



Modélisation, reconstruction et animation de personnages virtuels 3D à partir de dessins manuels 2D

Olfa Triki-Bchir

► To cite this version:

Olfa Triki-Bchir. Modélisation, reconstruction et animation de personnages virtuels 3D à partir de dessins manuels 2D. Informatique [cs]. Université René Descartes - Paris V, 2005. Français. NNT : . tel-00273248

HAL Id: tel-00273248

<https://theses.hal.science/tel-00273248>

Submitted on 14 Apr 2008

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.

UNIVERSITE RENE DESCARTES - PARIS V
Centre Universitaire des Saints-Pères
UFR DE MATHEMATIQUES ET INFORMATIQUE

*Thèse présentée en vue de l'obtention du grade de Docteur
de l'Université RENE DESCARTES - PARIS V*

Discipline : Sciences de la Vie et de la Matière
Spécialité : Mathématiques et Informatique

Par Olfa TRIKI BCHIR

Sujet de la thèse :

**Modélisation, reconstruction et animation
de personnages virtuels 3D
à partir de dessins manuels 2D**

Soutenue le 11 Octobre 2005, devant le jury composé de :

Monsieur le Professeur	Georges STAMON	Président
Monsieur le Maître de conférences	Henri HUDRISIER	Rapporteur
Monsieur le Maître de conférences	Hamid RAHBAR	Rapporteur
Madame le Professeur	Françoise PRÊTEUX	Directeur de thèse
Madame le Professeur	Christine GRAFFIGNE	Examineur
Monsieur le Maître de conférences	Titus ZAHARIA	Examineur
Monsieur	Olivier MARRE	Examineur

Remerciements

Je souhaite tout d'abord exprimer mes respectueux remerciements à Madame le Professeur Françoise Prêteux, responsable de l'Unité de Projets ARTEMIS de l'INT et directeur de cette thèse pour m'avoir accueillie dans son équipe et m'avoir ainsi permis de réaliser ce travail passionnant. Qu'elle soit assurée de ma profonde reconnaissance pour la qualité de la formation dont elle m'a fait bénéficier et pour ses conseils avisés.

Je tiens à remercier tout particulièrement Monsieur le Docteur Titus Zaharia, Maître de Conférences à l'Unité de Projets ARTEMIS pour m'avoir encadrée avec rigueur tout au long de cette formation doctorale. Ses suggestions scientifiques et recommandations pédagogiques ont toujours été fort précieuses.

A Monsieur Georges Stamon, Professeur à l'Université Paris V, qui me fait l'honneur de présider ce jury, je tiens à adresser mes chaleureux remerciements et à exprimer mon grand respect.

Que Messieurs Henri Hudrisier et Hamid Rahbar, Maîtres de Conférences à l'Université Paris VIII et à l'Université de Caen, rapporteurs de ce mémoire, trouvent ici l'expression de toute ma gratitude pour leur lecture minutieuse et l'intérêt qu'ils ont porté à cette recherche.

A Madame Christine Graffigne, Professeur à l'Université Paris V, je tiens à témoigner tous mes remerciements pour l'aide aimable avec laquelle elle m'a guidée dans le dédale universitaire.

Que Monsieur Olivier Marre, président de l'entreprise Quadaxis et promoteur du projet TOON, soit assuré de mes remerciements les plus chaleureux pour ces longues discussions fructueuses, que nous avons eues tout au long de cette thèse et qui m'ont fait découvrir le monde du dessin animé.

Que Monsieur Alain Tsarafidy de la société Quadraxis, sache combien je lui sais gré de toute l'aide qu'il m'a apportée et qui m'a permis de progresser plus efficacement.

A Nicola Rougon, Catalin Fetita et Marius Preda, Maîtres de Conférences à l'Unité de Projets ARTEMIS, j'exprime mes remerciements pour leur écoute attentive et leur aide stimulante.

Que ceux qui ont partagé mon quotidien avec bonne humeur, les Docteurs Mihai Mitrea, Caroline Petitjean, Son-Minh Tran, Sylvain Chevalier, Mathieu Carré et Frank Lepouliquen, soient ici remerciés bien vivement.

A mes collègues doctorants, Diane Perchet, Perrine Monjaux, Khaled Mamou, Amaury Saragaglia, Antoine Discher et Octavian Folea, au contact desquels je me suis enrichie culturellement et scientifiquement, je dis un grand MERCI.

Que mesdames Nicole Teste et Evelyne Taroni trouvent ici l'expression de mes remerciements.

Enfin, je ne remercierai jamais assez ma famille, mon mari, et mes amis pour leur précieux soutien tout au long de ces années.

A mes parents
A Hedi

Résumé

La production de dessins animés 2D qui suit actuellement un schéma mis en place dans les années 1920 fait intervenir un très grand nombre de compétences humaines. Par opposition à ce mode de travail traditionnel, la production de films 3D, en exploitant les technologies et outils les plus récents de modélisation et d'animation 3D, s'affranchit en bonne partie de cette composante artisanale et vient concurrencer l'industrie du dessin animé 2D en termes de temps et coûts de fabrication.

Les enjeux à relever par l'industrie du dessin animé 2D se posent donc en termes de :

1. Réutilisation des contenus selon le célèbre paradigme du “*Create once, render many*”,
2. Facilité d'échange et transmission des contenus ce qui nécessite de disposer d'un unique format de représentation,
3. Production efficace et économique des contenus requérant alors une animation automatisée par ordinateur.

Dans ce contexte compétitif, cette thèse, réalisée dans le cadre du projet industriel TOON financé par la société Quadraxis avec le support de l'Agence Nationale pour la Valorisation de la Recherche (ANVAR), a pour objectif de contribuer au développement d'une plate-forme de reconstruction, déformation et animation de modèles 3D pour les dessins animés 2D.

Un état de l'art des méthodes et outils contribuant à la reconstruction de modèles 3D et à leur animation est présenté et discuté au regard des contraintes spécifiques des règles de création des dessins animés 2D et de la chaîne de fabrication 2D.

Ayant identifié les verrous technologiques à lever, nos contributions ont porté sur :

- L'élaboration d'une de reconstruction des personnages virtuels 3D à partir de dessins 2D,
- La mise au point d'une méthode de modélisation 3D par surfaces maillées, compatible avec le standard MPEG-4/AFX,

Les développements réalisés, intégrés dans un premier prototype de la plate-forme TOON, montrent un gain en temps de 20% dans la chaîne de production et une complète interopérabilité des applications *via* le standard MPEG-4.

Table des matières

Introduction	i
1 Les chaînes de fabrication des dessins animés : 2D versus 3D	1
1.1 Chaîne traditionnelle de production de dessins animés 2D	2
1.1.1 La pré-production	2
1.1.1.1 L'écriture	3
1.1.1.2 Le <i>design</i>	3
1.1.1.3 Le <i>storyboard</i>	4
1.1.2 La production	4
1.1.2.1 Le décor	5
1.1.2.2 L'animation	5
1.1.2.3 La mise en couleur	6
1.1.3 La post-production	7
1.2 Chaîne actuelle de production 2D	7
1.2.1 Méthodes 2D	10
1.2.2 Méthodes 2,5D	13
1.2.3 Méthodes 3D	13
1.2.4 Conclusion	14
1.3 Chaîne de production de dessins animés 3D	15
1.3.1 La pré-production	15
1.3.2 La production	15
1.3.3 La post-production	16
1.4 Discussion	17
2 Modélisation 3D	21
2.1 Représentation par maillages polygonaux	22
2.1.1 Techniques de construction de maillages 3D	23
2.1.2 Conclusion	24
2.2 Surfaces implicites	24
2.2.1 Surfaces de squelette	25
2.2.1.1 Primitives ponctuelles	26
2.2.1.2 Surfaces de distance	28
2.2.1.3 Surfaces de convolution	29
2.2.2 Techniques de visualisation	29
2.2.3 Techniques de modélisation	31
2.2.4 Conclusion	32
2.3 Surfaces paramétriques	32
2.3.1 Surfaces de Bézier	33
2.3.2 Surfaces B-splines	33
2.3.3 Surfaces NURBS	35
2.3.4 Surfaces paramétriques par morceaux (<i>patches</i> paramétriques)	35
2.3.5 Trimmed NURBS	37

2.3.6	Conclusion	39
2.4	Surfaces de subdivision	39
2.4.1	Schéma de subdivision de Catmull-Clark	41
2.4.2	Schéma de subdivision de Doo-Sabin	43
2.4.3	Schéma de subdivision de Loop	43
2.4.4	Conclusion	44
2.5	Conclusion	46
3	Animation 3D de personnages virtuels	49
3.1	Modèles descriptifs	50
3.1.1	Modèles de déformation non rigide	50
3.1.1.1	Déformations de formes libres	51
3.1.1.2	Déformations par contrôleurs	53
3.1.2	Modèles par squelette cinématique	56
3.1.2.1	Cinématique directe	57
3.1.2.2	Cinématique inverse	57
3.1.2.3	Capture de mouvements	58
3.1.3	Techniques d'interpolation temporelle	59
3.1.3.1	Interpolation linéaire	60
3.1.3.2	Interpolation curviligne	60
3.1.3.3	Interpolation de paramètres de pose par quaternions	61
3.1.3.4	Techniques de métamorphose	62
3.1.4	Conclusion	63
3.2	Modèles générateurs	63
3.2.1	Méthodes procédurales	63
3.2.2	Modèles physiques	63
3.2.2.1	Modèles dynamiques	64
3.2.2.2	Systèmes de particules	64
3.2.3	Méthodes comportementales	64
3.2.4	Conclusion	65
3.3	Standards pour l'animation 3D	65
3.3.1	H-ANIM de VRML	65
3.3.2	FBA de MPEG-4	65
3.3.3	SMS de MPEG-4	67
3.3.3.1	La couche squelette	69
3.3.3.2	La couche muscle	70
3.3.3.3	Conclusion	71
3.4	Conclusion	71
4	Reconstruction 2D/3D de personnages virtuels	75
4.1	Reconstruction de modèles 3D à partir de dessins manuels 2D : état de l'art	76
4.2	Reconstruction de l'enveloppe visuelle	80
4.2.1	Implantation	81
4.2.2	Reconstruction volumique à partir de données de synthèse	83
4.2.3	Reconstruction volumique à partir de données réelles	84
4.3	Reconstruction 2D/3D par surfaces NURBS	86
4.3.1	Approche par interpolation	88
4.3.1.1	Détection des points de contrôle des coupes	88
4.3.1.2	Calcul des vecteurs de nœuds	89
4.3.1.3	Uniformisation des vecteurs de nœuds	90
4.3.1.4	Interpolation des courbes de section	91
4.3.1.5	Résultats expérimentaux	94

4.3.2	Approche par approximation	98
4.3.2.1	Harmonisation du nombre de sommets	98
4.3.2.2	Définition des points de contrôle	99
4.3.2.3	Résultats expérimentaux	100
4.4	Génération d'un maillage <i>seamless</i>	104
4.4.1	Critères d'assemblage des parties du modèle	105
4.4.2	Connexion hiérarchique	105
4.4.3	Connexion par <i>Rolling Ball</i>	108
4.5	Reconstruction 2D/3D par maillages	112
4.5.1	Assemblage volumique	112
4.5.2	Lissage des coupes	116
4.5.3	Triangulation	116
4.5.4	Résultats expérimentaux	117
4.6	Conclusion	119
5	Déformation 2D/3D de surfaces NURBS	121
5.1	Déformation sous contraintes 3D	122
5.2	Déformation sous contraintes 2D	126
5.2.1	Détermination des points sources 3D	128
5.2.2	Détermination des contraintes de projection	129
5.2.3	Résultats expérimentaux	131
5.3	Conclusion	139
	Conclusion et perspectives	141
	Références	143

Table des figures

1.1	Modèle de construction.	3
1.2	Turnaround.	4
1.3	Exemples d'expressions de visage.	4
1.4	Exemple de <i>storyboard</i> (extrait de [Dov00]).	5
1.5	Animation.	6
1.6	Caméra multiplan.	7
1.7	Les phases numériques d'une métamorphose réalisée par Peter Foldes (La Faim, 1974).	9
1.8	Modélisation 3D.	16
1.9	Organisation de la chaîne de production 2D.	18
1.10	Organisation de la chaîne de production 3D.	19
2.1	Modèles maillés.	23
2.2	Surfaces Implicites : Surfaces algébriques.	25
2.3	Exemples de surfaces de squelette.	25
2.4	Combinaison de primitives.	26
2.5	Fonctions de potentiel.	27
2.6	Combinaison de primitives ponctuelles : la fonction de potentiel générée présente un aspect bossu.	28
2.7	Main modélisée par surfaces de convolution [BS91].	30
2.8	Chien modélisé par surfaces de convolution [TZF03].	30
2.9	Définition récursive des fonctions de base B-splines.	34
2.10	Patch de Bezier.	36
2.11	Juxtaposition de patches de Bézier : conditions de continuité.	36
2.12	Visualisation de <i>patches</i> paramétriques : cracking.	37
2.13	Trimming NURBS : découpage de la surface.	37
2.14	Surface de <i>blending</i> définie à partir de courbes de découpage.	38
2.15	Exemple de surface paramétrique.	39
2.16	Subdivision d'une courbe selon l'algorithme de Chaikin.	40
2.17	Exemple de surface de subdivision.	41
2.18	Types de subdivision : Exemple de subdivision primale et duale appliquées à un maillage de base quadrilatéral.	42
2.19	Schéma de subdivision de Catmull-Clark.	42
2.20	Schéma de subdivision de Doo-Sabin.	43
2.21	Schéma de subdivision de Loop.	44
2.22	Subdivision : Effet de rétrécissement.	45
2.23	Influence du maillage de base en fonction du schéma de subdivision appliqué sur l'aspect de la surface finale.	46
2.24	Surface de subdivision : Le personnage Geri de Geri's Game (Pixar).	46
3.1	Déformations globales.	51
3.2	Déformation de formes fibres.	52

3.3	Effets de style en dessins animés.	52
3.4	FFD.	52
3.5	Déformation par EFFD [Coq90].	53
3.6	Déformation géométrique par points de contrôle.	54
3.7	Déformation à l'aide de courbes wires.	55
3.8	Déformation à l'aide de courbes géodésiques.	56
3.9	Animation d'un personnage par cinématique inverse.	58
3.10	Systèmes de capture de mouvements.	59
3.11	Interpolation de poses clefs.	60
3.12	Interpolation de quaternions : interpolation linéaire <i>versus</i> interpolation linéaire sphérique.	61
3.13	Interpolations sphériques des quaternions	62
3.14	Hierarchie d'un corps humain dans la norme MPEG-4.	66
3.15	Standard FBA : Points clefs définissant l'objet visage [MPE].	67
3.16	Modèle d'animation FBA.	67
3.17	Standard FBA : exemple d'utilisation des Tables de Déformation du Corps. . .	68
3.18	Contrôleurs d'animation BBA dans la spécification SMS de MPEG-4 [Pre02]. .	70
3.19	72
4.1	Incohérences de pose entre différentes vues du <i>turnaround</i>	76
4.2	Processus de reconstruction 2D/3D.	81
4.3	Visual Hull par coupe.	82
4.4	Volume d'intersection : Données de synthèse.	83
4.5	Intersection de volumes : Problème d'auto-occultation.	84
4.6	Intersection de volumes : Données réelles.	84
4.7	Enveloppe visuelle reconstruite pour le personnage Titeuf.	85
4.8	Reconstruction volumique à partir de vues segmentées.	86
4.9	Reprojections du modèle volumique de Titeuf	86
4.10	Reconstruction par intersection de volumes pour différents personnages de des- sins animés.	87
4.11	Images des sections horizontales du volume d'intersection obtenu pour le modèle humanoïde.	88
4.12	Détection de contour par codage de Freeman.	89
4.13	Directions discrètes définies par le codage de Freeman.	89
4.14	Points de contrôle extraits pour la coupe présentée Figure 4.12.	89
4.15	Polygone de contrôle et courbe NURBS de degré 3.	90
4.16	Uniformisation du vecteur des nœuds.	92
4.17	Raffinement des nœuds.	92
4.18	Aspect en escalier : l'effet de la discrétisation volumique se répercute sur la positions des points de contrôle et par conséquent sur la surface.	94
4.19	Atténuation de l'effet d'escalier par filtrage passe-bas.	94
4.20	Atténuation de l'effet d'escalier par sous-échantillonnage des coupes.	95
4.21	Approche par interpolation : modèle NURBS de Titeuf.	95
4.22	Reconstruction de surfaces NURBS par interpolation.	96
4.23	Reconstruction par interpolation NURBS : Résultats.	97
4.24	Correspondances entre les sommets de deux courbes polygonales.	99
4.25	Harmonisation du nombre de sommets	99
4.26	Définition des points de contrôle.	100
4.27	Surfaces NURBS reconstruites par approximation : tête, buste, et bras du per- sonnage Elfie.	100
4.28	Approche par approximation : modèle NURBS de la poire, de la chenille, du croissant et de l'avion.	101

4.29	Reconstructions par approximation NURBS : surfaces obtenues avec 3 niveaux de subdivision de Chaikin.	104
4.30	Critère d'assemblage des parties.	106
4.31	Choix des surfaces Parent et Enfant.	106
4.32	Processus de propagation à partir du pôle enfant.	107
4.33	Assemblage : Connexion hiérarchique.	108
4.34	Algorithme de <i>blending</i> de maillages par <i>Rolling Ball</i>	109
4.35	Détermination de la courbe d'intersection.	109
4.36	Procédure de classification des polygones générés par l'intersection.	110
4.37	Remaillage des triangles d'intersection.	111
4.38	Maillage seamless : Résultats.	112
4.39	Connexion du torse et du bras du personnage Elfie par <i>Rolling Ball</i>	113
4.40	Connexion du bras et de la main du personnage Elfie par <i>Rolling Ball</i>	113
4.41	Connexion de la jambe et du pied du personnage Elfie par <i>Rolling Ball</i>	114
4.42	Déconnexion de parties reconstruites en fonction des images d'entrée.	114
4.43	Vue globale de la méthodologie de reconstruction par maillages.	115
4.44	Assemblage des modèles volumiques.	115
4.45	Triangulation Delaunay 2D avec contraintes inter-coupes.	116
4.46	Reconstruction par maillage du personnage Elfie.	117
4.47	Détails du maillage obtenu pour le personnage Elfie.	118
4.48	Influence de la simplification des polygones sur la qualité de la triangulation.	118
5.1	Déformation sous contraintes 3D du nez d'un humanoïde.	125
5.2	Déformation sous contraintes 3D du dos d'un humanoïde.	125
5.3	Contrainte 2D de projection.	126
5.4	Détermination des points sources sur le contour générateur.	129
5.5	Contraintes de déformation	130
5.6	Réajustement du pull du personnage Titeuf selon la vue $\theta = 90^\circ$	131
5.7	Réajustement d'une jambe du personnage Titeuf selon la vue $\theta = 90^\circ$	131
5.8	Réajustement de la main du personnage Titeuf selon les vues $\theta = 0^\circ$ et $\theta = 90^\circ$	132
5.9	Déformation de la tête d'Elfie selon la vue $\theta = 30^\circ$	132
5.10	Déformation du buste D'Elfie selon la vue $\theta = 30^\circ$	133
5.11	Déformation de la tête du personnage Al selon la vue $\theta = 0^\circ$ pour obtenir un effet d'amincissement.	134
5.12	Déformation par contrainte projective : ajout d'oreilles à la tête du personnage Al en modifiant une unique vue projective.	134
5.13	Déformation de l'oreille du personnage Ickis selon la vue $\theta = 0^\circ$ pour lui donner plus d'importance.	134
5.14	Déformation de l'oreille du personnage Ickis selon la vue $\theta = 0^\circ$ pour l'amincir.	135
5.15	Déformation de la poire (croquée) selon la vue $\theta = 0^\circ$	135
5.16	Déformation d'une chaussure selon la vue $\theta = 90^\circ$	135
5.17	Déformation du corps du personnage Oblina selon la vue $\theta = 90^\circ$	136
5.18	Déformation d'une mandoline en un luth selon la vue $\theta = 90^\circ$	136
5.19	Déformation d'un narguilé selon les vues $0, 45, 90^\circ$ et 135	137
5.20	Déformation d'une ellipsoïde en tête selon les vues 0 et 90°	137
5.21	Construction d'une flèche à l'aide de déformations selon les vues 0 et 90°	137
5.22	Déformations d'un vase selon la vue $\theta = 90^\circ$	138
5.23	Procédure de déformation locale.	138
5.24	Déformation du vase pour différents niveaux de subdivision de reconstruction.	139

Liste des tableaux

2.1	<i>Blending</i> de primitives de squelette : apparition de bosses au niveau de la jonction entre deux éléments du squelette.	29
2.2	Caractéristiques des schémas de subdivision de Loop, Catmull-Clark et Doo-Sabin.	44
2.3	Comparaison des modèles de surface.	47
3.1	Tableau comparatif des standards FBA et SMS [Pre02].	71
3.2	Méthodes d'animation.	72

Introduction

“L’animation n’est pas l’art des images qui bougent, mais l’art des mouvements dessinés. Ce qu’il y a entre les images a beaucoup plus d’importance que ce que l’on voit sur l’image. L’animation est par conséquent l’art de se servir des interstices invisibles entre les images.”

Norman McLaren

Depuis quelques années déjà, l’animation de modèles 3D intervient de plus en plus souvent dans des domaines aussi variés et divers que la modélisation physique, la réalité virtuelle, les jeux vidéos, ou les productions cinématographiques. Les modèles de surfaces ainsi que les méthodes d’animation sont de plus en plus utilisés aussi bien au niveau de la communauté scientifique qu’à celui du grand public à travers les applications de divertissement.

Face à ces avancées significatives de la 3D, les dessins animés 2D se plient encore à des règles de fonctionnement et à un schéma de production traditionalistes, qui reposent fortement sur des compétences humaines. Ce mode de fabrication, actuellement en place à l’échelle internationale, requiert un nombre important d’intervenants et implique un temps de réalisation considérable, ce qui représente pour les maisons de production des coûts de fabrication de plus en plus élevés.

En dépit de l’apparition de quelques outils dédiés à l’animation 2D, et de l’amélioration des temps de fabrication, le processus reste entièrement hermétique aux technologies 3D, dépendant des intervenants humains non seulement pour les étapes de création des modèles, mais surtout pour l’étape d’animation réalisée exclusivement en 2D ce qui constitue un véritable goulot d’étranglement.

Dans ce contexte où les outils pour automatiser la production des dessins animés fait défaut, le projet industriel TOON, soutenu par l’Agence Nationale pour la Valorisation de la Recherche Française (ANVAR), a pour objectif de créer une plateforme logicielle unifiée pour la reconstruction, l’animation et la déformation de personnages virtuels, afin d’accélérer le processus de production en introduisant des techniques d’animation 3D.

Cette plateforme doit tenir compte des contraintes d’interactivité 2D, conformément au mode de travail des dessinateurs, et respecter la chaîne de production actuellement en place. En effet, celle-ci impose pour des raisons économiques la séparation géographique et temporelle des tâches, en particulier pour la création des personnages et leur animation. L’étape de reconstruction constitue par conséquent un élément clef pour la plateforme réalisée.

Cette étape de reconstruction s’inscrit dans le cadre de la problématique plus générale de la reconstruction de modèles 3D à partir de données 2D. En fait, les systèmes de reconstruction interactifs qui exploitent en entrée des dessins 2D manuels représentent un domaine de recherche très récent, en particulier dans le contexte de la reconstruction d’objets de forme libre pour des applications telles que le *design* ou la production d’animation.

La difficulté principale réside dans le passage de la 2D vers la 3D, qui représente un problème mathématiquement mal posé. Il est alors souvent nécessaire de développer des

méthodes qui exploitent les propriétés des applications considérées, afin d'en extraire les informations susceptibles d'aider à la reconstruction. Rappelons que dans le projet TOON, le choix des modèles de surfaces utilisés ainsi que des techniques d'animation doit répondre à une contrainte d'interactivité exclusivement 2D.

Le premier chapitre de ce manuscrit présente le contexte dans lequel s'inscrivent les développements que nous avons réalisés. En particulier, une analyse comparée des chaînes de production de dessins animés 2D et 3D est présentée afin de comprendre les raisons du fossé technologique qui sépare ces deux domaines *a priori* très proches et de mettre en évidence les besoins spécifiques des animations 2D en termes d'outils d'automatisation.

Le deuxième chapitre décrit les modèles de surface le plus communément utilisés en les classant suivant leur mode de représentation explicite (surfaces paramétriques) ou implicite (surfaces implicites). L'approximation des surfaces lisses par un ensemble de polygones (maillages et surfaces de subdivision) est également rappelée.

Ces différents modèles sont étudiés par rapport aux contraintes de stockage mémoire et de rapidité d'affichage, ou encore de facilité de manipulation, déformation et d'animation.

Le troisième chapitre est consacré aux principales méthodes d'animation, qui sont classées en méthodes descriptives et méthodes par modèles générateurs. Les modèles de surface auxquels chaque méthode peut être appliquée sont étudiés, ainsi que l'interactivité qu'elle nécessite. Une description des standards d'animation H-ANIM et MPEG-4 est également présentée.

L'étude de ces différentes méthodes d'animation, ainsi que la contrainte de conformité à la norme MPEG-4 nous conduisent à choisir pour modèle de représentation final les maillages pour lesquels des animations articulées conformes à la norme AFX de MPEG-4 peuvent être facilement appliquées. Les étapes de reconstruction et de déformation sont en revanche réalisées à partir de surface paramétriques NURBS qui offrent l'avantage d'une description compacte ainsi que d'un contrôle local.

Dans le quatrième chapitre, différentes méthodes de reconstruction de surfaces 3D, à partir de dessins manuels 2D segmentés sont décrites. Le principe retenu consiste en une première approximation par une enveloppe visuelle, calculée par intersection de volumes. A partir de ce volume, des surfaces NURBS correspondant à chaque partie segmentée du dessin sont générées. Ces surfaces sont ensuite converties en maillages triangulaires dont l'assemblage en un unique maillage *seamless* (sans coutures) est réalisé *a posteriori*. La procédure d'assemblage adoptée offre la possibilité de créer ou non des surfaces de *blending* entre les éléments assemblés.

Toutefois, les temps de calculs correspondant à cette étape d'assemblage sont importants au regard de celui nécessaire à la reconstruction des surfaces segmentées et les maillages obtenus sont denses au niveau des connexions. Par ailleurs, le lissage réalisé par les surfaces NURBS à partir du modèle volumique s'effectue de manière indépendante pour chaque partie. Cela engendre une perte de l'information de connexion entre les éléments. Afin de résoudre ces problèmes, nous proposons une deuxième approche de reconstruction plus directe, qui exploite l'enveloppe visuelle pour générer directement un maillage *seamless*.

Il arrive que les modèles reconstruits ne soient souvent pas exactement conformes aux dessins manuels fournis. Ces différences proviennent d'une part des erreurs de dessins, telles les incohérences de pose et les erreurs de projection, et d'autre part du processus de reconstruction qui implique des étapes de lissage et de sous-échantillonnage. Pour ces raisons, nous introduisons un module de raffinement et de déformation des modèles reconstruits, présenté Chapitre 5.

Enfin, nous concluons cet ensemble de recherche par une synthèse de nos contributions, esquissons quelques voies de recherche futures pour améliorer les performances des méthodes développées et ouvrons des perspectives d'application.

Chapitre 1

Les chaînes de fabrication des dessins animés : 2D versus 3D

Résumé

L'animation traditionnelle 2D et l'animation 3D semblent deux modes de production très différents, faisant appel à des technologies et des compétences distinctes. En effet, face à l'utilisation de plus en plus importante des techniques d'animation 3D dans les productions cinématographiques d'aujourd'hui, le monde des dessins animés reste encore très attaché aux approches d'animation 2D.

Dans ce chapitre, nous traçons une brève comparaison des chaînes de production d'animations 2D et 3D afin d'expliquer le fossé technologique qui les sépare actuellement, et de mettre en évidence les besoins spécifiques aux animations 2D.

L'étude des deux chaînes de production montre que les logiciels et outils actuellement disponibles et utilisés pour les dessins animés 2D ne permettent pas d'accélérer suffisamment le processus de fabrication. En outre, les fonctionnalités offertes restent pauvres par rapport au potentiel des techniques d'imagerie tridimensionnelles.

Toutefois, le mode de production des dessins animés 2D mis en place dans les années 1920 selon un schéma linéaire, utilisé à l'échelle internationale, se prête peu à l'introduction des outils 3D actuellement disponibles sur le marché. Celle-ci nécessiterait non seulement une modification profonde et coûteuse de la chaîne de production, mais aussi l'acquisition de nouvelles compétences de modélisation et d'animation.

Il est par conséquent nécessaire de développer des outils dédiés à l'animation 2D, qui répondent à ses besoins spécifiques tout en prenant en compte les contraintes liées au mode de production en place.

Mots clés

Chaîne de production, Dessins animés, Turnaround, Storyboard, Keyframing, Modélisation 3D, Animation 3D.

“In most instances, the driving forces behind the action is the mood, the personality, the attitude of the character or all three. Therefore, the mind is the pilot. We think of things before the body does them.”

Walt Disney

L’animation traditionnelle 2D et l’animation 3D semblent deux modes de production très différents, faisant appel à des technologies et des compétences distinctes. Il est certes vrai que l’avancée technologique en matière d’outils informatiques a permis la visualisation ainsi que la simulation de personnages virtuels tridimensionnels pouvant même gérer eux-mêmes leurs mouvements selon des lois physiques ou algorithmiques. Cela a suscité plus d’intérêt pour le monde de la 3D. Les maisons de production se sont alors lancées dans des films totalement réalisés en 3D tels que *Toy Story*¹, le premier long métrage entièrement réalisé sur ordinateur, ou encore des films mélangeant des scènes jouées par des acteurs réels avec des personnages virtuels tels que les dinosaures dans *Jurassic Park*², puis dans *Le Monde Perdu*³.

Face à cette utilisation de plus en plus importante des techniques d’animation 3D dans les productions cinématographiques d’aujourd’hui, le monde des dessins animés reste encore très attaché aux approches d’animation 2D. Or, celles-ci s’effectuent selon un schéma mis en place dès les années 1920-1930, impliquant un très grand nombre d’intervenants qui réalisent les dessins à la main, un par un. Malgré la délocalisation de certaines étapes de la fabrication dans le but de diminuer le coût de la main d’œuvre, les coûts de production d’un dessin animé 2D traditionnel continuent d’augmenter. Ce phénomène prend de plus en plus d’importance, notamment avec la place grandissante qu’occupent les animations 3D sur le marché, au point que la question de la survie des films d’animation traditionnels se pose [Eva03].

Afin de mieux appréhender ce phénomène et dans le but de mettre en évidence les besoins spécifiques aux dessins animés en matière d’outils d’automatisation, nous décrivons dans les sections suivantes chacune des deux types de chaîne de production.

1.1 Chaîne traditionnelle de production de dessins animés 2D

La fabrication d’un dessin animé [Tho60, Pon98, Dov00, JLD82, Cot01] est un processus long et complexe, nécessitant un grand nombre d’intervenants aussi divers que dessinateurs et concepteurs de personnages, scénaristes ou animateurs. Afin de gérer ce travail fortement collaboratif et d’assurer sa réussite, la chaîne de production traditionnelle fait l’objet d’une organisation précise dont les bases ont été mises en place dès l’avant-guerre. Généralement, la chaîne de fabrication de dessins animés comporte trois étapes : la pré-production, la production ou fabrication et la post-production.

1.1.1 La pré-production

Cette étape recouvre les phases de conception et de préparation. Elle est fortement créative et constitue la base à partir de laquelle les étapes suivantes seront effectuées. Elle consiste essentiellement en l’écriture de l’histoire, la création des différents modèles ou *design* et la mise en place du *storyboard*.

¹ John Lasseter, 1995

² Steven Spielberg, 1993

³ Steven Spielberg, 1997

1.1.1.1 L'écriture

L'écriture permet de mettre en place l'intrigue racontée, de définir et de caractériser les personnages, de préciser l'espace-temps de l'action, ainsi que les différents décors et dialogues de chaque plan. Le scénario ainsi réalisé constitue une référence pour les étapes suivantes, nécessaire pour garantir la cohérence entre les différentes scènes du film.

1.1.1.2 Le design

A partir du scénario, une liste des créations graphiques à réaliser est dressée. Elle englobe les personnages, les décors et les accessoires. L'étape de *design* consiste à réaliser l'ensemble des documents graphiques qui permettent de définir ces différents éléments de manière à fournir aux intervenants suivants une référence commune. Ces documents doivent donc être les plus précis et les plus nombreux possibles pour faciliter le travail ultérieur et garantir une cohérence des dessins et des animations réalisées.

Parmi ces documents, les *model-sheets* (feuilles de modèle) constituent un élément important puisqu'ils définissent chaque personnage de manière complète. Un *model-sheet* illustre généralement la structure du personnage ainsi que les mesures et les proportions de ses différentes parties. Il met en place un modèle de construction constitué par un ensemble de dessins de complexité croissante, parmi lesquels :

- le squelette du personnage, définissant les articulations et les os afin de préciser de façon rapide et schématique les différentes parties du corps et leurs proportions relatives (Figure 1.1(a)),
- une représentation par formes simplifiées qui permet de créer un support de dessin à partir duquel les détails seront graduellement raffinés (Figure 1.1(b)),
- la version finalisée du personnage, incluant l'ensemble des détails ainsi que parfois les couleurs et les textures (Figure 1.1(c)).

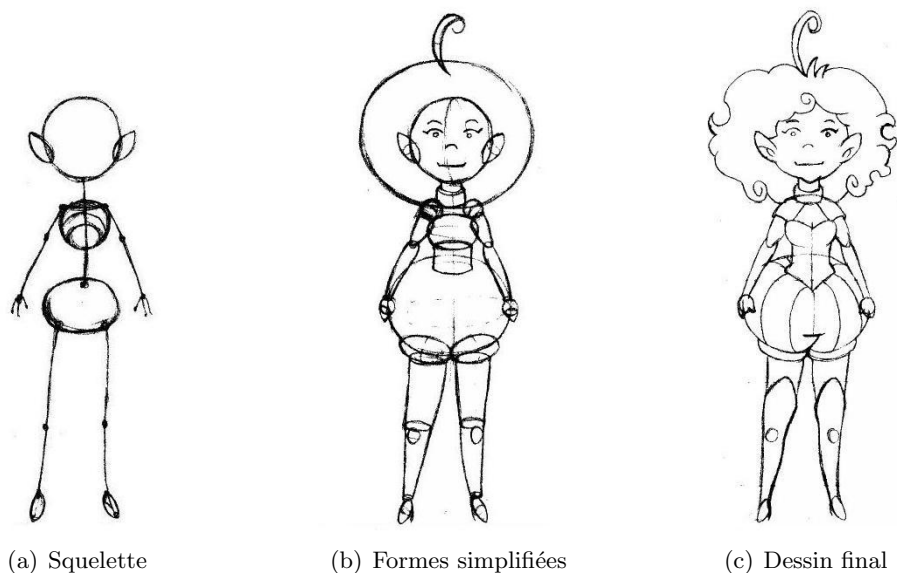


Figure 1.1 – Modèle de construction pour le personnage Elfie.

Afin de spécifier complètement le personnage, le *designer* fournit aussi une planche de dessins appelée *turnaround* (Figure 1.2) représentant le personnage dans la même pose selon différents angles de vue. Sont généralement créées les vues de face, profil et dos, auxquelles une vue de trois-quarts est parfois ajoutée.

Enfin, pour compléter le travail de définition du personnage, le *designer* fournit également une série de dessins spécifiant quelques expressions du visage (Figure 1.3), ainsi que des des-

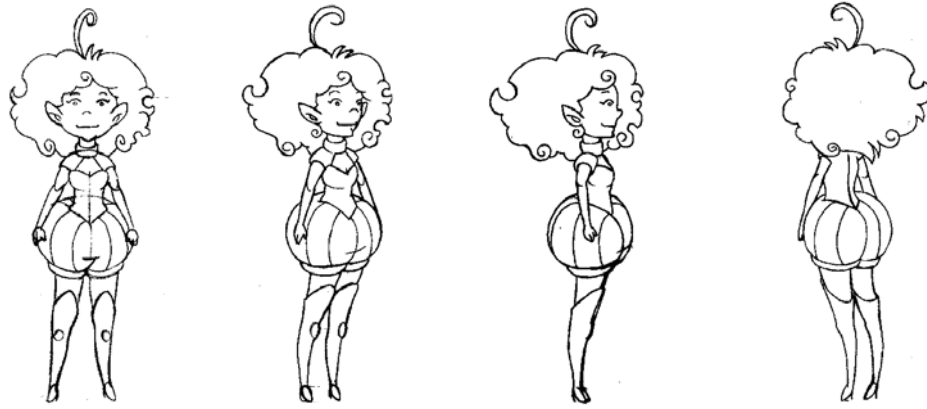


Figure 1.2 – Turnaround.

sins du personnage dans différentes positions. Cela permet de mieux cerner le caractère du personnage grâce à la description de son comportement.



Figure 1.3 – Exemples d'expressions de visage.

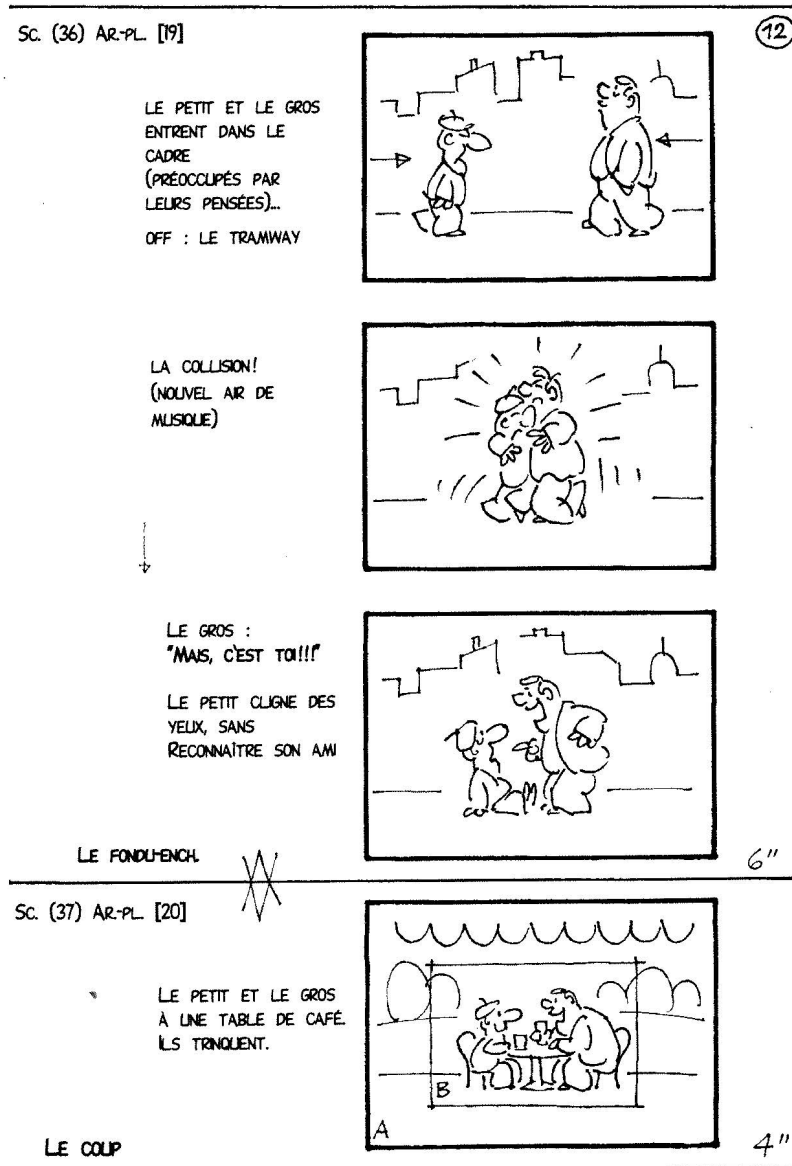
1.1.1.3 Le *storyboard*

Une fois les personnages définis, le scénario est spécifié de manière plus précise à l'aide du *storyboard* ou *scénarimage*. Équivalent du script en cinéma, le scénarimage décrit les actions des personnages, leurs dialogues ainsi que le décor dans lequel ils évoluent et les mouvements de la caméra. Contrairement aux scripts utilisés dans les productions cinématographiques, le scénarimage est entièrement spécifié par des dessins. Un exemple est illustré Figure 1.4.

En résumé, l'étape de pré-production a pour objectif d'établir un ensemble de règles permettant aux animateurs de produire des images qui représentent les personnages dans différentes positions de façon aisée et reproductible. Les images d'animation peuvent ainsi être réalisées par une équipe de plusieurs animateurs de façon transparente en suivant scrupuleusement le *storyboard* et les règles de dessin préalablement définies.

1.1.2 La production

La production est la fabrication des différents éléments qui composent les images nécessaires à la réalisation des séquences animées. Ces éléments comprennent principalement les décors et les personnages. La réalisation des images de chacun de ces éléments se déroule en deux étapes effectuées par deux équipes différentes : le dessin des contours, réalisé par une équipe de dessinateurs et d'*animateurs*, puis la mise en couleur par une équipe de *gouacheurs*.

Figure 1.4 – Exemple de *storyboard* (extrait de [Dov00]).

1.1.2.1 Le décor

Le décor est composé de tous les éléments graphiques non animés. Les images représentant les décors peuvent être réalisées sur plusieurs plans ou *layers* de manière à les placer par la suite à différentes distances de la caméra. Les dessins de l'arrière-plan sont réalisés sur papier dans le but d'obtenir un fond opaque sur lequel se superposent les différents *layers* de la scène.

1.1.2.2 L'animation

Afin d'obtenir à partir du *storyboard* une séquence d'animation complète, il est nécessaire de générer l'ensemble des trames vidéos qui correspondent généralement à une cadence de 25 images par seconde. La production de ces trames est effectuée en trois étapes successives :

1. Dans un premier temps, une équipe d'*animateurs en chef* produit, à partir du *storyboard*, une série d'images représentant des *poses clefs* de l'action, et précise les temps d'action correspondant, définis comme l'intervalle temporel séparant deux poses clefs successives.

2. Des *assistants* mettent au net les images clefs produites par les animateurs en chef.
3. Les poses clefs sont ensuite interpolées successivement par une équipe d'*animateurs associés* et une équipe d'*intervallistes* qui dessinent l'ensemble des images intermédiaires correspondant au temps d'action spécifié (Figure 1.1.2.2).



Figure 1.5 – Animation : images réalisées par l'animateur, puis complétées successivement par l'assistant et l'intervalliste (de haut en bas).

Sur l'ensemble d'une chaîne de fabrication de dessins animés, l'étape d'animation représente de loin le processus le plus coûteux, aussi bien en temps de réalisation qu'en moyens financiers impliqués. Cela est dû au très grand nombre d'images réalisées manuellement par de nombreuses équipes d'animateurs professionnels. En effet, une cadence de 24 images par seconde, norme la plus utilisée, nécessite 1.440 images par minute de film, soit 37.440 images pour un épisode d'une série animée de 26 mn. Cela représente par exemple pour une série d'animation de 26 épisodes de 26 minutes une quantité de travail estimée à 32.240 Jours/Homme !

Automatiser ou accélérer cette étape à l'aide d'outils informatiques devient alors un enjeu à l'impact économique considérable pour les producteurs de dessins animés.

1.1.2.3 La mise en couleur

Une fois les dessins des contours fournis, une équipe se charge de leur mise au net, de leur transfert sur le support final, puis de leur mise en couleur. Les décors qui doivent être placés en arrière plan sont réalisés sur papier, alors que les dessins des personnages et des parties mobiles du décor sont recopiés et mis au net sur des feuilles de celluloïd, qui est une matière plastique transparente.

Le gouachage est l'étape qui consiste à colorier manuellement les différents *layers* ainsi obtenus. Les images représentant les personnages sont colorées par des couches fines et opaques de gouache, de manière à éviter la présence d'ombres ou de transparences lorsque les différents celluloses sont superposés. La coloration des décors, en revanche, peut faire appel à des techniques plus sophistiquées, comme les dégradés et la superposition de couleurs à l'aide d'aquarelles.

1.1.3 La post-production

La post-production inclut les étapes finales qui permettent de générer le film animé qui sont principalement :

- Le tournage, ou la prise de vues, qui se fait à l'aide d'une caméra montée sur un banc-titre. Il s'agit de l'enregistrement, image par image, du film, où chaque image est obtenue en superposant les différents *cellulos* préparés. L'illusion de mouvement est créée en déplaçant les personnages et les décors dans des directions différentes pour simuler le défilement d'un paysage.
Il est possible de simuler un environnement 3D en plaçant les *layers* à des distances différentes de la caméra. Cette méthode a été introduite par les studios *Disney* grâce à une caméra multiplan où un système mécanique est prévu pour déplacer les *layers* (Figure 1.6).
- L'enregistrement et le mixage des différentes bandes sonores, constituées des voix, de la musique d'accompagnement, ainsi que des bruitages.
- Enfin, le montage qui consiste en l'assemblage des plans et des sons enregistrés, en fonction du découpage temporel précisé par le *storyboard*.

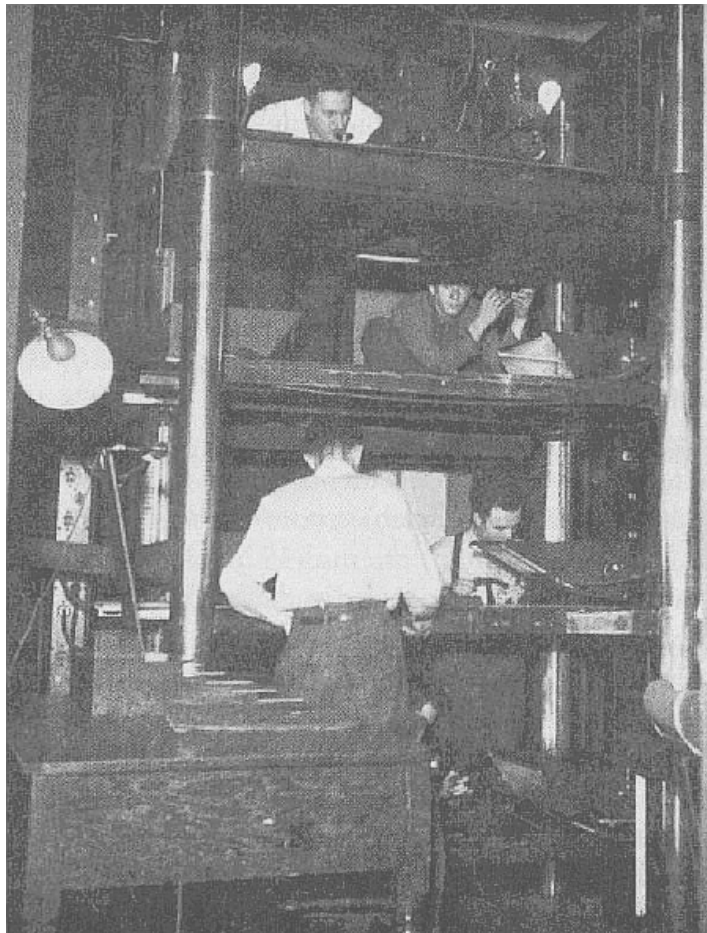


Figure 1.6 – Caméra multiplan.

1.2 Chaîne actuelle de production 2D

De nos jours, la production de dessins animés ne se fait plus de manière entièrement manuelle comme dans le cas traditionnel. En effet, dès l'apparition des ordinateurs, puis des scanners dans les années 1960, ceux-ci ont été graduellement introduits à différents stades de

la fabrication, en particulier au niveau de la mise en couleur des images, de l'animation et de la post-production.

