d'autoriser l'obtention de la complexité géométrique souhaitée F'.
8. Reprendre à l'étape 4.

Annexe B

Codage de la connectivité

Le pseudo-code suivant résume la séquence algorithmique mise en œuvre pour le codage de la connectivité.

Paramètres :

=====

Maillage M
Index de face germe
Index de sommet germe

Initialisation :

=====

1. Ajouter un sommet fantôme en présence de bords
2. Générer une pile de listes d'arêtes (une liste vide active au départ)
3. Allouer une liste de références aux sommets (VertexRun)
4. Générer un vecteur global des arêtes du maillage M
5. Générer une liste de références aux arêtes par sommet
6. Trier la liste des références aux arêtes autour de chaque sommet dans le sens anti-horaire
7. Marquer tous les sommets de M comme libres
8. Initialiser la liste active avec les arêtes de la face germe
9. Générer trois codes de valence correspondants
10. Initialiser le pivot de la liste active avec le sommet germe

Boucle globale de conquête :

=====

Tant que (la conquête se déroule)

```
{
  pEdge := arête libre du pivot courant de la liste active
  {
    si (aucune arête n'est libre ou
        l'arête libre pointe vers un fantôme)
    {
      pEdge est invalide
      sortir
    }
    si (il n'y a pas d'arête libre dans le secteur angulaire
        orienté formé par les arêtes précédant et suivant le pivot)
    {
      générer un code "contextuellement complet"
      pEdge est invalide
      sortir
    }
  }
}
```

```

    }
  }

si(pEdge est valide) -> il reste une arête libre à conquérir
{
  pVertex := sommet relié au pivot par l'arête pEdge
  flag := drapeau de pVertex (libre, conquis ou fantôme)

  Le sommet est non codé
  si(flag = libre)
  {
    insérer pVertex avant le pivot dans la liste
    marquer les deux arêtes correspondantes conquises
    marquer pVertex conquis
    générer un code de valence de pVertex
  }

  Le sommet correspond au fantôme
  si(flag = fantome)
  {
    générer un code spécifique au fantôme
    sa valence est codée globalement à part
  }

  Séparation de liste :
  si(flag = conquis et pVertex appartient à la liste courante)
  {
    marquer pEdge conquise
    séparer la liste active en deux listes
    pNewList := nouvelle liste créée à l'issue de la séparation
    ajouter un code spécifique à la séparation de listes
    générer un bit de signe de l'offset
    générer un code de la valeur absolue de l'offset
    ajouter pNewList sur la pile
  }

  Fusion de listes :
  si(flag = conquis et pVertex appartient à une autre liste de la pile)
  {
    marquer pEdge conquise
    fusionner les deux listes en une
    supprimer la liste fusionnée de la pile
    ajouter un code spécifique à la fusion de listes
    générer un code d'offset dans la pile
    générer un code d'offset dans la liste
  }
}
sinon -> le pivot courant est complet
{
  tant que(un pivot complet est supprimé de la liste)
  {
    si(le pivot courant n'est pas complet ni fantôme)
      sortir
  }
}

```

```

si(le pivot courant est complet et n'est pas un fantôme)
{
    marquer l'arête reliant le sommet précédant et /
    le sommet suivant le pivot comme conquise
    mémoriser la position du pivot dans la liste
    supprimer le pivot courant de la liste
    trouver un sommet complet comme candidat au pivot
    {
        lancer deux pointeurs simultanés avant et arrière
        si(deux sommets complets se présentent)
            donner la priorité au pointeur avant pour lever l'ambiguïté
    }
    si(un sommet complet se présente)
        affecter ce sommet comme pivot
    sinon
        affecter le sommet suivant le précédent pivot
        dans la liste
}
si(le pivot courant est fantôme)
{
    si(le fantôme est complet)
    {
        supprimer le fantôme de la liste
        affecter le sommet suivant non fantôme
        attention : il peut y avoir plusieurs occurrences
        d'un sommet fantôme dans la même liste
    }
    sinon
    {
        il ne peut rester pivot mais on le conserve
        affecter le sommet suivant non fantôme
    }
}
}
}

si(la liste active est vide)
{
    supprimer la liste active de la pile
    si(la pile est non vide)
        la nouvelle liste dans la pile devient active
    sinon
        la conquête est terminée
}
}

```

Nettoyage et génération des trains binaires :

- =====
- 1.Effacer la liste d'arêtes, utile seulement pour la conquête
 - 2.Supprimer le sommet pivot du maillage
 - 3.Générer le train binaire

Ecrire :

le nombre de sommets sur 32 bits

```

    un bit signalant la présence d'un fantôme
    le nombre maximal de valence (fantôme non compris)
Coder :
    le vecteur de valences par codage entropique à modèle adaptatif
Ecrire :
    un bit signalant la présence de codes de séparation de listes
    les bits de signe des offsets
Coder :
    le vecteur d'offsets par codage entropique
Ecrire :
    un bit signalant la présence de codes de fusion de listes
    les offsets de liste et de pile correspondants

```

Codage de la géométrie

Le pseudo-code suivant décrit la séquence algorithmique mise en œuvre pour le codage de la géométrie (les positions des sommets) :

```

Paramètres :
=====
Maillage M
Ordre sur les sommets fourni par le codeur de connectivité
NbBitsQuantize := Nombre de bits de quantification du niveau de base
SizeMobileMean := Largeur du support du calcul de la moyenne mobile
NbBitsRefine := Nombre de plans de bits de raffinement

Ecriture de l'en-tête
=====
Ecrire l'en-tête composée des paramètres
Ecrire les positions des trois premiers sommets

Codage du niveau de base
=====
Pour(le nombre de sommets à coder)
{
    trouver le mode de prédiction le plus adaptée (minimisant le résidu)
    pour(chaque coordonnée)
    {
        si(la prédiction n'est pas un saut d'adjacence)
            générer et quantifier un résidu
        sinon
            générer un ratio entier K et deux résidus différentiels
    }
}
Coder les ratios entiers Ki par codage entropique
Coder les modes de prédiction par codage entropique
Ecrire les bits de signe des résidus
Coder les valeurs absolues des résidus quantifiés par codage entropique

Codage des raffinements
=====

```

```
Mesurer et normaliser les résidus absolus
Ecrire les facteurs d'échelle
Ecrire et retirer les bits de signe des résidus absolus
Compresser les ordres de grandeur des raffinements par codage entropique
Pour(chaque plan de bit de raffinement)
    Pour(chaque sommet)
        Pour(chaque coordonnée)
            si(nécessaire)
                Ecrire le bit de raffinement
```


Annexe C

Raffinement adaptatif global

Le pseudo-code suivant résume la séquence algorithmique globale mise en œuvre pour le raffinement adaptatif.

M : maillage comportant F faces et S sommets
Requête : point de vue

Paramètres :

v : nombre d'itérations dans le champ de vision
s : nombre d'itérations sur la silhouette

```
1. Annuler le dernier raffinement adaptatif
2. Subdivision dans le champ de vision
pour(v itérations)
{
    marquer les faces de M dans le champ de vision
    lancer une subdivision hybride sur les faces marquées
}
3. Subdivision sur la silhouette
pour(s itérations)
{
    marquer les faces de M au voisinage de la silhouette
    restreindre la détection à l'ensemble des faces de la silhouette précédente
    lancer une subdivision hybride sur les faces marquées
}
```

Une itération de raffinement par subdivision

Le pseudo-code suivant décrit une itération de raffinement adaptatif :

M : maillage comportant F faces et S sommets
La structure de données ne comprend pas d'arêtes

1. Subdivision barycentrique

=====

Soit F' ($F' \leq F$) le nombre de faces à subdiviser

```
pour( $F'$  itérations)
{
    f : face courante à subdiviser composée du triplet ordonné {s1;s2;s3}
```

```

ajouter un sommet s au centre de la face f
ajouter une action à l'historique du sommet s
marquer les sommets s1, s2 et s3 à filtrer
ajouter une face f1 formée par le triplet {s2,s3,s}
ajouter une face f2 formée par le triplet {s3,s1,s}
modifier la face f, maintenant formée par le triplet {s1,s2,s}
ajouter une action à l'historique de chaque face f, f1 et f2
mettre à jour les adjacences sommets/faces et faces/faces
calculer les normales aux sommets
calculer les normales aux faces
marquer les faces f, f1 et f2 dont l'arête est à permuter
}
Le maillage comporte maintenant  $F'' = F + 2 * F'$  faces

```

2. Filtrage des positions

```

=====
dénombrer les sommets a filtrer
temp : structure temporaire de stockage des nouvelles positions

pour(le nombre de sommets à filtrer)
{
  s : sommet courant à filtrer
  n : valence du sommet s mémorisée avant subdivision un-vers-trois
  p : calculer la nouvelle position du sommet s à partir du
      coefficient alpha(n), de la position de s et de ses sommets adjacents
  stocker p dans la structure temporaire temp
}
pour(le nombre de sommets à filtrer)
{
  ajouter une action à l'historique du sommet courant
  appliquer la nouvelle position p au sommet courant
}

```

3. Permutation d'arêtes

```

=====
tant que(il reste des faces dont l'arête est à permuter)
{
  f : face courante dont les arêtes sont candidates à la permutation
  pour(chaque face f' adjacente à f non marquée)
    si(f' a été obtenue par subdivision barycentrique)
      {
        permuter l'arête partageant f et f' par échange de sommets
        ajouter une action à l'historique des faces f et f'
      }
  marquer la face f
}

```

4. Subdivision et approximation des arêtes vives régulières

Annexe D

Cette annexe consigne les calculs des coefficients du filtrage $\alpha(n)$ appliqué lors de la subdivision barycentrique des faces (voir chapitre 4).

Degré 3

La figure 5.1 illustre la construction de la matrice de subdivision globale $S(n)$ pour la valence 3.

$$A(3) = \begin{bmatrix} \frac{\alpha}{3+\alpha} & (3+\alpha)^{-1} & (3+\alpha)^{-1} & (3+\alpha)^{-1} \\ 1/3 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 \end{bmatrix}$$

$$B(3) = \begin{bmatrix} \frac{\alpha}{3+\alpha} & (3+\alpha)^{-1} & (3+\alpha)^{-1} & (3+\alpha)^{-1} \\ 1/3 & 1/3 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 1/3 & 1/3 \end{bmatrix}$$

$$S(3) = \begin{bmatrix} \frac{\alpha^2}{(3+\alpha)^2} + (3+\alpha)^{-1} & \frac{\alpha}{(3+\alpha)^2} + 2/3 (3+\alpha)^{-1} & \frac{\alpha}{(3+\alpha)^2} + 2/3 (3+\alpha)^{-1} & \frac{\alpha}{(3+\alpha)^2} + 2/3 (3+\alpha)^{-1} \\ 1/3 \frac{\alpha}{3+\alpha} + 2/9 & 1/3 (3+\alpha)^{-1} + 2/9 & 1/3 (3+\alpha)^{-1} + 1/9 & 1/3 (3+\alpha)^{-1} + 1/9 \\ 1/3 \frac{\alpha}{3+\alpha} + 2/9 & 1/3 (3+\alpha)^{-1} + 1/9 & 1/3 (3+\alpha)^{-1} + 2/9 & 1/3 (3+\alpha)^{-1} + 1/9 \\ 1/3 \frac{\alpha}{3+\alpha} + 2/9 & 1/3 (3+\alpha)^{-1} + 1/9 & 1/3 (3+\alpha)^{-1} + 1/9 & 1/3 (3+\alpha)^{-1} + 2/9 \end{bmatrix}$$

$$\lambda(3) = \left\{ 1, \frac{1}{9}, \frac{9-12\alpha+4\alpha^2}{9+6\alpha+\alpha^2}, \frac{1}{9}, \frac{1}{9} \right\}$$

$$\frac{(2\alpha-3)^2}{9(\alpha+3)^2} = \left(\frac{1}{9}\right)^2 \Rightarrow \alpha(3) = \frac{12}{5}$$

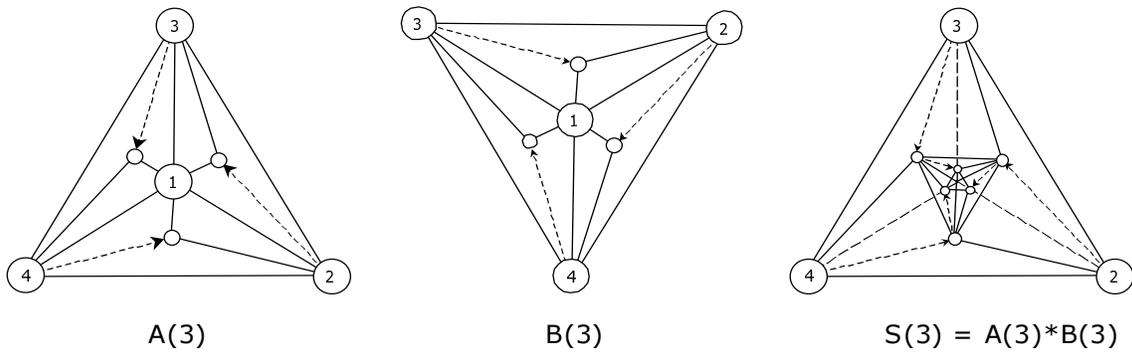


FIG. 5.1 : Matrice $A(3)$: une itération de subdivision dans le sens anti-horaire pour un sommet de valence 3. Matrice $B(3)$: une itération de subdivision dans le sens horaire afin d'éviter la définition d'une matrice de permutation. La matrice globale $S(3) = A(3) \cdot B(3)$ permet d'obtenir un schéma auto-répliatif nécessaire avant toute analyse spectrale.