Les premiers travaux et logiciels (*i.e.* CAPS de Disney et *Animo* de Cambridge animation System) avaient pour objectif l'accélération de la production de dessins animés. Ils ont d'abord proposé de numériser les images dessinées manuellement, en les scannant ou en les dessinant avec une tablette et un stylet ce qui permettait de les colorer numériquement. Ce simple passage à des images numériques a ouvert la porte à toutes les manipulations possibles. Ainsi, les cellullos anciennement utilisés pour générer les différents plans d'une image ont-ils été remplacés par des images pouvant être réutilisées ou modifiées. Les paysages sont par exemple repris, et les objets et personnages présents et immobiles sur plusieurs images successives ne doivent plus être redessinés. De plus, cela a permis de supprimer un grand nombre de difficultés techniques et chimiques liées aux cellullos tels que la présence de poussière ou l'apparition d'ombres dues à leur superposition. La numérisation des images a par conséquent, à elle seule, révolutionné la production de dessins animés en terme de changement d'outils et de temps de réalisation, celui-ci étant considérablement réduit.

Ainsi, un des premiers outils utilisés dans le but de créer des dessins animés est CAPS (*Computer Animation Production System*) développé par Pixar pour Disney. Les animateurs devaient alors scanner leurs dessins afin de pouvoir utiliser CAPS pour les colorier, vérifier la cohérence des dessins, puis composer les éléments des scènes. Disney a alors remplacé les *cellulos* "physiques" par des cellullos numériques. Utilisé pour la première fois en 1989, CAPS est devenu l'outil de base pour toutes les animations à partir du film "La belle et la bête" en 1992.

La description vectorielle des dessins a permis en particulier une résolution quasi-infinie des images, qu'elles soient réalisées sur ordinateur ou scannées. En effet, un dessin vectoriel est décrit par des composantes géométriques telles qu'une droite ou une courbe spline, ayant des attributs définissant sa forme, sa couleur ou encore sa position. Contrairement aux images de type bitmap décrites par des pixels, les images vectorielles peuvent subir des transformations de manière continue, comme des changements d'échelle.

Ainsi, grâce à la numérisation, et plus particulièrement à la 2D vectorielle, la production des images numérique peut-elle atteindre aujourd'hui 500 cellullos virtuels par jours, contre 200 cellullos physiques avec les techniques traditionnelles.

En effet, la 2D vectorielle est actuellement employée dans la majorité des logiciels utilisés pour la production de dessins animés, dont *Flash* de Macromedia [fla], *Toonz* de Digital Vidéo [toob], *Animo* de Cambridge animation System [toob], *Pegs* de Mediapegs [peg], et *TicTacToon* [FBC⁺95] ayant donné par la suite naissance à *Toon Boom* de Toon Boom Technologie [tooa].

De même, le gain en temps est très important pour l'étape de *mise en couleur*, celle-ci s'étant affranchie d'une utilisation de gouaches qui nécessitaient un travail très précis : la composition des couleurs et leur uniformité pour toutes les images, l'application de couches fines et homogènes sur les cellullos étant une étape particulièrement sensible.

Ce travail méticuleux est actuellement remplacé par la coloration numérique où une équipe d'intervenants dispose d'une palette de couleurs numériques communes et spécifiées pour chaque personnage. Les couleurs sont appliquées aux différentes images numérisées par un simple clic. Cette pratique est utilisée par les studios Disney depuis 1989 et a rapidement remplacé les techniques de gouachage traditionnelles dans la chaîne de production. En effet, le gain de temps est considérable, puisque le rythme de production passe de 30 *cellulos* par jour avec le gouachage traditionnel à 150/200 avec les techniques numériques.

Enfin, les techniques numériques sont utilisées au moment de la post-production :

- pour l'*assemblage* des différents *cellulos* représentant les décors et les différents personnages,

- ainsi que dans l'étape de *compositing* introduite avec la numérisation, qui consiste à réutiliser des *layers*, et à recomposer les images en rajoutant, déplaçant, ou supprimant des éléments, sans devoir redessiner toute l'image. Ces éléments peuvent être d'origines différentes telles que des modèles 3D, des acteurs humains filmés, ou encore des photographies 2D.
- Enfin, le tournage n'est plus effectué avec une caméra réelle mais virtuelle, procurant ainsi une plus grande souplesse en matière de réajustement des paramètres et des mouvements de la caméra ainsi que pour les recadrages.

L'utilisation de l'outil informatique pour l'étape de l'animation reste cependant un domaine encore peu exploré, malgré quelques tentatives expérimentales dès les années 1960. En effet, en 1963, Peter Foldes introduisit par exemple l'utilisation de l'ordinateur pour calculer des métamorphoses entre deux poses clés représentant des personnages différents. Ses films⁴ concrétisent les premiers pas de cette technologie [Gén03]. La Figure 1.7 représente un exemple de métamorphose réalisée dans le film *La Faim*.

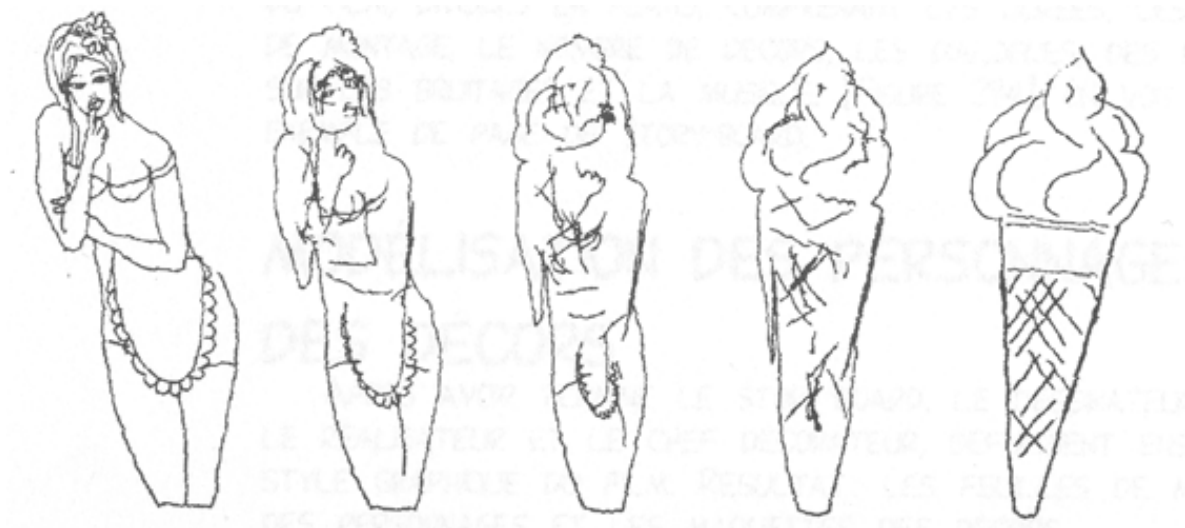


Figure 1.7 – Les phases numériques d'une métamorphose réalisée par Peter Foldes (*La Faim*, 1974).

Dans le but d'automatiser – même partiellement – l'animation d'images, les méthodes proposées ont souvent pris en compte la contrainte de la chaîne de production déjà bien établie. Cette contrainte implique l'utilisation de l'interpolation de poses clés comme processus d'animation, technique traditionnellement utilisée par les animateurs.

L'animation automatique à partir de poses clés peut être structurée en deux tâches :

1. L'établissement des correspondances entre les différents éléments devant être interpolés. Ceux-ci peuvent être des courbes splines ou NURBS, des sommets de lignes ou surfaces polygonales, ou encore les paramètres de position d'un squelette.
2. L'interpolation entre chaque couple de paramètres, *i.e.* la génération des états intermédiaires. Elle s'effectue suivant un chemin spatio-temporel définissant le trajet parcouru par l'élément considéré, ainsi que l'échantillonnage dans le temps des poses intermédiaires.

L'interpolation dépendant fortement de la correspondance établie, cette dernière est souvent effectuée manuellement ou de manière semi-automatique.

Actuellement, les logiciels disponibles sur le marché (tel que *the tab*, [the]) commencent à intégrer des outils d'animation qui interpolent des éléments plus ou moins simples sur deux

⁴MetaData (1970), *La Faim* (Prix du jury à Cannes 1974), *Rêves* (1975)

ou plusieurs images clefs. Les paramètres de ces éléments sont souvent mis en correspondance manuellement et la génération des images intermédiaires se fait ensuite automatiquement. Cependant, cette technique ne permet pas de gérer correctement les problèmes d'occlusion. Elle requiert que les images interpolées soient proches ce qui augmente le nombre de poses clefs nécessaires, et par conséquent le travail de la part de l'utilisateur. De telles limitations font que ces outils sont en fait surtout employés pour des animations simples qui nécessitent peu de calcul, typiquement pour des mini-séries TV ou une diffusion sur Internet.

Il existe dans la littérature plusieurs méthodes d'interpolation pouvant être utilisées pour générer les états intermédiaires, sous forme de courbes 2D, d'images texturées, ou de projections de modèles 3D. Parmi celles-ci, les interpolations linéaire et curviligne sont les plus utilisées, notamment dans les systèmes commercialisés.

Nous avons classé les méthodes existantes selon la représentation utilisée pour générer les images intermédiaires. La majorité des travaux, ainsi que les logiciels disponibles actuellement, résolvent le problème d'un point de vue purement bidimensionnel. Certains travaux ont proposé des méthodes d'animation dites 2,5D. Enfin, quelques auteurs proposent d'utiliser des modèles tridimensionnels simplifiés.

1.2.1 Méthodes 2D

Une manière directe et intuitive de réaliser l'interpolation pour l'animation sur des images dessinées en 2D est de considérer le problème sous un angle purement bidimensionnel. Les images d'entrée sont alors le plus souvent modélisées comme un ensemble de courbes qui peuvent être décrites par des splines, des NURBS, ou encore des lignes polygonales. Une correspondance entre les composants des différentes poses clefs fournies doit alors être établie, puis les images intermédiaires sont générées en les interpolant.

Mise en correspondance des paramètres

La détermination des correspondances est une étape critique pour la génération d'une animation. Elle dépend fortement de la représentation choisie pour modéliser les dessins.

Dans TicTacToon [FBC⁺95], les auteurs utilisent des courbes paramétriques. La segmentation de celles-ci est effectuée par l'utilisateur qui doit spécifier les points de début et de fin de chaque courbe sur les poses clefs, ainsi que les correspondances entre les courbes des différentes poses clefs, ce qui représente un niveau d'interaction important.

Afin d'automatiser la mise en correspondance, Burtnyk *et al.* utilisent dans [BW76] l'ordre dans lequel les courbes sont dessinées sur une tablette. Les points délimitant les courbes sont alors définis par les points de lever ou de pose du crayon sur la tablette. La correspondance est alors donnée par l'ordre dans lequel les courbes sont dessinées. Cela impose à l'utilisateur une contrainte non négligeable puisqu'il doit redessiner les lignes de toutes les poses clefs exactement dans le même ordre. La difficulté s'accroît lorsque les poses présentent des courbes qui apparaissent dans une image et pas dans l'autre, auquel cas toutes les correspondances seraient translatées en raison de la courbe insérée, pouvant ainsi conduire à des animations peu cohérentes.

Kort [Kor02] propose un schéma de définition des correspondances différent qui utilise une fonction de coût entre chaque couple de courbes. La minimisation de cette fonction détermine alors le choix des correspondances. La génération des courbes intermédiaires est aussi effectuée à l'aide d'une fonction de coût afin d'obtenir des animations fluides et réalistes.

Bien que cette technique permette d'alléger les contraintes imposées à l'utilisateur, elle dépend encore beaucoup de ce dernier. En effet, la définition des courbes dépend de la saisie de celles-ci et par conséquent du tracé de crayon. Un lever de crayon par exemple change l'assemblage des courbes et influe donc sur le résultat final.

Dans [MTSC04], les auteurs proposent une méthode d'interpolation fondée sur la polygonisation des courbes dessinées par l'utilisateur. L'originalité de leur approche réside dans le fait qu'ils effectuent l'interpolation sur trois paramètres du dessin : les courbes, l'épaisseur de celles-ci et la texture. L'interpolation séparée de ces trois aspects du dessin permet d'obtenir une animation plus complète. L'interpolation s'effectue segment par segment de manière linéaire.

Dans [SG92], Sederberg et Greenwood proposent une méthode d'interpolation de lignes polygonales. La correspondance entre les différents points des polygones respectivement présents sur les poses clefs est déduite à l'aide d'un nombre limité de paires de sommets spécifiés par l'utilisateur (que nous appellerons par la suite *sommets contrainte*). Les lignes polygonales sont ensuite paramétrées. Le nombre de sommets n'étant pas obligatoirement égal sur chaque partie de polygone délimité par deux sommets contrainte, la méthode procède à une insertion de nœuds multiples afin d'arriver au même nombre de paramètres. Deux méthodes sont proposées pour déterminer les sommets devant être dupliqués (ou multipliés) : la première selon des critères géométriques, et la seconde fondée sur une modélisation physique.

Dans [SGWM93], Sederberg *et al.* décrivent les polygones non plus en tant qu'ensemble de sommets, mais comme ensemble d'arêtes définies par leur longueur et l'angle entre deux arêtes adjacentes. Cette représentation permet de décrire la forme de la courbe de manière invariante par rapport à des transformations rigides telles que translation ou rotation. Elle est donc considérée par les auteurs comme une représentation intrinsèque des polygones. La mise en correspondance est effectuée selon la même procédure que dans [SG92].

Interpolation

L'interpolation linéaire est une méthode des plus intuitives, et par conséquent des plus utilisées, notamment par les logiciels actuellement disponibles sur le marché. Dans le cas de l'interpolation de poses clefs, celle-ci peut générer des animations peu fluides conduisant à des personnages aux mouvements mécaniques saccadés. De plus, le temps d'action entre chaque couple de poses clefs n'étant pas fixe, une interpolation linéaire aboutit à des changements de vitesse de l'animation brusques au niveau des poses clefs. Lorsque l'interpolation linéaire est appliquée à des points de contrôle ou à des sommets, les animations obtenues ne conservent pas la rigidité des parties animées. Un bras effectuant une rotation de 90° se voit par exemple rétrécir au cours de l'animation ce qui ne peut être acceptable. Une interpolation appliquée à une représentation telle que celle de Sederberg *et al.* dans [SGWM93] permettrait de remédier à ce problème. En effet, pour un objet rigide la longueur des segments reste la même que dans les poses clefs. Seuls les angles sont alors interpolés.

L'interpolation par courbes splines ou NURBS est toutefois plus appropriée puisqu'elle permet d'interpoler les poses clefs de manière lisse et de générer ainsi des mouvements plus fluides et plus naturels. Dans le cadre des dessins animés, ce type d'interpolation est particulièrement intéressant puisqu'il permet de configurer les vecteurs des nœuds de manière à simuler des effets d'animations caractéristiques des dessins animés décrits par Lasseter dans [Las87] tels que le *ease in ease out*.

Reeves propose dans [Ree81] une méthode d'interpolation plus adaptative que celle à partir des courbes splines, à partir de *points d'ancrage*. L'utilisateur précise les correspondances entre des points qu'il estime caractériser le mouvement, ainsi que le chemin spatio-temporel entre chaque couple de points d'ancrage. Ces chemins consistent en une courbe 2D définissant le déplacement du premier point vers le suivant, graduées de manière à préciser les pas de temps séparant les images intermédiaires. L'animation des points d'ancrage est en fait complètement définie par l'utilisateur. Ces courbes donnent naissance à des patchs définis chacun par deux courbes d'animation et deux courbes des poses clefs. L'interpolation est ensuite effectuée sur le réseau de patchs ainsi définis.

Bien que cette méthode confère à l'utilisateur un plus grand contrôle de l'animation, l'on

imagine aisément le travail qui doit être effectué pour un personnage réaliste tel qu'un humain, dont l'animation d'une main à elle seule nécessiterait la précision de tous les points d'inflexion du dessin ainsi que les chemins et les graduations ! De plus, estimer le mouvement d'un point donné, et maintenir la cohérence entre les différents chemins temporels d'un même modèle semblent délicats à réaliser.

Sederberg utilise dans [SG92] une modélisation physique pour générer les images intermédiaires. Les dessins sont représentés sous forme de polygones dont les arêtes se voient attribuer des raideurs. L'idée consiste ensuite à minimiser l'énergie nécessaire pour transformer le polygone source en polygone cible. L'énergie dépensée par un polygone se compose de deux termes : le premier décrivant l'étirement, et le second modélisant le recourbement. Les résultats obtenus sont très intéressants. Toutefois, les équations décrivant les deux termes d'énergie sont complexes, conduisant à des temps de calcul prohibitifs.

Une modification de cette méthode est proposée dans [Sli92] où l'auteur décrit les déformations et l'énergie qui en résulte par des équations plus simples, de manière à obtenir des résultats proches avec des temps de calcul acceptables.

Burtnyk et Wein [BW76] ont proposé une méthode différente fondée sur des squelettes 2D. Afin de minimiser l'intervention de l'utilisateur, celui-ci ne dessine que des squelettes (sous forme d'un ensemble de segments de droite attachés) pour les parties du personnage à animer. Une zone d'influence est définie par deux lignes, à droite et à gauche du squelette, délimitant les parties du dessin concernées par l'animation. L'animation est alors appliquée aux courbes incluses dans cette zone d'influence, grâce à un système de coordonnées locales défini par le squelette et les deux lignes d'influence.

Une méthode similaire a été introduite par Lazarus *et al.* [LCJ94] qui utilise également un squelette 2D. La déformation des parties, conformément au squelette, est cette fois générée à l'aide de la méthode de déformation de formes libres (FFD).

En conclusion, les techniques permettant d'interpoler des couples de courbes mises en correspondance peuvent donner des résultats intéressants lorsque les poses de départ et d'arrivée ne sont pas trop éloignées. Toutefois, dès que les poses considérées s'approchent d'un changement de position du caractère ou de la caméra, l'interpolation automatique risque de ne plus correspondre à l'animation voulue. En effet, les positions de deux courbes mises en correspondance ne correspondent en réalité pas forcément à une même courbe sur le personnage tridimensionnel.

Une autre difficulté réside dans le fait que, pour deux positions différentes d'un même personnage, certaines courbes peuvent être réduites à une longueur nulle sur une pose, et apparaître sur l'autre. La génération des courbes intermédiaires devient alors ambiguë.

Enfin, une difficulté intrinsèque au contexte du dessin animé est liée aux grandes déformations que subissent les personnages –allant d'un étirement à un aplatissement complet du personnage. Ces déformations rendent encore plus complexe la mise en correspondance entre les courbes, ainsi que l'interpolation. L'intervention requise de la part de l'animateur est alors lourde et le temps gagné par l'automatisation est compensé par un effort supplémentaire, nécessitant de surcroît des compétences nouvelles.

Les techniques présentées dans cette section se heurtent généralement à deux difficultés principales : l'occlusion et la rotation de l'objet ou du personnage autour d'un axe parallèle au plan de la caméra.

Une méthode suggérée par Hsu et Lee [HL94] tente de résoudre le problème de la rotation en introduisant le concept de *points d'ancrage*. Il s'agit de points ancrés en une position donnée lors de l'animation. Ceux-ci sont souvent les points d'extrémité des courbes. La déformation est alors appliquée à la courbe pendant que ses extrémités restent attachées à une position donnée. Cette technique permet de réaliser l'animation d'un visage tournant, tout en gardant

le contact du nez par exemple. Le résultat ainsi obtenu est certes intéressant par son réalisme, mais les angles de rotation doivent rester faibles pour que l'aspect du modèle animé (le visage dans notre exemple) soit ressemblant sur les deux poses clefs.

De manière générale, les méthodes qui considèrent le problème de l'interpolation entre les poses clefs sous un angle purement bidimensionnel sont limitées justement par cette vision 2D de l'animation, car le côté intrinsèquement tridimensionnel des objets est négligé. En effet, bien que les dessinateurs interpolent d'une manière naturelle et intuitive ce type d'informations, l'automatisation de l'interpolation reste difficile à réaliser. Ce problème a d'ailleurs déjà été signalé par Catmull dès les années 1970 [CC78].

Les logiciels actuellement disponibles sur le marché traitent cette problématique d'une manière à la fois simple et intelligente. La pose cible est produite en copiant la pose source, puis en la déformant [the]. La correspondance entre les paramétrisations des différentes courbes est dans ce cas directe, puisqu'elle découle d'un seul et même dessin. Toutefois, cette astuce nécessite la présence des mêmes courbes sur les deux poses. Par conséquent, le nombre de poses clefs nécessaires augmente. De plus, dans le cas d'un mouvement de rotation du personnage selon un axe parallèle au plan de la caméra, les déformations à appliquer deviennent trop importantes. Il serait alors plus facile de redessiner le personnage entièrement à la main.

Les techniques 2D se placent donc sous le signe de l'*interpolation assistée par ordinateur* de façon non *automatique*, étant donné l'effort encore important qu'elles requièrent de la part de l'utilisateur. De plus, le travail demandé, différent de celui habituellement attribué à un dessinateur, nécessite une période de familiarisation et d'apprentissage pour ce type d'outils.

1.2.2 Méthodes 2,5D

Le terme 2,5D désigne généralement des approches manipulant des dessins 2D dans l'espace 3D. Cette technique permet d'obtenir une illusion de 3D en plaçant par exemple le décor dans lequel évoluent les personnages sur différents plans parallèles plus ou moins éloignés, ou encore de simuler une légère rotation d'un personnage en l'appliquant au plan sur lequel son dessin est situé. Les animations sont donc réalisées dans un espace 3D, structuré par les courbes définies sur des supports bidimensionnels et doté d'une caméra dont la position est modifiable.

De telles techniques ont été utilisées dans le cadre d'applications aussi diverses que l'architecture comme dans [TDM99] où les courbes sont sur des sphères, le design et la haute couture comme dans [BCD01] où les dessins sont sur des plans, mais aussi dans le cadre du dessin animé comme dans [Ree96] où l'animation entre les poses clefs est réalisée sur des plans différents. Dans un même plan, l'interpolation est effectuée grâce à un squelette 2D précisé par l'utilisateur.

Le problème de l'occlusion est alors souvent traité en positionnant les différentes courbes sur des plans distincts [Kor02], [the] ce qui permet de spécifier quels contours sont apparents ou cachés. Cela est réalisé en conférant auxdits plans des positions invariantes tout au long de l'animation. Toutefois, cela alourdit la procédure d'animation puisque les positions relatives des plans doivent être modifiées manuellement et des images clefs doivent être ajoutées aux instants de changement de position des plans.

Bien que les techniques 2,5D permettent d'introduire une profondeur à la scène, la représentation des objets demeure bidimensionnelle. Cela constitue un manque d'information qui doit être complété par des dessins supplémentaires, ce qui augmente l'interaction requise par l'utilisateur.

1.2.3 Méthodes 3D

Les méthodes précédemment présentées laissent à la charge de l'utilisateur les tâches ne pouvant pas être réalisées automatiquement, ou dont le résultat n'est pas acceptable, comme le cas d'une courbe sur une pose clef qui n'a pas de correspondant sur la pose suivante.

Malheureusement, ce cas de figure se présente souvent et la quantité de travail demandée reste importante.

Afin de minimiser l'effort demandé, DiFiore *et al.* proposent dans [FSER01] une méthode utilisant des *modèles 3D simples*, dans le cadre d'une approche multi-couches. Les personnages sont spécifiés par des dessins 2D modélisés par des courbes paramétriques (niveau 0). Des structures simples, dites 2,5D, représentent la forme générale des personnages (niveau 1). L'information exploitée pour générer les animations est constituée par un squelette 3D (niveau 2). Enfin, une dernière couche permet de réaliser des déformations appliquées aux dessins 2D, pouvant être exploitées pour gérer les expressions faciales par exemple.

L'utilisateur précise les positions et angles des éléments du squelette 3D pour chacune des poses clefs. Les animations peuvent alors être appliquées en 3D, puis reprojétées afin d'obtenir les images intermédiaires nécessaires. Les difficultés liées à l'occlusion de différentes parties d'un personnage sont ainsi résolues en prenant en compte l'aspect tridimensionnel des personnages. Cependant, cette méthode nécessite un savoir-faire relatif au choix des paramètres de position du squelette, ce qui est peu adapté pour d'une application dédiée aux dessinateurs et animateurs 2D.

Animations Stylées pour le dessin animé

Une autre famille d'approches traite d'un problème spécifique au dessin animé qui consiste à représenter les personnages différemment en fonction du point de vue. Cet effet de style, qui est une caractéristique de l'animation traditionnelle, représente une difficulté supplémentaire puisqu'elle introduit une incohérence entre les dessins et le modèle 3D pouvant représenter le personnage.

Dans [Rad99], les auteurs utilisent un modèle 3D complet du personnage, préalablement fourni. A ce modèle sont associés des dessins de référence correspondant à son état pour un ensemble de points de vue donnés dits "points de vue clefs". La position de la caméra est précisée pour ces positions, ainsi que les déformations subies par le modèle 3D du personnage nécessaire pour correspondre aux dessins de référence fournis. Les positions intermédiaires sont ensuite calculées en interpolant à la fois le mouvement du personnage et la déformation due au changement de point de vue.

Dans [LGXS03] et [CKB04], les auteurs proposent des animations similaires, mais dont les déformations ne sont pas définies par le point de vue. L'effet de style est cette fois extrait des poses clefs, introduisant ainsi plus de flexibilité dans l'animation. L'interpolation se sépare donc en une interpolation du mouvement d'une part, et d'autre part une interpolation du style se traduisant par une interpolation des déformations subies par le personnage, telles que l'étirement d'une partie par exemple.

1.2.4 Conclusion

De manière générale, les méthodes fondées sur une vision 2D ou 2,5D du problème présentent des limitations intrinsèques au manque d'information puisqu'elles utilisent des données 2D pour décrire un phénomène tridimensionnel qu'est le mouvement d'un personnage dans l'espace 3D. De ce fait, les méthodes faisant appel à des modèles tridimensionnels nous semblent plus appropriées pour la génération automatique des images intermédiaires.

Toutefois, des méthodes telles que celle de Teddy [IMT99] imposant une interaction en 3D, sont peu adaptées au contexte de la production de dessins animés traditionnels. La génération des modèles virtuels, ainsi que leur interpolation et déformation doivent être effectuées de manière complètement transparente à l'utilisateur, afin que la chaîne de production conserve son mode de fonctionnement inchangé.

Pour conclure, les outils informatiques utilisés aujourd'hui aident certes les créateurs à accélérer le processus de fabrication des dessins animés. Néanmoins, les fonctionnalités offertes restent pauvres par rapport au grand potentiel offert aujourd'hui par les techniques d'imagerie

tridimensionnelle mises à profit dans les nouveaux films 3D qui ont su puiser de nouvelles technologies dans des domaines émergents comme la réalité virtuelle/augmentée.

1.3 Chaîne de production de dessins animés 3D

En raison de son jeune âge, la production de dessins animés 3D [Ker04a] obéit à des règles de fonctionnement beaucoup plus souples qui peuvent en outre varier d’une maison de production à l’autre. Nous nous contenterons ici de décrire rapidement les étapes principales de la chaîne de production.

Le processus de création d’animation par ordinateur varie en effet très souvent en fonction des buts créatifs, du budget disponible, ou encore du planning envisagé. Toutefois, on distingue comme pour les dessins animés 2D, trois étapes principales : la pré-production, la production, et la post-production.

1.3.1 La pré-production

La pré-production fait référence à l’ensemble des étapes de conceptualisation et de planification qui prennent place avant la production de l’animation par ordinateur. Elle inclut donc des tâches artistiques et créatives comme l’écriture du scénario et la détermination des acteurs ou personnages, ainsi que des tâches de planification relatives à l’organisation du déroulement du projet, et à la fixation des délais et des budgets nécessaires. Il s’agit d’une étape fondamentale pour la réussite du projet puisqu’elle vise à mettre en place les bases sur lesquelles il sera mené.

1.3.2 La production

La production de films d’animation 3D implique principalement des étapes de modélisation, de *rigging* (squelettisation), d’animation et de rendu. Chacune de ces étapes s’appuie sur des techniques spécifiques et est généralement réalisée par une équipe spécialisée. L’interaction est toutefois forte à la fois entre ces différentes équipes et au sein de chaque équipe d’une part et les scénaristes et créateurs des personnages d’autre part, ces derniers restant la référence pour chaque phase de production.

La modélisation consiste à créer en 3D les personnages, objets, et environnements qui seront utilisés dans l’animation. Cette étape peut faire appel à plusieurs techniques différentes. En effet, les modèles peuvent être d’abord physiquement sculptés avant d’être scannés pour produire le modèle virtuel 3D. Ils peuvent aussi être créés directement en 3D par des modelleurs à l’aide de logiciels dédiés tels que Maya de AliasWaveFront [MAY], 3DS Max de Discreet [MAX], ou XSI de Softimage [XSI].

Les personnages sont souvent définis à l’aide de dessins manuels 2D ressemblant aux *model-sheets* utilisés en dessins animés 2D. L’apparence du modèle 3D du personnage est caractérisée non seulement par son enveloppe surfacique, mais aussi par la façon dont il reflète la lumière, définie par un modèle de lumière, ainsi que par les modèles de couleurs et les textures qui lui sont appliqués.

Pour la plupart de ces méthodes, les personnages sont animés à l’aide d’un *squelette* qui représente une structure simplifiée du modèle, dont les déformations sont plus rapides à calculer et qui permet de hiérarchiser le modèle et donc ses mouvements. La création de ces squelettes fait partie des étapes sensibles de la production d’un film 3D. Elle détermine la qualité et la fluidité des mouvements des personnages dans toutes les scènes animées par la suite. Afin de définir au mieux le comportement de l’enveloppe surfacique lors des animations, celle-ci est “attachée” au squelette afin de préciser la dépendance de chaque partie envers les éléments du squelette.

La figure 1.8 illustre ces différentes étapes de modélisation pour le personnage nain inspiré d'un dessin manuel 2D.

Enfin, les animations peuvent être réalisées conformément au scénario, et ce à l'aide de plusieurs techniques : interpolation de poses clés, plaquage de mouvement obtenus par capture de mouvement sur des acteurs réels, ou encore élaboration de modèles physiques ou comportementaux des personnages.

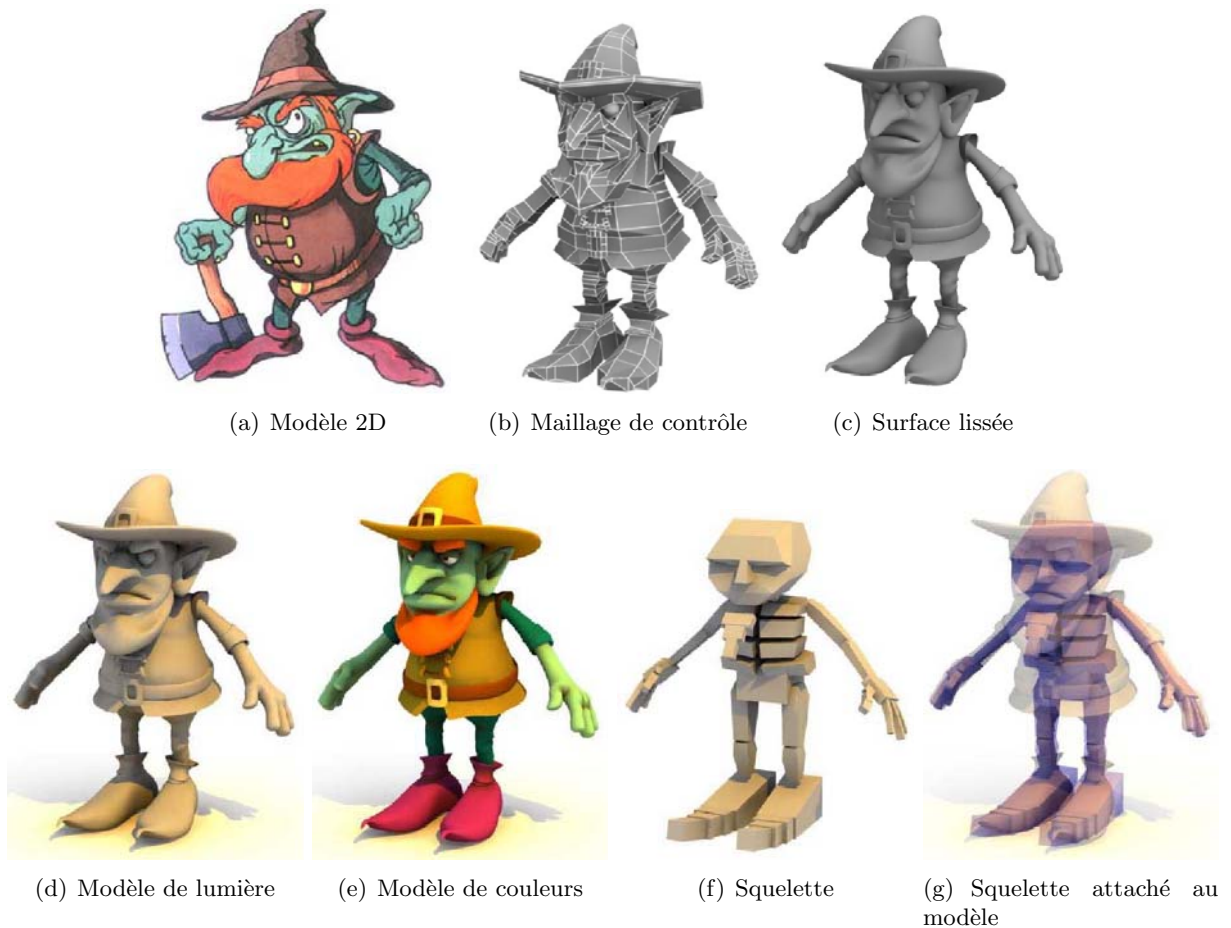


Figure 1.8 – Étapes de modélisation d'un personnage pour une production 3D.

1.3.3 La post-production

A partir des images filmées des modèles animés, différents traitements de post-production sont appliqués, dont la composition avec d'autres images générées par ordinateur représentant les décors par exemple, ou bien des images filmées avec des personnages ou des paysages réels. Des retouches comme des modifications de couleurs ou le recadrage de la caméra peuvent aussi être effectuées lors de cette étape.

Une fois les images prêtes, une étape très importante dans la fabrication d'un film est le mixage des sons qui consiste en l'insertion des dialogues, des musiques d'accompagnement, et des bruitages, et ce conformément aux temps d'action.

En fait, la post-production regroupe toutes les manipulations nécessaires pour la finalisation du film avant son enregistrement définitif en vue de sa distribution.

1.4 Discussion

La principale différence entre les deux chaînes de production réside dans l'agencement entre les étapes de création et d'animation. En effet, dans le cadre des dessins traditionnels, une fois le personnage spécifié par le concepteur, ses mouvements, ses textures et l'éclairage de la scène sont effectués directement lors de l'étape d'animation. Au contraire, dans le cas de la production 3D, le personnage virtuel doit d'abord être modélisé. Cette étape de modélisation est alors déterminante pour la suite. Elle doit répondre conjointement et de façon adaptée aux contraintes d'utilisation comme l'animation, le plaquage de texture et le rendu.

Ainsi, l'exploitation d'un modèle 3D engendre-t-elle de fortes inter-dépendances entre les étapes de modélisation et d'animation. Le modèle 3D est ainsi susceptible de subir des modifications et des ajustements en fonction du taux de conformité des effets d'animation obtenus à ceux souhaités. Cela implique des interactions fortes et répétitives entre les équipes de modéleurs et d'animateurs, tout au long du processus de production du film. La Figure 1.10 illustre l'organisation de la chaîne de production 3D, et montre en particulier l'intrication des différentes phases de la chaîne, contrairement à la chaîne 2D dont l'organisation est clairement linéaire (Figure 1.9).

Une contrainte d'interaction aussi importante est prohibitive dans le cadre des dessins animés 2D, puisque les différentes équipes sont souvent délocalisées. De plus, l'animateur 3D n'est plus un simple utilisateur du modèle du personnage. Il participe activement à sa création ce qui nécessite un spectre de compétences élargi et une maîtrise de logiciels complexes (3DS Max, Maya,...). En raison de toutes ces contraintes, les professionnels de l'animation 2D se montrent peu enclins à ce type d'approche mal adapté à la fois à l'organisation de la chaîne de production, à leur formation et expérience de travail sur les dessins 2D manuels.

Par ailleurs, quelles que soient les performances des outils d'animation 3D disponibles aujourd'hui, ils ne sont pas encore en mesure de remplacer complètement le facteur humain et l'aspect artistique des animations traditionnelles, en particulier pour l'introduction d'effets de style dans les mouvements comme l'anticipation ou l'exagération [Las87]. Les animations 3D sont généralement gérées à l'aide de modèles physiques ou paramétriques et leur objectif est de créer des mouvements réalistes. L'utilisation de ce type d'animation dans le contexte des dessins animés 2D ne peut donc être exclusive puisque cela représenterait une entrave à la touche artistique et caricaturale des dessins animés.

En résumé, l'adoption des techniques 3D dans le monde des dessins animés traditionnels semble aujourd'hui problématique, et ce pour trois raisons :

1. La première est liée au mode de production des dessins animés traditionnels. L'adoption des outils 3D nécessiterait en effet de grandes modifications de ce système de fonctionnement déjà bien établi. De plus, un élément handicapant provient de la délocalisation actuelle des équipes de dessinateurs, actuellement réparties sur divers sites géographiques du globe.
2. La seconde résistance est liée à l'utilisation de logiciels 3D qui passe par l'acquisition de nouvelles compétences de modélisation et d'animation 3D bien plus techniques que celles dispensées dans les formations essentiellement artistiques des dessinateurs traditionnels.
3. La troisième provient de l'inadéquation des méthodes d'animation 3D qui sont plus appropriées à des mouvements réalistes qu'aux déformations caricaturales souvent employées en dessins animés 2D.

En conclusion, la production des dessins animés 2D est actuellement confrontée à un manque d'outils dédiés qui permettraient d'en accélérer les temps de fabrication. En effet, la 2D vectorielle atteint ses limites lorsqu'il s'agit de réaliser des animations complexes, ce qui

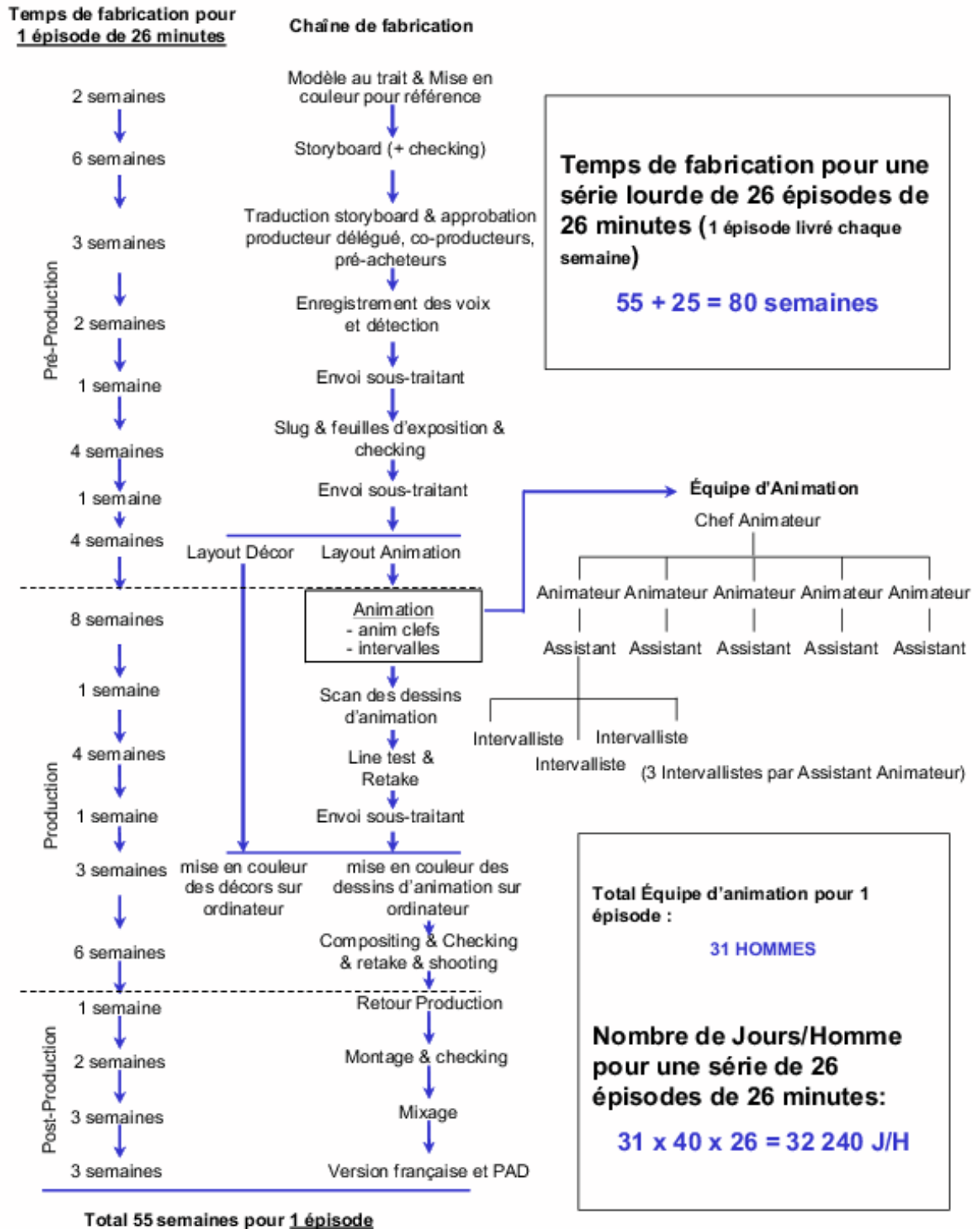


Figure 1.9 – Organisation de la chaîne de production 2D.

réduit son utilisation à des animations simples ou de petite durée comme les publicités. Les techniques 3D, quant à elles, sont mal adaptées aux besoins spécifiques de l'animation 2D, et nécessitent des compétences particulières.

La revue de l'état de l'art sur les techniques 2D dédiées aux dessins animés nous montre que l'utilisation de modèles 3D, même simplifiés, est nécessaire pour pouvoir répondre aux problèmes posés par l'interpolation de poses clefs. Nous proposons donc une approche par reconstruction 2D/3D à partir des dessins de *turnaround* qui prend en compte le mode de

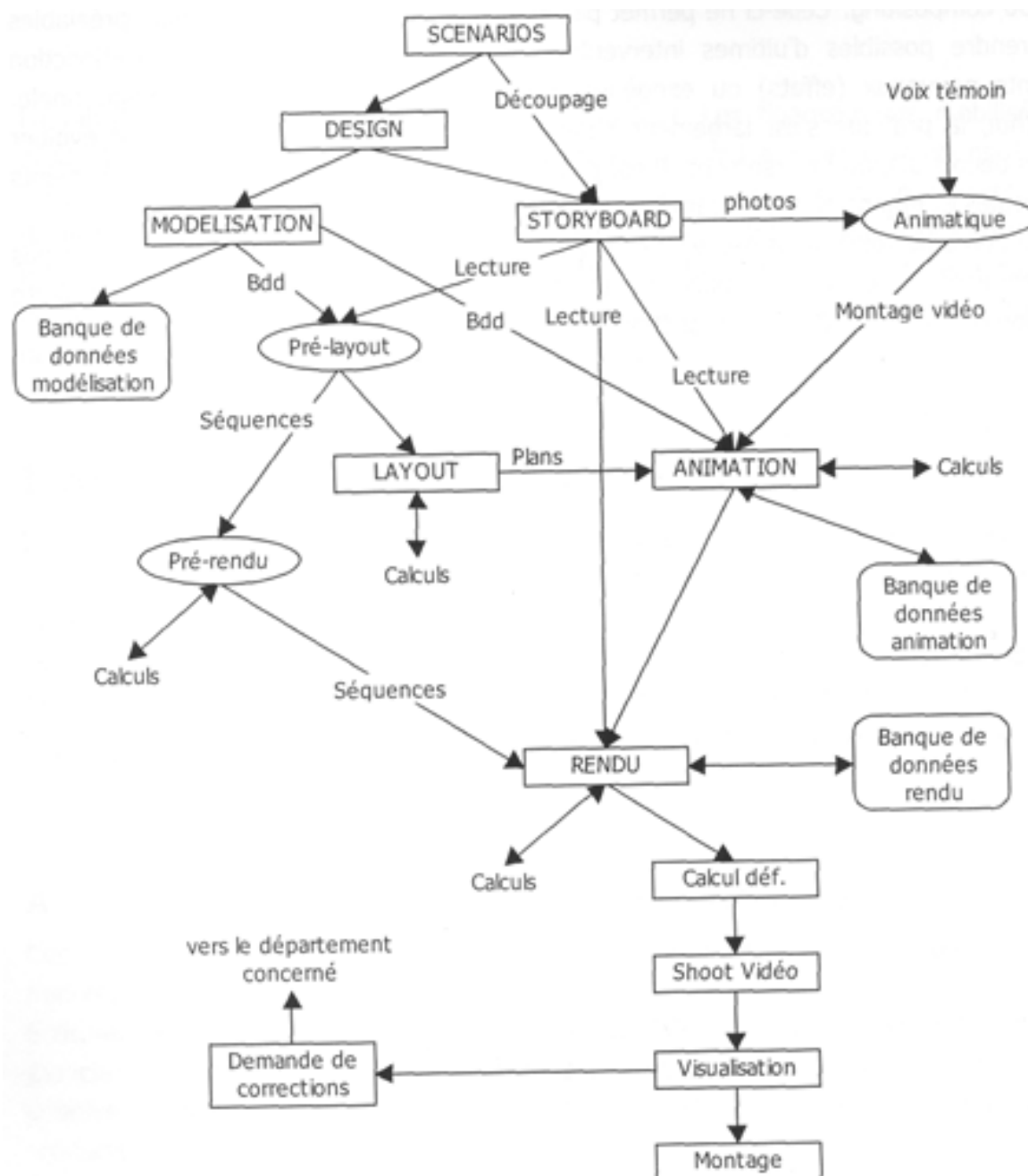


Figure 1.10 – Organisation de la chaîne de production 3D.

création des personnages. Les modèles 3D sont reconstruits de manière à pouvoir être animés en 3D, toujours selon des contraintes 2D.

Chapitre 2

Modélisation 3D

Résumé

Pour réaliser un outil de reconstruction dédié aux dessins animés 2D, il est nécessaire d'identifier tout d'abord les propriétés requises du modèle de représentation 3D à utiliser. Cela est effectué en fonction des facilités de reconstruction, de visualisation et d'animation nécessaires.

Ce chapitre présente un état de l'art des principaux modèles de représentation de surfaces 3D : maillages polygonaux, surfaces implicites, surfaces paramétriques et surfaces de subdivision.

L'étude de ces différents modèles montre que les maillages polygonaux sont les plus adaptés à une visualisation rapide. Correspondant au mode de représentation le plus communément utilisé, les modèles maillés sont bien adaptés à l'échange et la diffusion entre différentes équipes souvent délocalisées. Toutefois, les maillages constituent une structure irrégulière sans contraintes de continuité.

La continuité des surfaces NURBS et leur contrôle local en font une représentation particulièrement appropriée pour modéliser/déformer des objets 3D.

Les surfaces de subdivision correspondent quant à elles à un mode de représentation intéressant pour l'animation, notamment en raison de leur description sur plusieurs niveaux de détails. Le maillage de base doit toutefois répondre à des contraintes de régularité pour donner des surfaces lisses.

Les surfaces implicites peuvent entraîner des fusions non souhaitées de deux surfaces proches ce qui est un obstacle pour l'animation de personnages articulés.

Mots clés

Surfaces 3D, Maillages polygonaux, Surfaces de subdivision, Surfaces implicites, Surfaces paramétriques.

Les images de synthèse nécessaires à la réalisation d'un film d'animation sont obtenues par visualisation et rendu d'une scène 3D animée. Une scène est constituée de trois éléments essentiels positionnés virtuellement dans l'espace de la scène : des structures 3D représentant les acteurs et décors, des sources de lumière et une caméra.

Bien que la majorité des objets réels soient des volumes, les représentations utilisées pour les modéliser varient en fonction des applications. Dans la littérature, on distingue trois types de description d'un objet 3D : les nuages de points, les modèles surfaciques et les modèles volumiques.