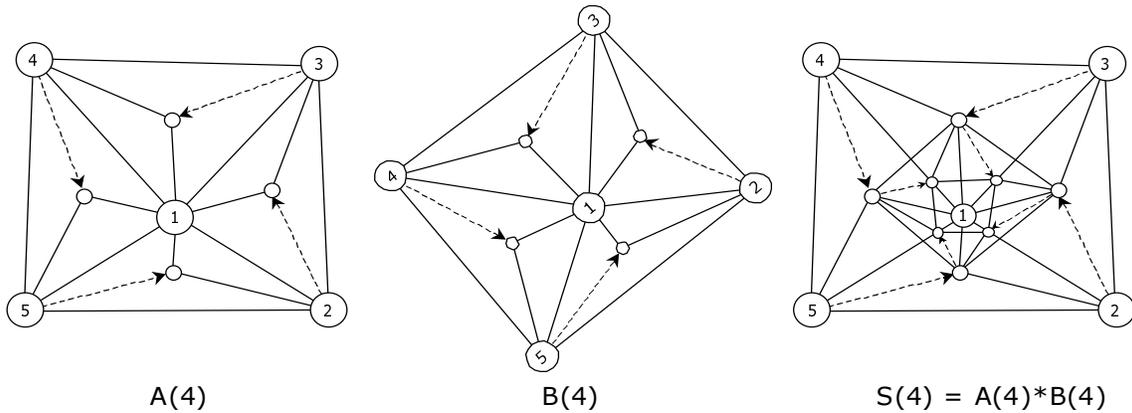


FIG. 5.2 : Matrice $A(4)$: une itération de subdivision dans le sens anti-horaire pour un sommet de valence 4. Matrice $B(4)$: une itération de subdivision dans le sens horaire. La matrice globale $S(4) = A(4) \cdot B(4)$ permet d'obtenir un schéma auto-répliatif nécessaire avant toute analyse spectrale.

Degré 4

La figure 5.2 illustre la construction de la matrice de subdivision globale $S(n)$ pour la valence 4.

$$A(4) = \begin{bmatrix} \frac{\alpha}{4+\alpha} & (4+\alpha)^{-1} & (4+\alpha)^{-1} & (4+\alpha)^{-1} & (4+\alpha)^{-1} \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 0 & 1/3 \end{bmatrix}$$

$$B(4) = \begin{bmatrix} \frac{\alpha}{4+\alpha} & (4+\alpha)^{-1} & (4+\alpha)^{-1} & (4+\alpha)^{-1} & (4+\alpha)^{-1} \\ 1/3 & 1/3 & 0 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 \end{bmatrix}$$

$$S(4) = A(4) \cdot B(4) = \begin{bmatrix} \frac{\alpha^2}{(4+\alpha)^2} + 4/3(4+\alpha)^{-1} & \frac{\alpha}{(4+\alpha)^2} + 2/3(4+\alpha)^{-1} & \frac{\alpha}{(4+\alpha)^2} + 2/3(4+\alpha)^{-1} & \frac{\alpha}{(4+\alpha)^2} + 2/3(4+\alpha)^{-1} & \frac{\alpha}{(4+\alpha)^2} + 2/3(4+\alpha)^{-1} \\ 1/3 \frac{\alpha}{4+\alpha} + 2/9 & 1/3(4+\alpha)^{-1} + 2/9 & 1/3(4+\alpha)^{-1} + 1/9 & 1/3(4+\alpha)^{-1} & 1/3(4+\alpha)^{-1} + 1/9 \\ 1/3 \frac{\alpha}{4+\alpha} + 2/9 & 1/3(4+\alpha)^{-1} + 1/9 & 1/3(4+\alpha)^{-1} + 2/9 & 1/3(4+\alpha)^{-1} + 1/9 & 1/3(4+\alpha)^{-1} \\ 1/3 \frac{\alpha}{4+\alpha} + 2/9 & 1/3(4+\alpha)^{-1} & 1/3(4+\alpha)^{-1} + 1/9 & 1/3(4+\alpha)^{-1} + 2/9 & 1/3(4+\alpha)^{-1} + 1/9 \\ 1/3 \frac{\alpha}{4+\alpha} + 2/9 & 1/3(4+\alpha)^{-1} + 1/9 & 1/3(4+\alpha)^{-1} & 1/3(4+\alpha)^{-1} + 1/9 & 1/3(4+\alpha)^{-1} + 2/9 \end{bmatrix}$$

$$\lambda(4) = \left\{ 0, 1, \frac{4}{9}, \frac{4-4\alpha+\alpha^2}{16+8\alpha+\alpha^2}, \frac{2}{9}, \frac{2}{9} \right\}$$

$$\frac{(2\alpha-4)^2}{9(\alpha+4)^2} = \left(\frac{2}{9}\right)^2 \Rightarrow \alpha(4) = 5$$

Degré 5

$$A(5) = \begin{bmatrix} \frac{\alpha}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} \end{bmatrix}$$

$$B(5) = \begin{bmatrix} \frac{\alpha}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} & \frac{1}{5+\alpha} \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

$$S(5) = A(5) \cdot B(5) = \begin{bmatrix} \frac{\alpha^2}{(5+\alpha)^2} + 5/3(5+\alpha)^{-1} & \frac{\alpha}{(5+\alpha)^2} + 2/3(5+\alpha)^{-1} \\ 1/3 \frac{\alpha}{5+\alpha} + 2/9 & 1/3(5+\alpha)^{-1} + 2/9 & 1/3(5+\alpha)^{-1} + 1/9 & 1/3(5+\alpha)^{-1} & 1/3(5+\alpha)^{-1} & 1/3(5+\alpha)^{-1} + 1/9 \\ 1/3 \frac{\alpha}{5+\alpha} + 2/9 & 1/3(5+\alpha)^{-1} + 1/9 & 1/3(5+\alpha)^{-1} + 2/9 & 1/3(5+\alpha)^{-1} + 1/9 & 1/3(5+\alpha)^{-1} & 1/3(5+\alpha)^{-1} \\ 1/3 \frac{\alpha}{5+\alpha} + 2/9 & 1/3(5+\alpha)^{-1} & 1/3(5+\alpha)^{-1} + 1/9 & 1/3(5+\alpha)^{-1} + 2/9 & 1/3(5+\alpha)^{-1} + 1/9 & 1/3(5+\alpha)^{-1} \\ 1/3 \frac{\alpha}{5+\alpha} + 2/9 & 1/3(5+\alpha)^{-1} & 1/3(5+\alpha)^{-1} & 1/3(5+\alpha)^{-1} + 1/9 & 1/3(5+\alpha)^{-1} + 2/9 & 1/3(5+\alpha)^{-1} + 1/9 \\ 1/3 \frac{\alpha}{5+\alpha} + 2/9 & 1/3(5+\alpha)^{-1} + 1/9 & 1/3(5+\alpha)^{-1} & 1/3(5+\alpha)^{-1} & 1/3(5+\alpha)^{-1} + 1/9 & 1/3(5+\alpha)^{-1} + 2/9 \end{bmatrix}$$

$$\lambda(5) = \left\{ 1, \frac{1}{9}, \frac{4\alpha^2 + 25 - 20\alpha}{25 + 10\alpha + \alpha^2}, \frac{1}{6} + \frac{1}{18}\sqrt{5}, \frac{1}{6} - \frac{1}{18}\sqrt{5}, \frac{1}{6} + \frac{1}{18}\sqrt{5}, \frac{1}{6} - \frac{1}{18}\sqrt{5} \right\}$$

$$\frac{(2\alpha - 5)^2}{9(\alpha + 5)^2} = \left(\frac{1}{6} + \frac{1}{18}\sqrt{5} \right)^2 \Rightarrow \alpha(5) = \frac{215}{38} + \frac{45}{38}\sqrt{5} = 8.305869973$$

Degré 6

$$A(6) = \begin{bmatrix} \frac{\alpha}{6+\alpha} & (6+\alpha)^{-1} & (6+\alpha)^{-1} & (6+\alpha)^{-1} & (6+\alpha)^{-1} & (6+\alpha)^{-1} & (6+\alpha)^{-1} \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 0 & 0 & 0 & 1/3 \end{bmatrix}$$

$$B(6) = \begin{bmatrix} \frac{\alpha}{6+\alpha} & (6+\alpha)^{-1} & (6+\alpha)^{-1} & (6+\alpha)^{-1} & (6+\alpha)^{-1} & (6+\alpha)^{-1} & (6+\alpha)^{-1} \\ 1/3 & 1/3 & 0 & 0 & 0 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 1/3 & 1/3 \end{bmatrix}$$

$$S(6) = \begin{bmatrix} \frac{\alpha^2}{(6+\alpha)^2} + 2(6+\alpha)^{-1} & \frac{\alpha}{(6+\alpha)^2} + 2/3(6+\alpha)^{-1} \\ 1/3 \frac{\alpha}{6+\alpha} + 2/9 & 1/3(6+\alpha)^{-1} + 2/9 & 1/3(6+\alpha)^{-1} + 1/9 & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} + 1/9 \\ 1/3 \frac{\alpha}{6+\alpha} + 2/9 & 1/3(6+\alpha)^{-1} + 1/9 & 1/3(6+\alpha)^{-1} + 2/9 & 1/3(6+\alpha)^{-1} + 1/9 & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} \\ 1/3 \frac{\alpha}{6+\alpha} + 2/9 & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} + 1/9 & 1/3(6+\alpha)^{-1} + 2/9 & 1/3(6+\alpha)^{-1} + 1/9 & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} \\ 1/3 \frac{\alpha}{6+\alpha} + 2/9 & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} + 1/9 & 1/3(6+\alpha)^{-1} + 2/9 & 1/3(6+\alpha)^{-1} + 1/9 & 1/3(6+\alpha)^{-1} \\ 1/3 \frac{\alpha}{6+\alpha} + 2/9 & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} + 1/9 & 1/3(6+\alpha)^{-1} + 2/9 & 1/3(6+\alpha)^{-1} + 1/9 \\ 1/3 \frac{\alpha}{6+\alpha} + 2/9 & 1/3(6+\alpha)^{-1} + 1/9 & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} & 1/3(6+\alpha)^{-1} + 1/9 & 1/3(6+\alpha)^{-1} + 2/9 \end{bmatrix}$$

$$\lambda(6) = \left\{ 0, 1, \frac{4}{9}, \frac{-6\alpha + 9 + \alpha^2}{36 + 12\alpha + \alpha^2}, \frac{1}{9}, \frac{1}{3}, \frac{1}{9}, \frac{1}{3} \right\}$$

$$\frac{(2\alpha - 6)^2}{9(\alpha + 6)^2} = \left(\frac{1}{3}\right)^2 \Rightarrow \alpha(6) = 12$$