La description par un ensemble de points est certes simple et directe, mais elle nécessite un très grand nombre de points pour restituer un modèle complexe et n'est pas adaptée à des manipulations de modèles articulés.

Les modèles volumiques décrivent l'objet modélisé comme un assemblage d'éléments volumiques de base plus ou moins simples : les *voxels* ou éléments de volume, l'*octree*, représentation hiérarchique qui décompose l'espace en éléments cubiques de tailles différentes de manière à décrire la forme de l'objet en réduisant le nombre d'éléments nécessaires. La *CSG* (Constructive Solid Geometry) utilise des primitives de forme plus complexe telles que des sphères, des cylindres ou des ellipsoïdes pour composer ainsi le volume global de l'objet modélisé. De manière générale, les modèles volumiques fournissent une description complète d'un objet, utile dans les applications nécessitant une connaissance de la composition intérieure de l'objet comme les simulations physiques. Dans le cadre des dessins animés où les animations doivent s'effectuer en temps réel, une représentation surfacique moins coûteuse en mémoire suffit à produire l'aspect extérieur du modèle.

Nous décrivons dans ce chapitre les modèles de surface les plus utilisés dans le cadre de l'animation 3D : descriptions mathématiques exactes pouvant être explicites (surfaces paramétriques) ou implicites (surfaces implicites), description approchée de surfaces lisses par un ensemble de polygones (maillages, surfaces de subdivision).

2.1 Représentation par maillages polygonaux

La représentation par maillage [FvDFH97] est une approximation linéaire par morceaux de la surface 3D. Un maillage est défini par un ensemble de polygones planaires appelés facettes. La position des sommets de ces facettes dans l'espace 3D, le plus souvent exprimée dans un repère cartésien, définit la *géométrie* du maillage, tandis que les relations d'adjacence entre sommets, arêtes et facettes définissent sa *topologie* (information de connexité).

D'une façon générale, les facettes d'un maillage peuvent avoir un nombre arbitraire de sommets. Toutefois, le plus souvent, les maillages sont composés de facettes triangulaires, en raison des techniques sous-jacentes de construction mises en œuvre qui impliquent souvent des procédures de triangulation (*cf.* scanners 3D).

Les maillages 3D offrent une grande puissance de représentation permettant de modéliser une large variété d'objets de formes et de complexités différentes, selon des topologies et géométries arbitraires.

Par définition, un maillage 3D est au mieux une surface C^0 -continue. Pour représenter alors de façon réaliste des objets lisses et de forme complexe, il est nécessaire de construire des maillages avec un nombre élevé de facettes. La Figure 2.1 illustre quelques exemples de maillages 3D représentant des humanoïdes à différents niveaux de détails.

Notons que les approximations réalistes d'un modèle aussi complexe qu'un humanoïde nécessitent au moins quelques milliers de facettes. Cela pose un certain nombre de problèmes relatifs à leur utilisation directe dans les systèmes de création et d'animation d'objets 3D.

Il s'agit notamment des aspects de construction : comment obtenir des maillages réalistes représentant les objets à modéliser ? Créer manuellement des représentations aussi complexes en positionnant des milliers de facettes dans l'espace 3D est bien sûr une tâche impossible.

Les modèles maillés complexes existant sont en fait obtenus à l'aide :

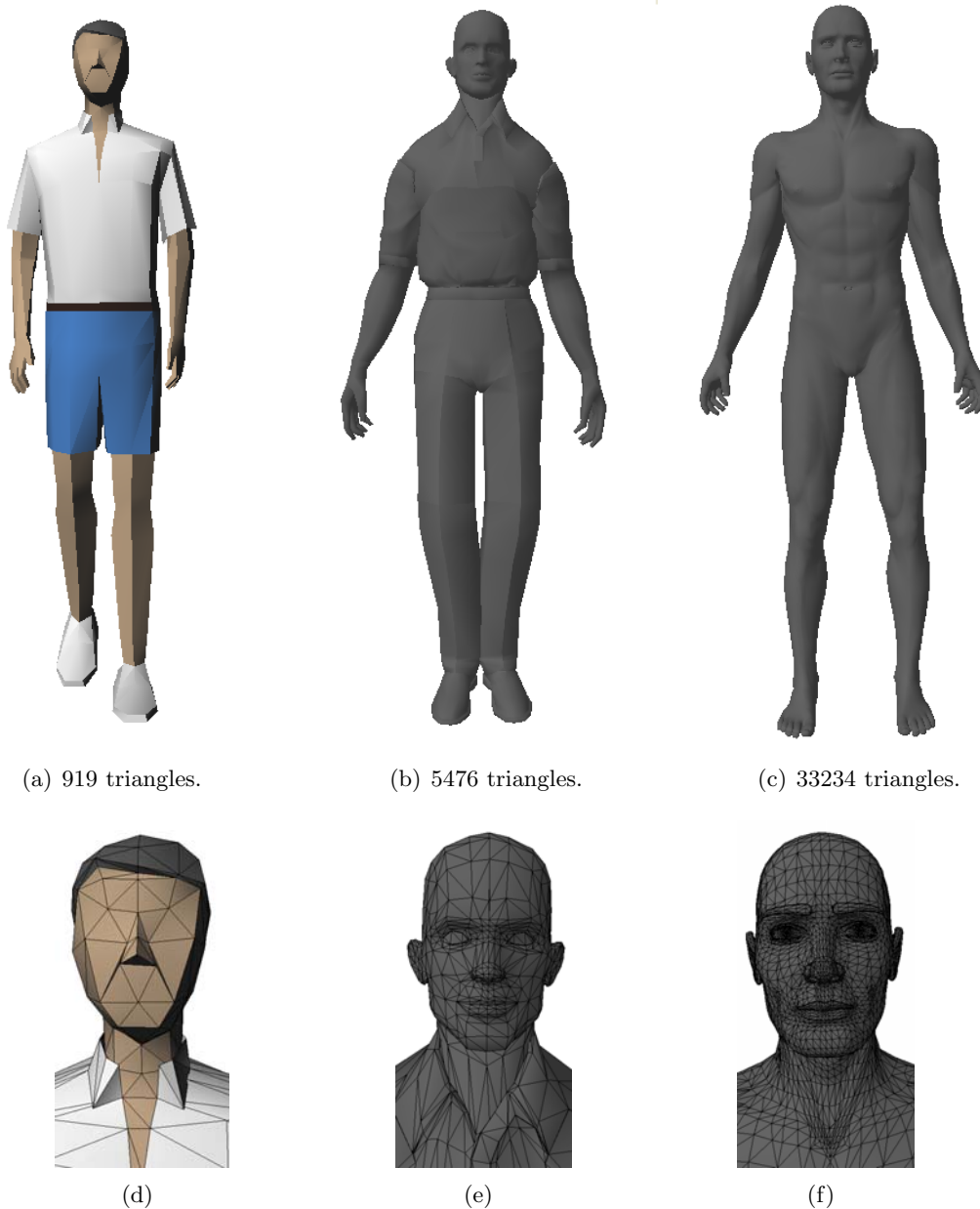


Figure 2.1 – Modèles maillés d’humanoïdes avec des résolutions différentes.

- Soit de systèmes d’acquisition 3D tels que les scanners numériques, ce qui implique l’existence d’un objet physique représentant le personnage que l’on souhaite modéliser. Cela limite l’application de cette technique dans le cadre du dessin animé où les personnages n’existent que sur papier.
- Soit de logiciels de modélisation 3D dédiés tels que Maya [MAY] et 3DS Max [MAX] qui proposent une approche incrémentale dans le cas de la construction de maillages.

2.1.1 Techniques de construction de maillages 3D

En partant d’une forme maillée simple, comme un cube ou une sphère, l’objet est modifié par étapes successives à l’aide d’une bibliothèque d’outils de manipulation de maillages qui permet d’insérer de nouveaux points, d’éliminer des sommets existants, de sélectionner et de déplacer dans l’espace des sous-parties de l’objet, etc.

Intuitivement, l’utilisateur vise à créer à chaque étape une représentation de l’objet la plus

fidèle possible, à la résolution donnée (*i.e.* en fonction du nombre de facettes du maillage). Le passage à un niveau de résolution plus fin est effectué par insertion de nouveaux points, bien localisés dans la partie de l'objet à affiner. Cette méthode permet de minimiser le temps de conception/construction.

Afin de produire des surfaces lisses, les modélisateurs commencent souvent par créer un modèle de surface paramétrique (*cf.* Paragraphe 2.3). Une fois ce dernier triangulé, il peut être affiné pour intégrer plus de détails selon les techniques décrites précédemment.

Remarque : Dans le contexte de la modélisation de personnages virtuels animés, les animateurs spécifient souvent des *boucles d'arêtes*. Il s'agit d'une liste d'arêtes successives définissant un contour fermé, correspondant à des éléments clefs pour l'animation tels que le contour de la bouche ou des yeux. Responsables du réalisme des expressions des personnages, ces éléments nécessitent en effet une finition de grande qualité.

2.1.2 Conclusion

Ce protocole de construction permet de générer des modèles plus ou moins simplifiés, pouvant avoir jusqu'à quelques centaines de facettes.

Toutefois, cette approche trouve ses limites dès lors qu'il s'agit de construire des modèles avec un degré élevé de réalisme. La solution consiste alors à exploiter des modèles de surface plus élaborés comme les surfaces implicites, qui permettent de représenter des objets 3D à l'aide d'un nombre plus réduit de paramètres.

2.2 Surfaces implicites

Une surface implicite est définie comme l'iso-surface d'un champ de potentiel scalaire défini sur l'espace \mathbb{R}^3 :

$$S = \{(x, y, z) \in \mathbb{R}^3 : f(x, y, z) = c\}. \quad (2.1)$$

où :

- $S \in \mathbb{R}^3$ est la surface implicite,
- $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ est la fonction de potentiel associée,
- $c \in \mathbb{R}$ est une valeur constante du potentiel, dite *seuil*.

Les fonctions de potentiel utilisées étant continues, les surfaces générées le sont aussi. Ces surfaces définissent par conséquent un partitionnement de l'espace en deux parties séparées par la surface : l'une où $f > c$ et l'autre où $f < c$. Par extension du cas particulier des surfaces fermées, ces parties sont respectivement appelées intérieur et extérieur de la surface S . En général, la valeur du seuil c est mise à zéro. On définit alors l'intérieur et l'extérieur de la surface par les zones où le champ prend des valeurs respectivement positives ou négatives.

La fonction de potentiel peut être quelconque et permet de décrire une multitude de formes. En modélisation 3D, les formes les plus utilisées sont choisies de manière à répondre à des critères de continuité, ainsi que de facilité de calcul de la fonction de potentiel, et des caractéristiques de la surface résultante. Nous distinguons deux familles principales : les surfaces algébriques et les surfaces définies par combinaison de primitives, dites aussi surfaces de squelette.

Les *surfaces algébriques* sont définies par des fonctions polynomiales. Parmi celles-ci, les plus utilisées en modélisation sont les quadriques (polynômes du second ordre) et les super-quadriques qui en sont une généralisation. Quelques exemples de ces surfaces sont présentés Figure 2.2.

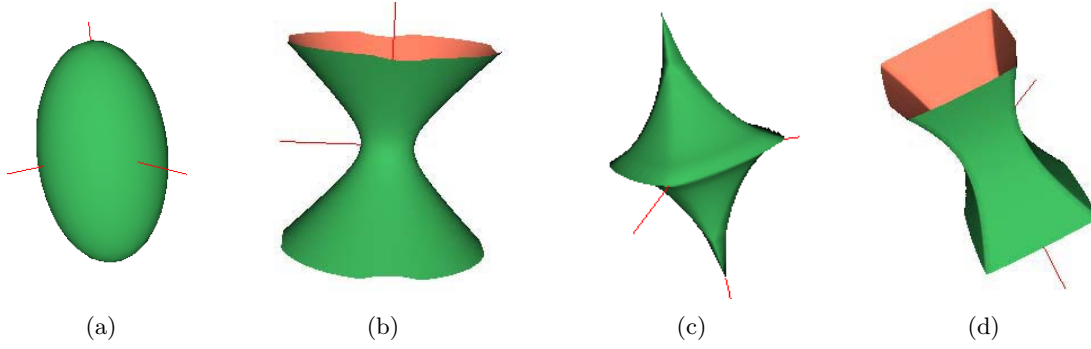


Figure 2.2 – Surfaces algébriques de type quadrique ((a) : ellipsoïde, (b) : hyperboloïde à une nappe) et superquadrique ((c) : superellipsoïde, (d) : superhyperboloïde à une nappe).

Les surfaces algébriques sont limitées en terme de variabilité des formes représentées. Modéliser des objets réels complexes nécessiterait en effet d'augmenter le degré de la fonction polynomiale considérée. Cela conduit à un grand nombre de paramètres dont l'influence sur la forme globale de l'objet est peu intuitive et donc difficile à gérer pour l'utilisateur.

Pour s'affranchir de ces limitations, les surfaces de squelette proposent une approche structurale, décrite dans le paragraphe suivant.

2.2.1 Surfaces de squelette

Pour modéliser des objets de forme libre de complexité arbitraire, les surfaces de squelette s'appuient sur une fonction de potentiel complexe, obtenue par superposition d'un ensemble de potentiels élémentaires appelés *primitives*.

Chaque primitive modélise une forme généralement simple (sphère, ellipsoïde,...) contrôlable à l'aide d'un nombre réduit de paramètres. Une primitive est associée à un sous-ensemble de \mathbb{R}^3 appelé source. Une source peut correspondre à un point, un ensemble de points, un ou plusieurs segments de droite, une courbe, un triangle, un polyèdre simple, etc. L'ensemble des sources considérées forme le *squelette* du modèle. La Figure 2.3 illustre les surfaces obtenues pour différents types de squelette, représentées en plan de coupe.

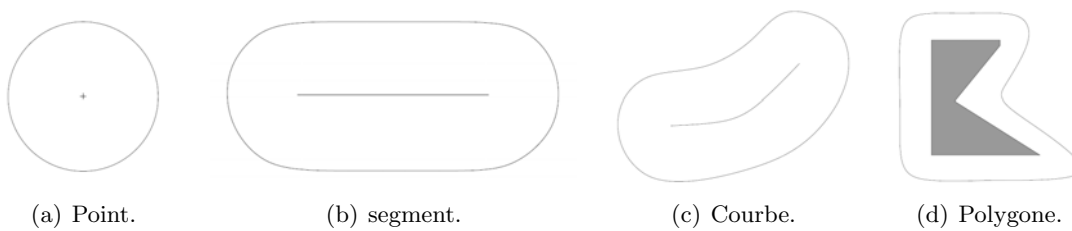


Figure 2.3 – Exemples de surfaces de squelette.

Notons que le plus souvent les potentiels définissant les primitives s'expriment comme une fonction décroissante de la distance d'un point de \mathbb{R}^3 à la source. Ainsi, le potentiel généré par une source S_i en un point p de \mathbb{R}^3 est-il donné par :

$$f(p, S_i) = g \circ \delta(p, S_i), \quad (2.2)$$

où δ est une mesure de distance définie sur \mathbb{R}^3 .

Nous distinguons dans ce qui suit trois catégories de surfaces implicites de squelette :

1. les primitives ponctuelles dont les sources sont réduites à des points et pour lesquelles la mesure de distance généralement appliquée est la distance euclidienne,

2. les surfaces de distance qui sont apparues par la suite comme une généralisation des primitives ponctuelles à des squelettes de forme plus complexe et dont les mesures de distance sont plus particulières,
3. les surfaces de convolution dont l'objectif est d'améliorer les mélanges entre primitives.

Les surfaces implicites sont bien appropriées pour des opérations de combinaison dérivées des techniques de modélisation par *CSG* (Constructive Solid Geometry). Ainsi, les opérations suivantes sont-elles souvent utilisées pour construire des modèles complexes :

Réunion : $f = \max(f_1, f_2)$,

Intersection : $f = \min(f_1, f_2)$,

Mélange (Blending) : $f = f_1 + f_2$,

Mélange négatif : $f = f_1 - f_2$.

La Figure 2.4 montre un exemple de combinaison de deux primitives.

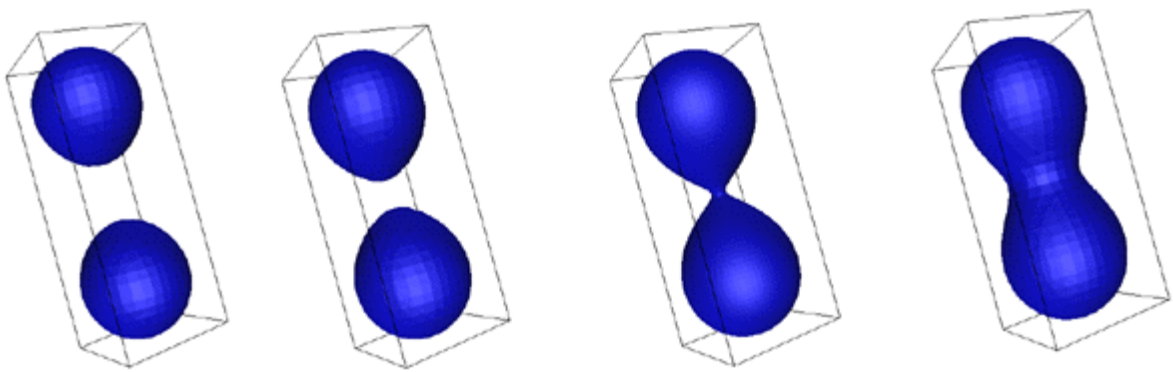


Figure 2.4 – Exemple de combinaisons de deux primitives implicites.

Une surface implicite de squelette est généralement définie par un champ de potentiel global de la forme :

$$\forall p \in \mathbb{R}^3, F(p) = \sum_{i \in N} \alpha_i f_i(\delta(p, S_i)), \quad (2.3)$$

où N est le nombre de primitives composant le modèle global, et α_i un paramètre permettant de contrôler les mélanges entre les différentes primitives.

2.2.1.1 Primitives ponctuelles

Une des prémices aux surfaces à primitives a été introduite au début des années 1980 par Blinn [Bli82], qui modélise par des *blobbies* la densité du nuage d'électrons d'une molécule. Ici, les primitives sont générées par des sources ponctuelles modélisant les atomes. La densité des électrons autour d'un atome isolé étant isotrope et fortement décroissante en fonction de la distance à l'atome considéré, Blinn la modélise par une exponentielle décroissante :

$$d_i(p) = b_i e^{-a_i \|p - c_i\|^2}, \quad (2.4)$$

où c_i est le centre de l'atome i . La densité des électrons est donc modélisée par une gaussienne centrée en c_i , de valeur maximale b_i et de variance $1/\sqrt{a_i}$. Physiquement, a_i caractérise la force d'attraction du noyau de l'atome considéré et permet de contrôler le rayon du *blob*, et b_i représente le poids de ce *blob* dans le mélange, correspondant au poids de l'atome dans la molécule.

Le concept de combinaison des primitives en sommant les potentiels a été introduit par Blinn [Bli82] dans sa modélisation de molécules. Dans une molécule, chaque électron subit

l'influence des différents atomes constitutifs et le nuage résultant n'est plus sphérique mais dépend de l'attraction de chaque atome. Blinn modélise ces nuages en utilisant les fonctions de densité d_i relatives à chacun des atomes. En supposant que les attractions des différents atomes se superposent pour chaque électron p , la densité résultante $D(p)$ est donnée par la somme des densités générées par chaque atome :

$$D(p) = \sum_i d_i(p). \quad (2.5)$$

Ce modèle présente l'inconvénient d'utiliser une fonction potentiel de support infini. Pour évaluer la valeur globale du champ potentiel en un point donné de l'espace, il faut alors calculer la somme de tous les potentiels présents, ce qui conduit à une grande complexité de calcul.

Afin de s'affranchir de cet inconvénient, Nishimura et Wyvill développent deux nouveaux modèles, respectivement les *métaballes* [NHK⁺85] et les *objets mous* [WMW86] qui s'appuient sur le même principe de superposition des densités individuelles. L'idée est d'approcher les potentiels gaussiens par des fonctions continues de support compact où la distance à l'atome à partir de laquelle la fonction potentiel s'annule est dite *rayon d'influence*. La Figure 2.5 illustre les fonctions de potentiel de ces différentes primitives ainsi que leurs définitions mathématiques.

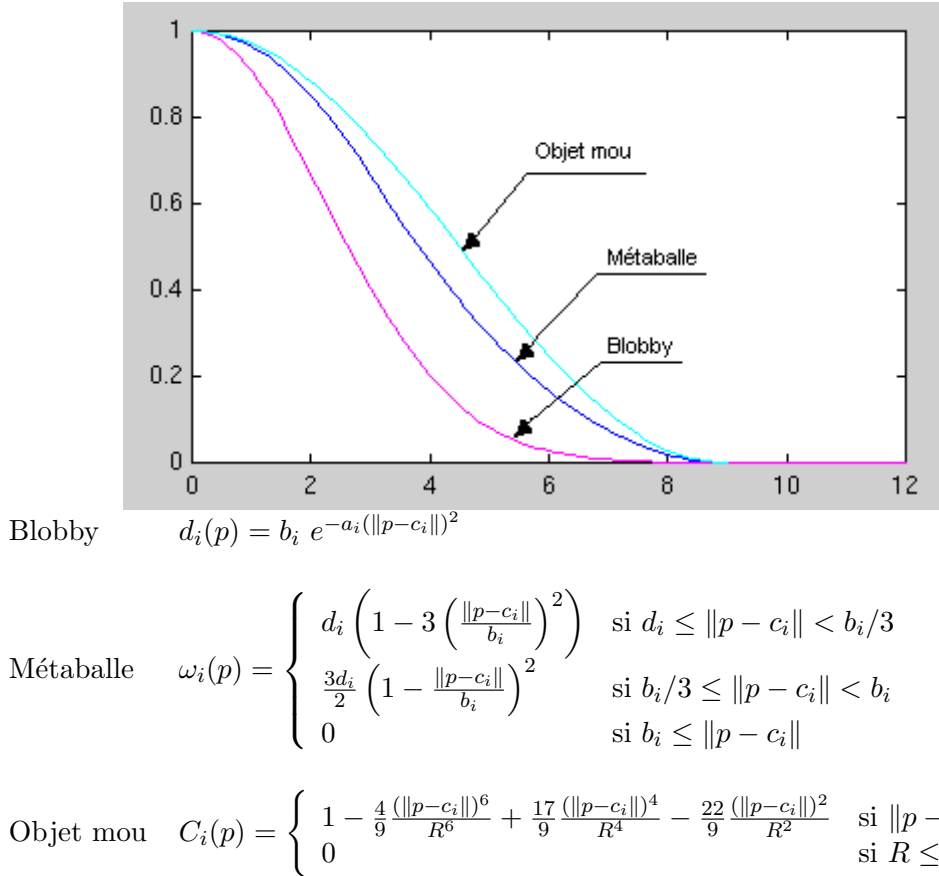


Figure 2.5 – Comparaison de quelques fonctions de potentiel centrées autour de $c_i = 0.0$ avec un rayon d'influence $R = 9.0$ pour la metaballes et l'objet mou.

L'utilisation de ces primitives ponctuelles permet de modéliser une grande diversité de formes et se montre particulièrement adaptée pour représenter des *objets organiques* tels que des animaux ou des humanoïdes. Toutefois, pour obtenir des modélisations de qualité représentant des objets de forme complexe, il faudrait augmenter considérablement le nombre de primitives utilisées. En particulier, si la densité des sources considérées n'est pas suffisamment élevée, la surface obtenue présente des "bosses" visuellement très désagréables. Cet effet

est dû à la combinaison des primitives effectuée en sommant les potentiels des différentes primitives, ce qui conduit à une apparence ondulée de la surface, les sources étant ponctuelles (Figure 2.2.1.1).

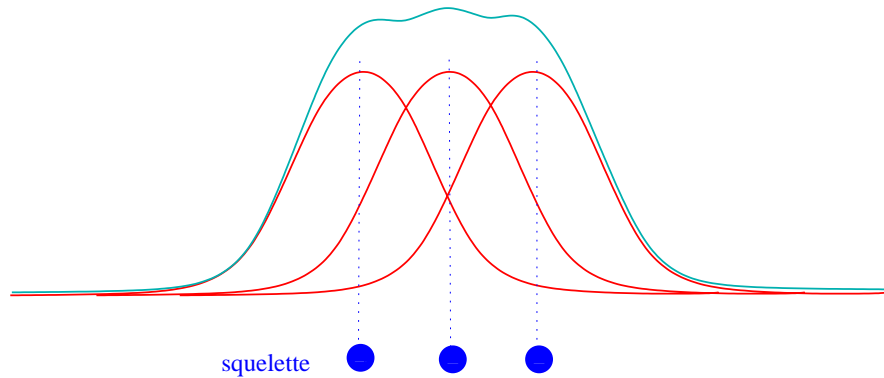


Figure 2.6 – Combinaison de primitives ponctuelles : la fonction de potentiel générée présente un aspect bossu.

Afin de réduire le nombre de primitives nécessaires, il est possible d'utiliser des primitives dont le squelette n'est plus ponctuel, mais défini par des courbes ou des éléments de surface.

2.2.1.2 Surfaces de distance

Pour évaluer la fonction de potentiel f en un point donné de l'espace, il est nécessaire de calculer la distance entre ce point et la primitive géométrique qui constitue le squelette. Les fonctions de potentiel utilisées sont de ce fait souvent exprimées en fonction de cette distance. On parle alors de surfaces de distance. Pour les primitives ponctuelles, on utilise généralement la distance euclidienne. Pour des squelettes plus complexes tels qu'un segment de droite ou un polygone, la fonction de distance est généralement donnée par le projeté du point P sur la primitive considérée.

Considérons par exemple le cas d'un segment $[AB]$ défini par :

$$s(t) = (1 - t).A + t.B, \text{ avec } t \in [0, 1]. \quad (2.6)$$

Soit $D_{[AB]}(P)$ la distance minimale entre le point P et le segment $[AB]$. Cette distance est donnée en fonction du projeté P' de P sur la droite (AB) défini par sa coordonnée paramétrique $t_{p'}$:

$$D_{[AB]}(P) = \begin{cases} \|P - P'\| & \text{si } 0 < t_{p'} < 1, \\ \|P - A\| & \text{si } t_{p'} \leq 0, \\ \|P - B\| & \text{si } t_{p'} \geq 1. \end{cases} \quad (2.7)$$

La fonction de potentiel utilise alors cette distance comme dans le cas des primitives ponctuelles pour calculer le champ de potentiel défini par de telles primitives.

En exploitant des primitives de forme plus élaborée, cette approche permet de réduire de manière significative le nombre de sources nécessaires. Toutefois, elle n'élimine pas complètement l'effet de bosse qui persiste au niveau des jonctions entre les sources. Ce phénomène s'explique par l'addition des potentiels primitifs qui s'accumulent au niveau de la jonction. Si l'on considère un squelette constitué par deux segments de droite se rejoignant au point A , les potentiels générés en un point $p \in \mathbb{R}^3$ qui se projette en A s'ajoutent, ce qui crée une bosse en raison de l'indépendance des calculs des potentiels associés.

2.2.1.3 Surfaces de convolution

Les surfaces de convolution, introduites par Bloomenthal [BS91], traitent de ce problème. Ici, le potentiel n'est pas donné en fonction d'une distance minimale entre le point P et le squelette, mais en sommant les potentiels par rapport à chaque point du squelette. Ce dernier étant continu, il s'agit en fait d'une convolution du squelette par la fonction de potentiel :

$$f(x, y, z) = \chi_s(x, y, z) * D(x, y, z) - c, \quad (2.8)$$

où χ_s est la fonction caractéristique du squelette, c est une constante, et $*$ est l'opérateur de convolution.

Les potentiels des surfaces de convolution sont définis de manière à ce que la somme des primitives ne conduise pas à l'apparition de bosses. En effet, la fonction de potentiel met en œuvre non pas une distance minimale, mais une intégration des distances à tous les points du squelette. La convolution étant un opérateur linéaire, la somme des potentiels générés par différents squelettes est alors égale à la convolution de la réunion des squelettes :

$$\sum_i F_i = \sum_i (\chi_s * D_i) = \chi_s * \sum_i D_i. \quad (2.9)$$

Les surfaces de convolution permettent ainsi d'obtenir des combinaisons où les bosses de jonction n'apparaissent plus (*cf.* Tableau 2.1), donnant de cette manière des modèles plus réalistes. Cette propriété est intéressante dans le cadre de l'animation d'objets articulés car il devient possible de segmenter le squelette d'un personnage en ajoutant des articulations sans affecter le modèle de surface résultant. Des exemples de surfaces obtenues par combinaison de primitives de convolution ainsi que les squelettes des différentes primitives utilisées sont illustrés Figures 2.7 et 2.8 .


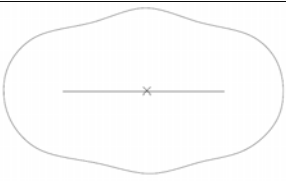

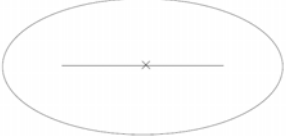
	1 segment	2 segments
Distance minimale (Surfaces de distance)		
Intégration (Surfaces de convolution)		

Tableau 2.1 – *Blending* de primitives de squelette : apparition de bosses au niveau de la jonction entre deux éléments du squelette.

Les surfaces de convolution semblent les surfaces implicites les plus appropriées pour modéliser de manière intuitive des modèles 3D de forme complexe, articulés et d'aspect lisse. Cependant, le produit de convolution nécessite des calculs d'intégration ce qui implique des temps de calcul prohibitifs notamment pour des applications en temps réel, avec des personnages réalistes dont le squelette est complexe.

2.2.2 Techniques de visualisation

Pour afficher une surface implicite, il faut trouver un ensemble de points P de coordonnées (x_p, y_p, z_p) vérifiant l'équation 2.1. Étant donnée la complexité des fonctions de potentiel

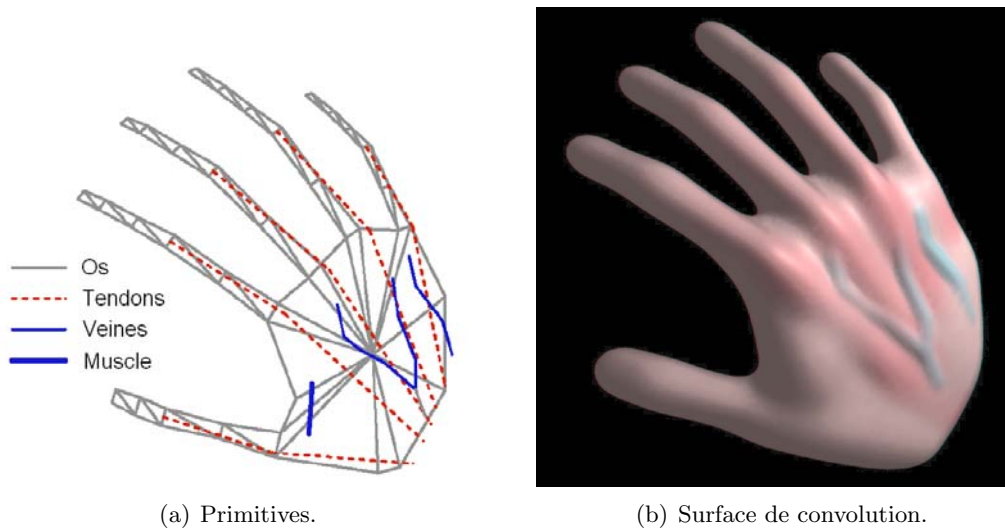


Figure 2.7 – Main modélisée par surfaces de convolution [BS91].

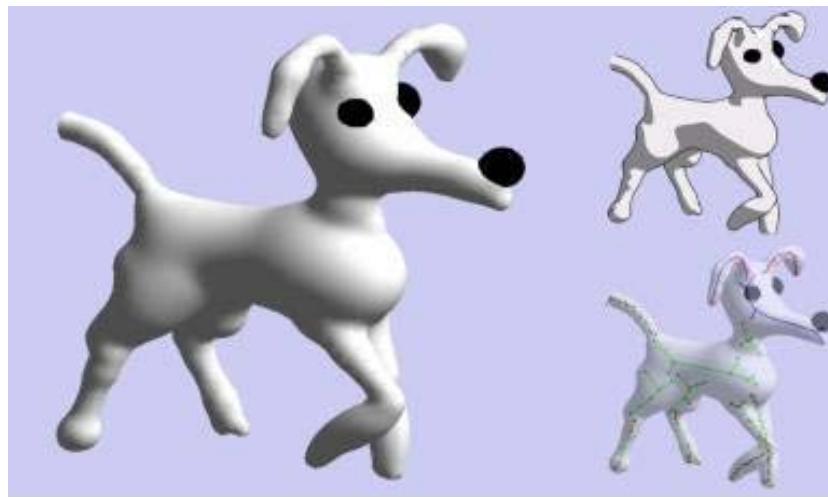


Figure 2.8 – Chien modélisé par surfaces de convolution [TZF03].

généralement utilisées pour modéliser des objets de forme libre, il est difficile d'obtenir cet ensemble de points de manière immédiate comme dans le cas des maillages polygonaux ou des surfaces paramétriques. Les techniques utilisées pour visualiser des surfaces implicites recourent alors généralement à des méthodes d'approximation.

Les surfaces implicites peuvent être directement tracées par *lancer de rayons* [Gla89], [FvDFH97]. En fonction des paramètres de la caméra et de la résolution d'affichage, des rayons sont lancés dans la direction de vue. L'expression des surfaces étant implicite, la détermination d'un point d'intersection entre un rayon et la surface est effectuée par dichotomie de manière à localiser le point d'intersection à une tolérance d'erreur près.

Étant donnée la complexité de l'équation à résoudre pour chaque pixel de l'image, le rendu des surfaces implicites par lancer de rayon reste coûteux en temps de calcul et est encore non envisageable dans le cadre d'applications interactives.

Certaines applications nécessitent une représentation polygonale. Celle-ci peut être construite à l'aide de la technique des *marching cubes* [LC87] qui échantillonne la fonction implicite aux sommets d'un maillage tridimensionnel régulier. Les valeurs de la fonction implicite sont alors calculées aux sommets du maillage, puis interpolées sur les arêtes de celui-ci pour estimer la

position de leur intersection avec la surface. Des polygones sont enfin construits dans chaque cellule du maillage à partir des points ainsi obtenus.

Cette méthode utilise un échantillonnage régulier de l'espace. Il devient alors nécessaire d'effectuer un raffinement important de cet échantillonnage afin que tous les détails de la surface soient représentés sur le maillage final ce qui augmente d'autant les temps de calcul.

2.2.3 Techniques de modélisation

Compte tenu des difficultés liées à la visualisation et au contrôle des surfaces implicites, un petit nombre d'outils de modélisation les ont introduites dans leurs interfaces. Toutefois, 3DS Max [MAX] a intégré récemment les metaballes dans le but de modéliser des phénomènes physiques tels que les écoulements de fluides ou le feu.

Au niveau de la recherche, quelques propositions de systèmes de modélisation fondés sur les surfaces implicites ont commencé à apparaître dans les années 1980, mettant en oeuvre essentiellement des surfaces algébriques. En particulier, Wyvill [WMW86] développe un système de modélisation à l'aide de quadriques comme primitives. La forme de ces dernières est contrôlée par les paramètres les définissant. Les objets sont créés en utilisant des outils de type CSG telles que réunion ou intersection.

Dans les années 1990, l'intérêt s'est porté vers les surfaces générées par un squelette. En effet, celles-ci sont plus adaptées à des applications d'animation que les surfaces algébriques, difficilement contrôlables pour l'utilisateur qui n'a pas une compréhension intuitive de l'effet de changement des paramètres les définissant.

Une première famille d'approches définit les squelettes dans un environnement d'interaction 3D. Le *blending* entre les différentes primitives ainsi que la forme des surfaces générées par celles-ci sont alors contrôlés à l'aide de paramètres. Tsingos et Gascuel [TG95] développent un modèleur fondé sur des surfaces de squelette où le contrôle des formes est effectué à l'aide de 3 paramètres associés à chaque squelette : un rayon d'influence au-delà duquel le potentiel s'annule, une épaisseur, et une raideur qui permet de contrôler le mélange des potentiels. Sherstyuk [She99] met en place un système similaire à partir de surfaces de convolution, où la génération des primitives peut être faite soit à la main par l'utilisateur, soit à l'aide de méthodes procédurales, soit en combinant les deux.

Dans les deux cas, l'interactivité de l'application est assurée en visualisant une forme simplifiée des modèles. Ainsi, dans [TG95] la visualisation est-elle effectuée à l'aide d'un ensemble d'éléments de surface dits "graines" se déplaçant aux alentours de la surface, et visualisés par des triangles. Le faible nombre de ces graines garantit une représentation interactive, mais approximative du modèle. Dans [She99], l'auteur propose de visualiser une forme plus simplifiée, qui est une surface d'épaisseur fixe autour des squelettes (un segment de droite donne par exemple un cylindre). La surface implicite finale n'est alors calculée qu'à la fin de la procédure de création à l'aide d'un tracé de rayon, en particulier pour les modèles utilisant un nombre important de primitives.

Dans [HAC03], les auteurs proposent une approche multirésolution où les surfaces peuvent être visualisées selon différents niveaux de détail. Celles-ci sont obtenues grâce aux squelettes générés par des courbes de subdivision. Les surfaces sont ensuite obtenues par convolution, ce qui assure un aspect lisse à tous les niveaux de détail.

Barbier *et al.* [BGA04] présentent un système de modélisation et d'animation de surfaces implicites fondé sur une structure hiérarchique de primitives gérée par un arbre dit *BlobTree*. Les opérations de combinaison des primitives telles que le *blending*, la réunion ou l'intersection, sont gérées par les nœuds de l'arbre, alors que les feuilles de celui-ci constituent les squelettes des différents éléments du modèle. L'interface propose un ensemble de primitives de base à partir desquelles l'utilisateur peut créer ses objets.

Les modélisations physiques ont été utilisées dans le cadre d'applications haptiques [HQ04] ou de sculpture virtuelle [KPA03] où les forces appliquées par des outils simples agissent sur un volume implicite qui est sculpté en conséquence.

Une méthode de création par déformation de formes simples est proposée dans [ACWK04] où les surfaces subissent des déformations sous la contrainte de conservation du volume. Les forces appliquées sont dérivées des déplacements de points de la surface suivant un chemin précisé par l'utilisateur.

Enfin, un modèle de création fondé sur une interactivité 2D est proposé dans [AGB04] qui s'inspire de l'interface Teddy [IMT99] pour déduire un squelette à partir des dessins 2D. Des surfaces *blobbies* sont ensuite générés, centrées en les points du squelette 2D.

Une approche similaire fondée sur les surfaces de convolution est décrite dans [TZF03] où les images d'entrée sont des silhouettes correspondant à des plans de coupe du personnage (Figure 2.8).

De manière générale, on constate que l'interactivité constitue le problème principal auquel se heurtent encore les surfaces implicites, et ce, sous deux aspects : la rapidité des algorithmes de visualisation d'une part, et le contrôle utilisateur d'autre part. Aussi les approches existant dans la littérature essaient-elles de résoudre ces deux difficultés en choisissant des fonctions de potentiel plus intuitives et des algorithmes de visualisation dont le calcul est de plus en plus rapide.

2.2.4 Conclusion

En combinant par superposition plusieurs formes élémentaires, les fonctions implicites permettent de modéliser des objets complexes de manière intuitive. Toutefois, le principal inconvénient des surfaces implicites est lié aux aspects de visualisation, nécessitant un temps de calcul important. En effet, les approches actuellement utilisées sont inexploitable dans le cadre d'une application interactive.

Par ailleurs, l'animation et la déformation de ces surfaces sont aussi peu intuitives puisqu'elles nécessitent soit la modification des fonctions de potentiel, soit le déplacement d'un grand nombre de sources. L'influence d'une modification du champ de potentiel sur la surface isopotentielle est en effet difficilement prévisible.

Enfin, il faut gérer les *mélanges indésirables* qui surviennent dès que deux primitives se rapprochent l'une de l'autre (par exemple, les doigts d'une main peuvent se confondre).

Au-delà de ces aspects pratiques, les surfaces implicites donnent toujours naissance à des modèles lisses qui conviennent pour des objets organiques, mais pas pour des objets rigides tels que des meubles. L'aspect arrondi des modèles semble restrictif pour une application de créativité telles que les dessins animés qui doivent pouvoir représenter tout type d'objet.

2.3 Surfaces paramétriques

Les surfaces paramétriques [Far96] sont définies par rapport à un repère 2D relatif à la surface S modélisée. La position 3D d'un point P de S , de coordonnées relatives u et v , s'exprime à l'aide de trois fonctions f , g , et h , définies sur un domaine D de \mathbb{R}^2 , permettant de calculer ses coordonnées cartésiennes (x, y, z) à partir de sa position relative $(u, v) \in D$:

$$S = \{(x, y, z) : x = f(u, v), y = g(u, v), z = h(u, v)\}, \quad (2.10)$$

La définition générale d'une surface paramétrique peut donc être considérée comme une déformation du plan.

Historiquement, l'intérêt pour les surfaces paramétriques a été suscité par l'introduction de la notion de *courbe à pôles* [Cas63], dont l'innovation consiste en la manière de définir une

courbe, et ultérieurement une surface. En effet, cette technique permet de créer des modèles lisses à l'aide d'un nombre restreint de points, et donc avec une interaction réduite.

Une surface à pôles est décrite par une équation de la forme :

$$S(u, v) = \sum_i \sum_j P_{i,j} f_{i,j}(u, v), \quad (2.11)$$

où $P_{i,j} \in \mathbb{R}^3$ sont les points de contrôle (ou pôles) et les $f_{i,j}(u, v)$ sont des *fonctions de base* caractérisant l'influence des différents points de contrôle en chaque coordonnée relative (u, v) de la surface S .

Parmi les surfaces paramétriques, les plus utilisées en CAO ¹ sont les surfaces définies par un produit tensoriel de deux courbes à pôles, telles que les surfaces de Bézier, les surfaces B-splines, et les NURBS (*Non Uniform Rational B-Splines*). Les pôles de ces surfaces présentent une certaine régularité, étant définis selon une *grille de contrôle* rectangulaire. Le nombre de points de contrôle selon les coordonnées u et v est donné par chacune des courbes génératrices. Les surfaces générées par un produit tensoriel s'écrivent sous la forme générale suivante :

$$S(u, v) = \sum_i \sum_j P_{i,j} f_i(u) f_j(v). \quad (2.12)$$

La forme et les propriétés d'une surface à pôles par produit tensoriel dépend des fonctions de base choisies. Ces fonctions sont généralement définies par morceaux, de manière à ce que l'influence d'un pôle donné soit locale. L'utilisation de fonctions polynomiales a permis d'assurer une continuité des formes obtenues, en fonction du degré des polynômes considérés. Parmi les surfaces à pôles exploitant des fonctions de base polynomiales, mentionnons les surfaces de Bézier, les B-splines et les NURBS.

2.3.1 Surfaces de Bézier

Les surfaces de Bézier ont été introduites par les ingénieurs français Paul de Faget de Casteljau et Pierre Bézier [Far96], [Cas63]. Pour ces surfaces (Équation 2.13), les fonctions de base sont données par les polynômes de Bernstein :

$$S(u, v) = \sum_i \sum_j P_{i,j} B_i^n(u) B_j^m(v), \quad (2.13)$$

où :

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i. \quad (2.14)$$

Les polynômes de Bernstein étant non nuls pour tout $t \in [0, 1]$, la position d'un point donné de la surface dépend de l'ensemble des points de contrôle. Ainsi, pour déterminer une courbe de Bézier interpolant un ensemble de n points, est-il nécessaire d'imposer un degré $n - 1$. De plus, le déplacement d'un des points de contrôle influe sur toute la courbe/surface. Cette globalité de la représentation constitue un inconvénient pour des applications de création de contenus 3D où l'on souhaite disposer de mécanismes de contrôle local de la surface.

2.3.2 Surfaces B-splines

Les courbes et surfaces B-splines permettent de s'affranchir de cet inconvénient, en utilisant une définition polynomiale par morceaux, ce qui réduit la zone d'influence des points de contrôle. Ici, le support des fonctions de base est restreint à un voisinage défini par le degré de la courbe considérée ainsi que par un *vecteur de nœuds*.

¹Conception Assistée par Ordinateur

Soient p le degré d'une courbe B-spline, et $n + 1$ le nombre de points de contrôle la définissant. Le vecteur des nœuds est défini comme un ensemble de paramètres ordonnés compris entre deux valeurs fixées a et b :

$$U = \{a, \dots, a, u_{p+1}, \dots, u_{r-p-1}, b, \dots, b\} \quad (2.15)$$

où $r = n + p + 1$, et a et b sont répétés $p + 1$ fois. Notons que le plus souvent, $a = 0$ et $b = 1$.

Pour une coordonnée relative $u \in [a, b]$, la fonction de base B-spline est définie de manière récursive, en exploitant le partitionnement du domaine paramétrique $[a, b]$ donné par le vecteur des nœuds :

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{si } u_{i-1} < u \leq u_i \\ 0 & \text{sinon} \end{cases}, \\ N_{i,k}(u) &= \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}. \end{aligned} \quad (2.16)$$

Par définition, les fonctions de bases B-splines $N_{i,p}(u)$ sont nulles en dehors de l'intervalle $[u_i, u_{i+k+1})$. Les courbes B-splines ainsi définies sont polynomiales par morceaux. La Figure 2.9 illustre la définition récursive des fonctions de base B-splines, ainsi que leur caractère local. Notons que le support des fonctions de bases augmente avec le degré. $N_{i,2}$ est non nulle sur l'intervalle $[u_i, u_{i+3}]$, tandis que $N_{i,3}$ est non nulle sur l'intervalle $[u_i, u_{i+4}]$.

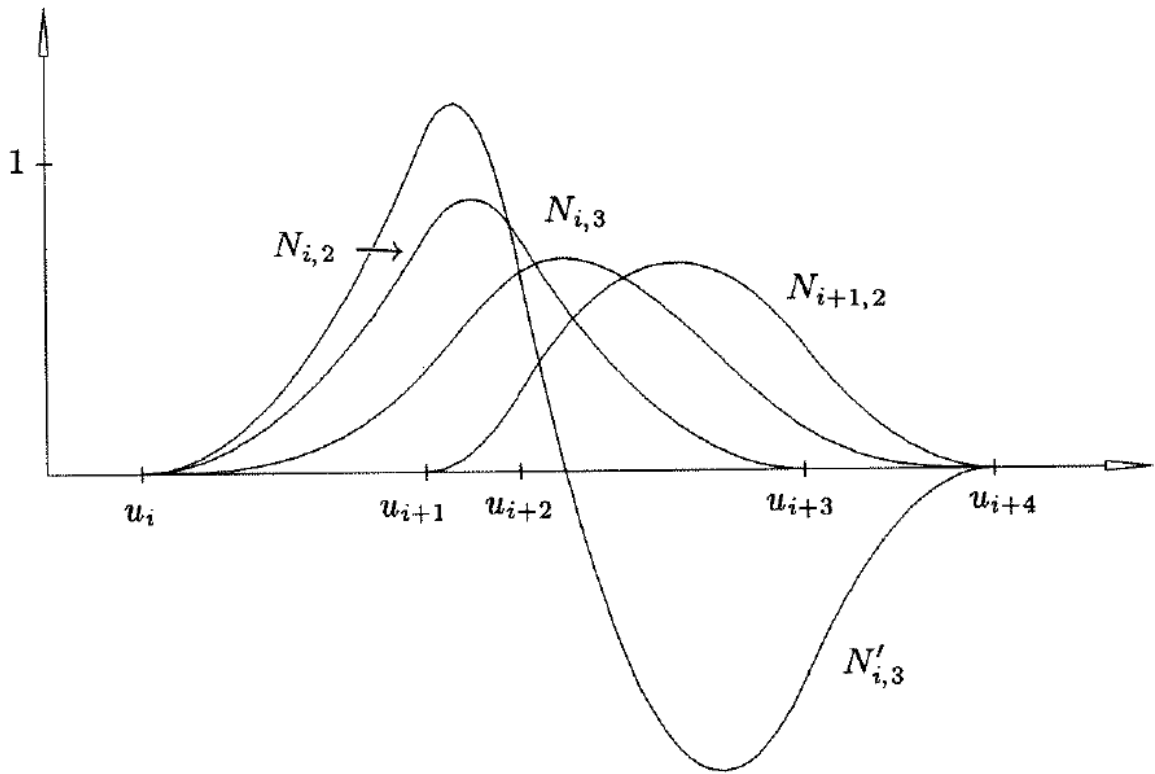


Figure 2.9 – Définition récursive des fonctions de base B-splines.

Les surfaces B-splines sont définies de manière séparable comme le produit tensoriel de deux courbes B-splines comme suit :

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) P_{i,j}, \quad (2.17)$$

où $n + 1$ et $m + 1$ sont les nombres de points de contrôle des courbes B-splines génératrices, et p et q sont leurs degrés. Ici, les fonctions de base $N_{i,p}$ et $N_{j,q}$ de degrés p et q respectivement

sont définies sur les vecteurs des nœuds :

$$\begin{aligned} U &= \{0, \dots, 0, u_{p+1}, \dots, u_{r-p-1}, 1, \dots, 1\}, \\ V &= \{0, \dots, 0, v_{q+1}, \dots, v_{s-q-1}, 1, \dots, 1\}, \end{aligned}$$

où $r = n + p + 1$ et $s = m + q + 1$.

La caractéristique locale des fonctions de base B-splines est intéressant : il facilite l'évaluation de la surface en un point (u, v) donné, ainsi que le contrôle de l'emplacement des points de contrôle dont la zone d'influence est restreinte en fonction du degré de la surface.

Les surfaces B-splines permettent de modéliser des objets lisses et de les déformer de manière intuitive. Afin d'accroître la flexibilité et le pouvoir de représentation des modèles B-splines, une généralisation a été introduite. Il s'agit des modèles NURBS, décrits dans le paragraphe suivant.

2.3.3 Surfaces NURBS

Les surfaces NURBS [PT95] généralisent les modèles B-splines en introduisant un degré de liberté supplémentaire, qui permet une plus grande flexibilité pour le même ensemble de points de contrôle. Le principe consiste à attribuer aux points de contrôle $P_{i,j}$ des poids $\omega_{i,j}$. Ces poids sont des coefficients réels déterminant le degré d'influence des points de contrôle sur la courbe.

Une surface NURBS donnée par le produit tensoriel de deux courbes de degrés p et q est alors définie par :

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \omega_{i,j} P_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \omega_{i,j}} \quad (2.18)$$

Intuitivement, les poids contrôlent l'attraction de la surface par le point de contrôle considéré. Ainsi, plus la valeur d'un poids $\omega_{i,j}$ augmente, plus les points de la surface de coordonnées relatives $(u, v) \in [u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$ s'approchent du point de contrôle $P_{i,j}$.

Les surfaces NURBS sont définies sur un domaine rectangulaire intrinsèquement lié à la paramétrisation. Cette propriété constitue une contrainte forte car ce type de surface ne permet pas de modéliser des objets de forme libre, de topologie arbitraire tels que des personnages articulés présentant des embranchements. Une solution consiste à créer des surfaces paramétriques par morceaux.

2.3.4 Surfaces paramétriques par morceaux (*patches* paramétriques)

Le principe des modèles paramétriques par morceaux consiste à juxtaposer des morceaux de surfaces paramétriques dits *patches*, pour modéliser la surface globale de l'objet. Quant au type de patches, tout modèle paramétrique peut être considéré. Il est donc possible de construire des patches de Bézier, B-splines, ou encore NURBS.

Les patches de Bézier [Sar90], [Baj94] sont les plus utilisés dans le cadre de la modélisation de personnages animés en raison de leur simplicité, facilité de manipulation et de contrôle.

Un exemple d'un patch de Bézier cubique est illustré Figure 2.10. Le patch est ici déterminé par un ensemble de $4 \times 4 = 16$ points de contrôle. Notons que chaque patch interpole les coins du maillage de contrôle, qui correspondent aux points de contrôle extrémité de chaque courbe de bord.

Cette technique de modélisation par patches nécessite d'imposer des conditions de compatibilité et de continuité qui se traduisent par des contraintes sur la configuration des points de contrôle. Pour une continuité C^0 , les courbes de bord doivent coïncider en tout point.

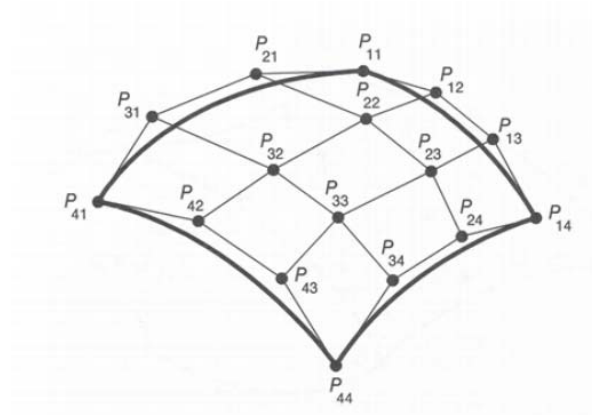


Figure 2.10 – Patch de Bezier : les courbes de bord sont des courbes de Bézier définies par les points de contrôle de bord, le patch interpole les points de contrôle des coins.

Cela implique que les points de contrôle et les vecteurs des nœuds doivent coïncider dans les deux patches (Figure 2.11(a)). La continuité C^1 conduit à une contrainte encore plus forte, qui consiste en la colinéarité des points de contrôle de part et d'autre des courbes de bord (Figure 2.11(b)). Au niveau des coins des patches, cela impose une coplanarité des 4 points de contrôle voisins. Ces fortes contraintes rendent difficile la modélisation d'objets de forme complexe qui nécessitent un nombre élevé de patches.

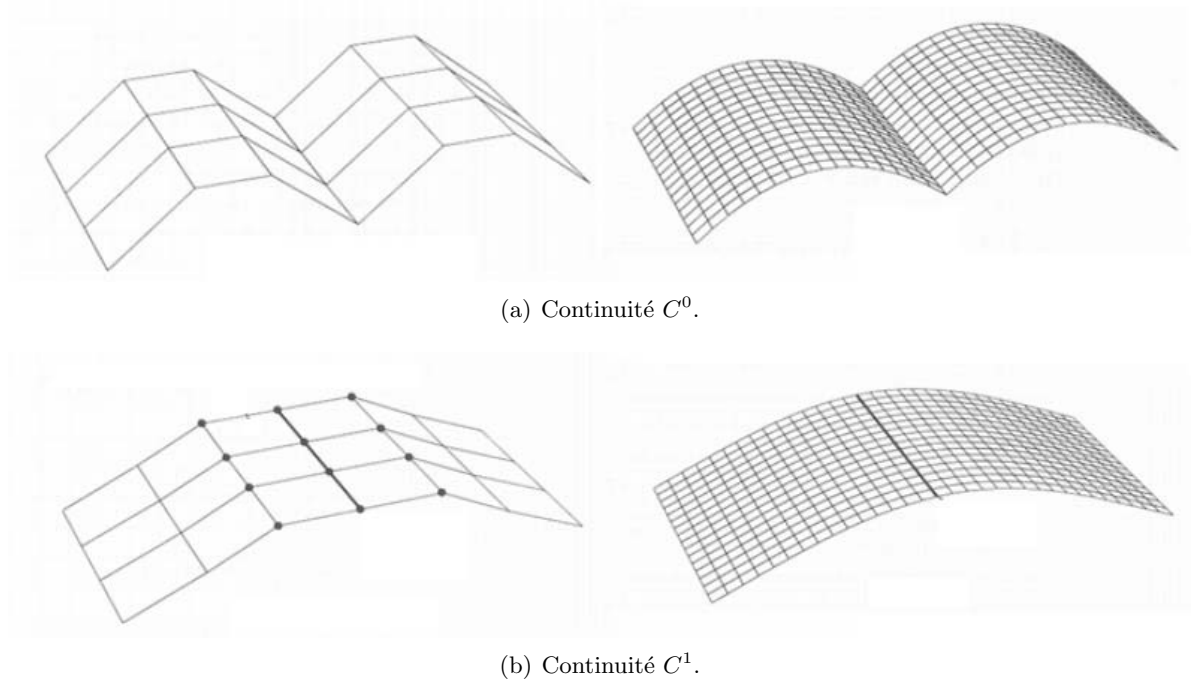


Figure 2.11 – Juxtaposition de patches de Bézier : conditions de continuité.

Une autre limitation concernant ce type d'approche est liée à la visualisation, qui est réalisée à l'aide d'une polygonisation de la surface paramétrique. Contrairement aux surfaces implicites, la polygonisation des surfaces paramétriques peut être effectuée de façon directe, en échantillonnant le domaine paramétrique selon une grille rectangulaire. Celle-ci peut correspondre par exemple à un simple échantillonnage uniforme du domaine paramétrique. Toutefois, pour pouvoir rendre des éléments de surface de forte courbure, il est nécessaire d'appliquer un pas d'échantillonnage suffisamment fin ce qui alourdit la représentation polygonale obtenue.

Les approches de subdivision adaptative [DM95][Sug02] visent notamment à pallier cet inconvénient en adoptant une procédure d'échantillonnage non uniforme qui prend en compte les caractéristiques locales de la surface. Elles permettent en outre d'approcher une surface paramétrique avec contrôle de l'erreur maximale d'approximation.

Cette technique conduit en revanche à l'apparition de déconnexions entre des patches voisins dont le niveau de subdivision n'est pas identique (problème de *cracking*). Ce problème est illustré Figure 2.12 où deux niveaux différents de subdivision ont été appliqués à deux patches voisins.

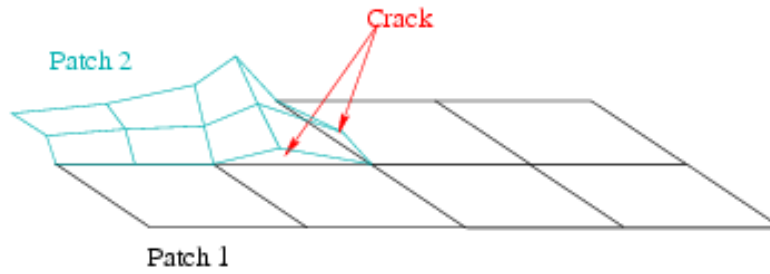


Figure 2.12 – Visualisation de patches paramétriques : la subdivision adaptative conduit à l'apparition de cracking.

Pour éviter les limitations liées aux contraintes de continuité au niveau des bords, une solution est apportée par les modèles *trimmed NURBS*.

2.3.5 Trimmed NURBS

Les modèles *trimmed NURBS* [Cas87], [PT98] proposent également une représentation de surface par patches paramétriques multiples. Afin d'éviter la gestion locale des aspects de continuité aux bords, le principe consiste à réaliser le découpage et le raccordement des différents patches selon un ensemble de courbes définies dans le domaine paramétrique et appelées courbes de découpage (*trimming curves*).

Cette méthode permet en particulier des patches avec des trous et avec des bords de forme arbitraire. Chaque courbe de découpage est orientée de manière à définir sur le patch deux domaines, l'un intérieur et l'autre extérieur. Au moment de la visualisation, seul un des deux domaines sera rendu. La Figure 2.13 illustre le choix de la partie visualisée en fonction de l'orientation de la courbe de découpage.

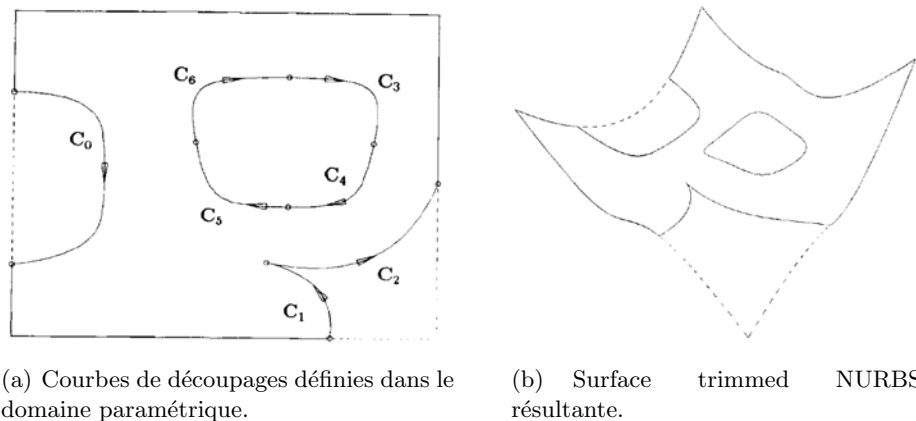


Figure 2.13 – Trimming NURBS : découpage de la surface.

Une courbe de découpage c (*trimming curve*) est une courbe géodésique définie sur le

domaine paramétrique D du patch de surface S considéré

$$c(t) : [0, 1] \rightarrow D. \quad (2.19)$$

Elle est alors exprimée par :

$$c(t) = (u(t), v(t)). \quad (2.20)$$

Soient deux patchs de surface différents $S_i(u_i, v_i)$, $i \in 1, 2$. Pour assembler ces deux *surfaces primaires* selon deux courbes de découpage, un opérateur de *blending* [Fil89], [KE95] est défini en fonction des deux courbes de découpage c_i et des deux courbes T_i décrivant les tangentes le long des courbes de c_i .

On définit par composition les courbes 3D dites *courbes rails* (*rail curves*) :

$$C_i(t) = S_i(c_i(t)) = S_i(u(t), v(t)). \quad (2.21)$$

Une surface de *blending* est alors construite comme surface cubique interpolante d'Hermite (Figure 2.14) :

$$H(s)(t) = h_{00}(s)C_1(t) + h_{01}(s)C_2(t) + h_{10}(s)T_1(t) + h_{11}(s)T_2(t), \quad (2.22)$$

où h_{00}, h_{01}, h_{10} et h_{11} sont les fonctions de base de l'interpolant cubique hermitien, définies par :

$$\begin{cases} h_{00}(s) = s^2(2s - 3) + 1, \\ h_{01}(s) = s^2(3 - 2s), \\ h_{10}(s) = s(s - 1)^2, \\ h_{11}(s) = s^2(s - 1). \end{cases} \quad (2.23)$$

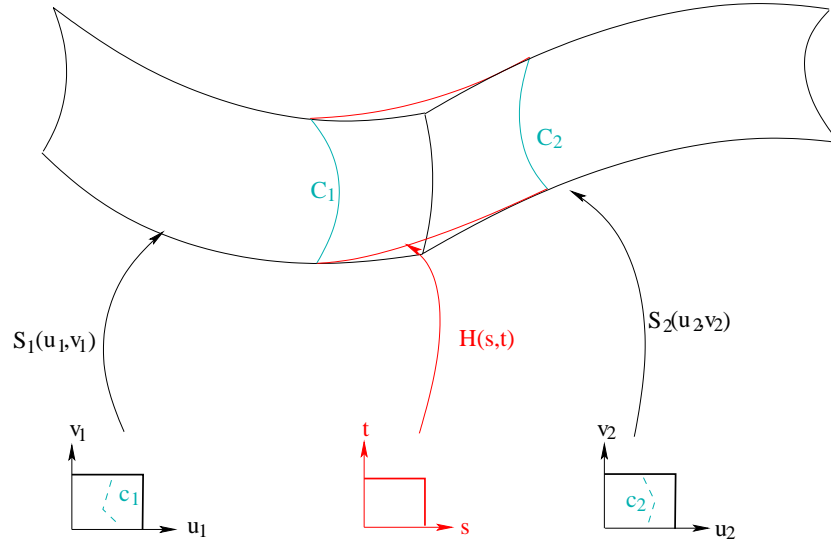


Figure 2.14 – Surface de *blending* définie à partir de courbes de découpage

Notons que ces fonctions de base permettent de vérifier les conditions aux bornes $s = 0$ et $s = 1$ pour tout $t \in D$: $H(0) = C_1$, $H(1) = C_2$, $H'(0) = T_1$ et $H'(1) = T_2$.

Les techniques rapides de visualisation de surfaces NURBS utilisent généralement une procédure de polygonisation. Dans le cadre des trimmed NURBS, cette méthode nécessite l'évaluation des courbes composées C_i , qui est coûteuse en temps de calcul. En outre, ces courbes étant définies dans le domaine paramétrique des patchs de surface, la polygonisation s'effectue de manière indépendante pour les surfaces primaires d'une part et la surface de blend d'autre part, ce qui conduit à l'apparition de cracks au niveau des courbes rails C_i . En effet, ces courbes de découpage sont échantillonnées régulièrement sur la surface de blend, selon le paramètre t . Sur les surfaces primaires en revanche, ces courbes sont évaluées par composition, ce qui ne permet pas d'effectuer directement l'échantillonnage des surfaces et des courbes rails.

2.3.6 Conclusion

Contrairement aux maillages, qui sont des approximations linéaires de surface, les modèles paramétriques fournissent des approximations polynomiales d'ordre supérieur qui conduisent à une diminution du nombre de points de contrôle nécessaires ainsi qu'à un degré de continuité supérieur. Cela est intéressant pour un certain nombre d'applications telles que la modélisation 3D où les points sont spécifiés par l'utilisateur. La Figure 2.15 représente un exemple de personnage réalisé à l'aide de deux surfaces (une pour la tête et une pour les cheveux) définies comme un produit tensoriel de B-splines.

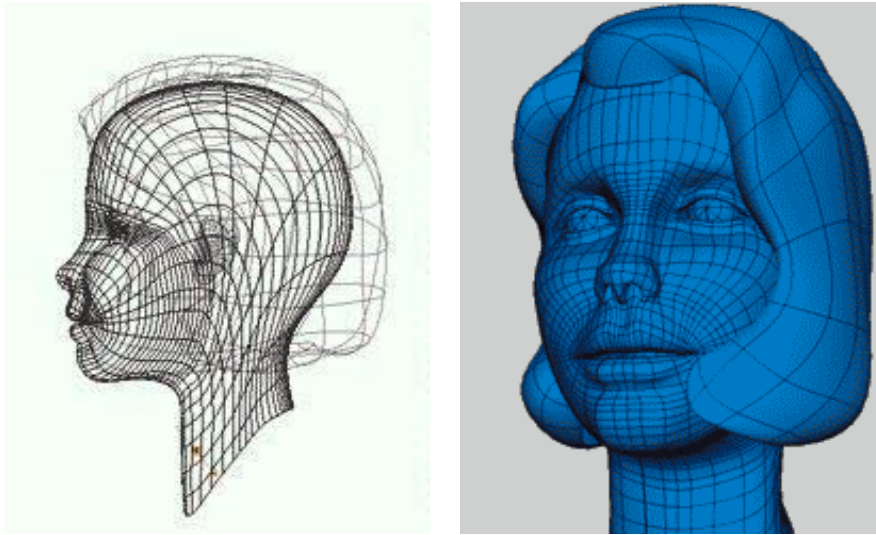


Figure 2.15 – Exemple de surface paramétrique, extrait de [Bir].

Par ailleurs, les surfaces paramétriques présentent l'avantage de produire des modèles définis de manière précise en tout point, puisqu'elles sont exprimées par une formule mathématique pouvant être évaluée en chaque couple de coordonnées relatives (u, v) . Cette propriété permet d'obtenir une résolution infinie des modèles créés. De plus, la forme paramétrique est intéressante pour des applications comme le plaquage de textures, et de manière générale pour le calcul de caractéristiques de surface (normale, dérivées, courbure, etc.) en un point donné.

Toutefois, des calculs tels que l'appartenance d'un point de l'espace à une surface paramétrique, les intersections avec une surface ou la distance à une surface sont plus compliqués, ce qui constitue un inconvénient pour des applications comme la détection de collision ou le calcul d'intersections entre différents modèles paramétriques.

Enfin, les surfaces paramétriques ne permettent pas de modéliser des formes complexes en utilisant un seul domaine paramétrique. L'utilisation de modèles plus élaborés comme les patches paramétriques ou les trimmed NURBS impose des contraintes pour obtenir des surfaces continues. Cette contrainte représente une difficulté au niveau de la visualisation où des cracks apparaissent. Ces modèles par patch semblent donc peu appropriés à des applications d'animation, où le maintien de la continuité C^1 sur toutes les séquences est important.

2.4 Surfaces de subdivision

Une surface de subdivision S est définie comme la limite d'une suite convergente de maillages $\{M_n\}_{n \in \mathbb{N}}$:

$$S = \lim_n M_n. \quad (2.24)$$

Le maillage initial M_0 est appelé *maillage de base* ou *maillage de contrôle*. Pour $n \geq 1$, le maillage M_n , appelé niveau de résolution n , est obtenu à partir du maillage M_{n-1} par une procédure de subdivision qui consiste en deux étapes :

- insertion de nouveaux points,
- filtrage passe-bas des coordonnées.

Historiquement, les surfaces de subdivision trouvent leur origine dans les techniques de sectionnement de coins (*corner cutting*) [ZS00] qui permettaient d'obtenir des courbes lisses à l'aide d'une procédure itérative de raffinement. Une telle technique, correspondant à l'algorithme de subdivision de Chaikin [Cha74] est illustrée Figure 2.16. A partir d'une courbe polygonale initiale (Figure 2.16(a)), une suite de courbes est itérativement construite en insérant sur chaque segment $[P_i, P_{i+1}]$ de la courbe courante deux points Q_i et R_i tels que :

$$\frac{\|P_i Q_i\|}{\|P_i P_{i+1}\|} = \frac{1}{4} \quad \text{et} \quad \frac{\|P_i R_i\|}{\|P_i P_{i+1}\|} = \frac{3}{4}. \quad (2.25)$$

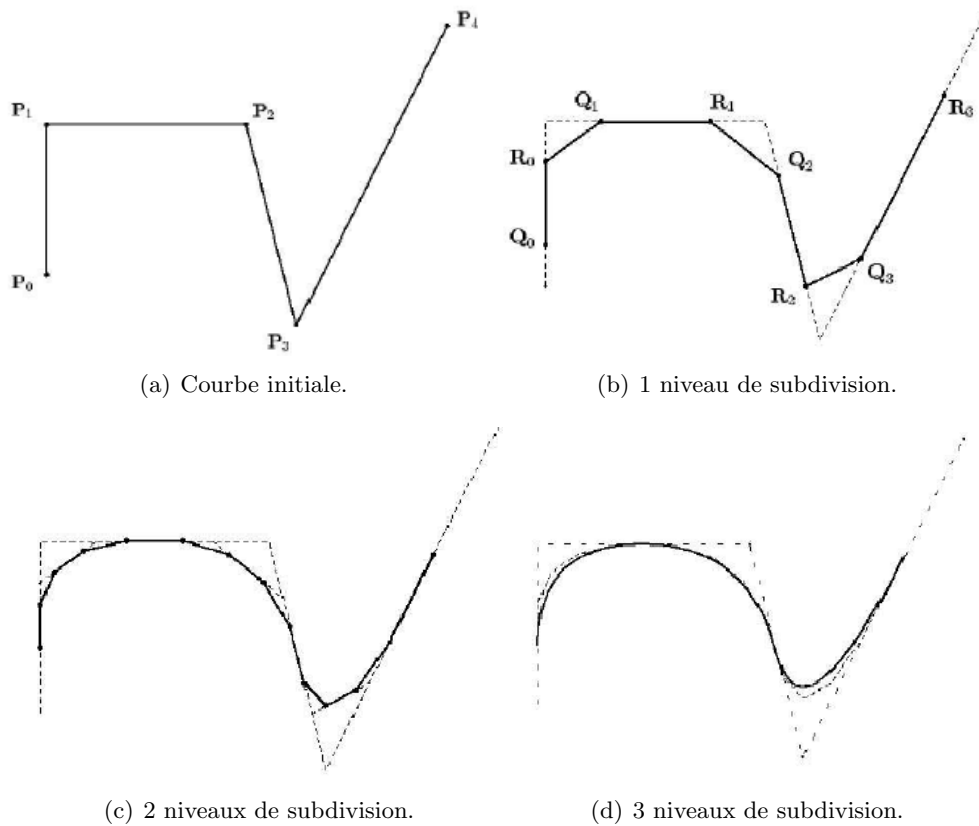


Figure 2.16 – Subdivision d'une courbe selon l'algorithme de Chaikin.

Les points P_i sont ensuite éliminés, la nouvelle courbe étant définie par la suite de points $\{Q_0, R_0, \dots, Q_i, R_i, Q_{i+1}, R_{i+1}, \dots\}$.

Dans [Rie75], Riesenfeld démontre que l'algorithme de Chaikin converge vers des B-splines quadriques uniformes. De manière plus générale, De Boor [Boo87] démontre que, en imposant quelques conditions sur le procédé de coupure des coins, les méthodes de sectionnement des coins convergent toujours vers des courbes lisses.

Les surfaces de subdivision, initialement introduites par Catmull et Clark [CC78] et Doo et Sabin [DS78], ont permis de dépasser la limitation principale des surfaces NURBS puisqu'elles peuvent modéliser une topologie arbitraire. Le principe de construction d'une surface de subdivision repose toujours sur un ensemble de points de contrôle, formant un maillage de base. Ce maillage – pouvant être considéré comme une représentation à basse résolution de la

surface étudiée – est ensuite raffiné selon des règles prédéfinies. L’objectif est de produire une surface visuellement équivalente à une surface paramétrique.

Les surfaces de subdivision représentent une solution flexible pour la modélisation de surfaces. Elles constituent un compromis entre les maillages et les surfaces paramétriques, et offrent ainsi la plupart des avantages de ces modèles traditionnels. En effet, les surfaces de subdivision jouissent de la généralité topologique des maillages, sans subir les contraintes imposées pour les surfaces paramétriques au niveau de la gestion des frontières entre les différents patches. Contrairement aux surfaces paramétriques, les surfaces de subdivision peuvent présenter une densité de sommets variable. Elles permettent ainsi de créer des niveaux de détail différents sur un même objet, tout en conservant une apparence lisse et en minimisant le nombre de sommets nécessaires.

La Figure 2.17 présente différents niveaux de résolution associés à une surface de subdivision.

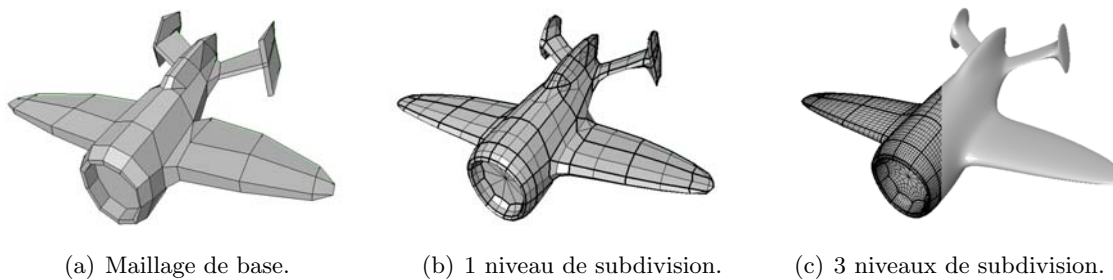


Figure 2.17 – Exemple de surface de subdivision.

Les différents schémas de subdivision de la littérature se distinguent selon les critères suivants :

- *le type du maillage généré* : triangulaire ou quadrilatéral par exemple,
- *la procédure d’insertion de points* : elle peut être effectuée soit en subdivisant les facettes du maillage (*face split*), on parle alors de *subdivision primale* (Figure 2.18(a)), soit en dupliquant les sommets (*vertex split*), auquel cas on parle de *subdivision duale* (Figure 2.18(b)),
- *le schéma de subdivision interpolatif ou approximatif* : dans un schéma d’interpolation, les sommets du maillage de base ainsi que ceux introduits par les différentes itérations de subdivision appartiennent à la surface limite. Une fois introduits, leur position ne change plus au cours des itérations suivantes. En revanche, dans un schéma d’approximation, la position des sommets est modifiée à chaque itération. Notons que cette classification est uniquement appliquée aux schémas de subdivision primaux, car dans un schéma dual les sommets du maillage d’un niveau de subdivision sont supprimés au niveau de subdivision suivant.

Détaillons à présent les schémas de subdivision les plus utilisés dans le domaine de la modélisation 3D. Il s’agit des schémas de Catmull-Clark, de Doo-Sabin, et enfin du schéma de Loop.

2.4.1 Schéma de subdivision de Catmull-Clark

Ce schéma a été introduit en 1978 [CC78]. Il s’agit d’un schéma d’approximation générique, s’appliquant à des maillages avec des facettes quelconques.

L’algorithme de subdivision procède comme suit :

- Pour chaque facette du maillage, insérer le nouveau *point de face*, qui représente le barycentre des sommets originaux de la facette considérée (Figure 2.19(b)).

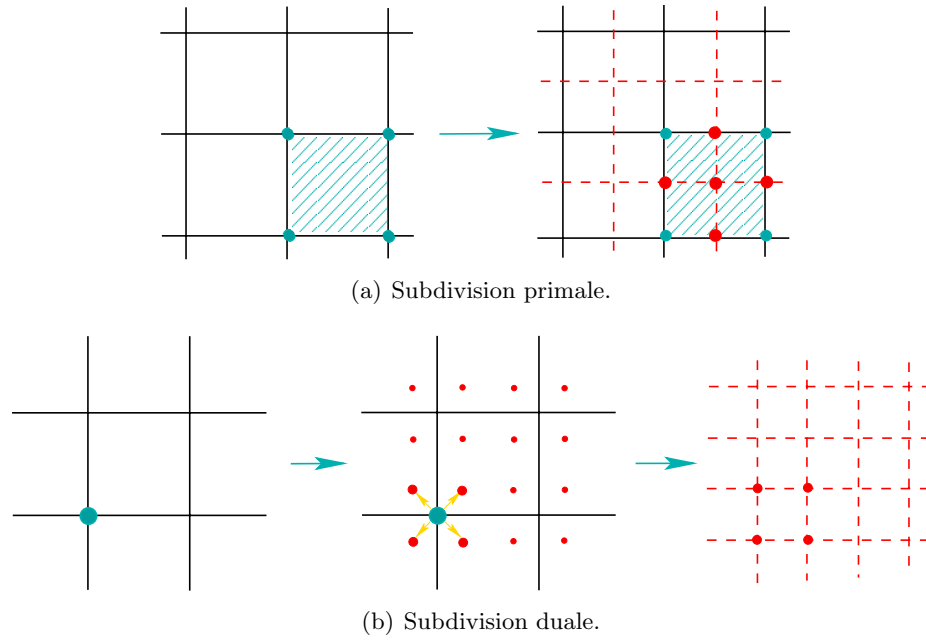


Figure 2.18 – Types de subdivision : Exemple de subdivision primale et duale appliquées à un maillage de base quadrilatéral.

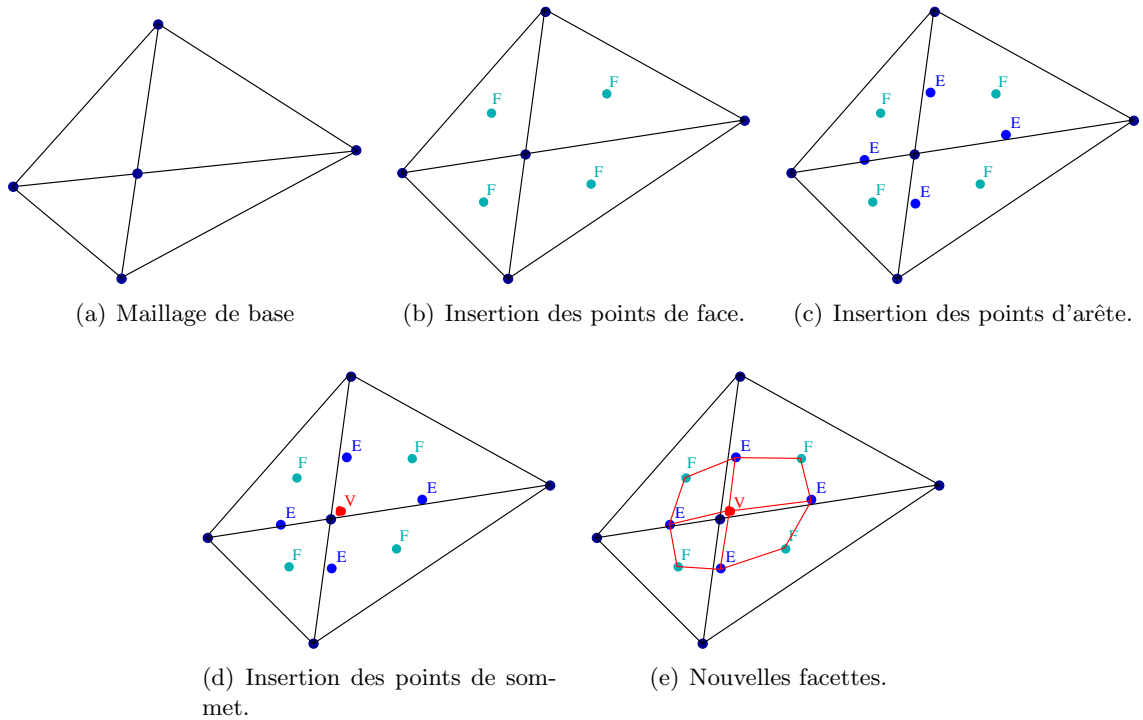


Figure 2.19 – Schéma de subdivision de Catmull-Clark.

- Pour chaque arête du maillage, insérer un nouveau point appelé *point d'arête*, obtenu en moyennant les sommets de l'arête considérée et les *points de face* des deux facettes adjacentes (Figure 2.19(c)).
- Actualiser les coordonnées des *points de sommet* (Figure 2.19(d)), dont la nouvelle position prend en compte les points des faces et les points d'arête précédemment insérés. Notons par S la position du sommet sur le maillage de base, par A la moyenne des points d'arête insérés à partir des arêtes adjacentes, et par F la moyenne des points de face

insérés sur les faces adjacentes. Soit n le nombre d'arêtes adjacentes au sommet S sur le maillage de base. La nouvelle position S' du point de sommet est donnée par :

$$S' = \frac{F + 2A + (n - 3)S}{n}. \quad (2.26)$$

- Enfin, le maillage est généré de la manière suivante (Figure 2.19(e)) :
 - chaque nouveau point de face est relié à chaque point d'arête correspondant à une arête adjacente à la face considérée sur le maillage de base,
 - chaque point de sommet déplacé est relié aux points d'arête insérés correspondant à une arête adjacente au sommet considéré.

2.4.2 Schéma de subdivision de Doo-Sabin

Doo et Sabin se sont inspirés de l'algorithme de subdivision de Chaikin pour les courbes pour en déduire un schéma applicable aux maillages [Doo78], [DS78].

L'algorithme de subdivision de ce schéma s'effectue selon les étapes suivantes :

- Calculer les milieux de chaque segment du maillage,
- Calculer les milieux de chaque face du maillage (définis par la moyenne des sommets de la face considérée),
- Pour chaque sommet de chaque face, insérer le point dont les coordonnées sont la moyenne du sommet considéré, des milieux des deux arêtes qui lui sont adjacentes et du milieu de face calculé précédemment. La Figure 2.20(a) illustre cette étape de l'algorithme.
- Les nouvelles faces du maillage sont obtenues en reliant ensemble les différents points insérés sur une même face d'un côté (Figure 2.20(b)) et les points insérés au voisinage d'un sommet original d'un autre côté (Figure 2.20(c)).

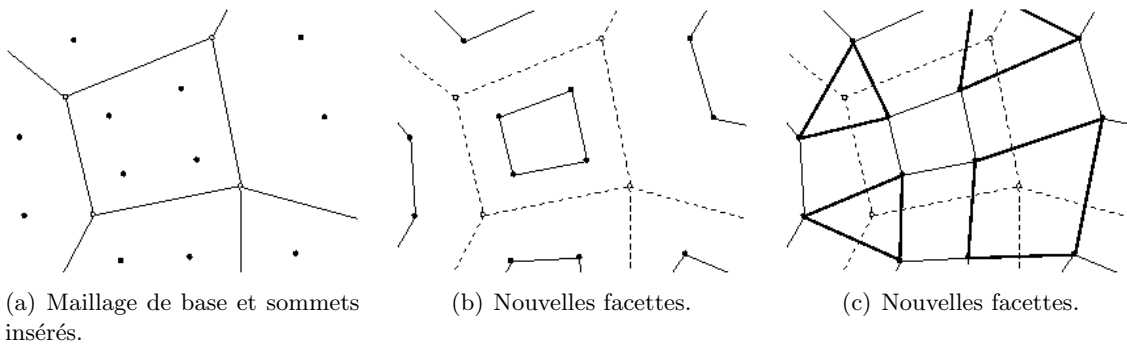


Figure 2.20 – Schéma de subdivision de Doo-Sabin.

2.4.3 Schéma de subdivision de Loop

Ce schéma introduit en 1987 par Charles Loop [Loo87] est un schéma de subdivision primal et approximant qui s'applique aux maillages triangulaires.

Dans le schéma de Loop, illustré Figure 2.21, le nouveau maillage est déduit suivant les deux étapes suivantes :

- Une étape de raffinement de la topologie qui consiste à insérer les nouveaux points. Ceux-ci correspondent dans le schéma de Loop aux milieux des arêtes. En reliant ces points, chaque triangle est alors subdivisé en 4 nouveaux triangles (Figure 2.21(b)).
- Les sommets sont ensuite déplacés selon des lois de positionnement prédéfinies, de manière à lisser le maillage résultant (Figure 2.21(c)). La position d'un nouveau sommet est donnée par une moyenne pondérée de ses sommets voisins appartenant au maillage

précédant. Les poids dépendent de la valence des sommets, *i.e.* du nombre de leurs voisins.

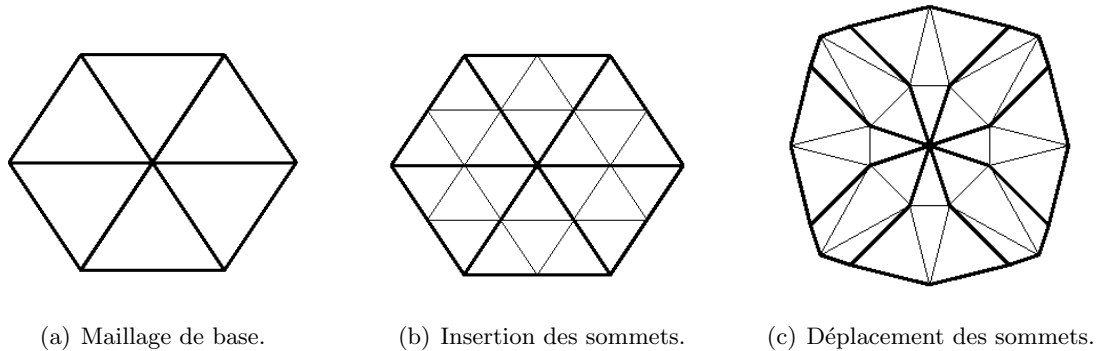


Figure 2.21 – Schéma de subdivision de Loop.

2.4.4 Conclusion

Les modèles de subdivision sont créés avec les mêmes techniques que les maillages. Ils permettent certes d'obtenir des modèles lisses et des niveaux de détail précis, mais cela alourdit la procédure d'animation étant donné le nombre croissant de sommets introduits par la subdivision. Afin d'éviter ce problème, les animateurs effectuent en fait l'animation avant la subdivision, celle-ci étant réalisée en dernière étape avant la visualisation.

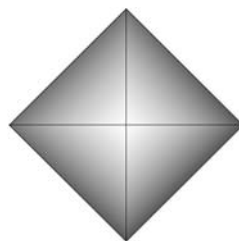
Les surfaces de subdivision présentent une difficulté qui leur est particulière dans le cadre des opérations booléennes. En effet, appliquer une subdivision sur un maillage de base obtenu par soustraction par exemple aboutit souvent à l'apparition d'artefacts au niveau du contour ainsi obtenu. Il est toutefois possible de contourner le problème en déformant les surfaces au lieu d'utiliser des opérateurs booléens.

Le schéma de subdivision de Loop utilise les sommets initiaux du maillage. Afin d'obtenir une forme lisse, une étape de filtrage est alors ajoutée qui déplace les positions des sommets. L'application d'un tel filtre conduit à un rétrécissement du volume du maillage de départ, qui est illustré sur la Figure 2.4.4. Les schémas de subdivision de Catmull-Clark et de Doo-Sabin procèdent par une subdivision duale où les nouveaux sommets du maillage sont directement sur les faces du niveau de subdivision précédent. De ce fait, une relation de tangence existe entre chaque niveau de subdivision et le niveau précédent. Dans le cas de la subdivision de Doo-Sabin où cette tangence est maintenue au niveau d'une face, le maillage final conserve la relation de tangence avec le maillage de base. La diminution de volume est par conséquent moindre (Figure 2.4.4).

	<i>Loop</i>	<i>Catmull-Clark</i>	<i>Doo-Sabin</i>
Régularité	C^2/C^1	C^1/C^0	C^2/C^1
Type de schéma	approximatif	approximatif	approximatif
Polygone	quadrilatère	quadrilatère	triangle
Subdivision	primal	dual	primal

Tableau 2.2 – Caractéristiques des schémas de subdivision de Loop, Catmull-Clark et Doo-Sabin.

La subdivision de Loop est cependant celle qui donne les surfaces les plus lisses, grâce à l'étape de filtrage. Elle est par conséquent moins sensible à la configuration des triangles sur le maillage de base, contrairement aux schémas de Catmull-Clark et Doo-Dabin qui peuvent



Maillage de base



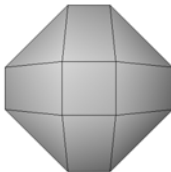

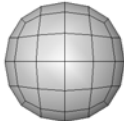
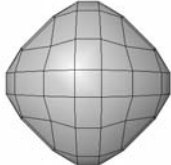

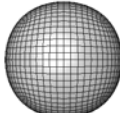
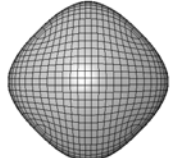
Schéma de subdivision	Loop	Catmull-Clark	Doo-Sabin
Un niveau de subdivision			
Deux niveaux de subdivision			
Quatre niveaux de subdivision			

Figure 2.22 – Comparaison des schémas de subdivision de Loop, Catmull-Clark et Doo-Sabin : L’effet de *rétrécissement* de la surface s’accroît au fur à mesure des niveaux de subdivision. Il est très important dans le cadre du schéma de Loop, alors que le schéma le plus proche de la surface de départ est le schéma de Doo-Sabin.

donner des surfaces moins lisses. La Figure 2.23 où les trois schémas de subdivision ont été appliqués au même maillage de base illustre ce cas de figure.

Une des raisons de la popularité des surfaces de subdivision est le fait qu’elles aboutissent à des surfaces maillées pouvant restituer de manière compacte des formes très complexes. Le personnage *Geri* (Figure 2.24) du célèbre court métrage “*Geri’s Game*”² illustre les capacités de représentation autant pour le visage du personnage qui restitue très bien les rides que pour les vêtements.

Les outils de visualisation, de compression, ainsi que les standards d’animation tels que H-Anim [HAN] et MPEG-4 [MPE] étant particulièrement adaptés aux maillages, l’introduction des surfaces de subdivision dans le monde de la 3D a été facilitée. De plus, l’aspect multirésolution de ces surfaces présente un intérêt particulier pour la compression, ainsi que

²Pixar, 1997

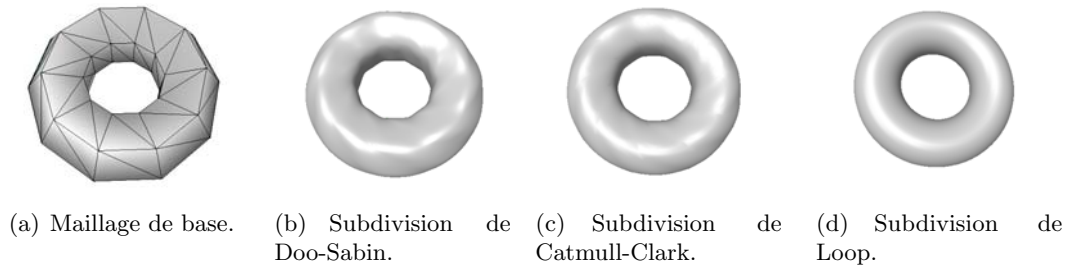


Figure 2.23 – Influence du maillage de base en fonction du schéma de subdivision appliqué sur l’aspect de la surface finale.

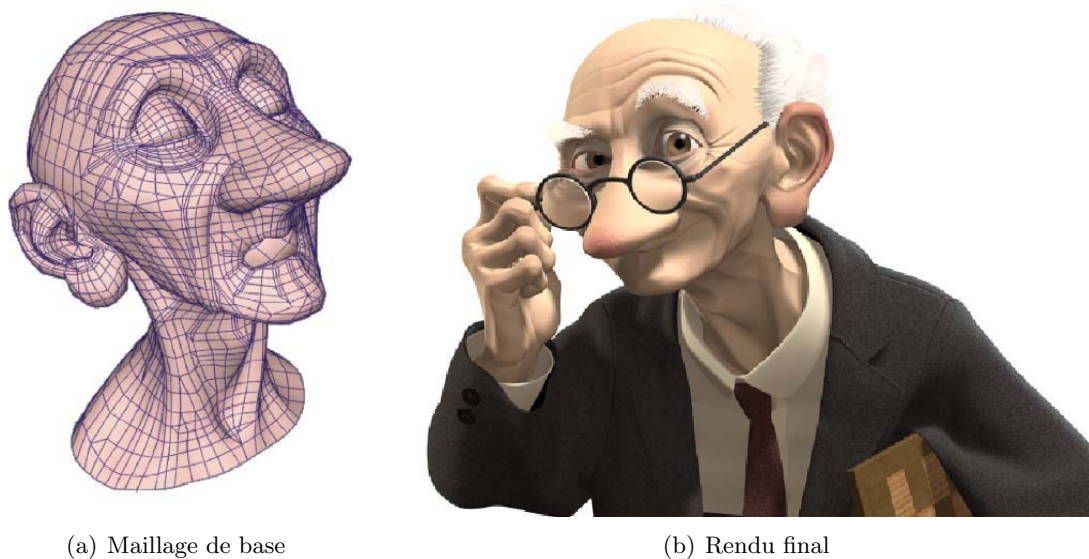


Figure 2.24 – Surface de subdivision : Le personnage Geri de Geri’s Game (Pixar) est créé par surfaces de subdivision : le modèle restitue très bien des détails de peau et de vêtement, pour un maillage de base relativement simple (extrait de [ZS00]).

pour la visualisation et l’animation.

La popularité de ces surfaces est accentuée par l’apparition de modifications des schémas de subdivision. Celles-ci permettent de modéliser un plus large spectre d’objets, en offrant la possibilité de conserver des arêtes vives [HDD⁺94] ou encore d’appliquer différents niveaux de subdivisions à un même objet [ZS00].

2.5 Conclusion

Afin de comparer les différents modèles de surface présentés dans ce chapitre, nous utilisons un ensemble de critères de comparaison :

- La *complétude* qui consiste en la capacité de la surface à décrire de manière complète la surface de l’objet modélisé. La modélisation par un nuage de points par exemple ne permet pas de préciser si un point de l’espace n’appartenant pas à la liste définissant l’objet est sur la surface ou non.
- La *concision* qui caractérise la quantité de mémoire nécessaire pour décrire la surface. Ce critère est important lors de la modélisation à des fins d’animation, car il détermine la rapidité des animations effectuées.
- La *facilité d’accès aux propriétés locales*. La détermination des propriétés de la surface en

un point donné telles que la normale à la surface ou la continuité est importante pour la visualisation et l'application de textures. Un temps de calcul important nécessaire pour déterminer ces propriétés locales implique en effet un temps de visualisation important.

- La *facilité de contrôle local*. La modélisation des personnages s'effectue souvent de manière incrémentale, et ce en déformant localement une surface de départ relativement simple sur plusieurs étapes successives. De la même manière, lors de l'animation, des déformations locales peuvent être réalisées, notamment dans le cas des dessins animés où les effets de style utilisent souvent ces déformations. Il est par conséquent impératif, pour une application de dessins animés, d'utiliser un modèle où les déformations locales sont non seulement possibles, mais surtout faciles.
- La *facilité de visualisation*.