Degré 7

$$A(7) = \begin{bmatrix} \frac{\alpha}{7+\alpha} & (7+\alpha)^{-1} \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 1/3 \end{bmatrix}$$

$$B(7) = \begin{bmatrix} \frac{\alpha}{7+\alpha} & (7+\alpha)^{-1} \\ 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 \end{bmatrix}$$

$$S(7) = A(7) \cdot B(7)$$

$$\lambda(7) = \left\{ 1, \frac{1}{9} \frac{-28\alpha + 49 + 4\alpha^2}{49 + 14\alpha + \alpha^2}, .3607755115, .02200691825, .1727731258, .3607755115, .02200691825, .1727731258 \right\}$$

$$\frac{(2\alpha - 7)^2}{9(\alpha + 7)^2} = 0.3607755115^2 \Rightarrow \alpha(7) = 15.88395686$$

Degré 8

$$A(8) = \begin{bmatrix} \frac{\alpha}{8+\alpha} & (8+\alpha)^{-1} \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 \end{bmatrix}$$

$$B(8) = \begin{bmatrix} \frac{\alpha}{8+\alpha} & (8+\alpha)^{-1} \\ 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 \end{bmatrix}$$

$$S(8) = A(8) \cdot B(8)$$

$$\lambda(8) = \left\{ 0, 1, \frac{4 - 8\alpha + 16 + \alpha^2}{9 \cdot 64 + 16\alpha + \alpha^2}, \frac{2}{9}, \frac{2}{9} + \frac{1}{9}\sqrt{2}, \frac{2}{9} - \frac{1}{9}\sqrt{2}, \frac{2}{9}, \frac{2}{9} + \frac{1}{9}\sqrt{2}, \frac{2}{9} - \frac{1}{9}\sqrt{2} \right\}$$

$$\frac{(2\alpha - 8)^2}{9(\alpha + 8)^2} = \left(\frac{2}{9} + \frac{1}{9}\sqrt{2} \right)^2 \Rightarrow \alpha(8) = 19.84452688$$

$$B(10) = \begin{bmatrix} \frac{\alpha}{10+\alpha} & (10+\alpha)^{-1} \\ 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 \\ 1/3 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 \end{bmatrix}$$

$$S(10) = A(10) \cdot B(10)$$

$$\lambda(10) = \{0, 1, \frac{4}{9} \frac{-10\alpha + 25 + \alpha^2}{100 + 20\alpha + \alpha^2}, \frac{5}{18} + \frac{1}{18}\sqrt{5}, \frac{5}{18} - \frac{1}{18}\sqrt{5}, \frac{1}{6} + \frac{1}{18}\sqrt{5}, \frac{1}{6} - \frac{1}{18}\sqrt{5}, \frac{5}{18} + \frac{1}{18}\sqrt{5}, \frac{5}{18} - \frac{1}{18}\sqrt{5}, \frac{1}{6} + \frac{1}{18}\sqrt{5}, \frac{1}{6} - \frac{1}{18}\sqrt{5}\}$$

$$\frac{(2\alpha - 10)^2}{9(\alpha + 10)^2} = \left(\frac{5}{18} + \frac{1}{18}\sqrt{5}\right)^2 \Rightarrow \alpha(10) = 27.78391446$$

Table des figures

1.1	La chaîne de création (design), de représentation (structuration, simplification / approximation, codage), de transmission et de visualisation d'une scène tridimensionnelle distribuée. Le plan de ce mémoire suit le chemin emprunté par une scène, depuis le serveur jusqu'au terminal. . .	17
1.2	Le concept de réseau maillé, ou maillage surfacique. Les infographistes se limitent à la frontière géométrique des objets, et approximent la surface d'un objet par un réseau de mailles. De gauche à droite : nous visualisons les arêtes du réseau maillé, les faces et arêtes superposées, les faces seules et un maillage muni d'un matériau.	18
1.3	Principe de reconstruction de maillages à partir d'un nuage de points non organisés, le plus souvent acquis par scanner tridimensionnel à base de laser.	18
1.4	Les radars produisent des cartes d'altitude de terrains (à gauche et au milieu), à partir desquelles une triangulation est reconstruite (à droite).	19
1.5	Des primitives géométriques paramétrables (rayon, longueurs, diamètres, etc...) sont combinées par des opérateurs booléens pour construire des modèles.	19
1.6	Différents matériaux : plastique noir, laiton, cuivre et étain. On remarque une interaction lumière/surface différente suivant le type de matériau défini.	20
1.7	Un exemple du post-traitement le plus couramment utilisé : l'interpolation linéaire de l'ombrage ou lissage de Gouraud. De gauche à droite : le maillage original en mode ligne, en mode face, la version ombrée avec une normale définie par face, puis la version avec interpolation de l'ombrage de Gouraud avec une normale définie par sommet.	21
1.8	La géométrie et la connectivité définissent un maillage. Sur cet exemple la face triangulaire F_1 est définie par l'ensemble ordonné $\{X_1, X_7, X_2\}$, tandis que le sommet X_1 est défini par les coordonnées $\{x_1, y_1, z_1\}$. La normale \vec{n}_{F_1} associée à la face F_1 peut être soit déduite de la géométrie, soit définie explicitement par l'infographiste. Il en est de même pour la normale \vec{n}_{X_5} associée au sommet X_5	21
1.9	L'adjacence est définie entre les éléments de type sommet, face ou arête d'un maillage. Adjacence face / face : deux faces triangulaires sont adjacentes dès qu'elles partagent une arête. Adjacence sommet / faces : un sommet S connaît la liste des faces le contenant. Adjacence sommet / sommet : un sommet S connaît la liste des sommets partageant une arête. Le nombre de sommets voisins d'un sommet est appelé <i>valence</i> ou degré. Adjacence arête / face : une arête connaît la liste des faces qu'elle partage.	22
1.10	Le genre d'une surface correspond au nombre maximal de coupures applicables à une surface tout en la conservant connexe.	23
1.11	Les coins et les arêtes vives appartiennent au groupe des singularités, visuellement pertinentes. La courbure fait également partie des critères à prendre en compte lors du développement des algorithmes.	24
1.12	La chaîne de transmission et de rendu d'une scène 3D.	24

2.1	La suppression d'un sommet entraîne la suppression de ses faces adjacentes, et nécessite de ce fait la retriangulation du trou résultant. A chaque opération élémentaire, un ensemble de n triangles est ainsi remplacé par $n - 2$ triangles, dont l'arrangement conditionne en partie la qualité de l'approximation.	30
2.2	La fusion d'arête agglomère deux sommets et supprime de ce fait un ou deux triangles adjacent(s) à l'arête concernée. Une phase d'optimisation locale ou une heuristique détermine la position optimale du sommet résultant de la fusion.	30
2.3	La demi-fusion d'arête est un cas particulier de la fusion d'arête puisque l'un des sommets est choisi comme accumulateur, tandis que les faces dégénérées par l'opération sont supprimées. Cette opération réduit le degré de liberté offert par la fusion d'arête, c'est à dire la conservation de la position du sommet.	30
2.4	La fusion de face supprime une face en agglomérant deux de ses sommets en un sommet, appelé accumulateur. Deux sommets et au plus quatre faces sont alors supprimées au cours de l'opération. Il en résulte un degré de liberté important : la position du sommet accumulateur, ou le choix de ce dernier si l'on contraint les sommets à conserver leur position initiale.	31
2.5	La fusion d'arêtes généralisée agglomère deux sommets séparés par une distance inférieure à un seuil prédéfini. L'intérêt d'un tel opérateur réside en sa capacité à simplifier par fusion des régions non connexes sur le maillage.	32
2.6	Simplification de Garland-Eckbert appliquée au maillage <i>cow</i> . Respectivement de gauche à droite et de haut en bas : le maillage original (5804 faces) puis les versions géométriquement simplifiées à 2000, 1000, 500, 200 et 100 faces. L'opérateur de fusion d'arête généralisé autorise la fusion de régions disjointes du maillage.	33
2.7	Simplification de Borouchaki (avec l'autorisation de l'auteur). Respectivement de gauche à droite : le maillage original (134000 faces) puis les versions géométriquement simplifiées à 22000, 8000, et 3400 faces. La qualité en forme des triangles est parfaitement contrôlée, ainsi que la distorsion géométrique exprimée en pourcentage de la diagonale de la boîte englobante.	35
2.8	L'opération de fusion de l'arête e agglomère les sommets v_0 et v_1 en un sommet cible v . Les triangles t_7 et t_8 sont supprimés à cette occasion et l'opération de fusion forme des volumes élémentaires signés (deux exemples à droite). Le sommet v est placé de telle sorte que la variation élémentaire de volume soit nulle afin de préserver le volume du maillage au cours de la décimation.	35
2.9	Représentation progressive de Yemez-Schmitt sur la statuette <i>Coignard</i> : le niveau de détail augmente progressivement avec le nombre de particules, lui-même lié au nombre de triangles : 164, 795, 3232, 12914, 51253, et 201792.	37
2.10	Le processus d'optimisation peut être appliqué globalement après une méthode de simplification géométrique ou localement en alternant décimation atomique et optimisation locale. Dans les deux cas l'optimisation fait référence au maillage original.	39
2.11	La déviation géométrique est définie par le volume compris entre les deux surfaces M et \hat{M}	39
2.12	La variation élémentaire de volume correspond au parallélépipède généré par $\delta\hat{M}(u, v)$ sur l'élément de surface correspondant à $dudv$. Une telle variation est signée : négative lorsque le volume décroît et positive dans le cas inverse. Cette constatation nous amène ensuite à remplacer la notion de volume intérieur au modèle \hat{M} par le volume enfermé entre \hat{M} et M	40
2.13	Les fonctions de forme sont déduites à partir d'une interpolation linéaire sur les sommets du triangle. Pour tout point (u, v) à l'intérieur du triangle, exactement trois des valeurs des coefficients sont différents de zéro. Ces valeurs sont les coordonnées barycentriques du point à l'intérieur du triangle. En pratique ces valeurs sont calculées par un rapport de surfaces (voir [ZT91]).	42

2.14 Le support du domaine de λ_i concerne seulement les faces adjacentes du sommet X_i , sur lesquelles on applique une méthode d'échantillonnage pour évaluer l'équation 2.9. 43

2.15 L'expression 2.9 est évaluée par échantillonnage régulier sur les triangles. Un repère d'échantillonnage est localement déterminé par l'arête de longueur maximale dénotée \vec{e}_{max} et la hauteur \vec{h} du triangle. Seuls les points localisés sur le triangle sont retenus pour le calcul (à droite), puis la contribution totale est pondérée par la surface du triangle. 43

2.16 Cette figure décrit le calcul local de l'orientation relative entre M et \hat{M} . Soit X_{uv} un point échantillonné sur une face F adjacente au sommet X_i , on considère une ligne L issue de X_{uv} et de direction le vecteur normal \vec{n} à F . On cherche la plus proche intersection notée X_M entre L et le maillage original M . Si l'intersection est trouvée dans le voisinage de X_{uv} , et que les normales \vec{n} et \vec{n}_M ont la même orientation, on compare alors l'orientation entre \vec{n} et $X_{uv}\vec{X}_M$ afin d'en déduire l'orientation relative locale entre M et \hat{M} , autrement dit le terme η 44

2.17 Le pas du gradient est initialisé par une fraction de la longueur moyenne des arêtes autour de chaque sommet ($\gamma_0 = 0.1$ sur cet exemple). Cette heuristique correspond à une grandeur géométrique locale. 45

2.18 Minimisation de volume entre une sphère et un cube. Deux maillages différents ont été choisis volontairement sur cet exemple afin de mettre en évidence les qualités intrinsèques de la métrique basée volume. On constate sur cet exemple une parfaite approximation de la surface originale, notamment sur les coins et les arêtes vives régulières. Les sommets convergent par oscillation autour de leur optimum local, et l'optimisation s'arrête soit lorsqu'un nombre d'itérations est atteint, soit lorsque les mouvements deviennent infimes par rapport aux pas initiaux du gradient. A condition de conserver une compacité raisonnable sur les triangles, l'optimum est atteint lorsque la solution initiale est valide et lorsque le volume enfermé entre les deux surfaces est nul, ce qui correspond à une parfaite approximation des singularités géométriques. Le sous-optimum serait atteint si le maillage initial ne disposait pas de la géométrie suffisante pour approximer correctement le maillage de référence. 46

2.19 Minimisation de volume entre un rhombicuboctaèdre et une version géométriquement simplifiée par fusion d'arêtes suivant un critère de longueur seul. 10 itérations sont effectuées entre deux maillages successifs sur cet exemple, et la version finale approxime parfaitement le maillage original. 47

2.20 Approximation L_2 entre un rhombicuboctaèdre et sa version géométriquement simplifiée. Le maillage original, puis les versions simplifiées puis optimisées pour différentes pondérations croissantes du terme de ressort associé à la métrique L_2 . On constate un effet de lissage global plus ou moins important en fonction de la pondération du ressort, ainsi que la mauvaise approximation des singularités comme les coins et les arêtes vives. 48

2.21 Le maillage *horse* (20 000 triangles), ses versions respectivement simplifiées et approximées (en haut : 6500 triangles, en bas : 4500 triangles). La phase d'optimisation en position restaure le volume du maillage initial, et on note une amélioration sensible sur la version 4500 triangles, le ventre et le dos voyant leur forme rétablie après la phase de décimation. 48

2.22 L'opération élémentaire de décimation choisie correspond à la fusion d'arête. A chaque décimation élémentaire un sommet et une ou deux faces sont supprimées. Cette opération ne nécessite pas de retriangulation, et correspond à une granularité fine de décimation puisque la résolution peut être choisie au sommet près. On obtient donc deux degrés de liberté à exploiter : l'ordre des arêtes à décimer en priorité et la position du sommet résultant de la fusion. Ce positionnement peut correspondre soit à un pré-positionnement avant optimisation, soit à une pseudo-optimisation qui placerait le sommet sur un optimum probable. 50

2.23	La queue de priorité combine deux critères organisés en une hiérarchie : la courbure locale et une grandeur géométrique (la longueur d'arêtes dans le cas présent). Au départ un seuil définissant la notion de faible courbure est paramétré, et les arêtes de longueur minimale sont décimées en priorité sur le sous-ensemble correspondant. Au cours du processus de décimation, la tolérance sur la courbure est relâchée afin d'atteindre la complexité géométrique visée. Le critère de longueur d'arête diminue la densité géométrique d'une part, et encourage la formation de triangles compacts sur les régions de faible courbure d'autre part.	51
2.24	De gauche à droite et de bas en haut respectivement : le maillage original, la courbure affichée en fausses couleurs, le maillage simplifié uniquement sur les régions de faible courbure, puis le maillage simplifié suivant le critère de longueur d'arête seul.	52
2.25	La phase de pré-positionnement peut être vue comme une sorte de prédiction de la position optimale. Les arêtes vives sont dénombrées autour des sommets d'une arête candidate à la fusion. Si les deux nombres sont différents, le sommet conservé après contraction est placé sur le sommet de plus grand nombre d'arêtes vives (colonne de gauche). Dans le cas inverse, le sommet est placé au milieu de l'arête (colonne de droite). On remarque sur cet exemple que les coins et arêtes vives sont correctement conservés, ainsi que la compacité des triangles situés soit sur une région plane, soit de part et d'autre d'une arête vive régulière.	53
2.26	Décimation du maillage <i>fandisk</i> . De gauche à droite, puis de haut en bas : respectivement 12946, 6500, 4000, 3000, 2600, 2000, 1000, et 800 faces. Temps d'exécution : environ 60 secondes sur un PII 266 MHz. L'algorithme est mis en œuvre par vagues successives de décimation en agissant sur le seuil de courbure. Les vagues de décimation retirent itérativement les arêtes de plus faible longueur sur les régions de faible courbure, tout en relâchant progressivement la tolérance en terme de courbure. La contrainte de courbure est ensuite quasi nulle afin d'atteindre la complexité géométrique souhaitée et de décimer le maillage le long des arêtes vives régulières.	54
2.27	Détail du maillage <i>fandisk</i> au cours de la simplification. On constate que seules les régions de faible courbure sont décimées, sur lesquelles la compacité des triangles est conservée, le critère de longueur d'arête tendant à générer des triangles équilatéraux. Le pré-positionnement des sommets après fusion permet de conserver les coins et de préserver une compacité satisfaisante de part et d'autre des arêtes régulières. Il est à noter que seule une augmentation de la tolérance en terme de courbure permet de décimer par la suite le maillage le long d'une arête vive.	55
3.1	La redondance principale contenue dans un fichier VRML97. Le nombre 6 correspond à la valence moyenne des sommets d'un maillage triangulaire. Sur cet exemple, chaque face f_i adjacente au sommet s_1 fait référence à ce sommet (souligné à droite). Il en résulte une redondance du même ordre pour la représentation de la connectivité.	61
3.2	Dans un train binaire échelonnable bien conçu, les données sont organisées suivant l'ordre décroissant de l'innovation. Idéalement, on devrait ainsi obtenir la meilleure qualité perceptuelle pour une certaine quantité de données, correspondant au fameux <i>ratio débit / distorsion</i>	65
3.3	L'enjeu consiste à transformer un objet 3D sous la forme d'un flux adaptable sur les réseaux. Idéalement, une image peut être reconstruite en tronquant une partie du train binaire. La scalabilité autorise ainsi la double adaptation réseau / terminal.	66
3.4	Un exemple bidimensionnel de codage des positions : la succession du nombre de points compris dans les sous-ensembles est transmise par le codeur, sur une longueur exprimée en nombre de bits. Les occurrences déductibles (donc non transmises) sont écrites entre parenthèses. Le code échelonnable résultant s'exprime sous la forme de la séquence suivante : {7,5,2,0,1,1,2,1,0,1,1,1}, chaque symbole étant codé sur un nombre de bits décroissant.	69

3.5	De gauche à droite et respectivement : une bande de triangles (triangle strip), une bande étoilée (triangle fan) et une bande de triangles généralisée pour le codage : les sommets sont référencés dans l'ordre de parcours sur le graphe, tandis qu'une chaîne de bits de permutation organise le maillage des sommets.	70
3.6	Un exemple de découpage du maillage <i>nefertiti</i> en bandes de triangles. La connectivité représente alors la triangulation des sommets sur chaque bande, la marche en zigzag correspondant au maillage par défaut.	71
3.7	Résultat de la compression géométrique progressive de Khodakovsky et Guskov [KG00]. Les ondelettes de Loop [Loo87] sont utilisées pour la reconstruction, ce qui explique la continuité obtenue sur le niveau de base.	73
3.8	La quantification globale des positions. Les positions de la première figure sont codées sur 2 bits, correspondant ainsi à quatre positions différentes sur chaque coordonnée. Sur la plupart des objets, un niveau de quantification situé entre 10 et 12 bits coïncide avec la notion de codage sans perte visuelle.	75
3.9	Quantification linéaire des résidus différentiels versus quantification globale. On constate qu'il est plus avantageux de quantifier les résidus différentiels sur cet exemple. En pratique, cela est vrai seulement lorsque l'excursion des valeurs résiduelles est inférieure à la taille de la boîte englobante du maillage.	76
3.10	Les algorithmes de codage et de décodage travaillent indépendamment sur les deux types de données (géométrie et connectivité), ce qui garantit la robustesse de l'algorithme.	77
3.11	Plus un maillage est régulier, et plus la courbe de distribution des valences des sommets présente un pic centré sur la valeur moyenne 6, ce qui est propice au codage entropique de ces valeurs. Si on arrive à représenter un maillage seulement sous la forme d'une liste de valences et de quelques données supplémentaires, les gains en compression deviennent très performants.	79
3.12	Une liste de conquête est formée par un ensemble ordonné de sommets successivement adjacents. Elle possède un sommet pivot, qui connaît l'orientation topologique de ses arêtes adjacentes dans le sens anti-horaire. Cette orientation autorise la conquête ordonnée des arêtes adjacentes au pivot.	80
3.13	A gauche : la conquête anti-horaire des arêtes autour de sommets pivots successifs d'une liste anti-horaire aboutit à une expansion de la liste. Au centre : la même conquête sur une liste horaire aboutit à une rétraction de la liste. A droite : le sommet pivot est contextuellement complet lorsque le secteur angulaire orienté formé par les arêtes de la liste est vide. L'arête restante non conquise (en trait double pointillé) ne sera pas conquise par ce pivot.	81
3.14	Conditions nécessaires au codage de la connectivité. Une arête contribue à la formation d'au moins une, et au plus deux faces, le maillage est dit <i>2-manifold</i> . La liste des arêtes reliant les sommets adjacents d'un sommet doit former un cycle, ce qui n'est plus vrai sur les bords ou les trous du maillage.	81
3.15	A gauche : la valence minimale d'un sommet vaut 2 (on fait l'hypothèse qu'un sommet contribue toujours à la construction d'une face), c'est le cas où un sommet se retrouve sur un coin du bord. Après ajout du sommet fantôme et de ses arêtes adjacentes, la valence minimale d'un sommet vaut 3 (au centre). A droite : un sommet fantôme est ajouté au maillage en présence de bords. Comme son nom l'indique, un tel sommet possède le don d'ubiquité, et régularise ainsi tous les sommets appartenant aux bords. D'un point de vue algorithmique, les arêtes et l'adjacence correspondante sont générées pour garantir la fermeture de la liste d'adjacence de ses sommets voisins, tandis qu'aucune face n'est générée. Notons qu'un sommet fantôme possède une valence équivalente à la taille des bords exprimée en nombre de sommets.	82

3.16 L'orientation topologique définie autour du sommet S est déduite de l'information d'orientation des faces et de l'hypothèse de multiplicité d'adjacence des arêtes (le maillage est considéré *2-manifold*). Observons la description des faces adjacentes au sommet s_1 dans le format VRML [vrml]. Pour chaque sommet S concerné, on commence par choisir arbitrairement une face F . Les faces sont définies par trois références aux sommets, et on fait l'hypothèse que l'ordre de description est anti-horaire et cohérent autour du sommet s_1 . Cet ordre va permettre le tri des arêtes adjacentes au sommet s_1 suivant une orientation anti-horaire. Au départ les sommets sont étiquetés comme suit : S_{ref} est le sommet concerné par le tri de sa liste d'arêtes adjacentes, S_{diff} et S_{curr} sont déduits de l'orientation définie sur la face F . Les arêtes e_1 et e_2 sont directement déduites de la construction de la face F , puis l'arête e_3 est déduite du sommet S_{suiv} , trouvé par relation d'adjacence entre les sommets (S_{suiv} est adjacent à S_{ref} et S_{curr} , et différent de S_{diff}). Il peut y avoir ambiguïté lorsque plusieurs sommets S_{suiv} répondent au test d'adjacence (en trait discontinu) ; ce cas survient en présence de sommets de valence 3. Cette ambiguïté est levée par un test sur les faces : on retient le sommet S_{suiv} qui forme une face avec S_{ref} et S_{curr} . En effet, si les sommets S_{ref} , S_{curr} et S_{suiv_2} formaient une face, le maillage ne serait plus *2-manifold*. A l'itération suivante les étiquettes de sommet se décalent suivant l'orientation anti-horaire calculée en cours de marche. Remarquons qu'aucune orientation ne peut être calculée autour d'un sommet fantôme puisqu'aucune face n'est connectée à ce dernier. En pratique ce sommet n'est jamais affecté comme pivot. 83