	<i>Maillages polygonaux</i>	<i>Surfaces paramétriques</i>	<i>Surfaces implicites</i>	<i>Surfaces de subdivision</i>
Complétude	oui	oui	oui	oui
Concision	non	oui	oui	oui
Propriétés locales	non	oui	oui	non
Contrôle local	oui	oui	non	oui
Visualisation	oui	oui	non	oui

Tableau 2.3 – Comparaison des modèles de surface.

Le tableau 2.5 établit une comparaison des modèles de surfaces précédemment présentés selon ces critères de comparaison. On remarque que les surfaces paramétriques sont les seules à satisfaire tous les critères. Cependant, pour les raisons citées plus haut (difficulté de maintenance de continuité de patches), ces dernières sont généralement utilisées par les animateurs comme un modèle de base, converti par la suite en un maillage plus facile à visualiser et à animer car sans coutures.

Les maillages présentent cependant l'inconvénient de nécessiter un espace mémoire important, notamment dans le cas de modèles complexes, ce qui induit un temps de calcul important pour les animations. Les surfaces de subdivision constituent un compromis entre les maillages et les surfaces paramétriques car elles utilisent une représentation hiérarchique fondée sur un maillage de base nécessitant peu de mémoire. De plus, cette définition hiérarchique est intéressante pour des utilisations comme la prévisualisation qui ne nécessite qu'une description approximative pour vérifier la fluidité et l'exactitude des animations.

Les surfaces implicites, de par leur description, présentent des difficultés au niveau de la visualisation, de la détermination des propriétés locales, ainsi que du contrôle local de la surface. Elles sont par conséquent peu adaptées à notre objectif.

Chapitre 3

Animation 3D de personnages virtuels

Résumé

Ce chapitre présente les principales techniques d'animation 3D de personnages virtuels. L'objectif est ici d'identifier les principales contraintes que les modèles 3D doivent satisfaire pour être adaptés à des fins d'animation.

Deux grandes familles d'approches sont identifiées et décrites en détail. Il s'agit des méthodes descriptives et des méthodes par modèles générateurs. Les premières s'appuient sur une description du mouvement spécifiée par l'animateur à des instants précis dits poses clefs. Celles-ci sont modifiées en appliquant soit des mécanismes de déformation non rigide, soit des méthodes d'animation par squelette cinématique. Les trames individuelles nécessaires pour produire les séquences d'animation sont ensuite obtenues par interpolation. Les méthodes par modèles générateurs visent à déterminer l'ensemble des trames de la séquence d'animation à l'aide d'une modélisation physique, procédurale ou comportementale, caractérisant le mouvement de l'objet.

Enfin, les principaux standards d'animation 3D (VRML, H-ANIM, MPRG-4/FBA et MPEG-4/AFX) sont également présentés. Une attention particulière est consacrée à la présentation du standard MPEG-4/AFX adopté dans le cadre du projet TOON en raison de ses fonctionnalités avancées d'animation de personnages virtuels *génériques*.

Mots clés

Déformation, Animation, Interpolation de poses clefs, Méthodes cinématiques, Méthodes dynamiques, Métamorphose, Méthodes procédurales, Méthodes comportementales, Capture de mouvement, H-ANIM, VRML, MPEG-4, Squelette, FBA, BBA.

“L’animation n’est pas faire mouvoir des dessins. C’est, par essence, dessiner le mouvement.”

R.Taylor. [Tay96]

L’animation met en œuvre des éléments qui varient dans le temps, qu’il s’agisse de la position et de la pose des objets 3D de la scène, des paramètres de la caméra ou encore des paramètres photométriques (couleurs, textures,...) des objets.

L’objectif est alors de générer, à partir de ce modèle, l’ensemble des trames nécessaires pour constituer une séquence d’animation à la cadence vidéo requise (*e.g.* 15 à 25 trames par seconde). Pour chaque trame, il est nécessaire de modifier la géométrie de l’objet. Réaliser cette tâche manuellement n’est bien sûr pas envisageable. L’objectif des techniques d’animation est notamment d’automatiser cette tâche en garantissant un contrôle simple et intuitif de la géométrie et une interaction utilisateur minimale.

La littérature fait état de deux grandes familles d’approches [WW], [Ker04b], [Par03], [MFCD99], [Fuc03]. Il s’agit des méthodes descriptives et des méthodes par modèles générateurs. Les premières sont fondées sur une description paramétrique du mouvement spécifiée par l’animateur en des instants précis dits poses clefs. Les images nécessaires à la production de l’animation sont ensuite obtenues par interpolation. Les méthodes par modèles générateurs mettent en œuvre un ensemble de données, de paramètres et de lois qui gouvernent les mouvements des objets de la scène et permettent de générer les séquences d’animation automatiquement.

3.1 Modèles descriptifs

Les méthodes descriptives transposent la technique du *keyframing* traditionnellement utilisée dans la fabrication de dessins animés 2D au cadre de l’animation de scènes virtuelles 3D. Les poses clefs dessinées par l’équipe des animateurs 2D sont ici remplacées par la spécification interactive d’un certain nombre de paramètres caractérisant la pose des objets 3D. L’interpolation des paramètres ou des géométries clefs obtenues permet ensuite de générer les images intermédiaires nécessaires (*cf.* Paragraphe 3.1.3).

L’animation descriptive peut donc être divisée en deux étapes :

1. la construction des poses clefs à partir d’un modèle de référence,
2. l’interpolation temporelle (*inbetweening*), qui vise à générer à partir des poses clefs l’ensemble des poses intermédiaires nécessaires pour obtenir la séquence d’animation à la cadence vidéo requise.

Bien sûr, obtenir de nouvelles géométries en modifiant un à un, manuellement, les éléments caractéristiques d’un objet 3D (sommets du maillage, ou points de contrôle d’une surface NURBS par exemple) n’est pas envisageable. L’idée est alors de créer des modèles paramétriques contrôlables par un nombre réduit de paramètres, de façon simple et intuitive.

En fonction de la structure du modèle utilisé, on distingue deux familles d’approche d’animation descriptive : les déformations non rigides établissent des mécanismes de déformation génériques applicables à tout type de modèle. Les méthodes cinématiques visent plus spécifiquement l’animation des modèles articulés et exploitent un concept hiérarchique de l’animation. Notons que les deux approches peuvent être également hybridées [FvdPT97].

3.1.1 Modèles de déformation non rigide

Afin de modifier la forme d’un objet, une matrice de transformations M est appliquée aux points le constituant. Les nouvelles coordonnées de chaque sommet $v = [x, y, z]$ sont ainsi

déduites en appliquant la matrice de transformation aux coordonnées actuelles :

$$\vec{v}' = M\vec{v}. \quad (3.1)$$

En partant de ce modèle, Barr a introduit la notion de déformation globale [Bar84] en proposant de faire varier la matrice de transformation en fonction de la position où elle est appliquée. Cela permet de générer des transformations plus complexes, à partir d'un ensemble de transformations de base telles que l'amincissement ou la torsion.

Dans le cas de l'amincissement par exemple, l'objet est mis à l'échelle selon un axe donné. Soit un sommet (x, y, z) d'un objet aminci selon l'axe (Ox) . Sa nouvelle position (x', y', z') est donnée par l'opérateur d'amincissement :

$$F(x, y, z) = (x, ry, rz), \quad (3.2)$$

où r est un scalaire définissant le facteur d'échelle.

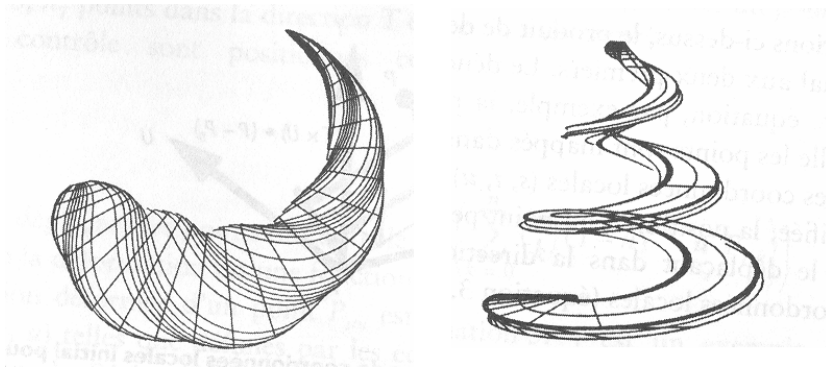


Figure 3.1 – Déformations globales : exemples de formes obtenues à partir d'un cylindre.

Ces opérateurs de base définissent des transformations simples et intuitives dont la combinaison permet d'obtenir une grande variété de formes à partir de modèles aussi simples qu'un cylindre.

Historiquement, les déformations de Barr (Figure 3.1) représentent un des premiers outils de modélisation conçus pour créer des modèles élaborés à partir de formes simples. Cependant, cette approche devient lourde et limitée lorsqu'il s'agit de définir des formes aussi complexes que des personnages articulés.

Afin de pallier ces inconvénients, les techniques de déformation libres (*FFD - Free Form Deformation*) généralisent le principe introduit par Barr en définissant un champ de déformation continu autour de l'objet.

3.1.1.1 Déformations de formes libres

La technique de déformation de formes libres *FFD* introduite par Sederberg et Parry [SP86] utilise un treillis parallélépipédique pour regrouper les sommets d'un maillage. Ce treillis représente un nouveau "repère" dans lequel les coordonnées des sommets considérés sont exprimés. Ces coordonnées étant définies en fonction de celles des sommets du treillis, une déformation de ce dernier implique automatiquement la déformation de tous les points du maillage qui lui sont associés.

Notons par $X = (x, y, z)$ le vecteur des coordonnées absolues d'un point du modèle 3D à déformer, par $\tilde{X} = (s, t, u)$ ses coordonnées dans le repère relatif du treillis. Soient (l, m, n) le nombre de points de contrôle du treillis pour chaque direction, et P_{ijk} les points de contrôle exprimés dans le repère absolu Cartésien. Les coordonnées déformées X_{FFD} de X sont alors exprimées dans le repère absolu par :

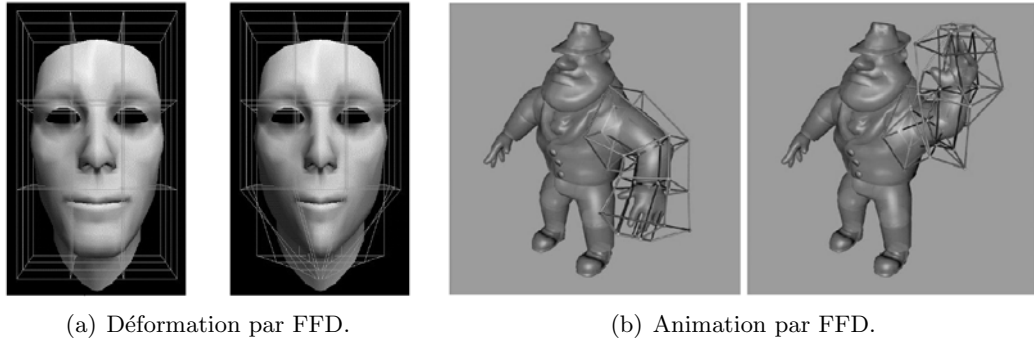


Figure 3.2 – Déformation de formes fibres.

$$X_{FFD} = \sum_{i=0}^l \binom{l}{i} (1-s)^{l-i} s^i \left[\sum_{j=0}^m \binom{m}{j} (1-t)^{m-j} t^j \left[\sum_{k=0}^n \binom{n}{k} (1-u)^{n-k} u^k P_{ijk} \right] \right] \quad (3.3)$$

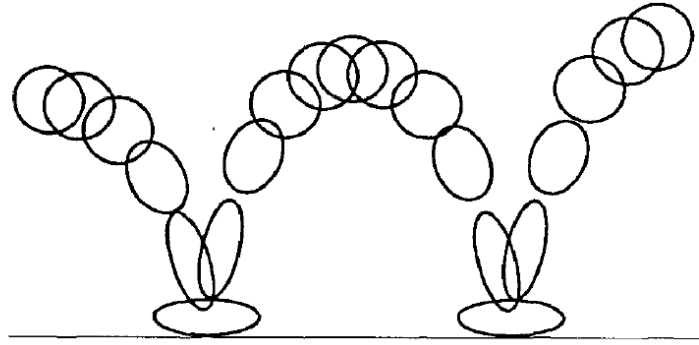


Figure 3.3 – Effets de style de dessins animés : étirement et écrasement subis par une balle lorsqu'elle rebondit.

L'utilisation de treillis est pratique pour réaliser des effets de style dans l'animation [Las87] en appliquant des distorsions telles que l'écrasement ou l'étirement (Figure 3.3). Cette méthode est aussi très utilisée par les modélisateurs 3D pour réaliser et/ou animer leurs personnages. La figure 3.4 illustre par exemple l'utilisation de la FFD pour déformer les yeux d'un personnage.

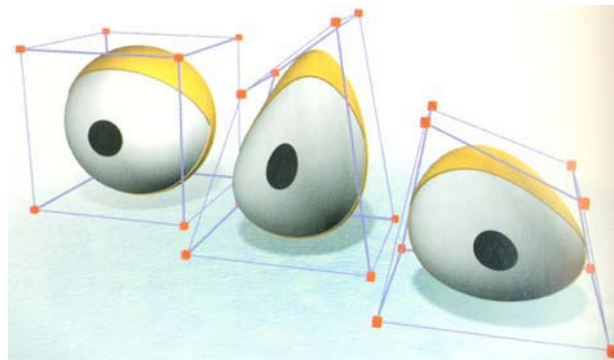


Figure 3.4 – Exemple d'utilisation de la FFD pour la création des yeux d'un personnage d'animation.

La force de cette méthode réside dans sa généralité, car elle peut s'appliquer à tout type

de surface. Toutefois, les treillis sont généralement de forme simple et régulière (souvent des cubes), ce qui ne correspond pas à la forme complexe des objets animés. De plus, il est difficile de “deviner” les déplacements des points de contrôle nécessaires pour obtenir la déformation souhaitée.

Dans [Coq90], Coquillart propose la déformation de formes libres étendue (*EFFD*, *Extended FFD*). L'idée est ici de généraliser le treillis parallélépipédique des FFD à une plus grande variété de formes. Le treillis est alors défini par l'utilisateur, qui peut ainsi construire une bibliothèque d'outils de déformation. La figure 3.5 illustre un exemple de EFFD obtenu à l'aide d'un treillis cylindrique.



(a) Treillis Cylindrique.

(b) Déformation du Treillis.

(c) Surface déformée.

Figure 3.5 – Déformation par EFFD [Coq90].

Une approche plus générale proposée dans [OCNF02] consiste à définir de manière automatique un treillis qui épouse la forme du modèle 3D en utilisant une structure en *Octree*. Le treillis ainsi obtenu peut alors être exploité dans un contexte multi-résolution, mais le nombre important de cellules rend la mise en œuvre laborieuse.

Quel que soit le type de treillis considéré, l'inconvénient majeur des approches par FFD vient de la manipulation indirecte de la surface par un réseau de points de contrôle dont l'influence sur la surface est peu intuitive.

Afin de s'affranchir de cet inconvénient, les approches par *contrôleurs* proposent une manipulation plus directe et intuitive, mise en œuvre à l'aide d'objets géométriques adaptés au type de déformation souhaitée et contrôlables directement.

3.1.1.2 Déformations par contrôleurs

Un *contrôleur* est défini comme un ensemble de points ou de courbes de l'espace avec une fonction d'influence associée. Une déformation appliquée au *contrôleur* engendre alors une fonction de déformation sur l'espace \mathbb{R}^3 dont l'amplitude est contrôlée pour tout point par la fonction d'influence.

En général, les fonctions d'influence considérées sont de support fini, afin de garantir un contrôle local de la surface. Le support de cette fonction est appelé *zone d'influence*.

Dans ce cadre, mentionnons tout d'abord l'approche de Yoshizawa *et al.* qui proposent une famille d'opérateurs définis à l'aide d'un ou de plusieurs points de contrôle [YBS02]. Ces opérateurs sont définis à partir d'un schéma de déformation de base, caractérisé par un point de contrôle et une fonction de déformation dont la forme et l'influence peuvent être modifiées à l'aide d'un certain nombre de paramètres.

Dans le schéma de base, un point P d'un maillage est déplacé vers sa nouvelle position P' à l'aide d'un vecteur de déplacement D qui est fonction de P et du point de contrôle C :

$$P' = P + D(C, P) \quad (3.4)$$

$$D(C, P) = \frac{\gamma}{\sigma} W(C, P) (P - C), \quad (3.5)$$

où $W(C, P)$ est une fonction de déformation prédéfinie, donnée par :

$$W(C, P) = \exp\left(-\frac{|C - P|^\alpha}{2\epsilon^2}\right) \quad (3.6)$$

$$\sigma = W(C, P_{min}), \quad (3.7)$$

où P_{min} est le sommet de la surface le plus proche du point de contrôle C , et γ , α et ϵ sont des paramètres.

En définissant plusieurs fonctions de déformation, les auteurs mettent ainsi en place un ensemble d'outils prédéfinis qui permettent de produire des déformations différentes et de les combiner. Un exemple de résultat illustré pour un visage d'humanoïde est présenté sur la figure 3.6.



Figure 3.6 – Déformation géométrique par points de contrôle [YBS02] : application à un visage humain.

Singh et Fiume introduisent en 1998 le concept des *wires*, [SF98], défini comme un ensemble de courbes utilisées pour déformer la surface d'un modèle.

Dans ce schéma, un contrôleur de type *wire* est défini par un quintuplet (C, R, s, r, f) où C et R sont des courbes paramétriques désignées respectivement par *courbe cible* et *courbe de référence*, s est un scalaire permettant d'effectuer une mise à l'échelle radiale, r est un scalaire définissant le rayon d'influence autour de la courbe wire, et $f : \mathbb{R}^+ \rightarrow [0, 1]$ est une fonction de densité caractérisant l'influence de la courbe de référence sur son voisinage. La fonction f et le rayon r définissent alors un volume d'influence autour de la courbe wire.

Soient P un point dans l'espace et p_R le paramètre de position sur R qui minimise la distance entre P et la courbe de référence R . La fonction F donnant l'influence de la courbe R sur P est alors définie par :

$$F(P, R) = f\left(\frac{\|P - R(p_R)\|}{r}\right). \quad (3.8)$$

Généralement, f est choisie décroissante sur $[0, 1]$ et nulle pour toute valeur supérieure à 1. La fonction d'influence F est alors nulle au delà du rayon d'influence r , et maximale et égale à 1 pour les points situés sur la courbe de référence R .

La position de chaque point P du volume d'influence est modifiée selon les étapes suivantes :

- Calculer le paramètre p_R minimisant la distance entre la courbe de référence R et le point P ,
- Effectuer une transformation de mise à l'échelle pour obtenir le point P_s :

$$P_s = P + (P - R(p_R)) \cdot (1 + (s - 1) \cdot F(P, R)), \quad (3.9)$$

- Appliquer au point P_s une rotation d'angle $\theta.F(P, R)$ où θ est l'angle formé par les tangentes $W'(p_R)$ et $R'(p_R)$ aux courbes W et R , et d'axe $W'(p_R) \times R'(p_R)$, donnant le point P_r ,
- Translater le point P_r , pour obtenir la position finale P_{def} :

$$P_{def} = P_r + (W(p_R) - R(p_R)) . F(P, R). \quad (3.10)$$

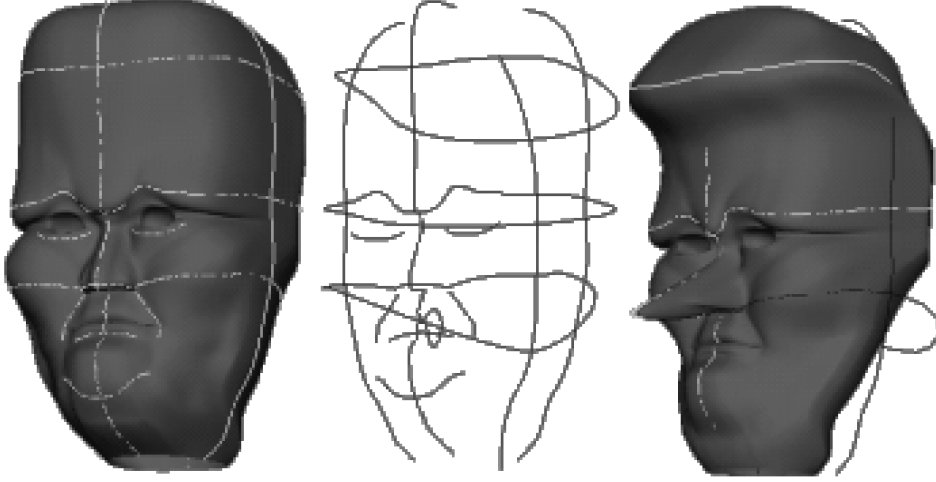


Figure 3.7 – Déformation à l'aide de courbes wires : application à un visage humain.

Comme dans le cas des FFD, les *wires* sont une méthode de déformation générale applicable à différents modèles de surface, qu'il s'agisse des sommets d'un maillage ou des points de contrôle d'une NURBS. La figure 3.7 présente un exemple de déformation obtenu en combinant plusieurs *wires* pour déformer le modèle d'un visage humain.

Dans [ZG98], Zelinka reprend cette idée en imposant l'utilisation de courbes de référence géodésiques à la surface S de l'objet à déformer. L'influence de la courbe de référence R sur un sommet v de la surface est alors définie par une fonction Gaussienne :

$$\omega_R = \text{Gaussienne} \left(\frac{d_R}{r} \right), \quad (3.11)$$

où d_R est la distance géodésique entre le point v et la courbe de référence R . Soit R_v le point de R le plus proche, au sens de la distance géodésique du point v . Soit C_v son correspondant (*i.e.* le point de la même coordonnée paramétrique) sur la courbe cible C . La nouvelle position v_{def} du sommet v est donnée par l'expression suivante :

$$v_{def} = v + \omega_R M_v^T M_u (C_v - R_v), \quad (3.12)$$

où M_p est une matrice définissant un repère local au point p :

$$M_p = [t_p b_p n_p]^T, \quad (3.13)$$

où t_p est la tangente à la courbe de référence au point p , n_p est la normale à la surface S en p , et $b_p = n_p \times t_p$.

Notons que les tangentes t_p sont calculées, pour tout point p de la surface à partir des tangentes des points de la courbe de référence, en appliquant un mécanisme de transport parallèle [Kre91].

Enfin, signalons qu'un point peut être affecté par plusieurs courbes *wires*. Dans ce cas, sa position finale est déterminée comme une moyenne des déplacements individuels associés à chaque *wire*, pondérée par les coefficients d'influence correspondants.

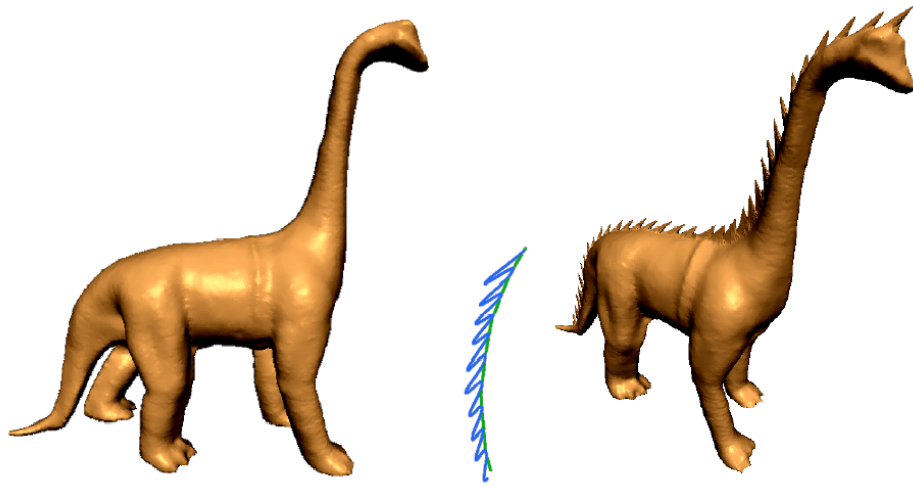


Figure 3.8 – Déformation à l'aide de courbes géodésiques : application à un dinosaure.

De manière générale, les déformations par contrôleurs ont l'avantage d'être rapides en temps de calcul, tout en offrant un contrôle aisé et intuitif. Les points situés sur les courbes de référence sont transformés en leurs correspondants sur les courbes cibles. Leurs déplacements définissent un champ de déformation continu et décroissant avec la distance à la courbe de référence.

Ce modèle est particulièrement utile pour réaliser des déformations quelconques. Toutefois, dans le cas spécifique des mouvements de personnages articulés, il est préférable de disposer de modèles plus adaptés à la réalité physique, prenant en compte la structure hiérarchique de ces objets pour en décrire la pose. C'est le cas des modèles par squelette cinématique.

3.1.2 Modèles par squelette cinématique

Le squelette d'un personnage articulé est un élément clé pour la gestion de ses mouvements. L'animation hiérarchique [Kis02], [MFCD99] reproduit pour les objets virtuels ce schéma intuitif emprunté au monde réel.

Le squelette d'un personnage articulé est une structure hiérarchique sous forme d'arbre composée d'un ensemble de *nœuds* correspondant aux articulations, reliés par des *arcs* correspondant aux os. Un nœud terminal, sans enfants, est dit *effecteur final*. Chaque nœud est caractérisé par un ensemble de paramètres de position et d'orientation (qui peuvent être des angles d'Euler, ou des quaternions [Par03], [Sho85], [Ple88]). Ces paramètres sont en général exprimés par rapport à un repère local, lié au nœud parent du nœud considéré. Les positions des effecteurs finaux sont alors déduites en traversant progressivement l'arborescence du squelette et en multipliant au fur et à mesure les matrices de transformation des nœuds traversés.

Soit $\theta = \{\theta_i\}_i$ l'ensemble des paramètres de pose du squelette, dit *vecteur d'état* du modèle articulé. Notons que, dans un contexte dynamique, le vecteur d'état peut également inclure les dérivées des paramètres θ_i par rapport au temps. On parle alors de coordonnées généralisées.

Réaliser l'animation d'un modèle par squelette revient alors à déterminer l'évolution du vecteur d'état dans le temps.

Les paramètres d'animation caractérisent complètement le mouvement du squelette. La surface de l'objet, appelée peau, doit être ensuite animée de manière cohérente [MPE], [CHP89], [TSC96], [Tur95]. Cela est réalisé en spécifiant l'influence de chaque élément du squelette sur la surface du modèle [MPE]. On parle alors d'un attachement du squelette à la peau. Ces aspects seront décrits en détail au Paragraphe 3.3.

3.1.2.1 Cinématique directe

La cinématique directe [WW], [Par03] consiste à spécifier le vecteur d'état à des instants donnés dits *poses clefs*. Pour calculer les positions absolues dans l'espace 3D de chacun des nœuds, le vecteur des positions, noté X , est alors donné par une fonction du vecteur d'état global :

$$X = f(\theta). \quad (3.14)$$

Spécifier des vecteurs d'état représente une tâche difficile et peu intuitive puisqu'il s'agit de définir les positions relatives des différentes articulations dans le but d'obtenir une pose donnée. Cette méthode peut devenir lourde pour des objets de structure complexe.

L'idée est alors de spécifier la position absolue d'un sous-ensemble de nœuds, à partir de laquelle déterminer automatiquement le vecteur des paramètres de pose. Dans ce contexte, les techniques les plus utilisées concernent la cinématique inverse et la capture de mouvement.

3.1.2.2 Cinématique inverse

En cinématique inverse [WW], [Par03], l'utilisateur spécifie uniquement les positions dans l'espace 3D d'un nombre limité de nœuds. Le plus souvent, il s'agit de spécifier les valeurs des effecteurs finaux.

Notons par X_{eff} le vecteur des positions 3D des nœuds spécifiés par l'utilisateur. Afin de réaliser l'animation, il faut alors calculer la valeur du vecteur d'état θ pour l'ensemble des nœuds du squelette vérifiant la condition :

$$X_{eff} = f(\theta). \quad (3.15)$$

Pour résoudre l'équation 3.15, plusieurs types d'approche sont disponibles. Lorsque le nombre d'articulation est réduit, et dans le cas de certaines configurations particulières, le système peut admettre une solution directe. Il s'agit des méthodes analytiques comme celles présentées dans [Pre02], [Par03], [Bae01], [Kor86].

Ces approches ont toutefois un domaine d'application très restreint. La solution consiste alors à linéariser le système 3.15 par un développement de Taylor autour d'une solution initiale supposée disponible et notée θ_0 :

$$f(\theta_0 + \delta\theta) = f(\theta_0) + J(\theta)\delta\theta + \dots \quad (3.16)$$

$$\Rightarrow f(\theta_0 + \delta\theta) - f(\theta_0) = J(\theta)\delta\theta + \dots \quad (3.17)$$

$$\Rightarrow \delta X \approx J(\theta)\delta\theta. \quad (3.18)$$

Ici, J désigne le Jacobien de f , dont les éléments sont définis par :

$$J_{ij} = \frac{\delta f_i}{\delta \theta_j}. \quad (3.19)$$

La résolution s'effectue ensuite de manière itérative à partir d'une valeur initiale de vecteur d'état θ selon l'équation :

$$\delta\theta = J^+ \delta X. \quad (3.20)$$

Ici, J^+ désigne la pseudo-inverse de la matrice J .

Toutefois, cette méthode se heurte aux difficultés liées aux méthodes de résolution itératives telles que la convergence vers des minima locaux et la sensibilité à l'initialisation.

Une autre technique consiste en la “descente cyclique des coordonnées”, qui procède par minimisation d'énergie itérative en parcourant les articulations dans le sens direct de l'arborescence. La méthode étant appliquée à chaque articulation séparément, la résolution est directe à chaque itération.

La figure 3.9 illustre une animation obtenue par cinématique inverse. Les positions des effecteurs finaux sont ici représentées par des sphères rouges. Le mouvement du genou gauche (respectivement du coude droit) est calculé automatiquement à partir de la nouvelle position du pied gauche (respectivement de la main droite).

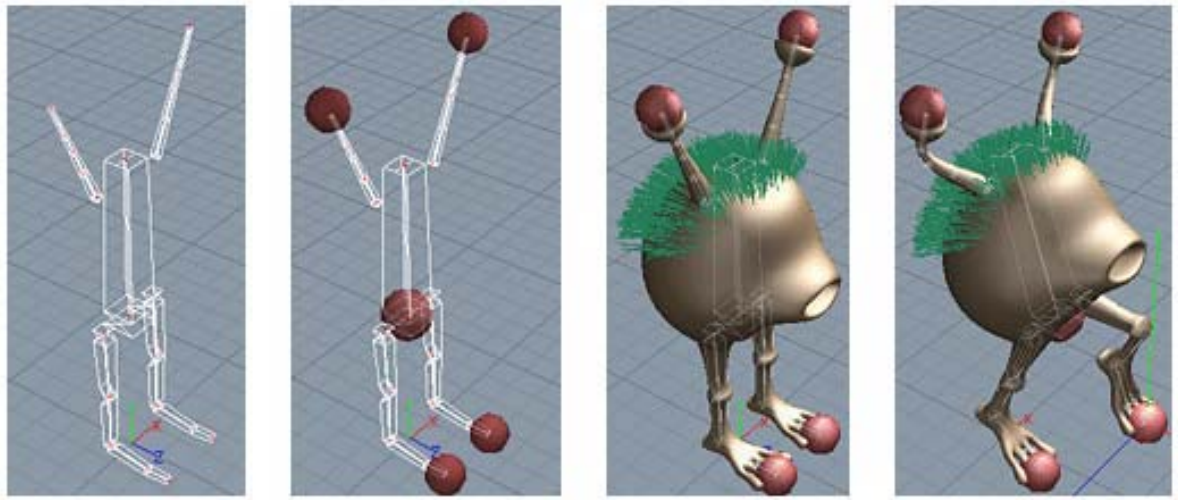


Figure 3.9 – Animation d'un personnage par cinématique inverse réalisée à l'aide du logiciel REALSOFT [REA].

3.1.2.3 Capture de mouvements

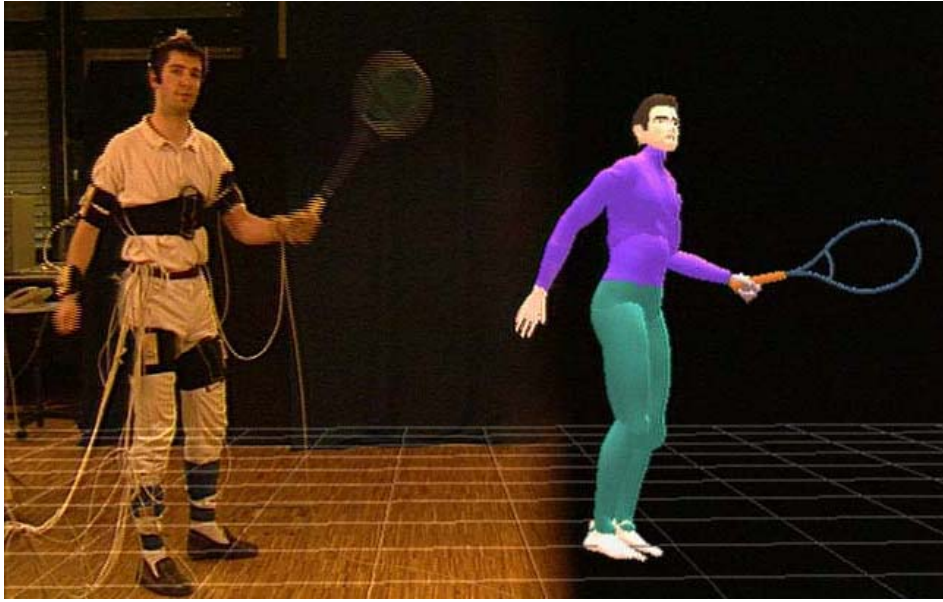
En fonction des applications, un degré de réalisme plus ou moins important du mouvement peut être requis. Plus ce critère est important, plus la tâche de l'animateur devient difficile et délicate, l'œil du spectateur étant très sensible aux éventuelles erreurs. Pour cette raison, dans le cas des films d'animation, ou des films mêlant des acteurs réels à des personnages virtuelles, les actions sont souvent réalisées à l'aide de la technique de capture de mouvements [Gle00], [ZH02], [LCR⁺02]. Cette technique consiste à enregistrer le mouvement effectué par un acteur réel (humain ou animal) à l'aide de capteurs magnétiques ou optiques.

Les *systèmes magnétiques* (Figure 3.10(a)) sont composés d'un émetteur central de champ magnétique et d'un ensemble de capteurs magnétiques que l'acteur doit porter. Ces capteurs sont capables de mesurer leur position et leur orientation par rapport à l'émetteur central.

Les capteurs magnétiques limitent les mouvements des acteurs, qui doivent gérer les fils reliant chaque capteur au transmetteur central. Un compromis doit être réalisé entre un nombre important de capteurs pour reproduire de manière plus complète les mouvements, mais qui gêne l'acteur, ou un nombre plus petit avec une capture moins précise. Dans le cas d'un acteur humain, on utilise généralement 11 capteurs (tête, bras, mains, poitrine, dos, chevilles, et pieds).

Les *systèmes optiques* (Figure 3.10(b)) exploitent des capteurs réflectifs appelés marqueurs. En plaçant ces marqueurs sur les vêtements de l'acteur, ainsi qu'à des endroits fixes de la scène, et en filmant à l'aide de plusieurs caméras (généralement 4 à 6), il est possible d'enregistrer les positions des capteurs, qui sont placés au niveau des articulations de la personne.

Les marqueurs optiques présentent l'avantage d'offrir à l'acteur une liberté de mouvement totale, et sont donc plus adaptés à des scènes où les actions décrivent des mouvements complexes. Malheureusement, l'utilisation de caméras se heurte à des problèmes d'occlusion. Des problèmes de distinction des marqueurs dans le cas de certains mouvements peuvent également apparaître.



(a) Système de capture magnétique (extrait de [mel]).



(b) Système de capture optique (extrait de [lor]).

Figure 3.10 – Systèmes de capture de mouvements.

3.1.3 Techniques d'interpolation temporelle

Les méthodes de déformation géométrique et hiérarchique permettent de générer des poses clefs qui décrivent les instants clefs du mouvement voulu. Les positions des sommets ou des paramètres décrivant la hiérarchie sont ensuite calculées par interpolation afin d'obtenir le nombre de poses nécessaire pour produire une animation complète.

De manière générale, les techniques d'interpolation temporelle concernent différents paramètres associés à un objet ou une scène 3D comme :

- la pose globale d'un objet modélisé dans une scène 3D,
- les positions de la caméra et des sources de lumière : les paramètres interpolés correspondent alors aux matrices de transformation définissant les position et orientation de la caméra au niveau des poses clefs,
- l'apparence d'un objet, comme ses paramètres géométriques (pour un maillage par exemple, la position des sommets) ou photométriques (texture, couleurs, etc.)

Soit α un paramètre variable dans le temps dont on connaît la valeur à des instants donnés t_i . Les couples (t_i, α_i) sont appelés poses clefs. Les instants t_i ne sont pas obligatoirement espacés régulièrement dans le temps et le nombre de poses intermédiaires nécessaires entre chaque couple de poses clefs n'est pas le même. Il est fonction de l'intervalle temporel $[t_i, t_{i+1}]$

précisé par l'animateur et du nombre de poses par secondes requises pour la vidéo finale (*e.g.* 25 trames par seconde).

Les méthodes d'interpolation visent à déterminer les valeurs du paramètre α à des instants différents de ceux des poses clefs. L'échantillonnage temporel est souvent régulier entre chaque couple de poses successives, avec un pas Δt_i donné par le *timing* de l'action précisé par l'utilisateur et défini comme le temps réel occupé par l'action dans le film final (Figure 3.11).

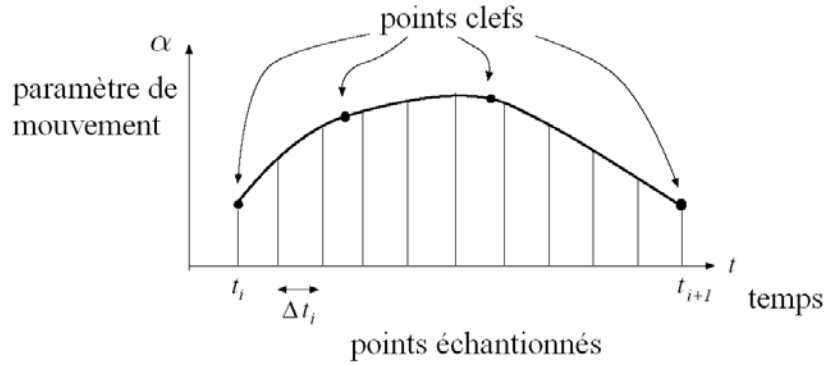


Figure 3.11 – Interpolation de poses clefs.

3.1.3.1 Interpolation linéaire

L'interpolation linéaire représente la méthode la plus simple pour obtenir les poses intermédiaires. La valeur du paramètre α animé à l'instant t s'exprime en fonction des poses clefs précédente α_{k-1} et suivante α_k comme suit :

$$\alpha(t) = (1 - t)\alpha_{k-1} + t\alpha_k. \quad (3.21)$$

Cette technique offre l'avantage de la simplicité, mais conduit à des animation mécaniques, qui présentent des trajectoires linéaires peu naturelles, ainsi qu'à des changements brusques de vitesse. En effet, le pas d'échantillonnage temporel entre les poses successives peut varier d'un intervalle à l'autre, conduisant alors à des vitesses différentes de part et d'autre des poses clefs.

Par ailleurs, les mouvements ainsi obtenus sont peu réalistes. Les mouvements des objets réels suivent en effet des trajectoires continues obéissant à la loi de Newton qui stipule que l'accélération d'un objet en mouvement est proportionnelle à la somme des forces qui lui sont appliquées. Cela implique que la dérivée seconde du vecteur de position des objets existe et est continue. Les fonctions d'interpolation doivent donc être C^2 pour représenter des mouvements réalistes. Ainsi, les fonctions polynomiales d'ordre supérieur, telles que les courbes de Bézier et les B-splines, fournissent-elles une solution adéquate.

3.1.3.2 Interpolation curviligne

L'utilisation des courbes de Bézier est cependant limitée par leur caractère global. En effet, la modification d'une pose clef modifie la trajectoire dans sa totalité. Les courbes B-Splines offrent en revanche un contrôle local qui permet d'ajuster localement les mouvements grâce à leur définition par morceaux.

Il est possible d'obtenir une courbe B-spline interpolant les poses clefs $[t_0, p_0; \dots; t_n, p_n]$ en résolvant le système suivant :

$$p(t_j) = \sum_{i=0}^n c_i N_{i,k}(t_j) = p_j, \quad (3.22)$$

où $c_i = \{c_0, \dots, c_n\}$ sont les coordonnées des points de contrôle à déterminer.

Ces méthodes d'interpolation sont adaptées aux caractéristiques géométriques d'un modèle, comme la position des sommets d'un maillage. Toutefois, lorsque l'on souhaite interpoler des paramètres de pose 3D, d'autres représentations sont préférables. C'est le cas des méthodes d'interpolation de quaternions.

3.1.3.3 Interpolation de paramètres de pose par quaternions

La pose 3D d'un objet articulé est définie souvent par des matrices de rotation 3D de taille 3×3 . Une interpolation directe des coefficients de ces matrices ne peut pas être applicable dans ce contexte étant donnée leur interdépendance. D'autre part, représenter la rotation par trois angles d'Euler et interpoler ces derniers de façon indépendante conduit à une décomposition du mouvement en trois rotations autour de trois axes et donc à des animations peu réalistes.

Les quaternions [Sho85], [Ple88] sont des représentations utilisées pour spécifier des rotations. Une rotation d'un angle α autour d'un axe \vec{d} s'exprime par le quaternion :

$$q = \left[\cos \frac{\alpha}{2}, \sin \frac{\alpha}{2} \vec{d} \right]. \quad (3.23)$$

Soient $u \in \mathbb{R}^3$ un vecteur de pose initial, et v le vecteur obtenu par rotation de u selon le quaternion q . En terme de quaternions, cette rotation s'exprime par la relation suivante :

$$[0, v] = q \cdot [0, \vec{u}] \cdot q^{-1}. \quad (3.24)$$

Notons que la composition de deux rotations q_1 et q_2 est donnée par le produit des quaternions [Vic01], [DKL98] :

$$q_2 \cdot (q_1 \cdot [0, \vec{u}] \cdot q_1^{-1}) \cdot q_2^{-1} = (q_2 \cdot q_1) \cdot [0, \vec{u}] \cdot (q_2 \cdot q_1)^{-1}. \quad (3.25)$$

Étudions à présent l'interpolation de poses clés décrites par des quaternions. Notons par q_1 et q_2 les quaternions décrivant les poses clés de départ et d'arrivée :

$$q_1 = \left[\cos \frac{\alpha_1}{2}, \sin \frac{\alpha_1}{2} \cdot \vec{d}_1 \right], \quad (3.26)$$

$$q_2 = \left[\cos \frac{\alpha_2}{2}, \sin \frac{\alpha_2}{2} \cdot \vec{d}_2 \right]. \quad (3.27)$$

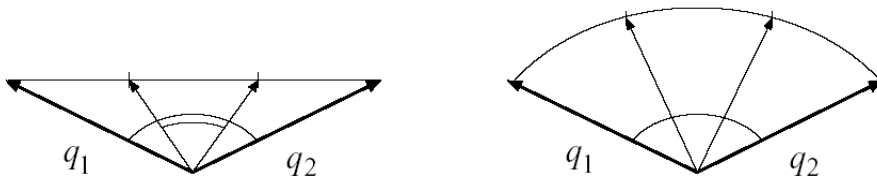


Figure 3.12 – Interpolation de quaternions : interpolation linéaire *versus* interpolation linéaire sphérique.

Une interpolation linéaire entre les quaternions q_1 et q_2 conduirait à des angles de rotation non constants, donnant lieu à un mouvement accéléré, puis décéléré. Ce phénomène est illustré Figure 3.12. Afin d'obtenir une rotation uniforme, il est nécessaire de maintenir une norme unitaire [Vic01], [DKL98] des quaternions pendant le mouvement, ce qui revient à une *interpolation sphérique*. Le quaternion à l'instant t est alors donné par :

$$q(t) = \frac{\sin(1-t)\theta}{\sin\theta} q_1 + \frac{\sin t\theta}{\sin\theta} q_2, \quad (3.28)$$

où l'angle θ est défini par :

$$\theta = \frac{\alpha_2 - \alpha_1}{2}. \quad (3.29)$$

Dans le cas de plusieurs poses clefs décrites par les quaternions q_1, \dots, q_n , l'interpolation sphérique linéaire, décrite précédemment conduirait à une discontinuité des vitesses et des trajectoires au niveau des poses clefs (Figure 3.13(a)).

Afin de résoudre ce problème, Shoemake [Sho85] a proposé une technique d'interpolation itérative inspirée de la définition des courbes B-splines, définie par une suite d'interpolations sphériques linéaires (Figure 3.13(b)).

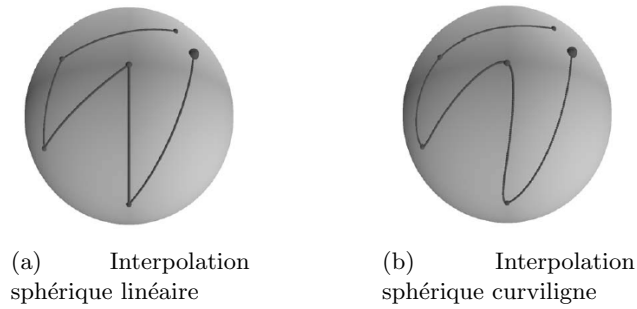


Figure 3.13 – Interpolations sphériques des quaternions : L'interpolation curviligne est continue au niveau des points clefs, contrairement à l'interpolation linéaire.

Les techniques d'interpolation de poses clefs concernent l'animation d'un même maillage dont les différents paramètres associés varient au cours du temps. Les techniques de métamorphose [Ale02] généralisent ce principe en abandonnant l'hypothèse d'une topologie fixe.

3.1.3.4 Techniques de métamorphose

Les techniques de métamorphose [LV98], [Ale02] visent à déformer graduellement un objet source en un objet cible de géométrie et de topologie différentes et arbitraires.

Notons par $M = (K, V)$ un maillage décrit par :

- sa géométrie représentée par l'ensemble des sommets $V = \{v_i\}_{i \in \{0, n\}}$,
- et sa topologie représentée par l'ensemble K des arêtes et des facettes du maillage.

Une métamorphose est une transformation d'un maillage source $M_1 = (K_1, V_1)$ en un maillage cible $M_2 = (K_2, V_2)$ qui nécessite la génération d'un ensemble de maillages de transition $M(t) = (K(t), V(t))$ de M_1 vers M_2 .

Ce problème est généralement résolu en définissant tout d'abord une correspondance entre les maillages M_1 et M_2 . Cette correspondance est définie par une topologie commune K et de nouvelles géométries $V(0)$ et $V(1)$ telles que les surfaces représentées par $M(0) = (K, V(0))$ et $M(1) = (K, V(1))$ soient respectivement identiques à celles représentées par M_1 et M_2 . L'interpolation entre les maillages $M(0)$ et $M(1)$ est ensuite effectuée pour $t \in]0, 1[$. La métamorphose implique donc trois étapes principales :

1. **Effectuer une mise en correspondance** des géométries des maillages M_1 et M_2 . Le plus souvent, celle-ci est réalisée à l'aide d'un domaine paramétrique commun, noté D , qui permet de spécifier les sommets des deux maillages par les coordonnées paramétriques 2D.

Dans ce contexte, mentionnons en particulier les techniques de paramétrisation sphérique [Flo97], [GGS03] qui consiste à définir une bijection entre la surface maillée et la sphère unité.