3.17 Conquête régulière par expansion. [A] Une face germe est sélectionnée sur le maillage, ses trois sommets sont ajoutés à la liste de conquête suivant une orientation anti-horaire. Trois codes {8,6,5} correspondant aux valences respectives des sommets sont générés. Le premier sommet est affecté comme pivot. [B] La liste active commence sa conquête en marchant sur les arêtes libres du pivot dans l'ordre anti-horaire. Le nouveau sommet est inséré avant le pivot dans la liste et deux arêtes sont conquises : code {5}. [C] Code {6}. [D] Code {6}. [E] Code {5}. [F] Code {6}. [G] Code {4}. [H] Le pivot est complet, l'arête reliant les sommets précédant et suivant le pivot, est marquée sans aucun code. [I] Le pivot est supprimé de la liste, le sommet suivant est affecté comme pivot, sauf si un sommet déjà complet se trouve dans la liste. Notez que cette opération de mouvement du pivot permet la génération d'une face par *maillage* sur le décodeur, avec un coût de codage nul. [J] Code {6}. [K] Code {5}, fantôme compris. [L] Code {5}. 85

3.18 Conquête par expansion, et rebouclage de liste. Une liste germe anti-horaire s'étend par conquête des arêtes autour de sommets pivots successifs. Un rebouclage de liste, considéré comme un accident, survient lorsqu'une arête reliant deux sommets de la liste active est conquise. La liste active est alors séparée en deux listes. En pratique le nombre de rebouclage de listes est lié aux variations de densité géométrique sur le maillage, la liste de conquête progressant plus lentement dans les régions denses. Remarquons l'orientation horaire d'une sous-liste déduite d'une séparation d'une liste anti-horaire. 86

- 3.19 Séparation de liste et conquête par rétraction. [A] Supposons une telle configuration de conquête réalisée. La prochaine arête à conquérir suivant l'orientation anti-horaire fait référence à un sommet déjà transmis, ce qui va à l'encontre d'un système idéal qui ne génère qu'une référence par sommet. Il s'agit donc de transmettre au décodeur une information suffisante pour permettre le maillage du pivot avec le sommet déjà transmis. A priori tous les sommets du maillage peuvent être candidats, en pratique la corrélation locale étant élevée, un offset non signé exprimé en nombre d'arêtes et un bit de signe sont générés : codes {split,+,5}. [B] La liste active est séparée en deux listes d'orientation inverse, un code réservé à la séparation est généré et la sous-liste est élevée dans la pile. Le nouveau sommet référencé est affecté comme pivot de la liste active, sauf si un sommet candidat complet se présente. Il en va de même pour la sous-liste qui possède aussi son pivot propre. [C] Code {5}, fantôme compris. [D] Code {5}. Le pivot n'est pas complet au sens strict, pourtant on le considère comme contextuellement complet puisque chaque liste doit conquérir sa région réservée pour garantir une bonne cohérence de l'algorithme. Un code réservé est généré à cet effet. [E] Supposons la conquête de la première liste terminée. La sous-liste est retirée de la pile et devient active. La conquête anti-horaire va se poursuivre autour du pivot, diminuant la liste. [F] Code {6}. [G] Code {4}. Le pivot est complet, l'arête correspondante est marquée, le pivot est supprimé et avance d'un rang. [H] Le pivot est complet. L'arête correspondante est marquée et le pivot n'est pas affecté au sommet suivant, ce qui créerait une séparation en deux listes et donc un code réservé, gage de surcoût. [I] Le sommet complet de la liste est préféré comme candidat au pivot. Cette modification de stratégie par rapport à la méthode originale [TG98] permet de terminer la conquête avec un coût nul. [J] [K] [L] [M] [N] Chaque sommet complet aboutit à la conquête d'une arête, encourageant ainsi la complétude des sommets voisins. [O] La liste active est vide, supprimée de la pile puis la liste suivante devient active. 87
- 3.20 Une liste germe horaire aboutit à une conquête par rétraction puisque la conquête sur les arêtes s'effectue toujours dans le sens anti-horaire, avec l'arête précédant le sommet pivot comme référence (dans la liste de conquête active). La rétraction garantit la conquête intégrale de la région enfermée par la liste horaire, jusqu'à l'obtention d'une liste vide. 88
- 3.21 [A] Supposons une telle conquête réalisée. Le sommet pivot est complet. L'arête reliant les sommets avant et après le pivot est marquée, puis le pivot avance d'un rang dans la liste de conquête, le tout sans aucun code. [B] Le pivot poursuit sa conquête dans le sens anti-horaire, la prochaine arête à conquérir est reliée au sommet fantôme. Notons qu'aucune face n'a été créée entre le fantôme et les bords. [C] Un code réservé {fantôme} est généré. [D] Le pivot est complet, une arête est marquée, ce qui rend le futur pivot complet et diminue le nombre d'arêtes libres du fantôme. [E,F,G,H,I,J,K,L] Chaque marquage d'arêtes successif à l'avancée du sommet pivot a rendu le pivot suivant complet, donc aucun code n'est nécessaire. [M,N] Le pivot est complet, et toutes les arêtes sont déjà marquées, le pivot avance d'un rang. [O,P] Le sommet fantôme est affecté comme pivot. Il s'agit d'un cas particulier puisqu'il est complet. Ce dernier est donc supprimé de la liste et la conquête de la liste active est terminée. A l'inverse un fantôme non complet ne peut être affecté comme pivot puisqu'aucune orientation cohérente ne peut être définie autour de ce dernier : un fantôme unique peut régulariser simultanément plusieurs trous et bords. 89

- 3.22 Une liste de conquête peut contenir plusieurs occurrences de fantômes. [A] Supposons l'état de conquête suivant. Chaque sommet du bord est relié au fantôme. Il en est de même pour chaque sommet de la liste de conquête. Le pivot est complet et avance d'un rang dans la liste en marquant une arête. [B] Le nouveau pivot est complet, avance d'un rang (on fait l'hypothèse qu'aucun sommet complet ne se trouve dans la liste). [C] Le pivot est encadré par deux occurrences du sommet fantôme dans la liste, et avance dans la liste jusqu'à trouver un sommet différent du fantôme (le fantôme ne peut être pivot). [D] plusieurs occurrences du fantôme coexistent dans la liste, ces dernières seront supprimées de la liste après le constat de la complétude du fantôme. 90
- 3.23 Codage de la connectivité du maillage *nefertiti* (562 faces, 299 sommets) par conquête d'arêtes. Les deux premières lignes illustrent la séquence de conquête avec notre technique tandis que les deux lignes suivantes illustrent la technique originale. Le sommet ajouté au centre correspond au sommet fantôme, le sommet germe étant choisi arbitrairement. Au cours du codage, la liste de conquête initiale progresse sur le maillage par conquête des arêtes adjacentes aux sommets pivots successifs. La technique présentée génère une seule séparation de liste et 302 codes de valences pour aboutir à 1.537 bit/face. La technique originale génère quant à elle 34 codes de séparation de liste et 344 codes de valences pour aboutir à 2.121 bit/face. Dans les deux cas le codage est effectué en 30 ms sur Pentium II 266 MHz (soit 18 733 faces/s). 91
- 3.24 L'algorithme de codage de la connectivité réordonne les sommets tout en favorisant la proximité géométrique de deux sommets successifs. A gauche : le maillage original. A droite : un réseau de segments relie les sommets tels qu'ils sont stockés dans le mode représentation avant (en haut), puis après (en bas) le codage de la connectivité. On visualise ainsi l'effet de la conquête sur les arêtes adjacentes aux sommets pivots successifs : on obtient une meilleure cohérence après réordonnement, ce qui incite au codage différentiel ou par prédiction. Sur cet exemple, les lignes reliant les sommets du bord au sommet fantôme ne sont pas représentées, et les grands segments obliques transitent par le sommet fantôme au cours de l'encodage. 92
- 3.25 Maillage *fandisk*. Au cours du codage de la connectivité, les sommets sont réordonnés à partir d'un sommet pivot germe. En traçant un réseau de segments reliant consécutivement tous les sommets dont la position est à coder, on observe des régions régulières et d'autres plus oscillantes. Les longs segments reliant deux régions éloignées du maillage correspondent à des accidents. 93
- 3.26 A gauche, le sommet pivot germe. On visualise au centre la conquête régulière par expansion sur un maillage parfaitement régulier. 93
- 3.27 Quatre modes de prédiction sont employés : parallélogramme, linéaire, différentiel et accident. Considérons la position i codée, la position suivante $i + 1$ est à coder. Pour les trois premiers modes de prédiction, seul le résidu r entre la position prédite et la position réelle du sommet $i + 1$ est à coder. La prédiction parallélogramme nécessite la connaissance des trois antécédents i , $i - 1$ et $i - 2$ et reporte le segment $[i - 2; i - 1]$ à partir du sommet i . La prédiction linéaire nécessite la connaissance des deux antécédents i et $i - 1$ et reporte le segment $[i - 1; i]$ à partir du sommet i . Le mode différentiel ne prédit rien, tandis qu'un accident génère deux résidus successifs et un ratio K entier. 94
- 3.28 Comparaison qualitative entre la quantification globale (repérée par la lettre G) et la quantification des résidus après prédiction (lettre R) entre 2 et 7 bits. On constate un gain sensible dès le premier niveau de quantification par le mode de quantification des résidus, et un codage quasi sans perte visuelle dès le niveau 6, 11 bits étant nécessaires pour le mode de quantification globale. 96

3.29 Maillage *fundisk*. [A][B][C] les distributions des valeurs absolues des résidus après quantification sur 4 bits, des modes de prédiction et des ordres de grandeur des expansions binaires sont non uniformes et donc compressées par codage entropique. [D] Les tailles des plans binaires successifs de raffinement augmentent progressivement avec le gain en précision. Les bits correspondants sont codés indépendamment et sans compression par ordre croissant dans l'expansion. 97

3.30 Maillage *fundisk*. De gauche à droite : 4 niveaux de quantification initiaux, respectivement 2, 3, 4 et 5 bits. La largeur des empilements illustre les quantités de données contenues dans les trains binaires suivants : connectivité, positions après le niveau de quantification initial (Base), premier raffinement R_0 puis chaque plan de raffinement R_1, \dots, R_9 . Les résultats associés (section 3.5.4) sur ce maillage montrent les différentes qualités de reconstruction obtenues après transmission des deux premiers trains binaires, puis des couches binaires successives de raffinement. La taille du premier raffinement R_0 s'explique par la présence dans le train binaire correspondant des ordres de grandeur des expansions binaires après codage entropique. Cette information correspond à l'ensemble des chiffres 0 situés à gauche des premiers 1 rencontrés dans l'expansion binaire (voir tableau précédent). Il est à noter que l'on pourrait obtenir une granularité plus fine en transmettant simplement les plans verticaux de raffinement au prix d'un coût de codage final plus élevé. 98

3.31 Décodage de la connectivité. Les trois premiers codes [8][6][5] permettent de créer une face, et d'informer les sommets de leur valence respective. Le premier sommet est nommé pivot. Les arêtes libres attendent une future connexion. A chaque *maillage* d'un sommet, son nombre d'arêtes vives est décrémenté. Sur cet exemple il reste 6 arêtes libres sur 8 en ce qui concerne le sommet pivot. 100

3.32 Les codes [8][6][5][5][6][6][5][6] permettent un maillage progressif des sommets de manière à reformer les faces originales du maillage. 101

3.33 Aucun code. Le pivot est complet, la face F comprenant P, S_{prev} et S_{suiv} est créée avec un coût nul. Le nouveau pivot est ensuite affecté au sommet suivant, ou au premier sommet complet rencontré dans la liste. 101

- 3.34 Reconstruction et séparation de liste. [A] Supposons une telle reconstruction réalisée. Les sommets affichés connaissent leur valence respective. A chaque code de valence, le décodeur retire un sommet du décodeur des positions. Les arêtes en trait discontinu représentent l'incertitude de connectivité sur les sommets tandis que les arêtes en pointillé symbolisent les arêtes reconstruites. A ce stade, il est impossible de prédire avec quels sommets seront connectées les trois arêtes libres du pivot, tous les sommets possédant des arêtes libres étant candidats à la connexion. Cette incertitude représente précisément l'information de *connectivité*. Le code {split,+5} indique une séparation de liste à la position +5 sur la liste active. [B] Cette information permet de connecter le sommet pivot avec le sommet repéré +5 dans la liste, d'affecter le sommet cible comme pivot de la liste courante et l'ancien comme pivot de la nouvelle liste élevée dans la pile, puis de mailler une face par connexion anti-horaire idoïne. Le pivot possède encore 3 arêtes libres. Rappelons qu'au décodeur ne réside aucune information d'arêtes, seul le nombre d'arêtes libres est affecté à chaque sommet. [C] le code {5} déclenche le maillage d'un nouveau sommet de valence 5 avec la liste active. [D] maillage d'un nouveau sommet de valence 5. Il reste une arête libre au pivot, pourtant un code spécifique le force à se comporter comme un pivot complet, ce qui déclenche la création d'une face par maillage anti-horaire du sommet précédent, suivant et le pivot. La reconstruction se poursuit ainsi jusqu'à la complétude de tous les pivots de la liste courante. [E] La liste générée par séparation devient active. Cette liste orientée dans le sens horaire aboutit à une reconstruction de sa région intérieure. La reconstruction s'effectue toujours dans le sens anti-horaire autour du pivot. [F] Le code {6} déclenche la création d'une face par maillage anti-horaire du sommet précédent le pivot, du nouveau sommet de valence 6 et du pivot. [G] Maillage d'un nouveau sommet de valence 4. [H] Le pivot est complet, ce qui déclenche le maillage d'une face et la réduction du nombre d'arêtes libres associé. [I] Le nouveau pivot est à nouveau complet. [J] Un sommet complet a été détecté dans la liste, il est affecté comme pivot, ce qui évite un code de séparation de liste, un bit de signe et un offset. [K][L][M][N][O] A chaque itération un sommet complet est détecté, maillant ainsi la région restante avec un coût nul. 102
- 3.35 En haut : décodage d'une position avec la prédiction parallélogramme et un résidu r . Le décodeur calcule en permanence la moyenne mobile M des arêtes reconstruites sur une largeur L exprimée en nombre d'arêtes précédant le sommet courant X_i . Le résidu r est signé, mis à l'échelle, multiplié par M puis appliqué au sommet X_p calculé par prédiction parallélogramme à partir des sommets reconstruits X_{i-2} , X_{i-1} et X_i . En bas : décodage d'une position avec une prédiction de type accident, un ratio entier K et deux résidus successifs r_1 et r_2 . Le résidu r_1 est signé, mis à l'échelle et multiplié par le ratio entier K (5 sur cet exemple). On constate une erreur de colinéarité liée à l'erreur de quantification du résidu r_1 , tandis que le second résidu r_2 permet de rattraper l'amplification de l'erreur de quantification provoquée par le ratio K 104
- 3.36 Courbe comparative des coûts de codage de la connectivité. Les coûts obtenus par la technique sont systématiquement inférieurs aux valeurs obtenues par la technique originale développée par Touma et Gotsman. 106
- 3.37 Première ligne : évolution du coût de codage en fonction du nombre de faces (échelle logarithmique). On constate une diminution sensible du coût, d'autant plus rapide sur un maillage régulier tel que le rhombicuboctaèdre (voir figure 3.42). Le nombre de codes de séparation de liste est réduit de manière drastique et n'augmente pas de manière significative avec le nombre de faces. Cette dernière caractéristique s'avère très intéressante et se distingue nettement du comportement de la méthode de Touma et Gotsman. En pratique cette amélioration s'explique par la suppression pure et simple des codes liés aux encoches formées par la conquête (les séparations de listes avec un offset de plus ou moins deux). 107

3.38	Codage de la connectivité sur deux maillages irréguliers (environ 4000 et 15000 faces), générés à partir d'une perturbation aléatoire de la densité.	108
3.39	A gauche : un maillage irrégulier de très haute complexité géométrique (environ 70 000 faces), dont la densité a été amplifiée sur les régions de forte courbure. A droite : un maillage plus régulier (5000 faces).	109
3.40	A gauche : <i>susie</i> , un maillage hautement irrégulier (900 faces) utilisé pour le codage des images. A droite : le maillage <i>fandisk</i> (13 000 faces), représentatif d'une pièce mécanique.	109
3.41	Deux maillages <i>bunny</i> et <i>venus</i> couramment utilisés comme références en compression de maillages triangulaires.	110
3.42	Un rhombicuboctaèdre parfaitement régulier suivant différents niveaux de complexité géométrique (44, 176, 704, 2816, 11264, et 45000 faces). On constate la suppression du plancher précédemment constaté à 0.2 bit/face.	111
3.43	Influence du choix d'une face germe sur le coût de codage du maillage <i>susie</i>	111
3.44	Reconstruction progressive du maillage <i>fandisk</i> pour différents niveaux de quantification initiaux. Respectivement et pour chaque ligne de haut en bas : quantification initialisée à 2, 3, 4 et 5 bits. Chaque colonne correspond respectivement à la reconstruction initiale (quantification des résidus de prédiction), au raffinement lié aux ordres de grandeurs des résidus finaux, puis aux 2 premiers plans binaires de raffinement successifs. Les coûts de codage indiqués comprennent la connectivité (0.53 bit/face) et les positions. On constate l'incidence du niveau de quantification de base sur la qualité finale après reconstruction : il est préférable d' <i>investir</i> quelques bits de plus lors de la quantification initiale (4 versus 2 bits) afin d'obtenir une meilleure reconstruction (voir les deux modèles pour un même coût de 11.30 bits / face).	113
3.45	En haut : maillage <i>fandisk</i> original, pour un coût de codage théorique de 144 bits/face (et 395 bits/face pour le fichier VRML correspondant). En bas, le maillage reconstruit après quantification des résidus de prédiction paramétrée à 6 bits, pour un coût global de 7.79 bits/face, dont 0.53 pour la connectivité.	114
3.46	Maillage <i>nefertiti</i> . De haut en bas : résultats de reconstruction du niveau de base (sans raffinement) pour respectivement 2, 3, 4 et 5 bits de quantification, comparés au maillage original. De gauche à droite : le même maillage affiché en différents modes : maillage, faces + maillage superposé, faces et faces avec lissage. Ces images démontrent qu'en dessous d'un niveau de quantification de 4 bits, la fidélité au maillage est fortement dégradée. Les coûts de codage indiqués comprennent la connectivité (0.66 bit/face) et les positions.	115
3.47	Maillage <i>nefertiti</i> . De haut en bas : résultats de reconstruction progressive pour respectivement 2, 3, 4 et 5 bits de quantification, comparés au maillage original. Les coûts de codage indiqués comprennent la connectivité et les positions.	116
3.48	Maillage <i>bunny</i> . Colonne de gauche : le maillage de base reconstruit avec un coût de codage de 7.86 bits / face pour un niveau de quantification fixé à 4 bits, comparé au maillage original (colonne de droite). La connectivité occupe 1.02 bit / face.	117
3.49	Principe du codage hiérarchique progressif d'une image : l'arbre est codé après détermination de sa profondeur, puis seules les valeurs différentielles de luminance / chrominance et position associées aux nœuds de l'arbre sont transmises. Les valeurs des nœuds associées aux branches non transmises de l'arbre sont alors interpolées.	118
3.50	Procédé de transformation d'un triangle quelconque en une matrice carrée symétrique en trois étapes : sélection d'un triangle de forme quelconque dans la partition du maillage, transformation affine par déformation du triangle en un triangle rectangle isocèle dont les deux plus courts côtés sont de longueur N , et remplissage de la matrice par interpolation bilinéaire. La matrice est ensuite symétrisée afin de permettre le calcul de la DCT.	118

3.51	Principe du codage hybride : interpolation affine et codage de texture.	119
3.52	Une image est codée à partir d'une hiérarchie de maillages triangulaires dont les sommets sont dotés d'attributs de luminance / chrominance. Les régions composées de basses fréquences sont codées par éléments finis tandis que les hautes fréquences (les éléments texturés) sont codés par transformée DCT. Sur cet exemple, le maillage de base est initialisé par un échantillonnage régulier en quinconce sur l'image, et l'opérateur de permutation est appliqué au niveau le plus fin. A gauche : l'image originale et la version après compression avec un ratio de 1/87. Les régions fortement texturées sont reconstruites à partir de coefficients DCT, tandis que les régions lisses par morceaux sont reconstruites par interpolation linéaire des attributs photométriques et colorimétriques associés aux sommets.	120
3.53	Un codage conjoint maillages triangulaires / éléments finis / DCT autorise la représentation échelonnée des textures.	120
3.54	Progressivité obtenue par la technique du <i>zerotree</i> sur les maillages (l'image originale est encadrée). Le décodeur est capable de reconstruire une image à tout moment au cours de la transmission, avec une granularité définie au sommet près.	121
3.55	Codage par régions d'intérêt. En haut à gauche : une région d'intérêt sélectionnée par l'infographiste lors de la préparation d'une texture. A droite : les éléments à coder dans l'arborescence sont déterminés par marquage récursif ascendant des éléments de l'arbre suivant la région sélectionnée. Seuls les éléments marqués sont pris en compte lors du codage, les autres étant simplement interpolés lors de la reconstruction. En bas à gauche : l'utilisateur associe un niveau de qualité (représenté en niveau de gris) à chaque région. Le niveau correspond ensuite à une profondeur du parcours descendant lors du codage. Chaque région ainsi sélectionnée donne lieu à la création d'un train binaire, où seule l'information différentielle entre deux niveaux emboîtés est codée.	122
3.56	Une carte de requête caméra est construite en fonction du point de vue et permet de relier la notion de niveau de qualité à la distance au point de vue. Cette carte est mise en coïncidence avec la hiérarchie de maillages lors de la reconstruction. Seules les régions orientées face à la caméra sont reconstruites par interpolation linéaire des attributs colorimétriques et photométriques affectés aux sommets, ou à partir des coefficients DCT sur les régions texturées. La distance à la caméra représentée en fausses couleurs est amenée à piloter l'algorithme de transmission progressive et de construction adaptative de la texture. La hiérarchie correspond alors à une structure de données hybride compacte susceptible de générer au vol une image lorsqu'une requête caméra lui est adressée.	123
4.1	Disposant du point de vue, la détection de la silhouette est réalisée de manière directe : une arête appartient à la silhouette si elle partage deux faces dont les normales sont d'orientation mutuellement opposées par rapport à la caméra.	127
4.2	De gauche à droite : le point de vue du maillage <i>nefertiti</i> , la silhouette correspondante représentée en couleur claire au centre et la zone d'intérêt correspondante élargie à une bande de triangles (en gris sur l'image de droite).	128
4.3	A gauche : les faces en dehors de la pyramide de vision sont invisibles, et les faces tournant le dos à la caméra ne sont pas affichées par convention avec les infographistes. A droite : la silhouette au sens de la définition donnée plus haut est à différencier des silhouettes visibles, la distinction nécessitant des calculs de masquage complexes.	128
4.4	Les principaux opérateurs de subdivision. Un triangle original, et ses versions subdivisés respectivement par les opérateurs 1 vers 2, 1 vers 3 et 1 vers 4. La subdivision barycentrique des faces ou des arêtes correspond au mode de raffinement le plus répandu.	130