2. **Générer une topologie K commune.** Cette étape est souvent réalisée en utilisant un supergraphe des topologies K_1, K_2 , qui constitue une topologie commune aux géométries des deux maillages ainsi que de l'ensemble de nouveaux points générés par les intersections des arêtes des deux maillages dans le domaine paramétrique commun D .
3. **Définir les chemins des sommets** du maillage $M(0)$ vers le maillage $M(1)$. La correspondance étant établie entre les différents sommets des deux maillages, notons par $V(0)$ la position d'un sommet sur $M(0)$ et $V(1)$ sa position sur $M(1)$. Il s'agit alors de déterminer la position de chaque sommet en chaque instant $t \in]0, 1[$, notée $V(t)$.

La solution la plus simple est l'interpolation linéaire, où

$$V(t) = (1 - t)V(0) + tV(1). \quad (3.30)$$

Bien entendu, des méthodes d'interpolation plus élaborées (*e.g.* sous contrainte de non auto-intersection) peuvent être également considérées pour obtenir des métamorphoses plus fluides.

3.1.4 Conclusion

De manière générale, les modèles d'animation descriptifs offrent une grande liberté de création. En effet, ils sont contrôlés en définissant manuellement les données qui déterminent les mouvements des objets. Ils nécessitent cependant une forte interaction et leurs résultats dépendent étroitement de l'habileté et de l'expérience de l'utilisateur.

Les modèles générateurs visent à alléger l'interactivité nécessaire en automatisant la génération des mouvements. Ils mettent en œuvre des algorithmes ou des loi prédéfinies.

3.2 Modèles générateurs

Les modèles d'animation générateurs ont pour objectif de produire des mouvements de manière automatique. Les mouvements sont calculés par l'ordinateur de manière à générer toutes les séquences de l'animation sans que l'utilisateur ne spécifie de poses clefs.

Les méthodes automatiques peuvent être classées en fonction des données sur lesquelles elles s'appuient pour construire les animations virtuelles : méthodes procédurales, simulations physiques et méthodes comportementales.

3.2.1 Méthodes procédurales

Les méthodes procédurales [Rey82] exploitent des algorithmes prédéfinis pour générer des animations. Ceux-ci constituent alors les lois selon lesquelles les objets sont régis.

Ces méthodes sont très utiles pour introduire de l'entropie dans les mouvement à l'aide de méthodes statistiques, et sont souvent utilisées pour modéliser des phénomènes naturels comme les vagues [HNC02] ou de l'herbe [PC01].

3.2.2 Modèles physiques

De façon générale, les *modèles physiques* [TT96] sont mis en œuvre pour réaliser des mouvements complexes qui obéissent à des lois physiques connues. Ils prennent en compte les propriétés physiques du modèle 3D comme sa masse ou son élasticité, ainsi que les différentes forces internes et externes qui lui sont appliquées telles que la gravité ou la collision avec un autre objet. Le mouvement des objets est alors calculé automatiquement en fonction des lois physiques précisées. Mentionnons parmi les modèles physiques les modèles dynamiques et les systèmes de particules.

3.2.2.1 Modèles dynamiques

Les méthodes dynamiques [BW97], [RGL05] procèdent selon le même principe que celui des méthodes cinématiques, à la différence que les paramètres précisés sont liés à la dynamique du personnage et non plus uniquement à sa cinétique.

Les méthodes dynamiques intègrent généralement des lois physiques relatives aux accélérations linéaires et angulaires. Le mouvement est décrit par les équations dynamiques, établies en prenant en compte les forces, les moments de force, les contraintes imposées au système, ainsi que les propriétés de masse des objets le constituant.

Le contrôle de l'animation à l'aide des lois dynamiques permet d'obtenir des mouvements réalistes et d'alléger le travail de l'animateur. Cette méthode ouvre également les possibilités pour générer des modèles qui réagissent de manière automatique à des contraintes externes telles qu'une collision.

Toutefois, les méthodes dynamiques requièrent une définition précise des forces et des propriétés des objets, ce qui représente une tâche peu intuitive pour un animateur dont l'objectif est de réaliser un mouvement donné et non pas de simuler une situation physique. Par ailleurs, le temps de calcul des animations devient prohibitif pour un modèle de hiérarchie complexe, ce qui réduit les possibilités d'utilisation des méthodes dynamiques. Enfin, les mouvements ainsi obtenus sont réguliers et n'intègrent pas la personnalité et le caractère des acteurs, ce qui leur donne un effet artificiel peu souhaitable.

3.2.2.2 Systèmes de particules

L'animation de *systèmes de particules* [FF01] est souvent utilisée pour modéliser des phénomènes naturels complexes constitués d'un grand nombre d'éléments individuels, comme la fumée ou l'eau. Les particules constituant ces systèmes ont généralement une apparence simple et se comportent selon des lois physiques relatives à l'environnement du système et indépendantes des autres particules.

La visualisation d'une agglomération de particules ainsi modélisées donne l'impression visuelle d'un modèle complexe. Ainsi, ces systèmes ont-ils été utilisés récemment pour animer des objets comme les cheveux ou la fourrure. Une application plus sophistiquée des systèmes de particules consiste en la modélisation de foules ou de groupes où les particules sont remplacées par des modèles 3D dont le comportement reste relativement simple.

De manière générale, les modèles physiques permettent donc de réaliser des animations complexes et réalistes. Toutefois, elles présentent l'inconvénient de produire invariablement le même mouvement pour une même situation, ce qui n'est pas le cas dans la réalité. Cela est dû à l'absence de variabilité entre les différents objets ou acteurs synthétiques. Afin de résoudre ce problème, les méthodes comportementales procèdent en introduisant des paramètres qui caractérisent chaque personnage virtuel.

3.2.3 Méthodes comportementales

Les méthodes comportementales [Tha96] prennent en compte la personnalité du personnage. Dans un système comportemental idéal, le même acteur réalisera le même type d'action de manière différente à chaque simulation, et deux acteurs différents la réaliseront différemment. Cette technique a été utilisée par exemple par Reynolds [Rey87] pour simuler le comportement de groupes d'oiseaux où l'utilisateur spécifie la trajectoire de l'oiseau en tête du groupe et les différentes contraintes de chaque individu tels que la distance minimale/maximale aux autres éléments du groupe. De manière plus générale, les méthodes comportementales sont très utiles pour animer des groupes ou des foules en introduisant une touche de réalisme.

3.2.4 Conclusion

Les modèles d'animation génériques permettent d'obtenir des animations à la fois complexes et réalistes. Néanmoins, elles sont peu adaptées dans le contexte des dessins animés 2D où le réalisme des mouvements n'est que peu respecté. En outre, ce domaine d'application est caractérisé par une forte dépendance vis-à-vis du style de chaque dessinateur/animateur, ce qui est peu compatible avec une animation régie par des lois prédéfinies où la touche artistique reste malgré tout absente.

Pour toutes ces raisons, nous nous sommes orientés vers les méthodes descriptives. Celles-ci sont d'ailleurs représentées dans les standards sur l'animation 3D.

3.3 Standards pour l'animation 3D

3.3.1 H-ANIM de VRML

Les premiers travaux de standardisation pour des modèles 3D de synthèse ont été initiés en 1997 par le groupe VRML [VRM]. L'approche d'animation adoptée est alors l'interpolation de positions clefs pour chaque transformation géométrique élémentaire (translation, rotation, ...) ou bien pour chaque sommet du modèle 3D. Les spécificités de l'animation de modèles d'humanoïdes ont été établies par le groupe H-Anim [HAN] qui a défini la géométrie du corps humain comme une structure hiérarchique constitué :

- d'un objet *humanoïd* qui constitue la racine du personnage H-Anim,
- d'un ensemble de *nœuds* représentant les articulations du personnage, et organisés hiérarchiquement en partant de la racine *humanoïd*,
- d'un ensemble de *segments* représentant les différentes parties du modèles (tels que la jambe ou le bras), et décrivent leur apparence et leur géométrie,
- d'un ensemble de *displaceurs* qui spécifient les contraintes de mouvement pour chaque segment,
- et d'un ensemble de *sites*, qui sont des emplacements particuliers pouvant servir pour attacher des accessoires ou des vêtements au personnage par exemple.

L'animation de ce modèle est effectuée de manière indépendante pour chaque segment et peut être raffinée pour obtenir des déformations locales en modifiant directement la position des sommets.

Cette approche a été reprise et complétée par le standard d'animation *Face and Body Animation* (FBA) de la norme MPEG-4, afin de pouvoir animer des modèles plus généraux et élaborés.

3.3.2 FBA de MPEG-4

La norme MPEG-4, développée par le groupe MPEG (*Moving Picture Experts Group*) [MPE] vise à créer un cadre commun pour des applications multimédia en définissant des scènes complexes de contenus audiovisuels et d'objets textuels ou graphiques 2D et 3D, ainsi qu'à assurer la transmission et la compression de ce type de scènes.

De manière générale, un personnage virtuel MPEG est décrit par un ensemble de nœuds organisés de manière hiérarchique. La surface d'un personnage virtuel conforme à la norme MPEG-4 peut être définie par un ensemble de segments ou parties indépendantes. La spécification FBA (*Face and Body Animation*) a mis en place une représentation précise de la géométrie et de l'animation d'un personnage virtuel qui utilise à cet effet deux groupes de paramètres :

- Un ensemble de paramètres relatifs à la géométrie, à la texture et à la hiérarchie du modèle FBA, constitué par les paramètres de définition du visage FDPs (*Face Definition Parameters*) d'une part, et par les paramètres de définition du corps BDPs (*Body Definition Parameters*) d'autre part. Ces paramètres permettent de créer et de décrire un modèle FBA en précisant sa forme et sa texture.

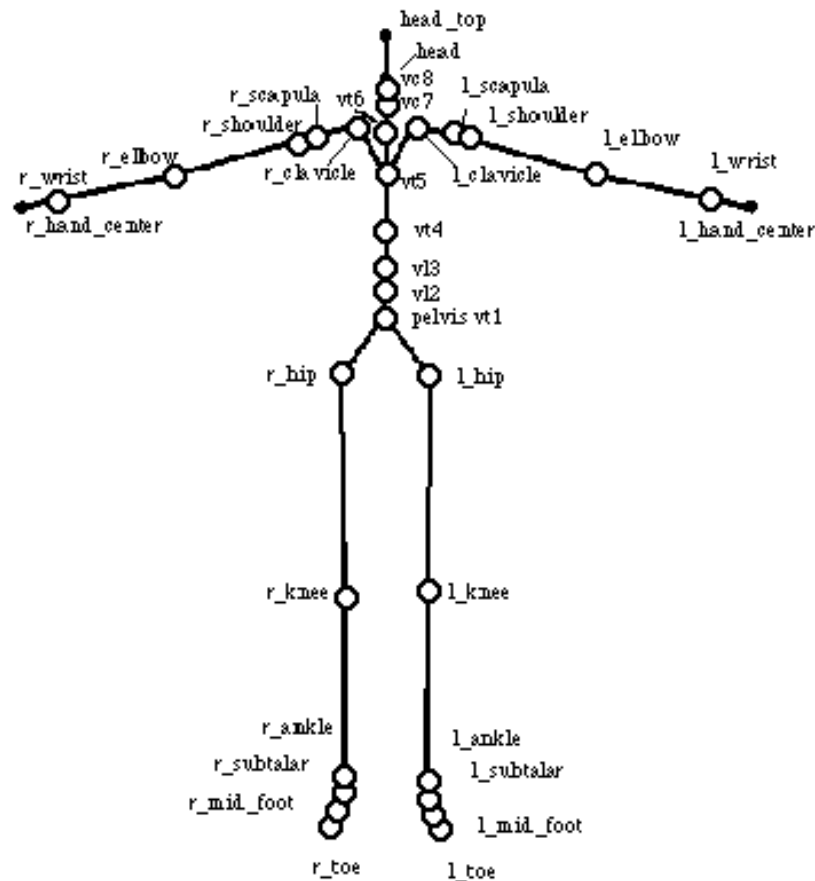


Figure 3.14 – Hiérarchie d'un corps humain dans la norme MPEG-4.

- Un ensemble définissant les paramètres d'animation du visage et du corps, qui sont les FAPs (*Face Animation Parameters*) et les BAPs (*Body Animation Parameters*). Ces paramètres définissent par exemple les degrés de liberté des articulations, ou des contraintes d'animation.

Le visage et le corps sont définis et traités de manière indépendante et différente dans la spécification FBA. En effet, celle-ci spécifie pour le visage 84 *points clefs* caractérisant sa géométrie et qui sont utilisés pour générer les animations. Ces points (Figure 3.15) correspondent à des éléments particuliers du visage humain (coins des lèvres par exemple). L'animation du visage s'effectue soit en utilisant ces positions standardisées, soit en modifiant manuellement la position des points clefs, et en déformant la surface au niveau des régions avoisinantes. En effet, il existe des FAP de bas niveau et de haut niveau. Les premiers mettent en œuvre des points clefs qui influencent le maillage lorsqu'ils sont déplacés afin de le déformer, alors que les deuxièmes sont constitués par 6 expressions et 14 visèmes prédéfinis. Les animations sont alors générées en combinant deux ou plusieurs de ces expressions et visèmes qui constituent une base.

Le corps du personnage est quant-à lui représenté par un assemblage arborescent de nœuds associés à des segments du modèle dont le nœud racine est le nœud *Body*, de la même manière que dans le schéma hiérarchique précisé par le standard H-ANIM (Figure 3.14). Les paramètres d'animation du corps (BAP) définissent les angles de rotation et les propriétés d'animation d'un segment du modèle relativement à son nœud parent. Afin de définir l'animation d'un modèle virtuel d'humanoïde, 296 paramètres de rotation sont utilisés. L'animation du maillage s'effectue alors en appliquant à chaque segment du modèle les transformations qui lui sont

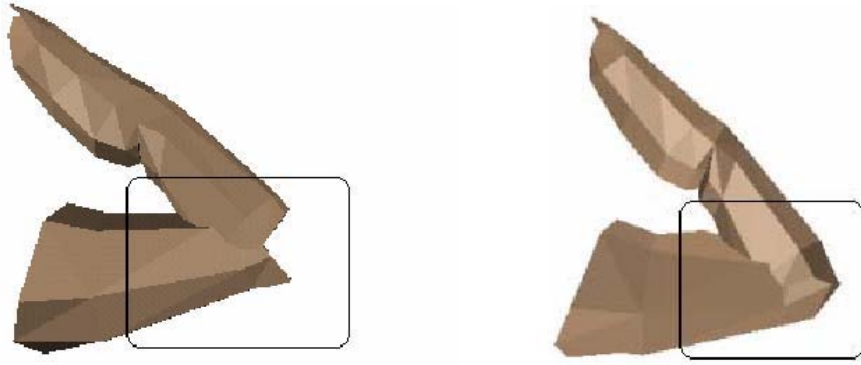


Figure 3.17 – Standard FBA : animation d’un doigt sans et avec utilisation des Tables de Déformation du Corps (BDT). L’utilisation des tables permet d’éliminer la déconnexion présentée dans le cadre d’une animation classique du modèle segmenté.

représentation plus générique que le schéma FBA, où un personnage est constitué de trois composants : un squelette, un ensemble de muscles et une peau ou enveloppe extérieure. Cette représentation en multi-couches a pour but de séparer les animations relatives aux articulations des déformations locales. Les animations ainsi obtenues étant inspirées des modèles humain et animal réels, elles donnent un résultat plus réaliste.

Afin d’effectuer l’animation, la surface d’un modèle n’est plus simplement segmentée en groupes de points, mais les sommets sont animés en fonction du ou des squelettes et du ou des muscles desquels ils dépendent. L’influence de chaque segment de squelette ou de muscle sur chaque sommet du modèle permet ainsi de déduire le déplacement de ce sommet à partir des transformations subies par les squelette et/ou les muscle du modèle.

Notons par $\Omega = v_0, v_1, \dots, v_n$ l’ensemble des sommet du maillage $M(\Omega)$, et par Ω_i un sous-ensemble non vide de Ω . Une *fonction de déformation* locale $\varphi_i : \Omega \rightarrow \mathbb{R}^3$ permet de définir le déplacement d’un sommet $v \in \Omega_i$ vers sa nouvelle position $v + \varphi_i(v)$. Les déformations ainsi définies s’appliquent aux sommets selon le principe de superposition de manière à ce qu’un sommet $v \in \Omega_i \cap \Omega_j$ soit déplacé de $\varphi_i(v) + \varphi_j(v)$.

Un modèle de déformation de base uniforme est utilisé pour les deux structures squelette et muscle. Ce modèle exploite une structure générale : les *contrôleurs de déformation*, définis par

- un *support* S associé à un objet géométrique de dimension n qui contrôle la déformation (os ou muscle par exemple),
- un *volume d’influence* $V(S)$ associé à S ,
- et une *mesure de propagation d’influence* μ , définie sur $V(S)$ et caractérisant les propriétés de déformation dans ce volume d’influence.

Soit $\psi_i(v)$ l’ensemble des éléments de S_i affectant le sommet $v \in \Omega_i$. Appliquer une transformation T_i au contrôleur C_i induit la fonction de déformation ϕ_i définie sur Ω_i par :

$$\forall v \in \Omega_i, \phi_i(v) = \mu_i(v) \sum_{\xi_k \in \psi_i(v)} \omega_k [T_i(\xi_k) - \xi_k]. \quad (3.31)$$

Ce modèle de déformation permet de gérer de manière homogène différents types d’animation obtenues pour différentes dimensions du support de contrôleur utilisé. Le choix d’un contrôleur de déformation approprié est par conséquent déterminé par deux éléments importants : la dimension et la complexité de représentation de son support d’une part, et la forme du volume d’influence qui lui est associé. Les contrôleurs utilisés par le standard SMS sont de dimension 1 : il s’agit de segments de droite dans le cas du squelette et de courbes NURBS pour les muscles. Les déformations du maillage sont alors obtenues par transformation rigides du squelette et/ou par déformation des muscles.

3.3.3.1 La couche squelette

Le squelette d'un personnage virtuel SMS est composé d'un ensemble d'os, définis comme un segment de droite de longueur l et caractérisé par :

- les transformations géométriques, qui définissent la position de l'os relative à son parent dans la hiérarchie du squelette,
- un modèle d'influence définissant la relation entre les mouvements du squelette et ceux de la couche peau,
- et des contraintes de cinématique inverse.

L'influence d'un os sur les sommets du maillage dépend de deux paramètres : la mesure de propagation d'influence, et le volume d'influence. Dans le cas du squelette, la définition de la *mesure d'influence* est partitionnée sur trois domaines 3.32 :

- un domaine intérieur Z_{int} qui représente le noyau où les sommets se voient attribuer une influence maximale et où ils suivent par conséquent de manière rigide les mouvements du squelette,
- un domaine extérieur Z_{ext} où l'influence de l'os considéré s'annule,
- et un domaine médian Z_{med} où la mesure d'influence évolue graduellement. Cette propagation s'effectue de manière symétrique par rapport au segment représentant l'os, et dépend de la distance du sommet à ce dernier.

Pour un point de l'os situé à la distance d de l'origine de ce dernier, la mesure d'influence générée en la position $x \in \mathbb{R}^3$ est alors décrite par :

$$\forall d \in]0, 1[, \mu_d(x) = \begin{cases} 1 & \text{si } x \in Z_{int} \\ f\left(\frac{\delta(x, Z_{ext})}{R_d - r_d}\right) & \text{si } x \in Z_{med} \\ 0 & \text{si } x \in Z_{ext} \end{cases}, \quad (3.32)$$

où $\delta(x, Z_{ext})$ est la distance euclidienne entre x et Z_{ext} , r_d et R_d caractérisent les domaines d'influence au niveau d de l'os, et f est une fonction choisie par l'utilisateur parmi un ensemble de fonctions prédéfinies : x^3 , x^2 , x , $\sin(\frac{\pi}{2}x)$, \sqrt{x} et $\sqrt[3]{x}$.

La mesure d'influence générée par les extrémités de l'os (pour $d = 0$ ou $d = 1$) se propage dans l'espace délimité par deux demi-sphères de centre l'extrémité considérée et de rayons r_0 et R_0 (respectivement r_1 et R_1) comme l'illustre la figure 3.18(a).

Un contrôleur de déformation de type os subit uniquement des *transformations* affines. Dans le cadre de l'animation de caractères virtuels, la transformation la plus utilisée est la rotation, qui définit l'orientation d'un os par rapport à son parent hiérarchique sur le squelette. Des transformations moins courantes telles qu'une mise à l'échelle peuvent cependant être utilisées, en particulier dans le cadre des dessins animés où des effets de style comme l'aplatissement ou l'inflation d'une partie du modèle, voire du modèle entier sont courants.

Dans le standard SMS, les transformation géométriques d'un os b_i sont décrites de manière générique par une matrice T_i de dimension 4 obtenue par la formule suivante :

$$T_i^w = TR^w b_i . R^w b_i . S^w b_i, \quad (3.33)$$

où $TR^w b_i$, $R^w b_i$ et $S^w b_i$ définissent respectivement la translation, la rotation, et la mise en échelle de l'os exprimées dans le système de coordonnées absolu.

A partir des transformations définies grâce au squelette, et des volumes et mesures d'influence associés à chaque os, les déplacements des sommets appartenant à la couche peau peuvent être déduites selon le principe de superposition en appliquant l'équation 3.31.

Cette technique permet de gérer de manière continue et graduelle l'animation du modèle, en particulier au niveau des articulation, et d'éliminer ainsi l'effet mécanique et discontinu des animations présentes dans le schéma FBA.

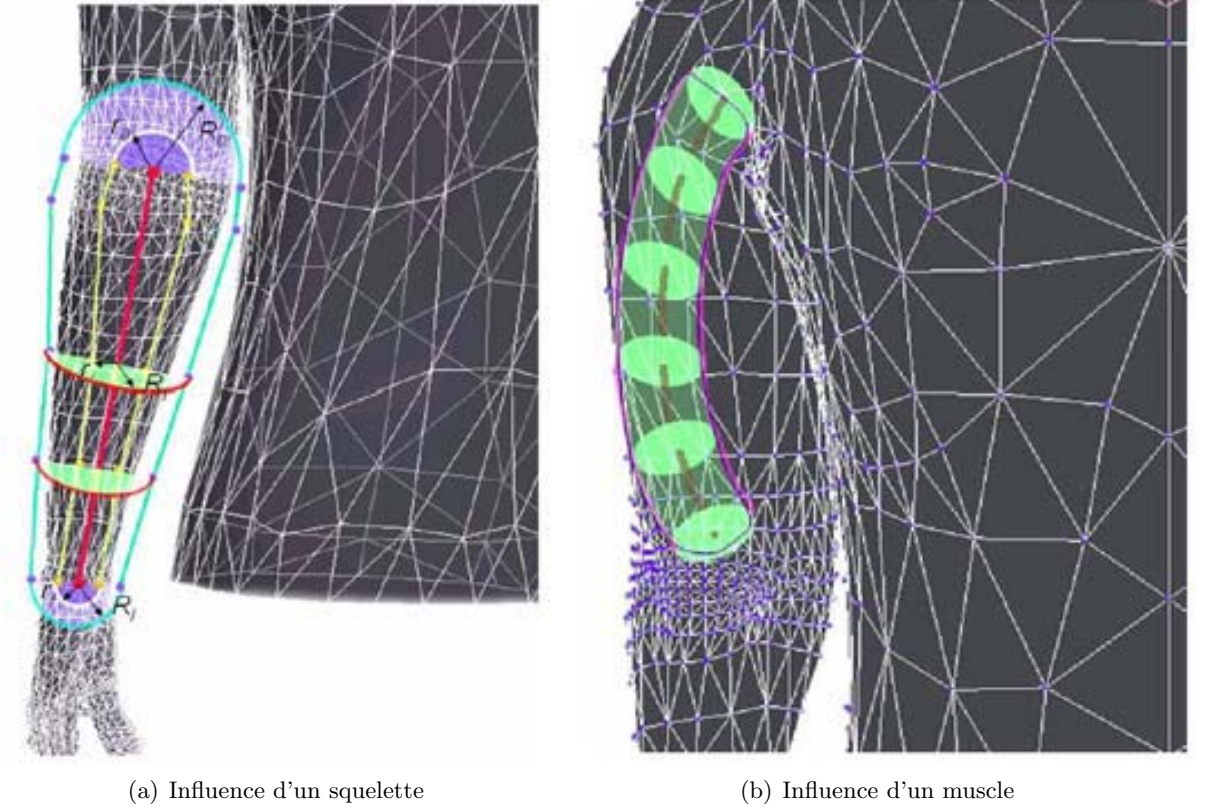


Figure 3.18 – Contrôleurs d'animation BBA dans la spécification SMS de MPEG-4 [Pre02].

3.3.3.2 La couche muscle

Les déformations locales sont gérées par la couche muscle qui bénéficie d'une modélisation plus complexe et plus souple que le squelette. En effet, un muscle est représenté par une courbe NURBS ; le volume d'influence correspondant est généré par un cercle de rayon r se déplaçant le long de cette courbe. La mesure d'influence est alors définie sur deux zones, et est donnée en un point x de l'espace par :

$$\mu(x) = \begin{cases} f\left(\frac{r - \delta(x, \psi(x))}{r}\right) & \text{si } \delta(x, \psi(x)) \leq r \\ 0 & \text{si } \delta(x, \psi(x)) > r \end{cases}, \quad (3.34)$$

où $\delta(x, Z_{ext})$ est la distance euclidienne, f est une fonction choisie par l'utilisateur parmi la même famille de fonctions prédéfinies que dans le cas du squelette, et ψ est la fonction attribuant à x un point correspondant sur la courbe du muscle. La figure 3.18(b) illustre un muscle et sa zone d'influence.

Le muscle étant modélisé par une courbe NURBS, il jouit d'un certain nombre de paramètres qui peuvent être modifiés indépendamment les uns des autres, et dont l'influence sur la forme finale du muscle est différente. Il s'agit en effet de la position des points de contrôle, des poids associés à ces derniers, et du vecteur des nœuds. Une étude détaillée de l'influence de chacun de ces éléments sur la courbe NURBS est présentée dans [PT95]. Notons que le positionnement des points de contrôle est un moyen intuitif et rapide de déformer une courbe NURBS, celle-ci étant par définition une approximation de la poly-ligne définie par ces points. La modification des valeurs des poids et des nœuds affecte quant-à elle la manière dont les points de contrôle sont approchés.

Les déformations subies par la courbe de muscle se répercutent ensuite sur la couche peau au niveau des sommets contenus dans la zone d'influence, et ce, selon le schéma général défini par l'équation 3.31.

3.3.3.3 Conclusion

Afin de pouvoir compléter les standards l'ayant précédé, le schéma SMS intègre les modèles segmentés en utilisant comme segments de squelette l'enveloppe d'un segment anatomique. Ces derniers étant animés de manière rigide, ils peuvent en effet être considérés comme des squelettes.

	<i>FBA</i>	<i>SMS</i>
Type de modèle	Modèle segmenté	Modèle <i>seamless</i>
Hiérarchie	Standardisée	Générique
Méthodes de déformation	Cinématique directe	Cinématique directe, Cinématique inverse, Interpolation de poses clefs
Paramètres d'animation	296 pour le corps, 68 pour le visage	Nombre de paramètres libre

Tableau 3.1 – Tableau comparatif des standards FBA et SMS [Pre02].

De plus, d'un point de vue animation, la représentation hiérarchique permet de décrire des personnages de forme quelconque et de s'affranchir des contraintes spécifiques aux humanoïdes du standard H-ANIM. Cela procure à ce standard plus de souplesse et de généralité, d'autant qu'il intègre les modèles segmentés du type FBA ainsi que les structures hiérarchiques humanoïdes du type H-ANIM. MPEG-4 est le standard le mieux adapté à des applications du type dessins animés où les personnages peuvent être purement imaginaires, et où la personification est souvent utilisée pour animer des objets comme une armoire ou un arbre de la même manière qu'un humain.

Un autre avantage offert par le schéma d'animation par squelette (*Bone Based Animation* ou BBA) consiste en la gestion des articulations qui s'effectue de manière continue et naturelle (Figure 3.19). Les animations ainsi obtenues sont plus fluides et semblent réalistes. Par ailleurs, l'introduction de la couche muscle élargit le spectre des déformations possibles et facilite le procédé de déformation qui s'effectuait généralement en modifiant les sommets manuellement autant dans le cadre des standards précédents qu'en utilisant les logiciels de modélisation et d'animation [MAY][MAX].

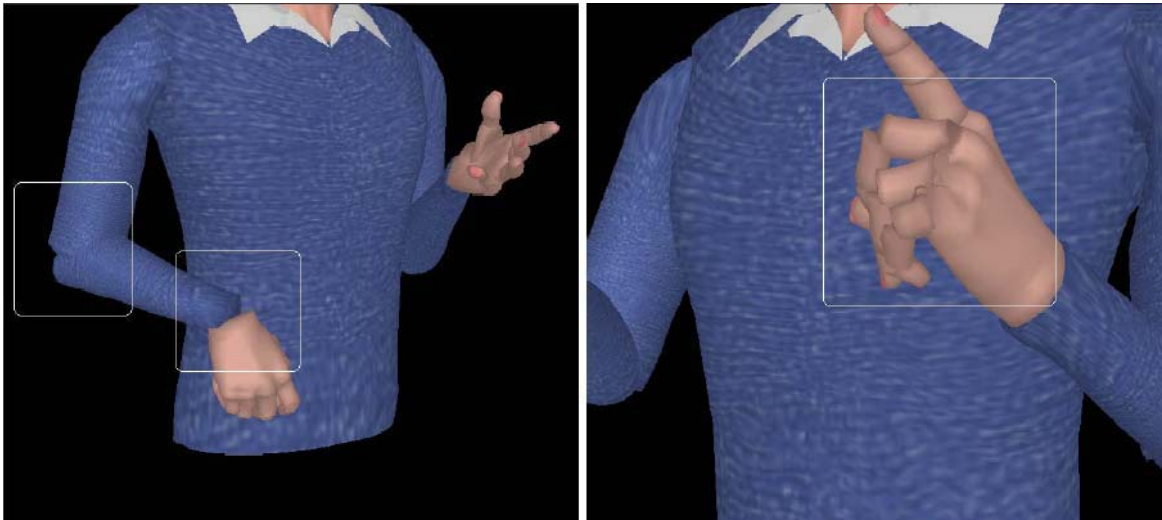
Enfin, contrairement au standard FBA dont les animations sont uniquement effectuées à l'aide de la cinématique directe, SMS est compatible avec la cinématique inverse et l'interpolation de poses clefs. L'intégration de l'animation par métamorphose est quant-à elle en cours de réalisation.

En résumé, SMS est le standard le plus complet comme le montre le tableau 3.1, offrant un plus grand nombre de possibilités à la fois en termes de modélisation géométrique, de choix de la représentation des surfaces, et en d'animation. Cela en fait l'outil le plus approprié et le plus adapté pour des applications de dessins animés. Nous le retenons donc pour notre étude.

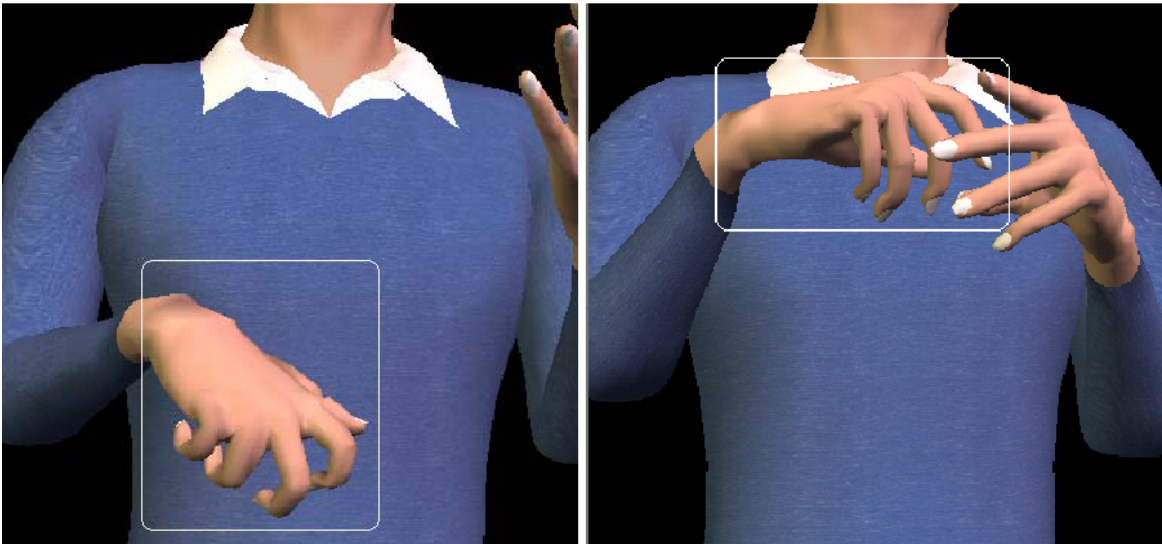
3.4 Conclusion

Prenant en considération la contrainte de conformité à un standard d'animation, et étant donnée la supériorité du standard AFX et son adéquation à nos objectifs, il devient nécessaire de prendre en compte les restrictions que cela impose d'un point de vue modèle de surface utilisé. En effet, l'animation BBA n'est actuellement prise en charge par le standard que pour les surfaces maillées. Il en découle que, quel que soit le choix de méthode de reconstruction effectué, les modèles doivent être convertis en maillages avant de procéder aux étapes d'animation.

D'un point de vue animation, les méthodes descriptives semblent être les plus appropriées aux dessins animés 2D puisqu'elles utilisent des poses clefs (*c.f.* Tableau récapitulatif 3.4),



(a)



(b)

Figure 3.19 – Animation d’un personnage virtuel : Gestion des articulations pour un modèle segmenté (3.19(a)), et pour un modèle seamless animé selon le standard SMS (3.19(b)) [Pre02].

	<i>Méthode d’animation</i>	<i>Données d’entrée</i>	<i>Loi d’évolution</i>
Modèles descriptifs	Interpolation de poses clefs	Poses clefs des paramètres	Interpolation (curviligne)
	Métamorphose	Poses clefs des modèles	Métamorphose
Modèles générateurs	Capture de mouvement	Mouvements réels capturés	Plaquage par correspondance
	Méthodes dynamiques	Forces	Lois physiques
	Méthodes procédurales	Objectifs	Lois/Algorithmes

Tableau 3.2 – Méthodes d’animation.

ce qui concorde parfaitement avec le mode de travail traditionnel des animateurs 2D. Ces différentes méthodes sont appliquées le plus souvent aux maillages ce qui confirme notre choix de représentation finale des surfaces.

Chapitre 4

Reconstruction 2D/3D de personnages virtuels

Résumé

Ce chapitre présente les différentes méthodes développées pour la reconstruction de personnages virtuels 3D à partir de dessins 2D. Ces approches s'appuient sur une première approximation grossière du modèle 3D, obtenue à l'aide d'une enveloppe visuelle générée par intersection de volumes à partir des dessins du *turnaround*. Le processus de reconstruction volumique est réalisé par partie de l'objet et exploite une segmentation fournie par les dessinateurs. Afin de construire à partir de cette enveloppe visuelle un modèle maillé sans coutures, deux approches différentes ont été développées.

La première utilise une représentation intermédiaire par surfaces NURBS, triangulées par la suite et assemblées en un unique maillage *seamless*, compatible avec le standard d'animation MPEG-4. La modélisation NURBS est bien adaptée à des objectifs de raffinement et de déformation locale de l'objet reconstruit. En revanche, générer des maillages sans couture à partir de modèles NURBS à multiples composantes nécessite la mise en œuvre de techniques d'assemblage, coûteuses en temps de calcul.

Afin de s'affranchir de cet inconvénient, une deuxième approche est proposée. Des modèles sans couture sont générés directement à partir du modèle volumique en exploitant les mécanismes de subdivision et de triangulation de Delaunay.

Dans les deux cas, un squelette d'animation MPEG-4 est utilisé pour spécifier la hiérarchie et les relations d'adjacence entre les différentes parties de l'objet.

Mots clés

Reconstruction 2D/3D, Shape from Silhouette, Enveloppe visuelle (Visual Hull), Intersection de volumes, Modèle volumique, Surfaces NURBS, Maillage seamless, Blending, Triangulation de Delaunay, Subdivision.

L'objectif de l'étape de reconstruction 2D/3D est d'obtenir à partir des images du *turnaround* fournies par les dessinateurs un maillage 3D sans couture, compatible et adapté à une animation MPEG-4.

La problématique de reconstruction de modèles 3D de personnages virtuels à partir de dessins 2D manuels présente un certain nombre de spécificités qui la distinguent nettement d'autres domaines d'application impliquant des méthodologies de reconstruction 2D/3D.

Parmi les contraintes fortes liées à notre application, mentionnons que :

- N'est disponible qu'un nombre très réduit de dessins (2 à 4 généralement) représentant le personnage sous différents angles de vue.
- Les éventuelles interactions utilisateur doivent être entièrement 2D, pour respecter les pratiques traditionnelles de travail des artistes créateurs.
- Par leur nature artisanale, les dessins utilisés présentent un certain nombre d'incohérences (Figure 4.1), ce qui nécessite un processus de reconstruction très souple, permettant de corriger les erreurs induites par ces incohérences.

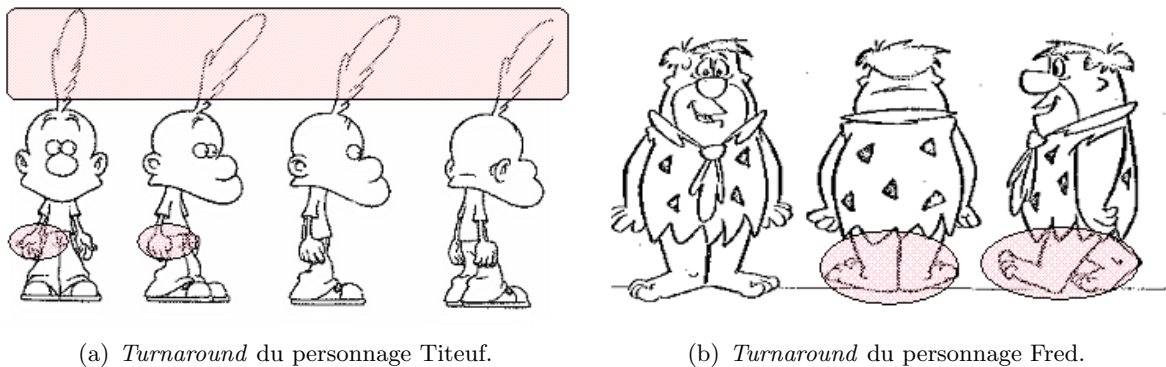


Figure 4.1 – Incohérences entre poses : les mains de Titeuf sont dessinées avec les paumes tournées vers l'avant sur la vue de face alors qu'elles sont collées au corps sur les autres vues. Sa crête est redessinée presque à l'identique sur les 3 vues différentes. Fred marche dans la vue de profil alors qu'il est dans une pose statique dans les vues de face et de dos.

Ces contraintes spécifiques rendent les techniques de reconstruction classiques peu adaptées à cette problématique. En effet, les méthodes de reconstruction de la littérature exploitent différents types d'information. Il s'agit soit de l'information de texture (dans le cadre des applications de type *shape from texture*) [Gar93], [MR97], soit de l'information de mouvement, dans le contexte des applications de suivi et de reconstruction 2D/3D à partir de séquences vidéo (approches *shape from motion*) [BWW94], [YS03].

Dans tous ces cas, un nombre élevé d'images est supposé disponible, ainsi qu'un modèle exact de caméra et de projection. Dans notre cas, aucune de ces informations n'est disponible : les personnages dessinés sont statiques et la texture est le plus souvent absente.

Une autre famille d'approches exploite une bibliothèque de modèles génériques, qui sont ensuite déformés de manière à correspondre aux images disponibles. Dans le cadre des dessins animés, la grande variabilité des modèles dessinés rend ce type d'approche inenvisageable.

Cet ensemble de difficultés a contribué à l'émergence d'une nouvelle famille d'approches en reconstruction 2D/3D, spécifique aux dessins manuels. Le paragraphe suivant présente un état de l'art de ces différentes approches.

4.1 Reconstruction de modèles 3D à partir de dessins manuels 2D : état de l'art

La reconstruction de modèles 3D de forme libre à partir de dessins manuels 2D représente un domaine de recherche très récent, avec des applications aussi diverses que la conception

assistée par ordinateur, les productions d'animation ou les outils d'illustration pour la mode, la botanique ou l'architecture. La difficulté principale consiste à déterminer un modèle 3D à partir d'une information 2D incomplète, ce qui est un problème mathématiquement mal posé.

Les premiers outils développés proposent des systèmes interactifs dédiés à la reconstruction ou à la reconnaissance de modèles géométriques dans le cadre d'applications telles que l'architecture ou la conception de pièces manufacturées. Ces interfaces utilisent souvent des techniques de modélisation gestuelle où les mouvements effectués par la souris ou le stylet d'une tablette sont classés et interprétés à l'aide d'une bibliothèque de gestes prédéfinis.

Ainsi, dans [MSK02], des configurations prédéfinies de dessins 2D sont-elles reconnues par un système de reconstruction 2D/3D qui interprète le dessin et génère un maillage 3D. Ici, une configuration de dessin est définie par un graphe d'arêtes constitué d'un nombre déterminé de courbes se rejoignant en des points de jonction (qui correspondent aux sommets de l'objet 3D dessiné). Une analyse des dessins est effectuée pour déterminer le nombre de courbes et leurs positions relatives afin de construire le graphe d'arêtes associé. Cela permet de vérifier si le dessin correspond à une configuration prédéfinie et de calculer alors les positions 3D des sommets de l'objet. Les faces sont ensuite définies à l'aide des courbes 2D dessinées en employant des contraintes de symétrie. Bien que la bibliothèque des graphes d'arêtes puisse être étendue, les objets ainsi reconstruits restent toutefois simples en raison des contraintes imposées sur les modèles (symétrie du modèle, face arrière et socle plans). Cela limite considérablement son usage pour la création de personnages virtuels de dessin animé.

Le système SKETCH [ZHH96] combine l'utilisation des gestes de la souris avec la reconnaissance géométrique pour créer et modifier des modèles 3D. SKETCH définit une grammaire gestuelle qui permet de créer des primitives d'extrusion et d'appliquer des opérations de type CSG¹, pour combiner à l'aide d'opérateurs booléens (*i.e.* réunion, intersection, soustraction) des primitives de forme élémentaire en objets plus complexes.

Quick-Sketch [EHBE97] est un système similaire exploitant une modélisation de surface paramétrique. Les modèles sont reconstruits à l'aide de primitives d'extrusion, des surfaces de révolution, ou des surfaces réglées² [FvDFH97], ce qui permet de créer une plus grande diversité de formes. La complexité des objets créés avec ce système reste toutefois limitée.

D'une façon générale, la forme des objets reconstruits à l'aide de ce type de systèmes reste simple, ce qui les rend peu intéressants pour des applications telles que le dessin animé.

Une deuxième famille d'approches a pour objectif de construire des modèles de forme libre, en s'appuyant sur des hypothèses plus ou moins arbitraires concernant la forme des objets dessinés : symétrie, épaisseur...

Dans ce cadre, mentionnons tout d'abord le système Teddy [IMT99], qui permet de reconstruire un modèle 3D maillé à partir d'un unique dessin 2D. L'information de profondeur est ici déterminée à partir du squelette de l'objet sous l'hypothèse que les objets dessinés vérifient toujours une relation de proportionnalité entre leur épaisseur et leur largeur.

A partir d'une forme simple, le système affine de manière incrémentale le personnage grâce à une boîte d'outils de manipulation incluant : navigation 3D, découpage de modèles, ajout de nouvelles parties, greffe de composantes prédéfinies ou disponibles dans d'autres modèles.

Bien que le système arrive à créer des personnages de forme libre, leur forme reste toutefois très élémentaire. En outre, créer des personnages plus élaborés nécessite un volume important d'interaction utilisateur.

Owada *et al.* [ONNI03] proposent une interface similaire à celle de Teddy, pour modéliser des objets de forme libre, présentant en particulier des structures internes. Les modèles 3D sont ici traités sous la forme volumique, de manière à faciliter les opérations d'édition et de construction de type CSG.

¹Constructive Solid Geometry

²Cas particulier de surfaces Hemitiennes

Dans [Kuk04], Kúkelová propose une interface similaire qui reconstruit des surfaces de convolution à partir d'un unique dessin 2D. Une analyse des courbes 2D de contour permet de déterminer un squelette polygonal utilisé pour définir une surface de convolution. L'approche met en œuvre une étape d'ajustement global minimisant une fonctionnelle d'énergie qui caractérise la distance entre la surface globale (définie par la somme pondérée des surfaces préalablement calculées) et les positions 3D des points du contour. Cette étape permet ainsi de contrôler des différentes primitives de surface associées aux segments indépendants du squelette.

Cette approche présente l'avantage d'hybrider une modélisation de surface par primitives et une construction hiérarchique par squelette morphologique. En revanche, la procédure finale d'ajustement global, réalisée par minimisation aux moindres carrés non linéaire est d'autant plus coûteuse en temps de calcul que le nombre d'éléments du modèle est élevé.

Dans [KHR02] et [AJ03] (système BlobMaker), des *blobs* sont reconstruits à l'aide de surfaces implicites variationnelles [TO99]. Dans [KHR02] la scène est organisée en une structure hiérarchique, de manière à déplacer et éditer plusieurs objets en même temps. Un outil de combinaison de primitives est également disponible. Il permet, à l'aide de courbes guides, de préciser la forme du *blending* au voisinage des surfaces assemblées. Blobmaker réalise la reconstruction 3D à l'aide d'une technique inspirée du système Teddy par dilatation des courbes dessinées. Des opérateurs d'assemblage et de correction des contours offrent une grande liberté de manipulation.

Dans [Kar01], la reconstruction 2D/3D utilise les surfaces implicites variationnelles. Ici, l'étape de dilatation est fondée sur des règles heuristiques, qui exploitent une analyse préalable des contours 2D, au travers d'une détection de points caractéristiques. Ici, deux classes de points sont définies : les points en T, correspondant à l'intersection de deux lignes qui forment la lettre "T", et les points corne (*cusp points*), sont des points du contour dont la tangente à la surface coïncide avec la direction de vue. Ces points sont utilisés afin de compléter les contours cachés à l'aide d'une minimisation d'énergie. Les contours cachés obtenus sont ensuite combinés aux contours initiaux sous forme de contraintes lors de la reconstruction 3D.

Les surfaces générées sont toutefois peu lisses au niveau des contours cachés et des jonctions entre primitives. Malgré l'exploitation des contours cachés qui apporte une information supplémentaire quant à la forme de l'objet dessiné, disposer d'une seule vue est trop limitatif puisque les défauts de reconstruction sont localisés sur les parties non visibles.

Dans le cadre spécifique des dessins animés, DiFiore *et al.* proposent dans [FR02] une méthode de reconstruction 2D/3D par modèles 3D simplifiés. L'idée est d'exploiter les esquisses par formes simplifiées de personnages, fournies par les créateurs dans la phase de spécification (*cf.* Chapitre 1, Section 1.1, Figure 1.1). Ces formes sont utilisées pour générer des surfaces de révolution, ensuite assemblées en un modèle 3D simplifié du personnage.

Dans [ZS03], les auteurs exploitent une modélisation par surfaces implicites (*blobs*). La méthode offre l'avantage de pouvoir attacher les différents éléments de détail du dessin initial (moustache du chat, ou ridicule...) au modèle 3D simplifié qui joue alors un rôle de support. Les reprojections du modèle ainsi créé conservent donc les détails du dessin initial, ainsi que son aspect manuel. Afin d'attacher les courbes manuellement dessinées, celles-ci sont d'abord classées en courbes de silhouette (contour), courbes géodésiques, ou courbes sortant de la surface. Un point ou un ensemble de points d'attache sont alors générés en fonction de la classe de la courbe considérée. Pour une courbe de silhouette, les points d'attache sont des points choisis sur le contour apparent de la surface 3D. Pour une courbe géodésique, un échantillon de points uniformément distribués sur la partie visible de la surface est exploité. Enfin, pour une courbe sortante, un seul point d'attache est positionné sur la surface.

De manière générale, ces approches présentent l'inconvénient d'être peu adaptées pour modéliser des formes pointues (*sharp features*).

Certaines approches de reconstruction 2D/3D à partir de dessins manuels offrent à l'utilisateur la possibilité de définir la profondeur du modèle. Le principe consiste à exploiter une courbe centrale, deux courbes de contour, et une troisième courbe de profil à partir de laquelle la profondeur du modèle est déterminée.

Dans [Gri99] et [KHR04], les auteurs appliquent ce concept aux surfaces implicites en utilisant des courbes de profil dessinées en 3D par l'utilisateur.

Dans [Gri99], Grimm propose un modèle de construction de surfaces implicites fondé sur des cylindres généralisés implicites. Cette représentation est décrite par une ou deux courbes 3D centrales, couplées soit à des courbes de section, soit à des profils.

Dans [KHR04], l'approche de reconstruction multi-vues proposée exploite le même modèle de surface implicite, pour réaliser des opérations de type CSG. Ici, la courbe centrale est reconstruite en 3D à partir de 2 vues de projection à l'aide de techniques de géométrie épipolaire. Deux courbes de section sont ensuite dessinées par l'utilisateur sur deux plans orthogonaux à la courbe centrale. Les courbes de section sont interpolées de l'ensemble d'échantillons ainsi calculés, un blob est reconstruit par surfaces implicites variationnelles.

Ces deux approches offrent un large éventail de surfaces à faible coût d'interaction. Cependant, elles impliquent une manipulation en 3D des courbes dessinées, nécessaire pour définir les courbes de section.

Afin de s'affranchir de cet inconvénient, dans [IITS04] et [CSSJ05], les courbes de section sont calculées à partir de dessins uniquement 2D.

Dans [IITS04], Ijiri *et al.* construisent d'abord un modèle 3D à partir des courbes centrale et de contour. Le modèle est ensuite déformé selon la direction de vue à l'aide de la courbe de profil, déduite par symétrisation à partir d'une courbe 2D dessinée de manière à relier deux points des courbes de contour. L'interface est proposée dans un contexte de création de modèles botaniques. Elle permet de générer rapidement des feuilles et des pétales.

Dans [CSSJ05], un modèle de surface paramétrique est défini à la fois par la courbe centrale, les courbes de contour, et les courbes de section. Ici, les courbes de section sont calculées en utilisant les courbes centrale et de contour. La forme de la courbe de section est par défaut circulaire, mais peut être modifiée conformément à une courbe 2D plus complexe, de la même manière que dans [IITS04]. Des outils interactifs de déformation sont également proposés. Ils s'appliquent soit au plan de dessin soit aux courbes de section calculées. Ainsi, une plus grande variété de formes peut-elle être obtenue.

Cette approche de reconstruction est très attractive, en particulier en raison de l'interactivité simple et purement 2D proposée. Néanmoins, la définition des objets à l'aide de deux courbes de contour ne permet pas de modéliser des objets articulés. En outre, le mode de définition du profil par une unique courbe simple est trop élémentaire pour spécifier des formes plus complexes, notamment dans le cas où le profil n'est pas régulier. Un visage par exemple présente des variations du profil au niveau des yeux, du nez et du menton, ce qu'une seule courbe de profil ne peut pas représenter.

Enfin, dans [BCCD04], une approche de reconstruction 2D/3D à partir d'images de profondeur (*i.e.* des images dont les niveaux de gris représentent l'information de profondeur) est décrite. Les formes dessinées sont tout d'abord échantillonnées régulièrement, puis des surfaces maillées sont générées à partir des points ainsi obtenus. Ces maillages sont ensuite déformés par des déplacements donnés par les niveaux de gris des images.

Bien que la technique offre une plus grande liberté de dessin que les approches précédemment discutées (pas de contraintes sur les contours, par exemple), elle présente peu d'intérêt d'un point de vue de la création artistique : il est en effet difficile de générer des images de profondeur pour des formes libres. En outre, les surfaces reconstruites ne sont pas continues, et la qualité du maillage reste médiocre en raison de l'échantillonnage régulier et grossier appliqué.

De manière générale, les méthodes existantes traitent le problème de la reconstruction

à partir de dessins manuels 2D soit à partir d'une vue unique, soit en utilisant des modes d'interaction 3D. Dans les deux cas, la difficulté principale, liée au manque d'information 2D, est résolue à l'aide d'hypothèses relatives à la forme des objets reconstruits, souvent supposée ronde et simple. Ces hypothèses sont restrictives dans le contexte du dessin animé où les formes représentées sont très variables. Or, les *turnaround* sont composés généralement de deux à quatre vues, ce qui constitue une information supplémentaire importante. L'utilisation de ces vues permettrait de modéliser une plus grande variété de formes.

Notons que, quels que soient le nombre de vues et la technique de reconstruction utilisée, l'information 2D restera trop incomplète pour pouvoir reconstruire à l'identique un modèle de synthèse à partir de ses projections 2D. En outre, la création d'un personnage est souvent un processus multi-étapes où les dessins sont modifiés localement pour ajouter des détails ou modifier un modèle à partir d'un autre. Pour cette raison, il est indispensable de disposer d'une représentation de surface aisément contrôlable, permettant de réajuster/rectifier interactivement le modèle 3D. Dans ce contexte, les surfaces paramétriques, en particulier les surfaces NURBS nous semblent, contrairement aux surfaces implicites, être appropriées grâce au contrôle local qu'elles offrent, ainsi qu'à leur continuité intrinsèque. De plus, la propriété de l'enveloppe connexe (volume contenu dans le polyèdre convexe défini par ses points de contrôle) pourra être exploitée pour définir le maillage de contrôle à partir de l'enveloppe visuelle volumique.

Nous proposons une méthode de reconstruction qui met à profit les différentes vues disponibles. Elle implique les trois étapes suivantes (Figure 4.2) :

1. Il s'agit tout d'abord de déterminer un volume englobant du personnage, constitué par son enveloppe visuelle (*Visual Hull* - VH) [SCMS01], [Lau99], définie comme l'intersection des volumes d'extrusion associés à chaque projection. Ce volume d'intersection inclut l'ensemble des points susceptibles d'appartenir à l'objet projeté.
2. La seconde étape consiste à construire une surface NURBS approchant l'enveloppe volumique ainsi obtenue. Cette étape assure la flexibilité du processus de reconstruction, les modèles NURBS étant fortement adaptées à la procédure interactive de raffinement et de déformation.
3. Enfin, la dernière étape concerne la construction, à partir des surfaces NURBS ainsi créées, d'un modèle maillé *seamless*, adapté aux objectifs d'animation MPEG-4 et de rendu.

Détaillons à présent l'étape de reconstruction volumique.

4.2 Reconstruction de l'enveloppe visuelle

Une première approximation de la forme de l'objet est constituée par l'enveloppe visuelle, construite comme suit. Notons par $I_j, 1 < j \leq N$ l'ensemble des vues de projection du *turnaround*. Le modèle de projection est ici orthogonal, et les angles de vue de chaque projection supposés connus. A chaque image de projection I_j , on associe un volume d'extrusion, noté $V_{\theta_j}^{extrusion}$ et défini par :

$$V_{\theta_j}^{extrusion} = \{(x, y, z) \in \mathbb{R}^3 | \pi_{\theta_j}(x, y, z) \in O_j\}, \quad (4.1)$$

où :

- θ_j représente l'angle de vue associé à l'image de projection I_j ,
- π_{θ_j} désigne l'opérateur de projection selon l'angle de vue θ_j ,
- O_j représente la région support de l'objet dans l'image de projection I_j :

$$O_j = \{(u, v) | I_j(u, v) = 1\}. \quad (4.2)$$

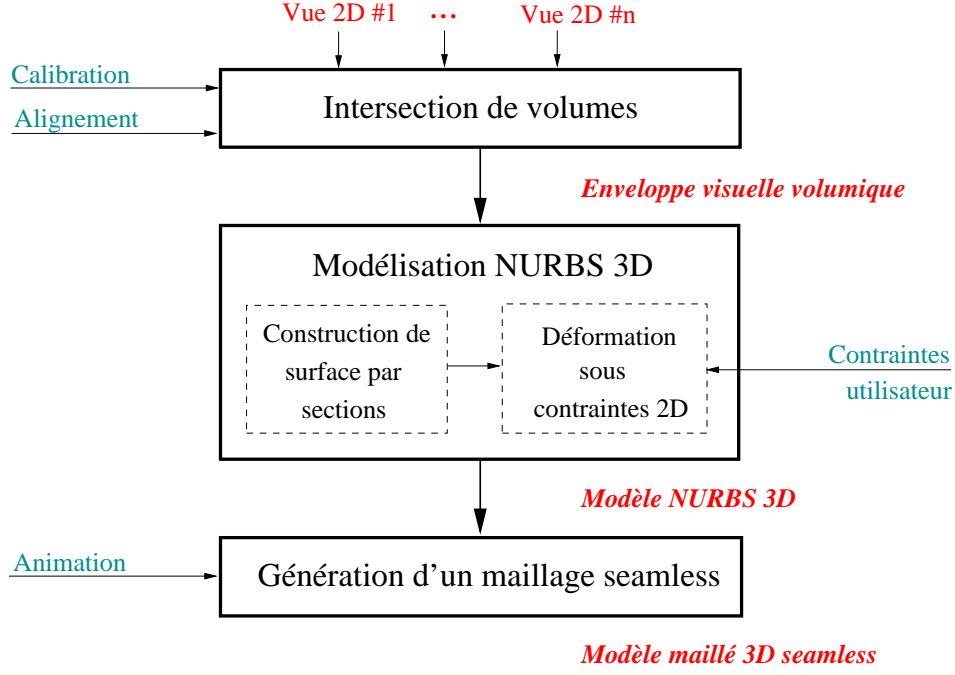


Figure 4.2 – Processus de reconstruction 2D/3D.

Le volume d'intersection V est défini par l'intersection de tous les volumes individuels d'ex-trusion :

$$V = \bigcap_{j \in J} V_{\theta_j}^{extrusion}. \quad (4.3)$$

4.2.1 Implantation

Le volume d'intersection est calculé à partir des images discrétisées déduites des dessins fournis.

Approche volumique

Soient L et H les dimensions des images fournies (respectivement largeur et hauteur). Nous supposons ici que les images sont de dimensions identiques et sont calibrées. Les dessins étant manuels et approximatifs, nous optons pour une simple calibration manuelle qui consiste à spécifier un axe vertical central pour chacune des vues. Les images sont ensuite centrées sur celui-ci. Une deuxième calibration est réalisée automatiquement afin de garantir une cohérence entre les hauteurs d'un même objet sur chaque vue en étirant les dessins.

Le volume discret contient $L \times L \times H$ voxels obtenu par étiquetage de la manière suivante :

- 1 si toutes leurs projections selon les angles de vue des images fournies se trouvent à l'intérieur de toutes les régions de support,
- 0 si au moins une projection du voxel se trouve à l'extérieur d'une région de support O_j .

On obtient ainsi un modèle volumique du VH, qui peut être triangulé par la suite grâce à la technique des *marching cubes* par exemple afin de le visualiser (Figure 4.8(b)).

Cette méthode nécessite le parcours de tous les pixels constituant le volume, ainsi que leur projection sur toutes les vues, d'où un nombre d'opérations à effectuer de l'ordre de $L \times L \times H \times N$, où N est le nombre de vues utilisées.

Même en négligeant N (de l'ordre de 3 ou 4) devant la dimension des images (supérieure à 100), la complexité de l'algorithme est en $O(L^3)$, ce qui conduit à un temps de calcul prohibitif

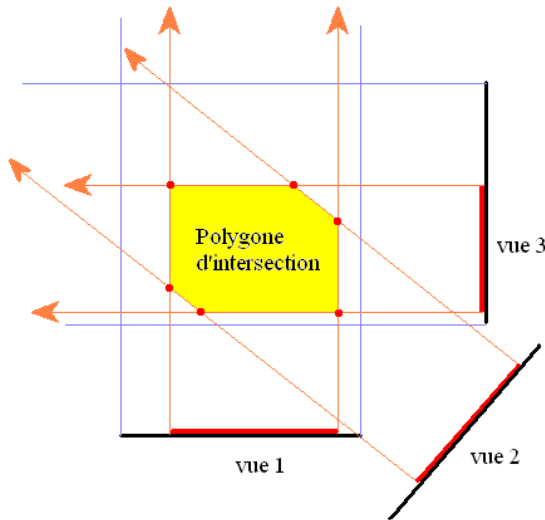
pour une application interactive, et notamment lors d'une étape de première approximation. Pour cette raison, une deuxième approche par contours est proposée.

Approche par contours polygonaux

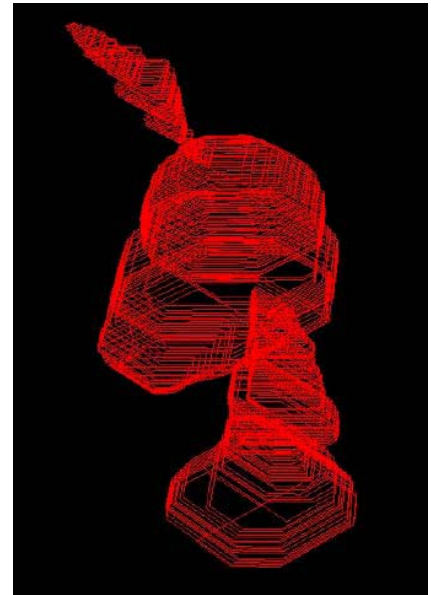
Le modèle de projection considéré étant orthogonal, une coupe horizontale du modèle volumique précédemment calculé découle exclusivement des contours inclus dans le plan de la coupe considérée.

Dans cette approche, nous utilisons cette propriété pour décrire le VH^3 par un ensemble de polygones horizontaux qui correspondent à l'intersection de volume en chaque coupe. Ces dernières sont données par la coordonnée verticale y des images de départ.

Les polygones sont calculés par une procédure itérative. Le polygone initial est un carré dont les dimensions sont données par la largeur des images de départ. Pour chaque ligne de chaque image, les coordonnées des points du contour sont projetées selon l'angle de vue de l'image considérée (Figure 4.3(a)). L'intersection de ce rayon avec les arêtes du polygone courant donne les nouveaux sommets du polygone, ce dernier est alors rogné en conséquence. La figure 4.3(b) illustre la série de polygones ainsi obtenue.



(a) Intersection de volume par coupe.



(b) Visual Hull par coupes de Titeuf.

Figure 4.3 – Visual Hull par coupe.

Cet algorithme nécessite, pour chaque ligne d'image, le parcours de chaque vue et un calcul d'intersection avec le polygone courant. Or, pour un nombre de vues N , le polygone final contient au maximum $2N$ segments. Pour une ligne avec deux points de contour, il y a $2N \times 2N \times H$ opérations. En négligeant le terme N par rapport aux dimensions des images, et en considérant que H et L sont du même ordre de grandeur, nous obtenons une complexité de l'algorithme en $O(L)$, ce qui représente un gain en temps de calcul considérable par rapport à la méthode volumique.

Lors du processus de reconstruction, un ensemble de points caractéristiques est extrait du modèle volumique afin de construire le maillage de contrôle de la surface NURBS. Ces points correspondent aux sommets des polygones de chaque coupe. L'approche par contours polygonaux permet de les calculer directement. En revanche, ces points doivent être détectés à partir du modèle volumique par l'algorithme de détection de contour de Freeman (*cf.* paragraphe

³Visual Hull (enveloppe visuelle)

4.3.1.1), ce qui constitue un temps de calcul supplémentaire. Par ailleurs, les coordonnées des sommets étant calculées exactement dans l'approche par contours polygonaux, l'effet de la discrétisation longitudinal est réduit.

4.2.2 Reconstruction volumique à partir de données de synthèse

Pour évaluer la capacité de ce processus de reconstruction 2D/3D par intersection de volumes, nous avons tout d'abord considéré des données 3D de synthèse, pour lesquels les images de projection peuvent être générées automatiquement, selon un modèle de projection exact. Pour illustrer la construction du volume d'intersection, nous avons projeté un modèle 3D d'humanoïde (Figure 4.4) au format VRML [VRM] selon huit angles de vue (Figure 4.4(b)) correspondant à des rotations de la caméra autour de l'axe vertical z de 0° , 45° , 90° , 135° , 180° , 225° , 270° et 315° . Le volume d'intersection obtenu est présenté Figure 4.4(c).

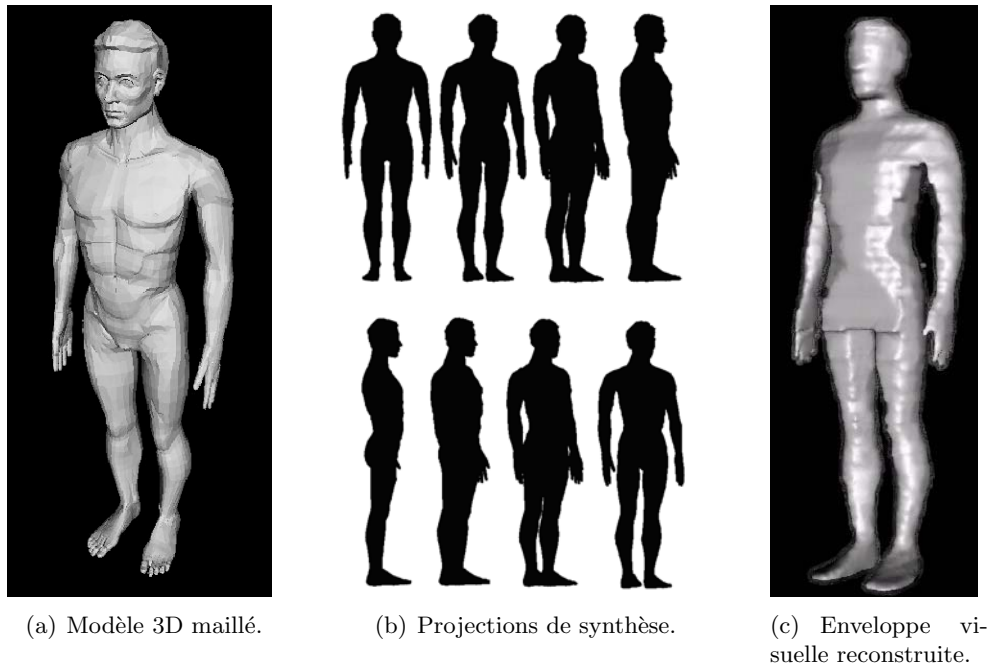


Figure 4.4 – Images de projection de synthèse obtenues à partir d'un modèle d'humanoïde maillé, et VH reconstruit par intersection de volume.

Le VH reconstruit fournit une première approximation grossière de l'objet 3D. Toutefois, même dans le cas idéal où le modèle initial et le modèle de projection sont connus, la méthode de reconstruction par intersection de volumes présente un certain nombre de limitations liées aux problèmes d'auto-occultation. Ce problème est illustré Figure 4.5.

Ici, dans la vue de profil, la main recouvre une partie de la jambe, ce qui se traduit au niveau du volume d'intersection reconstruit par l'apparition d'une "bosse" à la hauteur des cuisses. Cela est dû à la nature articulée des personnages virtuels.

Dans le cadre du projet TOON, pour éviter ce problème, un protocole d'acquisition multi-couches a été mis au point. Les différentes composantes corporelles sont représentées dans des couches (*layer*) séparées. Ce protocole de travail est parfaitement compatible avec les pratiques traditionnelles des dessinateurs. A noter que la segmentation est *de facto* réalisée puisque intrinsèquement liée à la création des images.

Notons que cette solution n'est pas la seule envisageable, une autre possibilité étant d'utiliser une tablette de dessin numérique pour re-tracer les contours et séparer ainsi les différentes parties de l'objet.

Une fois la segmentation disponible, le processus de reconstruction volumique peut être

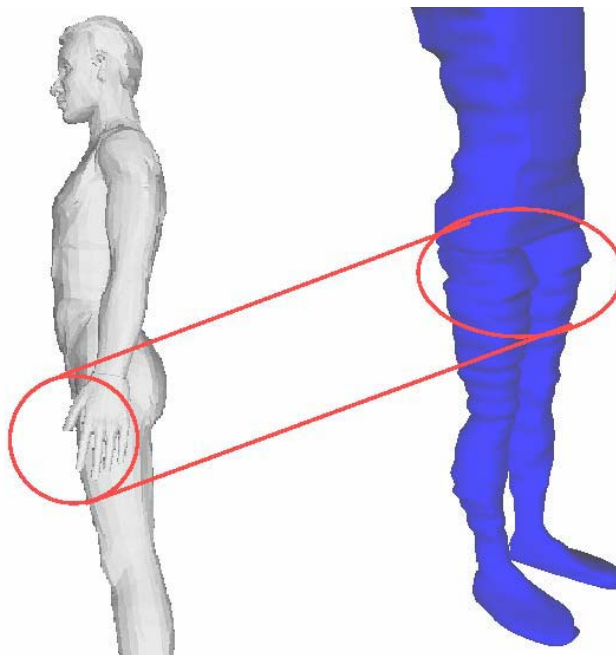


Figure 4.5 – Intersection de volumes : la main occulte les jambes, ce qui conduit à une mauvaise reconstruction.

effectué individuellement pour chaque composante, ce qui permet de s'affranchir des problèmes liés aux auto-occultations.

4.2.3 Reconstruction volumique à partir de données réelles

Présentons maintenant un deuxième exemple de reconstruction, réalisé cette fois à partir d'un jeu de données réelles, correspondant au célèbre personnage Titeuf (Figure 4.6).

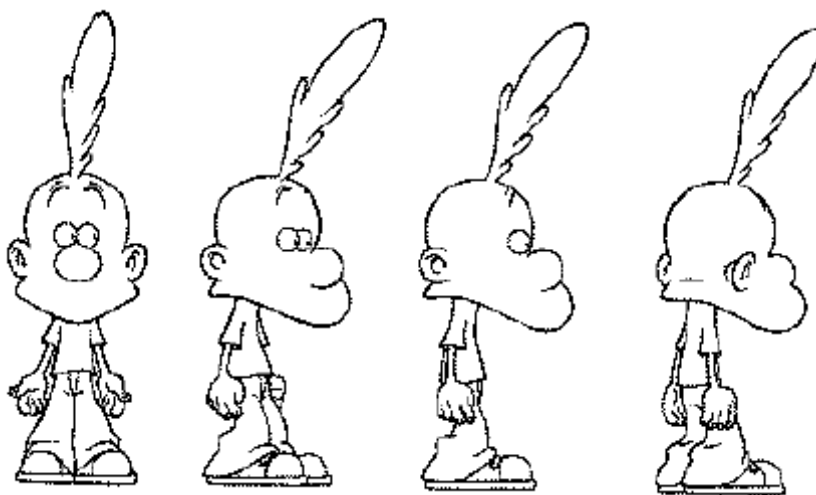


Figure 4.6 – Intersection de volumes : dessins du turnaround du personnage Titeuf.

La reconstruction à partir de données réelles nécessite une phase préliminaire de calibration : spécification des angles de projection associés à chaque vue, axe de vue, axe vertical de rotation, mise à l'échelle des images de projection...

En l'absence d'un vrai modèle de projection, d'information de texture ou de mouvement, cette étape est réalisée manuellement. Les différents paramètres estimés ne sont alors que des

approximations plus ou moins fiables.

Dans le cas des images de Titeuf, les angles de vue considérés sont : 0° , 45° , 90° et 135° . Notons que, dans le cas d'images réelles, il existe toujours une certaine incertitude sur les angles de vue. Cela risque d'introduire certaines déformations dans le modèle reconstruit mais qui restent négligeables, dans cette première étape d'approximation grossière.

A partir de ces images, par détection des contours extérieurs et remplissage, nous avons obtenu les régions support de l'objet, présentées Figure 4.7(a). La symétrie du personnage a été exploitée pour générer, à partir du *turnaround* initial, quatre vues supplémentaires (Figure 4.7(c)), obtenues par réflexion en miroir autour de l'axe vertical de rotation.

Le VH reconstruit est présenté Figure 4.7(b).

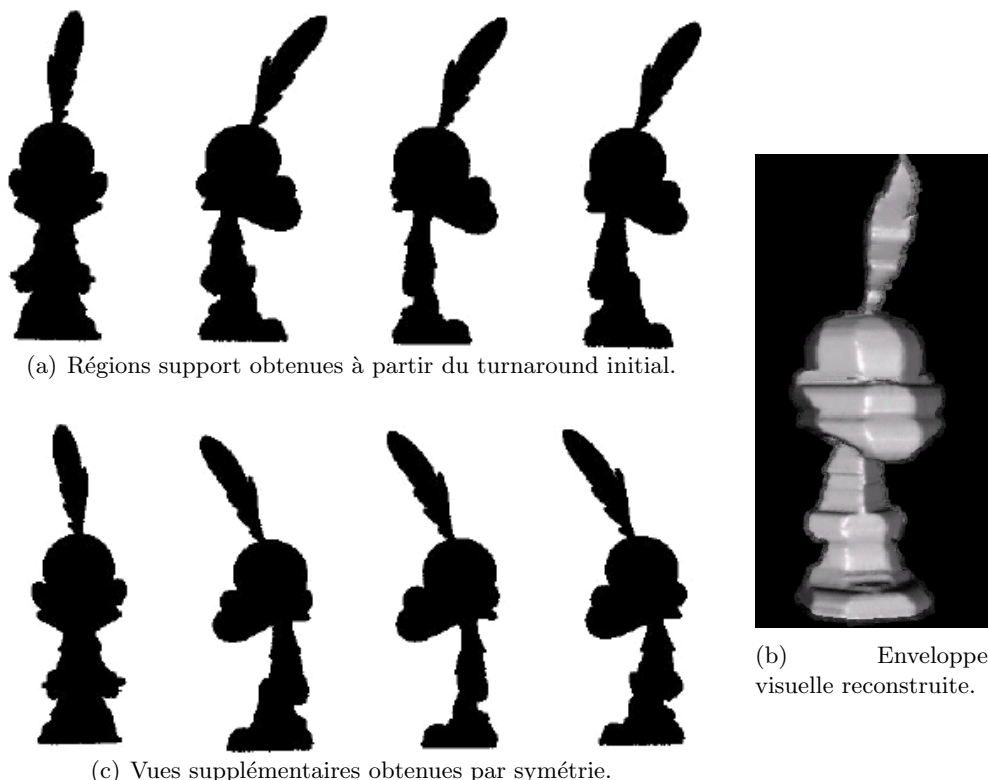


Figure 4.7 – Régions support du modèle Titeuf pour différents angles de vue, et VH obtenu par intersection de volumes.

Le résultat obtenu se heurte ici encore au problème de l'auto-occultation. De plus, l'aspect "monolithique" du volume reconstruit est dû au fait que les contours intérieurs ne sont pas pris en compte dans cette étape. Cela confirme la nécessité d'introduire un mécanisme de reconstruction volumique par composante.

Afin de tester l'apport d'une reconstruction par composante, nous avons segmenté manuellement le personnage Titeuf (Figure 4.8(a)). A l'évidence, la qualité de la reconstruction augmente considérablement, même si le volume obtenu reste une approximation grossière de l'objet.

Notons toutefois que, malgré l'imperfection de l'objet volumique reconstruit, les images 2D obtenues par projection du volume selon de nouveaux angles de vue représentent déjà une bonne approximation 2D du personnage. ces nouvelles vues pourraient être exploitées par les dessinateurs et les animateurs dans un contexte d'animation 2D.

Les résultats de reconstruction volumique obtenus pour différents personnages, de formes et de complexités variées, sont présentés Figure 4.10. Les couleurs correspondent ici à la segmentation considérée.

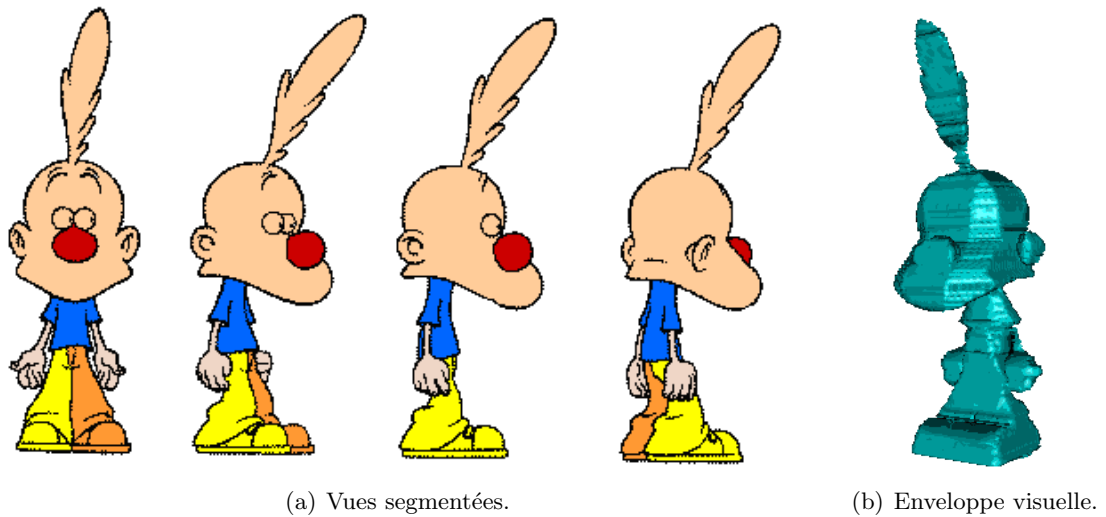
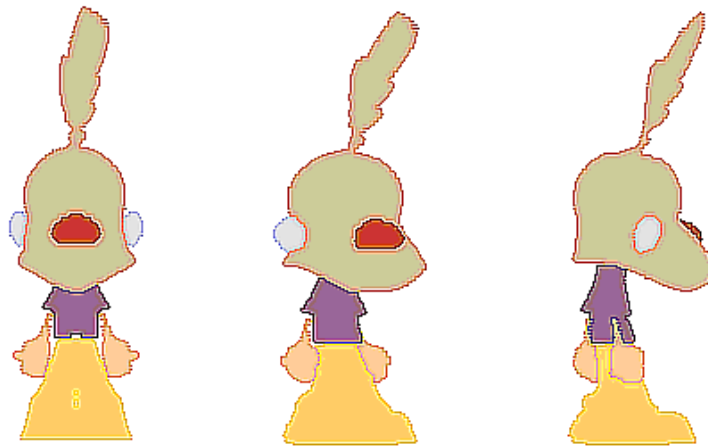


Figure 4.8 – Reconstruction volumique à partir de vues segmentées.

Figure 4.9 – Reprojections du modèle volumique de Titeuf obtenu à partir des vues segmentées selon de nouveaux angles de vue (de gauche à droite : 0° , 30° et 120°).

4.3 Reconstruction 2D/3D par surfaces NURBS

Pour générer, à partir de l'enveloppe visuelle, une représentation surfacique par NURBS, nous avons adopté une approche par sections. Pour obtenir les sections, le volume d'intersection est balayé selon l'axe vertical, noté z , qui correspond à l'axe de rotation des vues du *turnaround*. Pour chaque z , on obtient ainsi une image binaire qui peut être constituée d'une ou de plusieurs composantes connexes.

Le contour de chaque composante est alors approché par une courbe NURBS. La Figure 4.11 présente quelques images de section pour l'objet de type humanoïde de la Figure 4.4 correspondant à un échantillonnage uniforme de l'axe des z . Ici, de haut en bas et de gauche à droite, les images à unique composante convexe correspondent à la partie haute du modèle (tête et partie supérieure du thorax), celles à trois composantes connexes au tronc et aux bras, et enfin celles à deux composantes connexes aux jambes de l'humanoïde.

Pour générer des surfaces NURBS à partir de ces coupes horizontales, nous avons élaboré deux approches différentes. La première consiste en une procédure d'interpolation des courbes NURBS de section et vise à garantir une cohérence maximale par rapport aux images d'entrée. La seconde adopte une technique d'approximation afin d'obtenir des représentations plus lisses.

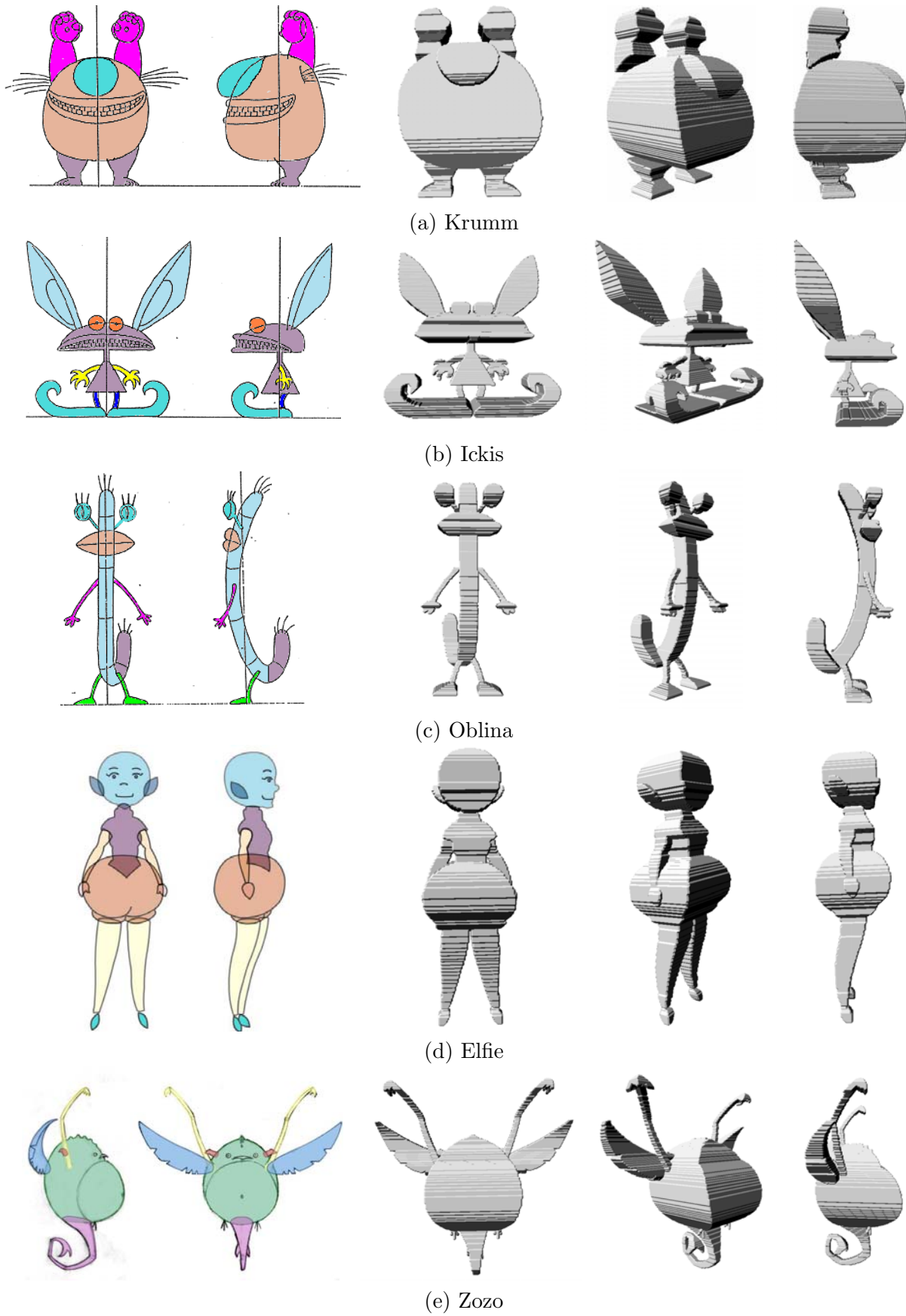


Figure 4.10 – Reconstruction par intersection de volumes pour différents personnages de dessins animés.

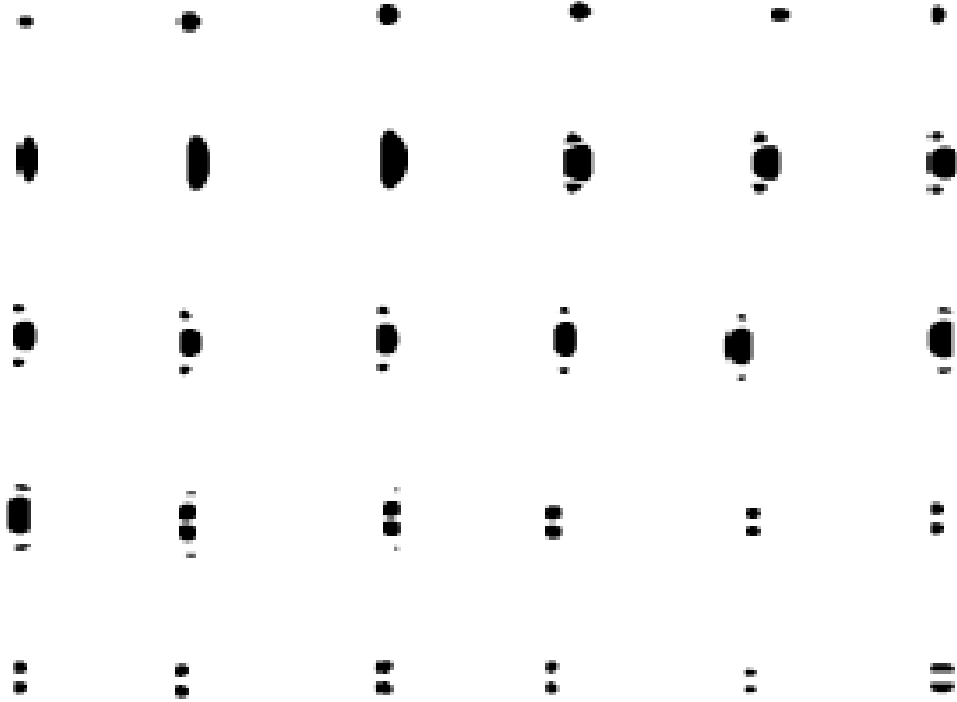


Figure 4.11 – Images des sections horizontales du volume d’intersection obtenu pour le modèle humanoïde.

4.3.1 Approche par interpolation

Le VH obtenu présente des angles vifs correspondant aux intersections des volumes d’extrusion. Dans la direction verticale en revanche, il correspond aux vues projectives de départ, et seul l’effet de discrétisation dû à l’utilisation de voxels est gênant.

Pour cette raison, nous effectuons dans un premier temps un lissage des coupes. Chaque section du modèle volumique du VH est tout d’abord approchée par une courbe NURBS, dont les points de contrôle sont déterminés selon la méthode décrite ci-dessous.

4.3.1.1 Détection des points de contrôle des coupes

Les coupes du volume d’intersection obtenu sont représentées par des images binaires. À partir de la région objet de la coupe considérée, nous effectuons tout d’abord une détection des points de contour à l’aide d’un codage de Freeman. Un balayage dans l’ordre lexicographique des pixels de l’image est réalisé pour détecter le premier pixel objet. Ensuite, les pixels de contour sont parcourus successivement et stockés dans une liste.

La Figure 4.12 présente un exemple de région support et le contour correspondant détecté alors que la Figure 4.13 indique les directions d_n dans un voisinage (3×3) et (5×5) .

Les points de contrôle sont ensuite déterminés comme les points du contour correspondant à un changement de direction du contour. Nous utilisons ici des directions discrètes définies dans des voisinages de différentes tailles. Les points de contrôle ainsi obtenus pour le contour présenté Figure 4.12 sont illustrés dans le cas de l’utilisation d’un voisinage (3×3) (Figure 4.14(a)) et (5×5) (Figure 4.14(b)).

Ayant obtenu les points de contrôle, le vecteur des nœuds est calculé à partir de ces derniers de manière uniforme.



Figure 4.12 – Détection de contour par codage de Freeman.

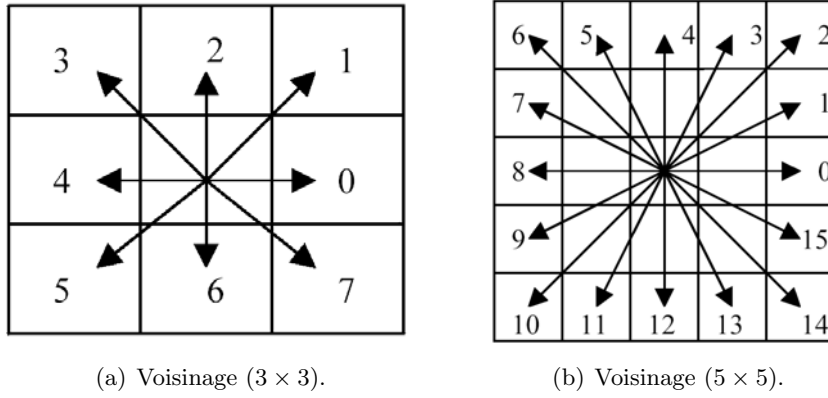


Figure 4.13 – Directions discrètes définies par le codage de Freeman.

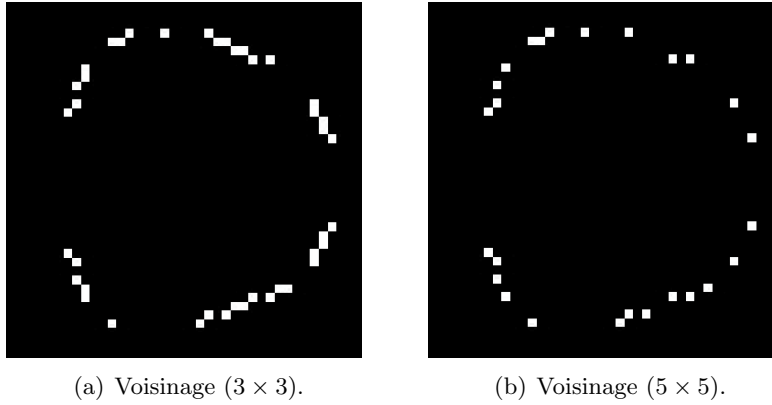


Figure 4.14 – Points de contrôle extraits pour la coupe présentée Figure 4.12.

4.3.1.2 Calcul des vecteurs de nœuds

Considérons un ensemble de points de contrôle $\{P_k\}_{k=0}^n$. Nous construisons dans un premier temps un vecteur $\bar{U} = (\bar{u}_0, \bar{u}_1, \dots, \bar{u}_n)$ défini comme suit :

$$\begin{cases} \bar{u}_0 = 0, \\ \bar{u}_k = \bar{u}_{k-1} + \frac{|P_k - P_{k-1}|}{d}, 1 \leq k \leq n, \end{cases} \quad (4.4)$$

où $d = \sum_{k=1}^n |P_k - P_{k-1}|$ est la longueur totale du polygone formé par les points de contrôle. Notons que $\bar{u}_n = 1$ par construction.

Le vecteur de nœuds final U est alors obtenu de la manière suivante :

$$\begin{cases} u_0 = \dots = u_p = 0, \\ u_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} \bar{u}_i, 1 \leq j \leq n-p, \\ u_{r-p} = \dots = u_r = 1. \end{cases} \quad (4.5)$$

où p est le degré de la courbe, et $r = n + p + 1$.

La Figure 4.15 présente la courbe NURBS obtenue pour la section de la Figure 4.12 conformément à cette procédure pour un degré 3.

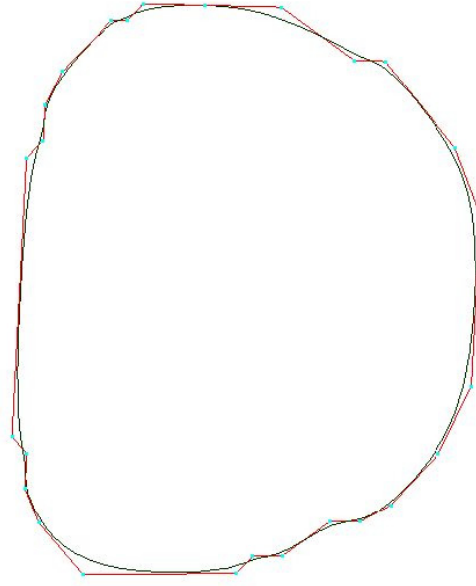


Figure 4.15 – Polygone de contrôle et courbe NURBS de degré 3 correspondant à la coupe Figure 4.12.

Pour chaque coupe $c \in \{0, 1, \dots, C\}$, est construit :

- un ensemble de points de contrôle $\{P_k^c\}_{k=0}^{n_c}$ où $n_c + 1$ est le nombre de points de contrôle de la coupe c ,
- un vecteur de nœuds associé U^c , constitué par $m_c = n_c + p + 1$ composantes.

Ayant calculé les courbes NURBS des différentes coupes, nous obtenons un ensemble de vecteurs de nœuds de tailles différentes. Cependant, une surface NURBS est définie par une grille de contrôle régulière.

Il est alors nécessaire d'avoir le même nombre de points de contrôle ainsi que le même vecteur de nœuds pour l'ensemble des coupes. Afin de vérifier cette condition, une procédure de *raffinement des nœuds* est nécessaire. Pour cela, nous avons adopté l'approche recommandée dans [PT95] et détaillée ci-dessous.

4.3.1.3 Uniformisation des vecteurs de nœuds

Pour uniformiser les vecteurs de nœuds, un vecteur commun de nœuds \hat{U} , donné par la réunion de l'ensemble des nœuds des différentes coupes est tout d'abord déterminé :

$$\hat{U} = \bigcup_{c \in \{0, 1, \dots, C\}} U^c. \quad (4.6)$$

Les valeurs des nœuds sont triées par ordre croissant, ce qui nous donne :

$$\hat{U} = \{\hat{u}_0 \leq \hat{u}_1 \leq \dots \leq \hat{u}_M\}, \quad (4.7)$$

où M est le nombre total de nœuds.

Notons que :

$$\forall i \in \{0, 1, \dots, C\}, U^c \subset \hat{U}. \quad (4.8)$$

L'objectif est de redéfinir chaque courbe NURBS de section sur le vecteur de nœuds commun \hat{U} de façon à ce qu'elle reste identique à la courbe initiale. Cela est réalisé en insérant un à un les nœuds de \hat{U}

U^c et en ajoutant à chaque insertion un nouveau point de contrôle. Soient $\{P_k^c\}_{k=0}^{n_c}$ les points de contrôle initiaux, et $\{Q_k^c\}_{k=0}^{n_c+1}$ les nouveaux points de contrôle après insertion d'un nouveau nœud \bar{u} . L'identité entre la courbe initiale et la nouvelle s'exprime par la relation suivante :

$$\forall u \in [0, 1], \sum_{k=0}^{n_c} N_{k,p}(u) P_k^c = \sum_{k=0}^{n_c+1} N_{k,p}(u) Q_k^c. \quad (4.9)$$

La solution de ce système d'équations est donnée par [PT95] :

$$Q_k^c = \alpha_k P_k^c + (1 - \alpha_k) P_{k-1}^c, \quad (4.10)$$

où :

$$\alpha_k = \begin{cases} 1 & \text{si } k \leq i - p, \\ \frac{\bar{u} - u_k}{u_{k+p} - u_k} & \text{si } i - p + 1 \leq k \leq i, \\ 0 & \text{si } k \geq i + 1, \end{cases} \quad (4.11)$$

où i est l'indice correspondant à l'intervalle $[u_i, u_{i+1})$ où le nœud \bar{u} est inséré.

Observons que le nombre de nœuds de \hat{U} augmente rapidement avec le nombre de coupes considérées. Un sous-échantillonnage des nœuds est alors appliqué en gardant uniquement ceux satisfaisant la condition suivante :

$$\forall i \in \{0, 1, \dots, n\}, u_i \geq \frac{i}{n} \text{ et } u_{i-1} < \frac{i}{n}, \quad (4.12)$$

où n est le nombre de nœuds fixé pour toutes les coupes.