4.5	La surface est reconstruite par approximation (à distinguer de l'interpolation) à partir d'une surface de référence. Sur cet exemple la subdivision de Loop est globale.	131
4.6	Subdivision régulière 1 vers 4 par bisection des arêtes. Le nombre d'éléments est multiplié par 4 à chaque itération. La réversibilité et la robustesse de l'opérateur autorisent une modification au vol de la résolution d'un maillage. Les nouveaux sommets générés par subdivision possèdent une valence 6, générant ainsi un maillage semi-régulier.	132
4.7	Subdivision de Loop sur le maillage <i>nefertiti</i> . L'amélioration visuelle est sensible sur des objets composés de régions courbes car il en résulte une meilleure approximation de la géométrie et surtout des normales, celles-ci étant le support du calcul de l'ombrage. On rappelle que les normales jouent un rôle prépondérant dans la perception (et parfois l'illusion) de la géométrie.	132
4.8	Subdivision de Loop [Loo87] sur le maillage <i>star</i> . On remarque un lissage des discontinuités et l'obtention d'une surface possédant certaines propriétés différentielles après une infinité de subdivision. Plus intuitivement, la notion de continuité est liée à une <i>douceur</i> d'aspect du modèle. Sur cet exemple la surface limite est C^2 continue en chaque point, excepté sur un nombre fini de points appelé sommets extraordinaires (i.e. avec une valence différente de 6 sur le maillage original).	133
4.9	L'opérateur élémentaire de subdivision combine une subdivision barycentrique des triangles et la permutation des arêtes du maillage initial. Chaque triangle est ainsi découpé en trois triangles superposables par l'insertion d'un sommet en son barycentre, puis les arêtes du triangle original sont échangées entre les sommets initiaux et les nouveaux sommets générés par la subdivision.	136
4.10	Le maillage initial, puis trois itérations successives de subdivision barycentrique des triangles. Chaque nouveau sommet inséré lors d'une étape de subdivision possède une valence égale à 3, et augmente d'une unité la valence de chaque sommet adjacent appartenant au maillage initial. Il en résulte la formation de triangles très peu compacts et une forte accumulation des valences sur les sommets initiaux.	137
4.11	La permutation des arêtes du maillage initial opère une redistribution des valences sur le maillage. Sur cet exemple le sommet central possède une valence égale à 5. Après subdivision barycentrique, la valence du sommet initial est multipliée par 2 tandis que tous les nouveaux sommets insérés ont une valence égale à 3. Après permutation localisée des arêtes du maillage initial, 6 sommets possèdent la valence 5. On remarque sur la dernière figure que tous les nouveaux sommets auraient une valence 6 si la permutation d'arêtes n'était pas limitée à la région composée des faces adjacentes au sommet central.	138
4.12	L'opérateur de subdivision choisi, autorise une cohabitation harmonieuse entre une région localement subdivisée et le maillage original. Le raffinement est en pratique très localisé sur la bande de triangles adjacents à l'ensemble des arêtes constituant la silhouette.	138
4.13	Le raffinement est appliqué localement et de manière itérative sur les régions d'intérêt. Sur cet exemple la région d'intérêt est progressivement réduite afin de simuler un raffinement localisé sur une zone appartenant à la silhouette. Plus intuitivement, il s'agit d'opérer le <i>branchement</i> entre une région très dense du maillage raffiné avec le maillage initial. La faible granularité de l'opérateur de subdivision (facteur 3) et l'opérateur de permutation d'arêtes sont à l'origine du raffinement progressif d'une région éparse vers une région dense.	139
4.14	Masque de pondération en position pour un sommet de valence $n = 5$. L'enjeu consiste à déterminer les valeurs $\alpha(n)$ pour chaque valence afin d'obtenir des contraintes différentielles sur la surface après une infinité de subdivisions d'un maillage de contrôle irrégulier. Sur le terminal, le maillage de contrôle correspond au modèle décodé au cours de la transmission.	139

- 4.15 Matrice A : une itération de subdivision 1 vers 3 orientée dans le sens anti-horaire. Matrice B : une itération de subdivision 1 vers 3 orientée dans le sens horaire. La matrice globale $S = A * B$ permet d'obtenir un schéma auto-répliatif nécessaire avant toute analyse spectrale. Il est à noter que l'application de deux matrices de subdivision dans le sens horaire aurait nécessité l'application d'une matrice de permutation, inutile dans le cas présent. 140
- 4.16 En haut, et respectivement de gauche à droite : maillage initial, première itération de subdivision hybride avec filtrage des positions, seconde itération avec filtrage, et détail sur la silhouette d'un point de vue différent. En bas : même descriptif mais sans filtrage des positions. On constate dans ce cas les artefacts liés à la permutation d'arêtes. 142
- 4.17 Le modèle de base (20 faces) correspond à un ensemble de sommets de contrôle et d'une connectivité définie sur les sommets. Sur cet exemple, la technique de subdivision est comparée avec la subdivision de Loop. Dans les deux cas le modèle converge vers une surface C_2 continue sur les sommets de valence 6 (les sommets ordinaires) et C_1 continue ailleurs (les sommets extraordinaires). Chaque nouveau sommet inséré possédant la valence 6, on peut en conclure que la surface finale est C_1 continue sur un nombre fini de points. Après 6 itérations de subdivision globale, l'algorithme 1 vers 4 atteint 81920 faces via 6 itérations en 7.711 s sur un Pentium II 266 MHz (soit 10623 faces/s) tandis que la méthode 1 vers 3 atteint 14580 faces en 2.018 s (soit 7224 faces/s). Cette différence de vitesse s'explique par la permutation d'arêtes et la mise à jour des adjacences dans le cas 1 vers 3. 142
- 4.18 La subdivision non contrainte d'un maillage comportant des singularités géométriques peut entraîner un lissage indésirable. Sur cet exemple, les coins et les arêtes vives sont lissés par la combinaison du filtrage de la géométrie et de la permutation d'arêtes. 143
- 4.19 [A] le maillage initial. Une classification des sommets est opérée suivant leur nombre d'arêtes vives adjacentes. Un sommet appartient à une arête vive s'il possède exactement deux arêtes vives adjacentes, un coin possède au moins trois arêtes vives adjacentes et un sommet lisse réside sur une région quasi plane en l'absence d'arête vive. Les sommets de dards possèdent exactement une arête vive adjacente et sont classés avec les sommets lisses en l'état actuel des développements. [B] la subdivision barycentrique s'opère simultanément avec un filtrage systématique des sommets, excepté sur les coins ou les arêtes vives. [C] La permutation d'arête n'est pas systématique. Remarquons que l'arête vive e_2 n'est pas permutée, à l'inverse de l'arête e_1 non vive formée par deux sommets possédant pourtant deux arêtes vives adjacentes. 145
- 4.20 Sur cet exemple les positions des sommets situés sur les arêtes vives et les coins ne sont pas filtrées, et les arêtes vives (en gras) ne sont pas permutées. L'absence de permutation d'arête localisée sur les arêtes vives aboutit à une dégénérescence des éléments composant les triangles adjacents à une arête vive. On remarque également un aspect polygonal des arêtes après plusieurs itérations de subdivision globale du maillage. L'enjeu consiste maintenant à reconstruire les arêtes vives régulières par subdivision et filtrage anisotrope. 146
- 4.21 la subdivision des courbes et des surfaces constitue l'outil de prédilection pour la reconstruction. Le lissage adaptatif des maillages est ainsi obtenu par la subdivision barycentrique des arêtes vives, associée à un filtrage anisotrope des positions des sommets la composant. La méthode s'hybride ainsi avec la technique de subdivision régulière un vers quatre dans sa version adaptative [HDD⁺94]. 147

4.22 A gauche : le maillage initial. La subdivision barycentrique de l'arête vive régulière est définie par le masque de pondération B appliqué avant le filtrage des sommets initiaux. Le filtrage du sommet sur l'arête est donné par le masque de pondération A , qui annule l'influence des sommets situés de part et d'autre d'une arête vive. Ainsi la courbe formée par les arêtes vives initiales est lissée par filtrage de ses points de contrôle représentés par les sommets situés sur l'arête vive. Le masque de filtrage d'un coin C illustre l'absence de déplacement par annulation de toute influence ; ainsi les coins sont conservés au cours de la reconstruction. 148

4.23 Sur cet exemple, deux itérations successives de subdivision adaptative sont appliquées sur un maillage comportant des coins, des arêtes vives et des régions courbes. On constate à droite la conservation des arêtes vives tout en obtenant un lissage des arêtes. 148

4.24 Deux itérations successives de subdivision barycentrique et permutation d'arêtes correspondent à une subdivision un vers neuf régulière, associée à une trisection des arêtes du maillage original. En pratique, les arêtes vives sont subdivisées à chaque itération par bissection afin de conserver une granularité unitaire de raffinement. 149

4.25 Interpolation conjointe géométrie / normales. Première ligne : la subdivision barycentrique se décompose en plusieurs étapes : insertion d'un sommet S_c au centre de la face formée par S_1, S_2 et S_3 , modification de la face (le sommet S_c remplace S_1), création de deux faces $\{S_1, S_c, S_2\}$ et $\{S_1, S_3, S_c\}$, et filtrage des positions ($\{S_1, S_2, S_3\}$ deviennent $\{S'_1, S'_2, S'_3\}$). Le calcul des positions finales donne également lieu à un calcul de normales cibles. Seconde ligne : à l'instant $t = 0$ les nouveaux éléments (faces et sommets) sont générés tandis que le sommet S_c (dénomé S_t temporairement le temps de l'interpolation) est superposé sur S_1 . Les deux nouvelles faces créées sont donc dégénérées car de surface nulle. La continuité est ensuite obtenue par interpolation linéaire conjointe géométrie / normale entre les valeurs initiales et les valeurs cibles. L'interpolation des normales peut s'effectuer sous une forme linéaire ou suivant tout autre principe plus sophistiqué comme l'interpolation le long d'une géodésique sur la sphère unité. Notons qu'une telle interpolation ne résout toutefois pas la continuité visuelle de la permutation d'arêtes. . . . 151

4.26 Les nouvelles positions et la permutation d'arêtes donnent lieu à un nouveau calcul de normales cibles. Lors de l'interpolation, les normales ne sont volontairement pas en accord avec la géométrie afin de rendre imperceptible la permutation opérée lors de la dernière étape de l'interpolation (entre $t = 0.9$ et $t = 1$ sur cet exemple). 152

4.27 Les deux premières lignes illustrent le principe de l'interpolation géométrique au cours du raffinement géométrique par subdivision barycentrique un vers trois des faces originales. Pour chaque face originale, deux faces dégénérées (i.e. de surface de nulle) sont créées. En gras : une arête adjacente à deux faces adjacentes et dégénérées, mise en évidence après permutation. La permutation d'une telle arête est imperceptible lorsqu'elle est effectuée en position dégénérée. La piste consiste alors à permuter initialement et systématiquement les arêtes du maillage original afin de *masquer* cette opération (troisième et quatrième ligne). En pratique, cette opération est impossible puisqu'elle nécessiterait la présence de deux faces dégénérées adjacentes à chaque arête originale, soit un maillage entièrement dégénéré. Cette constatation explique la présence de permutations visibles dès la première itération d'interpolation (en pointillé), et renforce le choix d'une application partielle de la technique, combinée à une interpolation conjointe géométrie / normales. 153

4.28 La progressivité de l'opérateur de subdivision utilisé est obtenue de manière correcte sur les arêtes adjacentes à deux faces dégénérées. La face mise en évidence est subdivisée en son centre, tandis que les arêtes du maillage original sont systématiquement permutes avant l'interpolation. 154

- 4.29 Première ligne : la permutation d'arêtes peut être interpolée par une transformation des triangles en quadrilatères due au déplacement progressif le long des arêtes des sommets S_1 et S_2 . En pratique, il est nécessaire de générer une copie des sommets afin d'éviter la dégénérescence des triangles au cours de l'interpolation. Seconde ligne : les quadrilatères sont triangulés par deux faces additionnelles se substituant aux faces originales à la dernière étape de l'interpolation. 155
- 4.30 La subdivision barycentrique d'une face génère 4 actions élémentaires sur les faces et les sommets. La première action (labellisée par l'instant 1) concerne la face F_1 et signifie que son sommet indexé 1 pointait sur le sommet V_i avant la subdivision. De cette manière, il sera possible de restaurer cette face dans sa position initiale. Les faces F_2, F_3 et le sommet V_n sont ajoutés au maillage courant et se voient simultanément affectés d'une action de type création. Ainsi l'application antichronologique de la première tranche d'historique labellisée 1 supprimera ces éléments de la mémoire. Le filtrage des positions des sommets originaux se caractérise par l'ajout d'une action de type mouvement sur ces éléments. 157
- 4.31 La permutation d'arête adjacente à deux faces se caractérise simplement par l'ajout de deux actions élémentaires de type *affectation*. Sur cet exemple, le sommet indexé 0 de la face F_4 pointait sur V_j , tandis que le second sommet de la face F_1 pointait sur V_k 157
- 4.32 A gauche : le maillage initial est successivement raffiné par la technique de subdivision de Charles Loop [Loo87]. Le nombre de triangles est multiplié par 4 à chaque itération pour aboutir à la progression suivante : 838, 3352, 13408 et 53632 faces. A droite : l'opérateur de subdivision 1 vers 3 combiné à l'opérateur de permutation d'arêtes : 838, 2514, 7542 et 22626 faces. Une telle subdivision est intrinsèquement échelonnable et possède une granularité avantageuse, notamment pour l'adaptation à la puissance d'un terminal graphique. 160
- 4.33 Statistiques liées à la figure 4.32. Les temps de calcul sont mesurés sur un PII 266 MHz. Le maillage *mannequin* est raffiné par subdivision globale 1 vers 3 ou 1 vers 4. On constate une granularité plus fine avec le raffinement 1 vers 3, mais une vitesse de raffinement inférieure principalement liée à l'opérateur de permutation d'arêtes nécessitant une mise à jour locale de l'adjacence. La vitesse ne correspond toutefois pas au critère prioritaire dans le cas présent puisque le raffinement 1 vers 3 permet d'obtenir une qualité de rendu équivalente dès la quatrième itération de subdivision globale, soit après 6.8 s contre 14.474 s pour la subdivision de Loop. 161
- 4.34 La qualité du rendu est sensiblement améliorée par la technique de subdivision globale sur les objets composés de surfaces courbes. Toutefois les singularités géométriques telles que les coins et les arêtes vives sont dégradées par le filtrage des positions et la permutation systématique des arêtes. 162
- 4.35 Maillage *venus*. 6 itérations de subdivision sont successivement appliquées sur la région composée des faces adjacentes aux arêtes de la silhouette. A gauche : l'aspect polygonal de la silhouette est supprimé par le raffinement localisé, et l'opérateur choisi autorise une cohabitation harmonieuse des niveaux de résolution sur le maillage. La complexité géométrique augmente de 1396 à 21544 triangles en 3,14 s. La détection de la silhouette est effectuée après chaque subdivision, les faces candidates à la silhouette étant limitées à la zone de silhouette de la précédente détection afin d'accélérer les temps de calcul. 163
- 4.36 Maillage *nefertiti*. 4 itérations de raffinement sont itérativement appliquées sur la silhouette du modèle. Colonne 2 : la bande formée par les triangles adjacents aux sommets des arêtes de la silhouette se rétrécit à chaque raffinement par subdivision. Colonne 3 : raffinement localisé résultant de la détection précédente de la silhouette. Colonne 4 : la densité géométrique augmente progressivement sur la région d'intérêt. 164

- 4.37 Maillage *mannequin*. En haut et de gauche à droite : le maillage original, sa version raffinée sur la silhouette et les régions orientées face à la caméra pour le point de vue courant et le maillage raffiné tracé sous un point de vue différent afin de mettre en évidence la densité obtenue sur la silhouette. Le nombre de faces varie de 838 à 43294 faces en 6,4 s lorsque trois itérations de subdivision sont appliquées sur les régions face à la caméra. En bas à gauche : le résultat final après lissage de Gouraud : l'aspect polygonal des silhouettes a totalement disparu, il en est de même en cas de zoom sur une silhouette puisque le raffinement est obtenu jusqu'à l'obtention d'une longueur d'arête projetée inférieure à la résolution de l'écran. En moyenne trois itérations additionnelles sont appliquées sur la silhouette pour cet exemple, totalisant ainsi 6 itérations de subdivision très localisées. A titre de comparaison, 6 itérations de subdivision globale équivalentes aboutissent à un maillage comportant environ 600 000 faces, pour une qualité d'image équivalente. A droite : le détail d'une silhouette localisée sur le lobe de l'oreille et le maillage raffiné vu de dos. 165
- 4.38 Ligne 1 : le maillage *venus* est raffiné sur la silhouette et les régions orientées face à la caméra. Le sous-ensemble d'intérêt correspond aux faces incluses dans la pyramide de vision définie par les coins de la fenêtre de visualisation et la caméra virtuelle. Ligne 2 : la région d'intérêt est explicitement définie par l'utilisateur sous la forme d'une pyramide. Notez que tout autre mode de sélection pourrait être utilisé. Ligne 3 : le maillage *nefertiti* avant puis après raffinement adaptatif, appliqué sur la région d'intérêt (mise en évidence sur la dernière colonne). La forme de la zone d'intérêt est liée à la pyramide de vision déduite du mode de projection perspective utilisé par le point de vue (première colonne). 166
- 4.39 Le maillage initial comporte des arêtes vives régulières, des coins et des régions lisses. L'algorithme adaptatif autorise la reconstruction de la surface et des singularités géométriques. Ligne 1, et respectivement de gauche à droite : maillage initial, détection des arêtes vives, aspect polygonal des arêtes, et classification des sommets suivant leur nombre d'arêtes vives adjacentes (0 : sommet lisse, ≥ 3 : coin, 2 : sommet situé sur une arête vive régulière). Colonne 1, et respectivement verticalement : trois itérations successives de raffinement adaptatif. Colonne 2 : détail sur l'approximation de l'arête vive de la boucle du caractère *a* par filtrage anisotrope (l'aspect polygonal de l'arête est supprimé dans une direction et sa densité augmente plus rapidement que les régions lisses). Ligne 2, colonnes 3 et 4 : superposition de l'arête vive initiale et de la surface après approximation. En bas à droite : la surface finale après reconstruction. . . 167
- 4.40 [A] Le maillage original (854 faces). [B,C] L'ordre initial sur les sommets [D,E,F] L'ordre sur les sommets après codage de la connectivité. Le pivot germe est représenté sur l'image E. On note à droite la cohérence géométrique obtenue. [G,H,I] Les niveaux de base après quantification de la géométrie avec respectivement 2, 3 et 4 bits. [J,K,L] Les résultats obtenus après trois itérations de raffinement global. 168
- 4.41 [A] Le maillage *mannequin* avant codage. [B] La géométrie du maillage a été progressivement codée avec un niveau de quantification initialisé à 3 bits (le niveau de base). Les raffinements par plans de bits sont superposés en mode maille. [C] Les sphères d'erreur correspondent à la quantification du niveau de base. Le niveau de base après décodage : 6.8 kbit au total, soit un ratio de 41.6 par rapport au fichier vrmf correspondant. [D] Le niveau de base [E,F,G,H] Raffinement progressif de la géométrie par intégration des ordres de grandeur et des plans de bits : respectivement 6.8, 10.8, 11.5 et 13.5 kbit. [I,J,K,L] après une itération de raffinement global. [M,N,O,P] après trois itérations de reconstruction globale et lissage de Gouraud activé. 169

- 4.42 Les sommets sont classés par type suivant le nombre d'arêtes vives adjacentes (lisse, dard, pli, ou coin) puis la méthode de reconstruction garantit une C_2 continuité sur les sommets ordinaires de l'arête vive, et une C_1 continuité ailleurs. Un sommet ordinaire possède deux arêtes vives adjacentes, et exactement deux arêtes lisses de part et d'autre du pli formé par l'arête (voir [HDD⁺94]). 171
- 4.43 DeRose et al [DKT98] définissent la notion d'arête semi-vive par le biais d'une caractérisation entière. Une arête est caractérisée une fois pour toutes sur le maillage de contrôle par sa *dureté* exprimée en nombre d'itérations κ avant filtrage. Sur cet exemple, les arêtes vives subissent globalement une subdivision barycentrique sur les κ premières itérations tandis que le maillage est raffiné par subdivision barycentrique des faces et permutation d'arêtes ensuite. Les auteurs [DKT98] proposent également la définition d'un κ scalaire par interpolation linéaire de la géométrie reconstruite entre les deux itérations encadrant cette valeur. L'optimisation avant codage pourrait ainsi concerner le κ propre à chaque arête du maillage de base, quelques valeurs s'avérant suffisantes pour reconstruire fidèlement la géométrie (6 itérations de subdivision correspondant à une multiplication du nombre de faces par 729 dans le cas présent), typ. $\{\infty, 0, 0.5, 1, 2, 3, 4, 5\}$ 172
- 5.1 Matrice $A(3)$: une itération de subdivision dans le sens anti-horaire pour un sommet de valence 3. Matrice $B(3)$: une itération de subdivision dans le sens horaire afin d'éviter la définition d'une matrice de permutation. La matrice globale $S(3) = A(3) \cdot B(3)$ permet d'obtenir un schéma auto-répliatif nécessaire avant toute analyse spectrale. 206
- 5.2 Matrice $A(4)$: une itération de subdivision dans le sens anti-horaire pour un sommet de valence 4. Matrice $B(4)$: une itération de subdivision dans le sens horaire. La matrice globale $S(4) = A(4) \cdot B(4)$ permet d'obtenir un schéma auto-répliatif nécessaire avant toute analyse spectrale. 206

Liste des tableaux

3.1	Distribution des valences des maillages représentés par la figure 3.11. On remarque une dispersion autour de la valeur 6, qui correspond à la valence moyenne d'un maillage triangulaire. . .	78
3.2	Coût mémoire pour l'encodage. Les hypothèses suivantes sont émises : l'adjacence quasi complète est nécessaire pour l'encodage, la valence moyenne d'un sommet vaut six, et chaque face possède au plus trois faces adjacentes. Les références sont représentées sous la forme de points stockés sur 32 bits, à l'instar des coordonnées de position.	84
3.3	Expansion binaire de nombres normalisés dans l'intervalle unité. L'ordre du premier 1 rencontré dans le train binaire de chaque nombre est appelé ordre de grandeur. Les bits suivants correspondent à des bits de raffinement équiprobables. La taille des plans verticaux de raffinement augmente en même temps que la précision souhaitée. Il s'avère ainsi possible de reconstruire un nombre compris dans l'intervalle unité à partir de son ordre de grandeur et des bits de raffinements.	95
3.4	Décodage des raffinements : chaque ordre de grandeur permet de reconstruire un train binaire. Le bit 1 est ajouté directement après le bit de l'ordre de grandeur afin de favoriser le milieu de l'intervalle plutôt qu'une extrémité. Le bit réceptionné suivant peut remettre en cause ce choix.	105
3.5	Résultats du codage de la connectivité. La modification de la stratégie de conquête réduit parfois de manière drastique le nombre de codes de séparation de liste (colonnes 3 et 4) par rapport à la technique originale de Touma et Gotsman [TG98] et la complexité algorithmique associée (séparation, élévation d'une nouvelle liste dans la pile, etc...). Un tel code générant également un bit de signe et un offset, le gain en terme de coût de codage est sensible (voir colonnes 5, 6 et 7 et courbe 3.36).	106
3.6	Vitesse de codage / décodage de la connectivité.	108
3.7	Coûts mémoire du codage / décodage de la connectivité.	112
4.1	Coefficients de filtrage de la géométrie calculés pour les valences les plus répandues.	141