Cette condition vise à obtenir un échantillon de nœuds plus ou moins uniformément répartis sur $[0, 1]$. En effet, les nœuds étant calculés à partir des positions des points de contrôle, et ces derniers étant proches sur les coupes successives puisque provenant de l'intersection de volumes, le vecteur de nœuds global \hat{U} présente localement de fortes densités de nœuds (Figures 4.16(a) et 4.16(b)). Cette procédure de sous-échantillonnage permet de garder un seul échantillon par "agglomération" de nœuds (Figure 4.16(c)) en choisissant un critère en fonction de la distribution au lieu d'un sous-échantillonnage régulier en fonction du nombre des nœuds.

Les courbes NURBS obtenues pour la coupe de la Figure 4.12 sont présentées Figure 4.17(a) pour un vecteur commun de 752 nœuds et un vecteur sous-échantillonné à 37 nœuds (Figure 4.17(b)).

4.3.1.4 Interpolation des courbes de section

Pour calculer une surface NURBS interpolant les courbes de section, il reste à déterminer deux éléments : le vecteur de nœuds V dans la direction longitudinale et l'ensemble des points de contrôle de la surface.

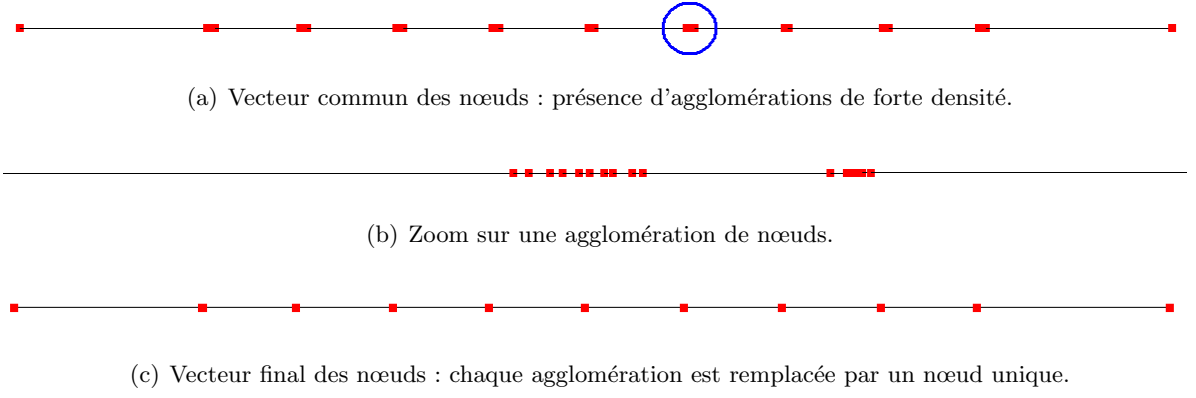


Figure 4.16 – Uniformisation du vecteur des nœuds.

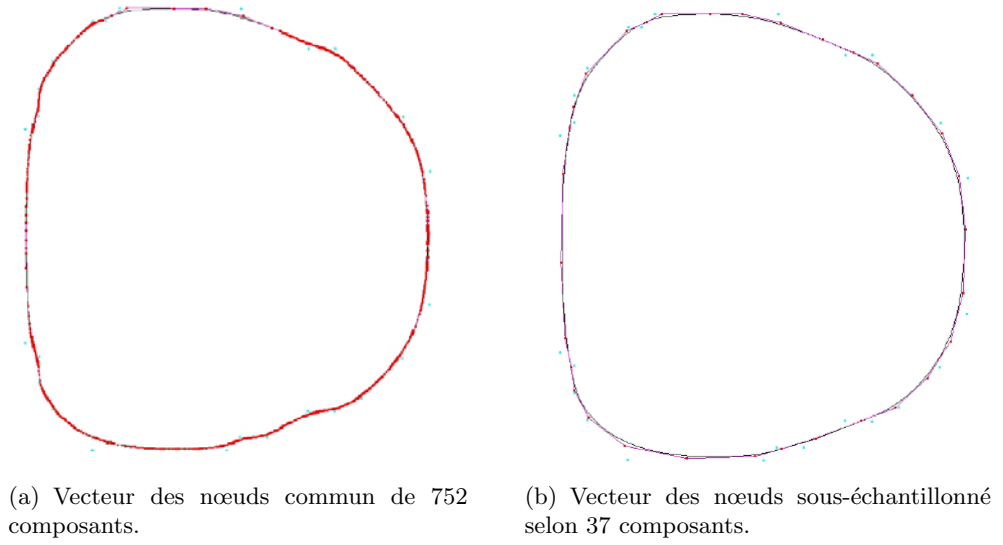


Figure 4.17 – Raffinement des nœuds.

Le vecteur de nœuds v est obtenu de façon similaire avec la construction du vecteur u (4.4). Il s'agit de calculer tout d'abord le vecteur $\{\bar{v}_0, \bar{v}_1, \dots, \bar{v}_C\}$:

$$\begin{cases} \bar{v}_0 = 0, \\ \bar{v}_c = \bar{v}_{c-1} + \frac{1}{n+1} \sum_{k=0}^n \frac{|P_k^c - P_k^{c-1}|}{d_k}, 1 < c \leq C, \end{cases} \quad (4.13)$$

où d_k représente la longueur totale de la polygline $P_k^0, P_k^1, \dots, P_k^C$, et $C+1$ le nombre des coupes.

Le vecteur de nœuds commun V est ensuite donné par :

$$\begin{cases} v_0 = \dots = v_q = 0, \\ v_{j+q} = \frac{1}{q} \sum_{i=j}^{j+q-1} \bar{v}_i, 1 \leq j \leq C-q, \\ v_{s-q} = \dots = v_s = 1. \end{cases} \quad (4.14)$$

où $s = C + q + 1$ et q est le degré de la surface NURBS dans la direction v .

Afin de définir totalement la surface NURBS interpolant l'ensemble des coupes, il reste à déterminer ses points de contrôle $\{Q_{i,j}\}_{i \in \{0, \dots, n\}, j \in \{0, \dots, C\}}$.

La condition d'interpolation de la coupe j correspondant à v_j s'écrit comme suit :

$$S(u, v_j) = C_j(u), \quad (4.15)$$

où

$$C_j(u) = \frac{\sum_{i=0}^n N_{i,p}(u) P_i^j}{\sum_{i=0}^n N_{i,p}(u)}, \quad (4.16)$$

et

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^C N_{i,p}(u) N_{j,q}(v) Q_{i,j}}{\sum_{i=0}^n \sum_{j=0}^C N_{i,p}(u) N_{j,q}(v)}. \quad (4.17)$$

Ainsi, pour v_j fixé, nous obtenons :

$$\begin{aligned} S(u, v_j) &= \frac{\sum_{i=0}^n \sum_{j=0}^C N_{i,p}(u) N_{j,q}(v_j) Q_{i,j}}{\sum_{i=0}^n \sum_{j=0}^C N_{i,p}(u) N_{j,q}(v_j)}, \\ &= \frac{\sum_{i=0}^n N_{i,p}(u) \left(\sum_{j=0}^C N_{j,q}(v_j) Q_{i,j} \right)}{\sum_{i=0}^n N_{i,p}(u) \left(\sum_{j=0}^C N_{j,q}(v_j) \right)}. \end{aligned} \quad (4.18)$$

Or

$$\sum_{j=0}^C N_{j,q}(v_j) = 1, \quad (4.19)$$

ce qui conduit au système d'équations suivant :

$$\forall i \in \{0, \dots, n\}, \quad \sum_{j=0}^C N_{j,q}(v_j) Q_{i,j} = P_i^j, \quad (4.20)$$

qui s'exprime sous la forme plus compacte :

$$\forall i \in \{0, \dots, n\}, \quad N Q_i = P_i, \quad (4.21)$$

où

$$P_i = \begin{pmatrix} P_{i,0} \\ P_{i,1} \\ \vdots \\ P_{i,C} \end{pmatrix}, \quad Q_i = \begin{pmatrix} Q_{i,0} \\ Q_{i,1} \\ \vdots \\ Q_{i,C} \end{pmatrix}, \quad (4.22)$$

et

$$N = \begin{pmatrix} 1 & 0 & \dots & 0 \\ N_{0,p}(v_1) & N_{0,p}(v_1) & \dots & N_{n,p}(v_1) \\ N_{0,p}(v_2) & N_{0,p}(v_2) & \dots & N_{n,p}(v_2) \\ \vdots & \vdots & \dots & \vdots \\ N_{0,p}(v_{n-1}) & N_{0,p}(v_{n-1}) & \dots & N_{n,p}(v_{n-1}) \\ 0 & 0 & \dots & 1 \end{pmatrix}. \quad (4.23)$$

La résolution du système linéaire 4.21 est alors effectuée par calcul de la pseudo-inverse de la matrice N , réalisé à l'aide de l'algorithme de SVD (*Singular Value Decomposition*) [PTVF92]

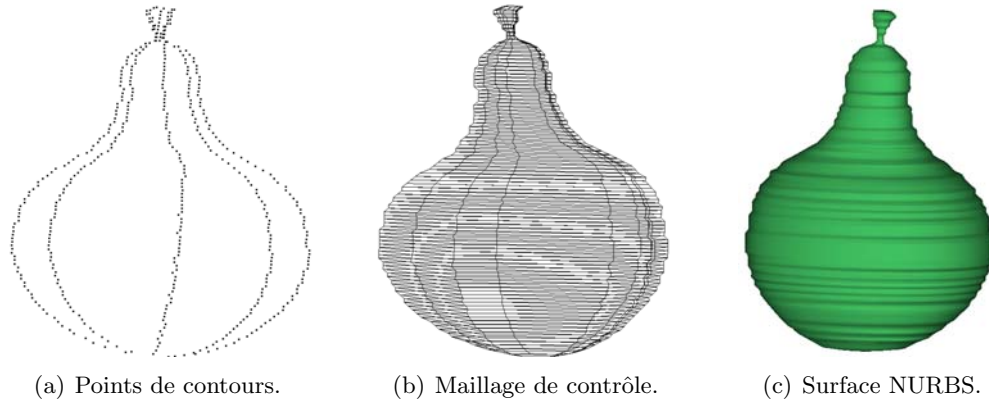


Figure 4.18 – Aspect en escalier : l'effet de la discrétisation volumique se répercute sur la positions des points de contrôle et par conséquent sur la surface.

4.3.1.5 Résultats expérimentaux

La figure 4.18 présente un exemple de reconstruction pour une forme simple correspondant à un modèle de poire. La surface reconstruite présente un effet d'escalier. Cela est dû à la résolution des images d'entrée (des vues 2D), qui est dans ce cas de 100×100 pixels. Bien sûr, une résolution plus fine permettrait de diminuer cet effet, mais les ressources de mémoire nécessaires pour la construction de l'enveloppe visuelle et le temps de calcul correspondant augmenteraient considérablement.

Afin de s'affranchir de cet inconvénient tout en gardant une résolution basse des vues 2D, nous avons adopté une procédure de filtrage passe-bas selon la direction longitudinale. Pour une coupe donnée, nous moyennons la valeur des coordonnées de ses points de contrôle avec celles de leurs correspondants sur les coupes précédente et suivante, en leur attribuant les poids de 0.5, 0.25 et 0.25 respectivement :

$$P_{i,j} = 0.25 \cdot P_i^{j-1} + 0.5 \cdot P_i^j + 0.25 \cdot P_i^{j+1}. \quad (4.24)$$

Notons que ce filtrage peut être appliqué de manière itérative pour obtenir une surface suffisamment lisse. Le nombre d'itérations nécessaire dépend de la résolution des images utilisées en entrée. La Figure 4.19 illustre l'effet du filtrage sur les surfaces obtenues.

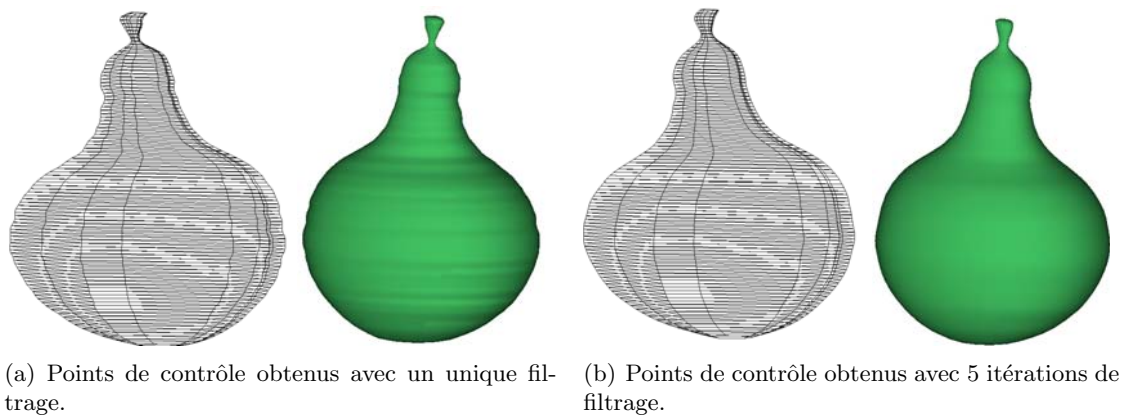
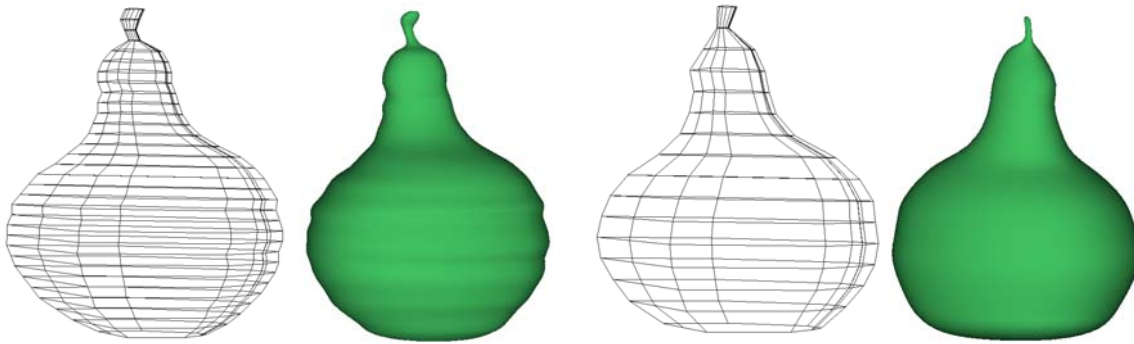


Figure 4.19 – Atténuation de l'effet d'escalier par filtrage passe-bas.

Le même effet de lissage peut être obtenu en appliquant un sous-échantillonnage des coupes, toujours selon la direction longitudinale. Des exemples de surfaces obtenues après sous-échantillonnage sont présentés Figure 4.20. La Figure 4.21 illustre des résultats pour

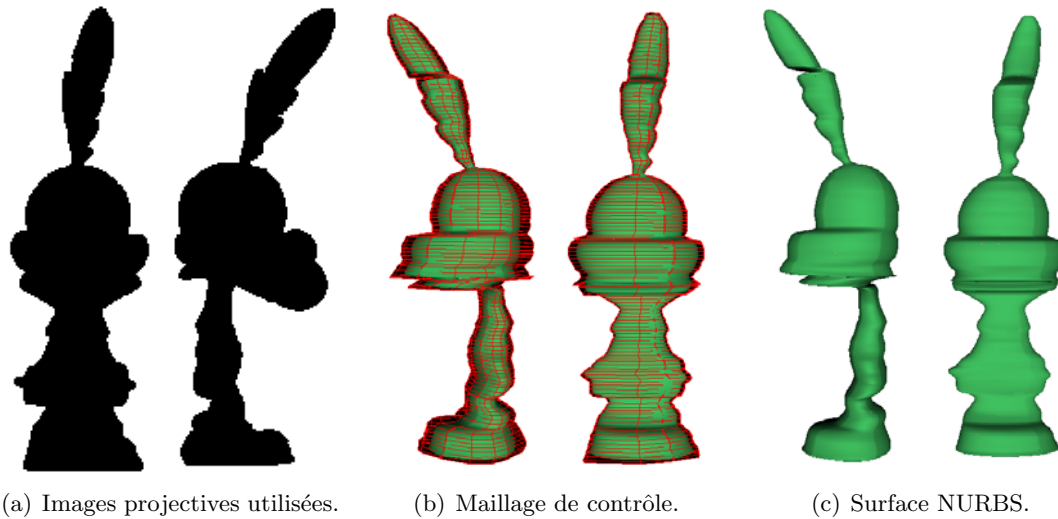
différents pas de sous-échantillonnage pour le modèle du personnage Titeuf, reconstruit sans segmentation préalable.



(a) Points de contrôle obtenus sans filtrage, avec un sous-échantillonnage de $pas = 3$.

(b) Points de contrôle obtenus sans filtrage, avec un sous-échantillonnage de $pas = 6$.

Figure 4.20 – Atténuation de l'effet d'escalier par sous-échantillonnage des coupes.



(a) Images projectives utilisées.

(b) Maillage de contrôle.

(c) Surface NURBS.

Figure 4.21 – Approche par interpolation : modèle NURBS de Titeuf obtenu pour un nombre de points de contrôle égal à 10 par coupe, et un degré égal à 3 dans les deux directions.

Toutefois, la procédure de sous-échantillonnage permet en même temps de diminuer le nombre de points de contrôles utilisés et d'obtenir ainsi une surface plus facile à manipuler et à déformer. Afin de lisser la surface tout en allégeant sa structure, nous optons pour une solution hybride qui consiste à appliquer à la fois un filtrage et un sous-échantillonnage.

Notons que, malgré les étapes de filtrage et de sous-échantillonnage, les surfaces obtenues présentent encore un faible aspect ridé. En effet, la procédure de raffinement des vecteurs des nœuds représente un inconvénient de l'approche de reconstruction par interpolation, car elle traite de manière globale les nœuds, sans tenir compte, pour une coupe donnée, de leur appartenance à la courbe de départ ou non. Un décalage entre les nœuds (et donc entre les points de contrôle des coupes successives) est introduit par cette étape de l'algorithme de reconstruction. Ce décalage se traduit par l'apparition d'un aspect en zig-zag du maillage de contrôle, qui fait que la surface se vrille légèrement, donnant naissance aux rides observées.

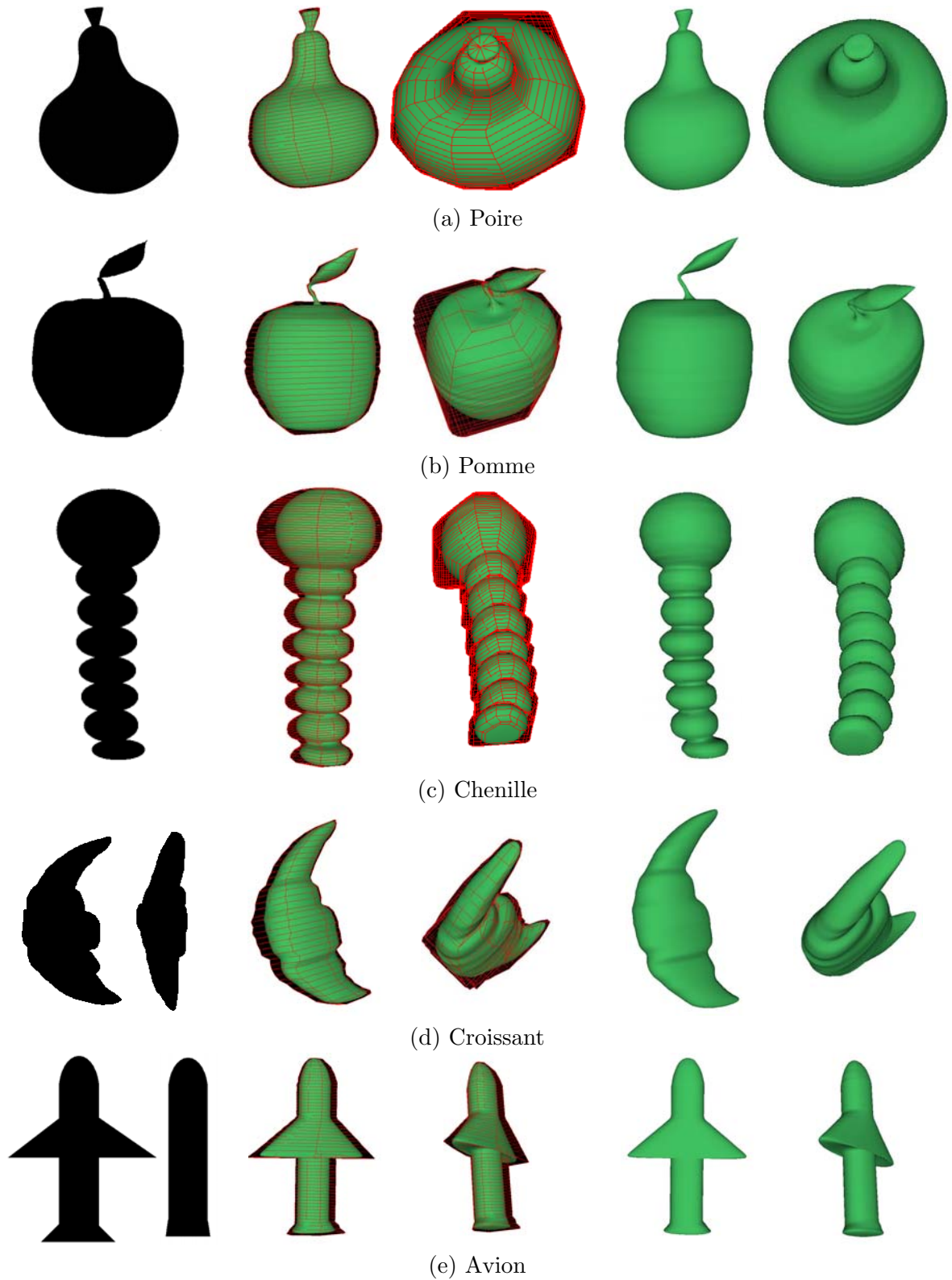


Figure 4.22 – Reconstruction de surfaces NURBS par interpolation pour différents objets dessinés manuellement, obtenues avec $NP = 10$ points de contrôle par coupe. Les surfaces NURBS sont de degrés $p = 3$ et $q = 3$. Les images d'entrée sont les mêmes pour les angles de vue 0 et 90 dans le cas de la poire, la pomme et la chenille.

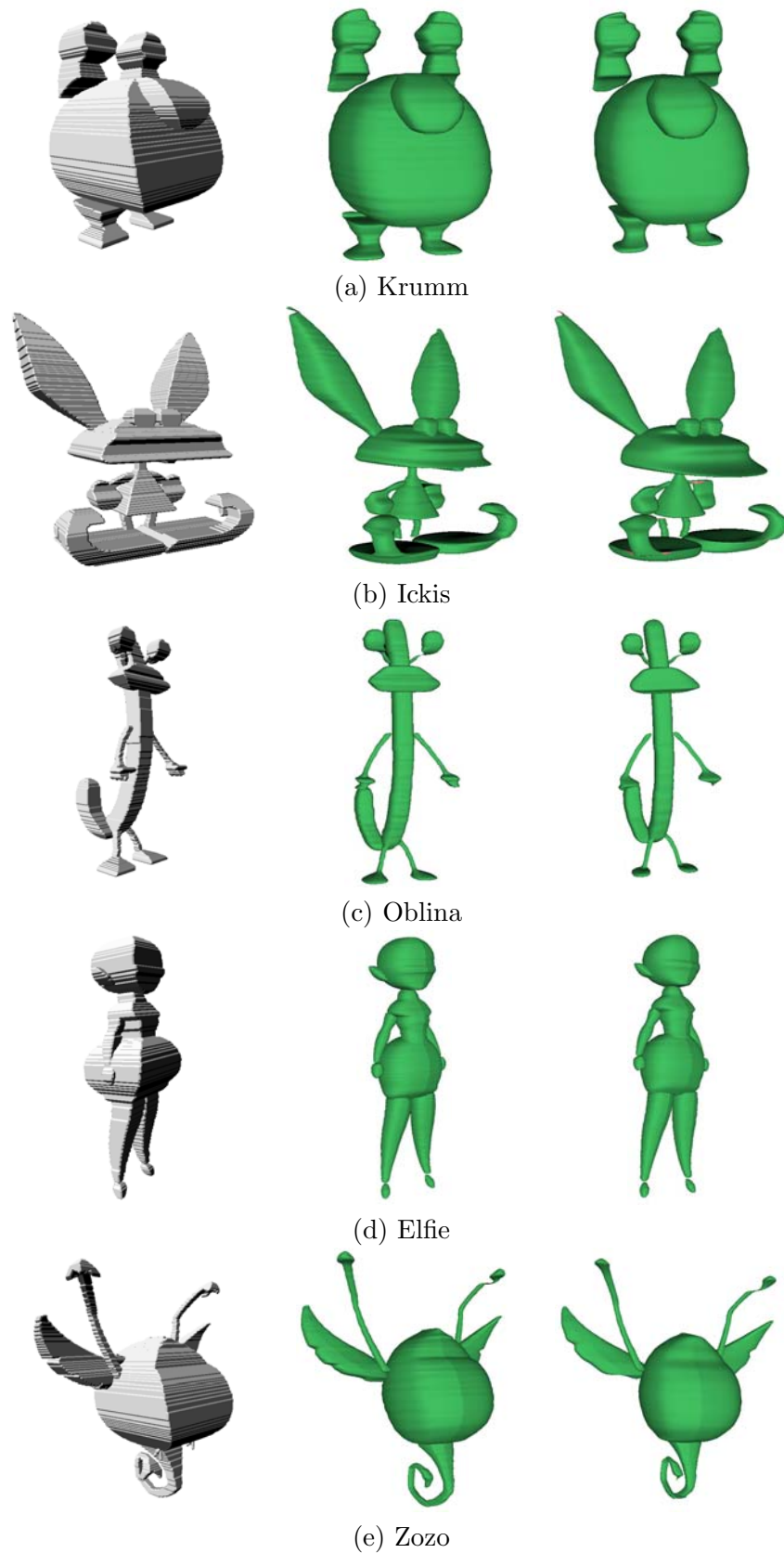


Figure 4.23 – Reconstruction par interpolation NURBS : Résultats (de gauche à droite : Visual Hull, Surfaces NURBS reconstruites avec 10 points de contrôle par coupe et des pas de sous-échantillonnage de 2 et 4).

4.3.2 Approche par approximation

Afin d'éviter l'effet "ridé" précédemment décrit, nous optons pour une approche de reconstruction différente, où les points de contrôle de la surface sont déduits directement lors de la procédure d'intersection de volumes.

L'intersection de volumes est réalisée en calculant pour chaque coupe le polygone d'intersection généré par les vues de projection (*cf.* Section 4.2.1). Ces polygones définissent une coupe exacte du volume englobant, contrairement à l'approche volumique qui passe par un échantillonnage préalable en voxels. Les sommets des polygones ainsi obtenus sont alors utilisés comme points de contrôle de courbe NURBS sur la coupe considérée.

Toutefois, la procédure d'intersection de volumes ne garantit pas le même nombre de sommets pour toutes les coupes. Une première étape d'harmonisation du nombre de sommets par coupe est alors nécessaire pour pouvoir définir la surface NURBS.

4.3.2.1 Harmonisation du nombre de sommets

Le principe de cette méthode d'harmonisation consiste à préserver les sommets initiaux des polygones dérivés de l'intersection de volumes. Leur conservation combinée à l'utilisation d'une méthode de subdivision par sectionnement des coins permet de garantir la tangence avec les polygones d'intersection et d'assurer ainsi la cohérence avec les vues de projection.

Soit C une coupe dont le polygone généré par intersection de volumes est composé de N_C sommets. Le nombre maximal (ou de référence) de sommets par coupe obtenue par intersection de volume est de $N_{ref} = 2 * \text{Nombre de vues}$. La procédure d'harmonisation doit donc être appliquée pour toute coupe dont le nombre de sommets N_C est inférieur au nombre de sommets de référence :

$$N_C < N_{ref}. \quad (4.25)$$

Afin de pouvoir insérer le nombre de sommets manquants sur la coupe C , nous nous référons à une coupe voisine ayant le nombre de sommets de référence N_{ref} . Les formes des polygones étant proches entre coupes successives, nous procédons à une recherche descendante de la coupe C_{ref} la plus proche ayant N_{ref} sommets.

L'algorithme d'insertion des sommets comporte les étapes suivantes :

1. Calculer les coordonnées relatives de chaque sommet par rapport au polygone auquel il appartient. Soient $C_0, C_1, \dots, C_{N_C-1}$ les sommets constituant un polygone. La coordonnée d'un sommet C_i est alors définie par son abscisse curviligne $u(C_i)$, exprimée comme :

$$u(C_i) = \frac{\sum_{j=1}^i \|C_j - C_{j-1}\|}{\sum_{j=1}^n \|C_j - C_{j-1}\|}. \quad (4.26)$$

2. A chaque sommet C_i^{ref} du polygone de référence C_{ref} est attribué le sommet le plus proche (en terme de distance entre les abscisses curvilignes) du polygone C (Figure 4.24). Soit $F_{C_{ref} \rightarrow C}$ l'application attribuant à chaque sommet $C_i^{ref} \in C_{ref}$ un correspondant sur C :

$$F_{C_{ref} \rightarrow C} (C_i^{ref}) = \arg \min_{C_j \in C} \left| u(C_j) - u(C_i^{ref}) \right|. \quad (4.27)$$

et $G_{C_{ref}}$ l'application attribuant au sommet C_i^{ref} la distance entre son abscisse curviligne et celle de son image sur C par $F_{C_{ref} \rightarrow C}$:

$$G_{C_{ref}} (C_i^{ref}) = \left| u(F_{C_{ref} \rightarrow C} (C_i^{ref})) - u(C_i^{ref}) \right| \quad (4.28)$$

Soulignons que la correspondance est ici fondée sur la distance entre coordonnées curvilignes et non pas entre les coordonnées 3D des sommets. Pour éviter tout décalage entre les sommets des coupes mises en correspondance, il est nécessaire d'ordonner les sommets selon l'ordre des vues intersectées.

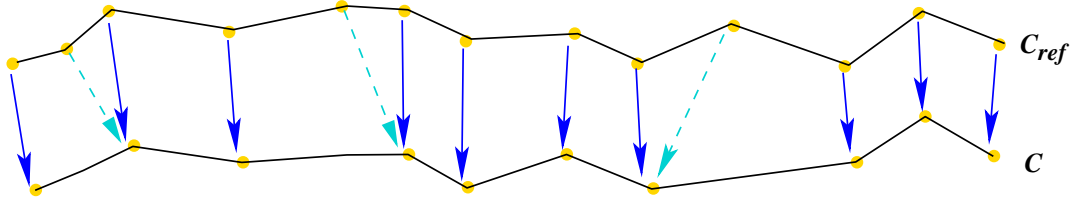


Figure 4.24 – Correspondances entre les sommets de deux courbes polygonales à l'aide de leurs coordonnées relatives.

3. Les sommets de C_{ref} sont alors ordonnés en fonction de la valeur qui leur est attribuée par $G_{C_{ref}}$. A la fin de cette procédure, on détermine les $N_{ref} - N_C$ sommets de C_{ref} dont les images par $G_{C_{ref}}$ sont les plus grandes (représentées Figure 4.24 par les flèches pointillées vertes). Un correspondant de chacun de ces sommets, de même coordonnée relative, est alors inséré sur le polygone C .

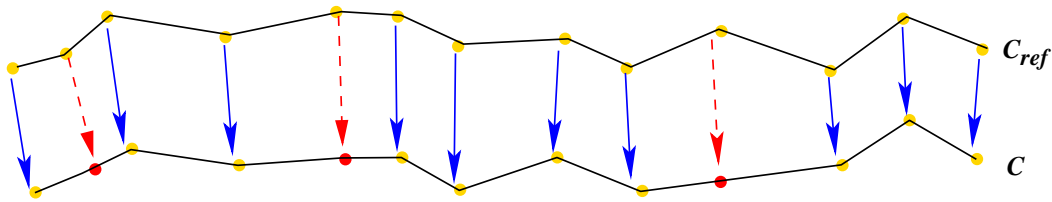


Figure 4.25 – Insertion de nouveaux sommets à l'aide des coordonnées relatives.

Notons que la forme des polygones de départ est ainsi préservée à l'identique.

Une fois le nombre de sommets par polygone harmonisé sur l'ensemble des coupes, les surfaces NURBS sont construites comme décrit dans la section suivante.

4.3.2.2 Définition des points de contrôle

Les polygones obtenus par intersection de volumes représentent une approximation du modèle 3D que l'on cherche à reconstruire. Cette approximation a la particularité de constituer un volume englobant de notre modèle. Cette propriété est vérifiée par le polygone de contrôle d'une courbe NURBS lorsque celui-ci est convexe. Il s'agit de la propriété bien connue de l'enveloppe convexe.

Afin de définir la surface NURBS de manière à ce qu'elle soit contenue dans le volume d'intersection décrit par les polygones de section, nous exploitons cette propriété. Cependant, l'utilisation directe des sommets des polygones d'intersection comme points de contrôle de la surface donnerait une surface trop proche du polygone de contrôle. Le nombre de vues généralement disponibles dans notre cas étant très limité, cette surface aurait par conséquent un aspect peu réaliste.

Nous procédons alors à une première étape de subdivision des polygones d'intersection en appliquant l'algorithme de Chaikin avec le masque $(\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$ (Figure 4.26). Les sommets des polygones ainsi obtenus constituent alors les points de contrôle de la surface NURBS.

La subdivision est justifiée par une hypothèse relative à l'application considérée qui consiste à supposer que les dessins fournis représentent généralement des formes arrondies. Cette hypothèse est satisfaite lorsqu'il s'agit de personnages dits organiques.

Par ailleurs, l'intersection de volumes est fondée sur le principe d'un volume maximal, alors que nous cherchons à reconstruire un volume représentatif du personnage.

Le principe de la procédure d'intersection de volumes découle en fait d'une inversion du processus de projection. Les contours de projection utilisés sont en réalité les *contours apparents* du modèle du personnage imaginé par le dessinateur. Le volume d'intersection est

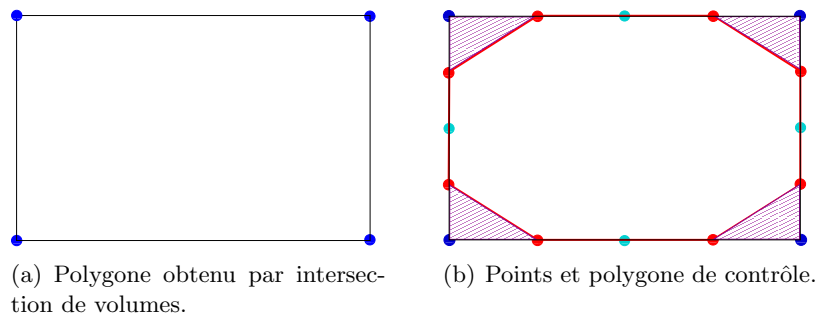


Figure 4.26 – Choix des points de contrôle par subdivision.

donc tangent à la surface de l'objet et les arêtes des polygones obtenus représentent le lieu géométrique des points de tangence.

Afin d'assurer la tangence de la surface NURBS reconstruite avec le volume d'intersection, des points de contrôle supplémentaires sont insérés au milieu de chaque arête du polygone de contrôle obtenu par subdivision de Chaikin provenant d'une arête du polygone initial, donné par l'intersection de volume. Ces points sont illustrés Figure 4.26 en couleur verte.

Enfin, les vecteurs de nœuds sont calculés en moyennant les abscisses curvilignes sur l'ensemble des coupes, de la même manière que pour la méthode de reconstruction par interpolation (cf. Section 4.3.1, Paragraphe 4.3.1.2).

4.3.2.3 Résultats expérimentaux

La Figure 4.27 présente quelques résultats de surfaces NURBS reconstruites à l'aide de la méthode d'approximation pour le personnage Elfie.

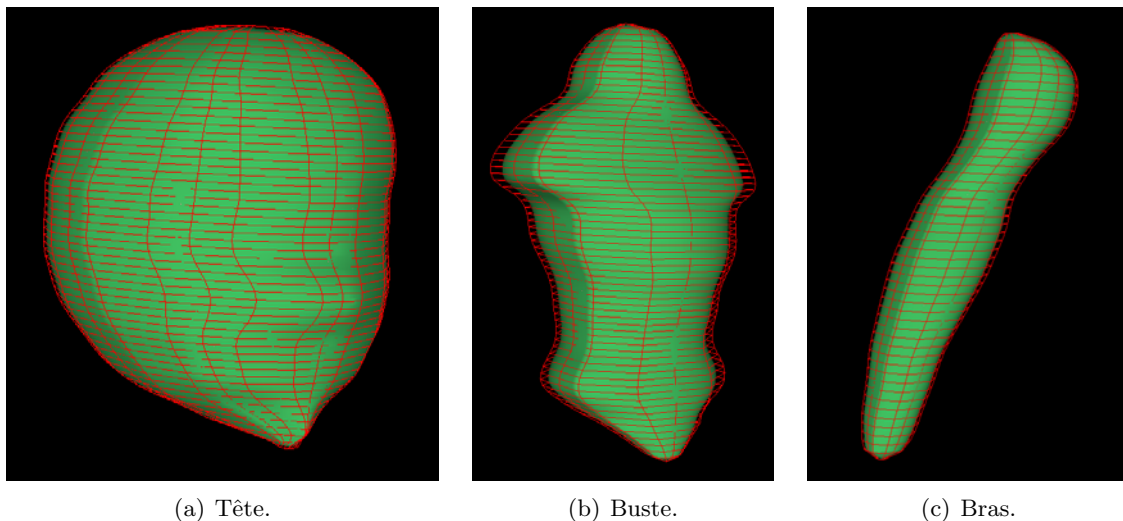
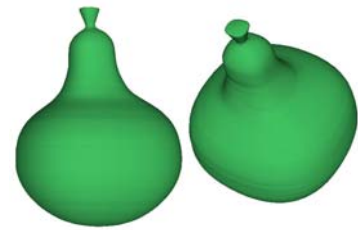
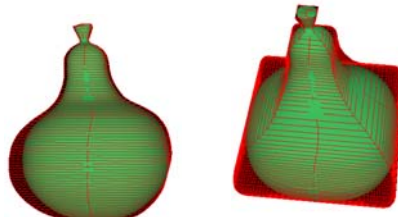
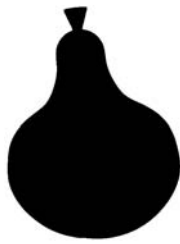
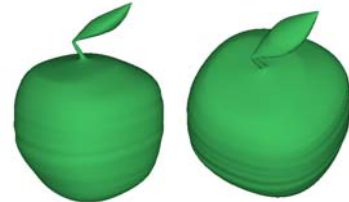
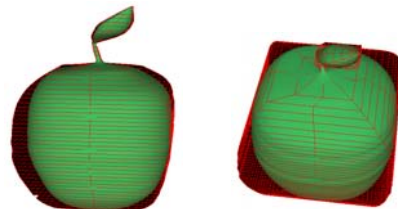


Figure 4.27 – Surfaces NURBS reconstruites par approximation : tête, buste, et bras du personnage Elfie.

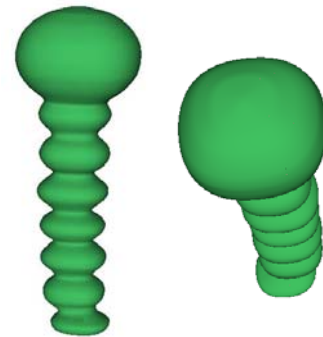
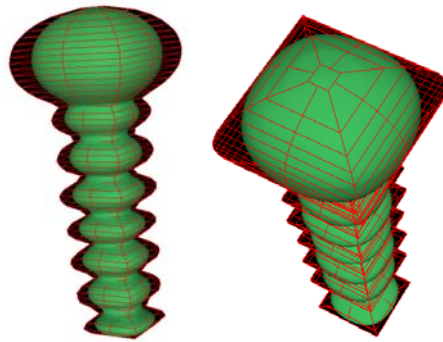
Un avantage de cette méthode réside dans le fait que le paramètre du masque de subdivision appliqué peuvent être modifiés pour modéliser des objets plus ou moins arrondis. Ainsi, la même approche de reconstruction peut-elle être appliquée pour construire des formes dites organiques, aussi bien que des objets plus rigides tels que des meubles ou des bâtiments, en modifiant le masque de subdivision, exprimé par le vecteur $(\alpha, 1 - \alpha, \alpha)$ avec $\alpha \in [0, 1]$. Ainsi, plus le paramètre α tend vers 0, plus la forme de l'objet reconstruit est carrée.



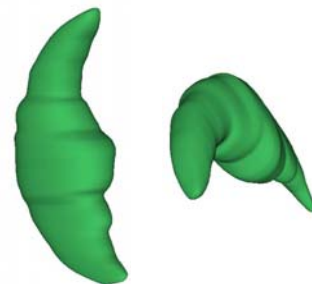
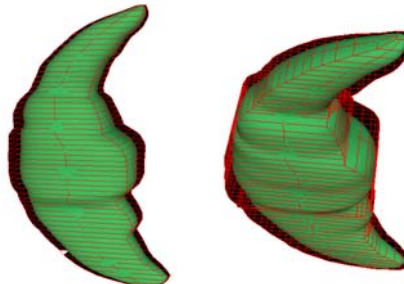
(a) Poire.



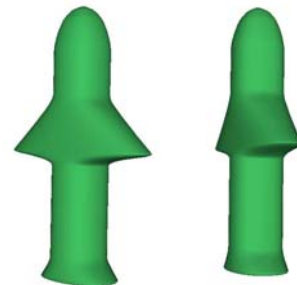
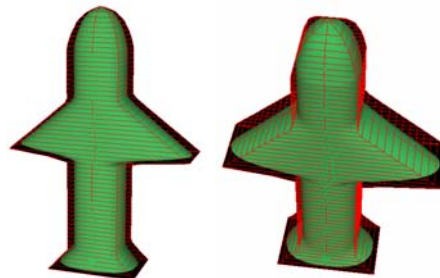
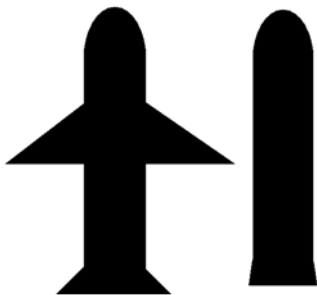
(b) Pomme.



(c) Chenille.

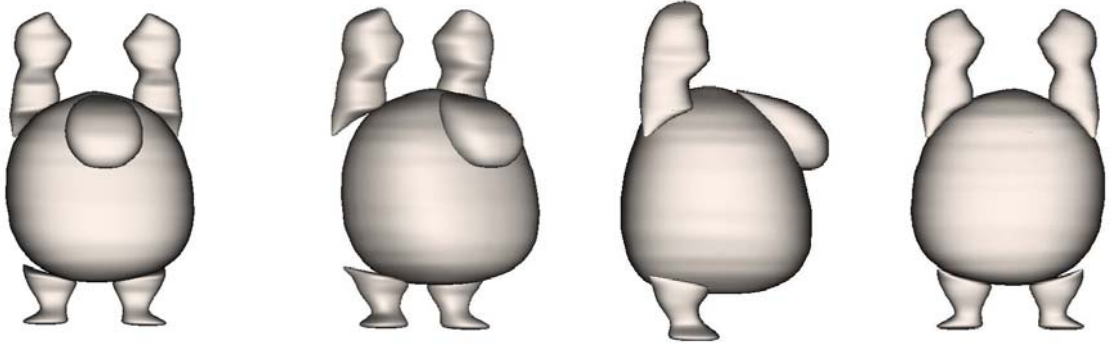


(d) Croissant.

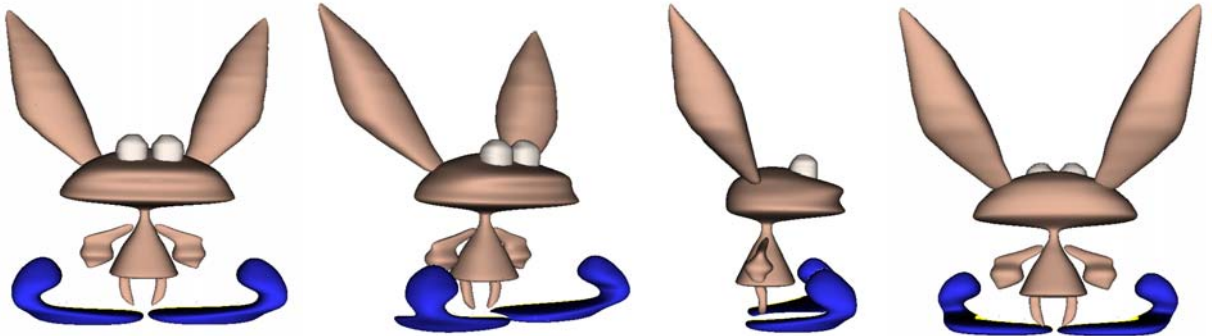


(e) Avion.

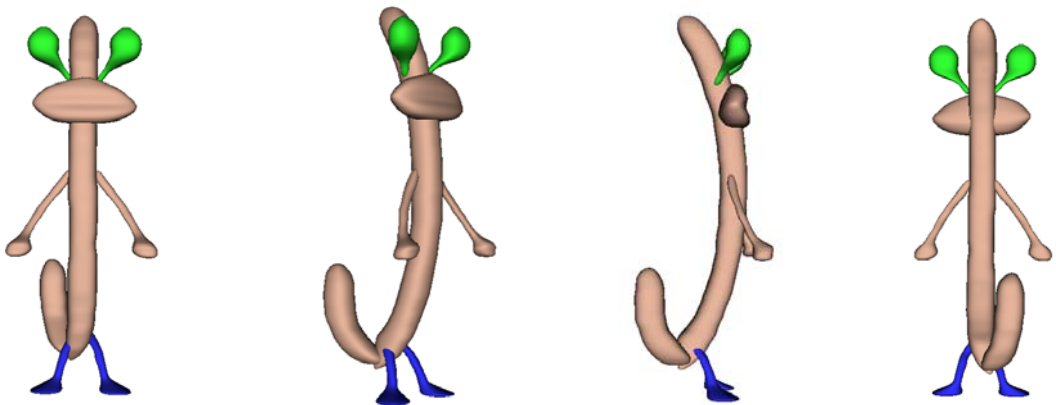
Figure 4.28 – Approche par approximation : modèle NURBS de la poire, de la chenille et de l'avion obtenus (surface de degré 3 dans les deux directions). Les images d'entrée sont les mêmes pour les angles de vue 0 et 90 dans le cas de la poire, la pomme et la chenille.



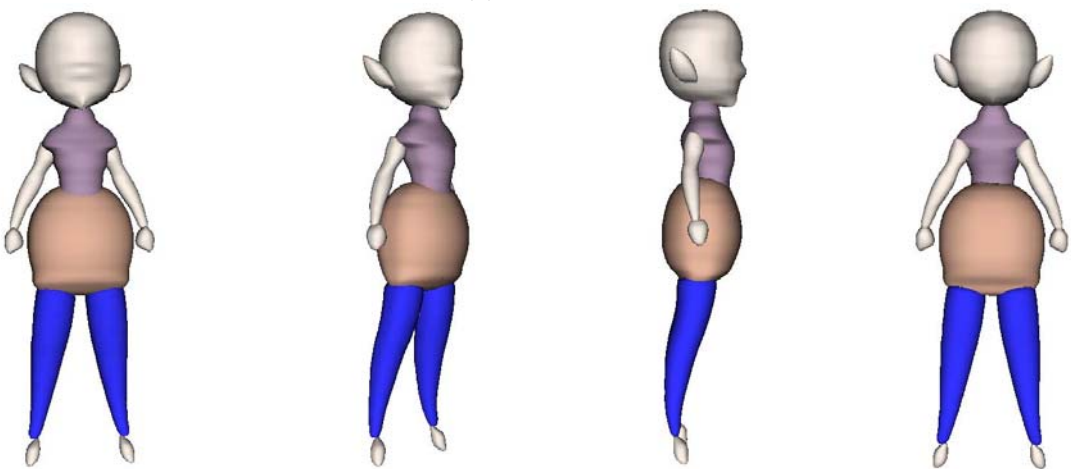
(a) Krumm.



(b) Ickis.



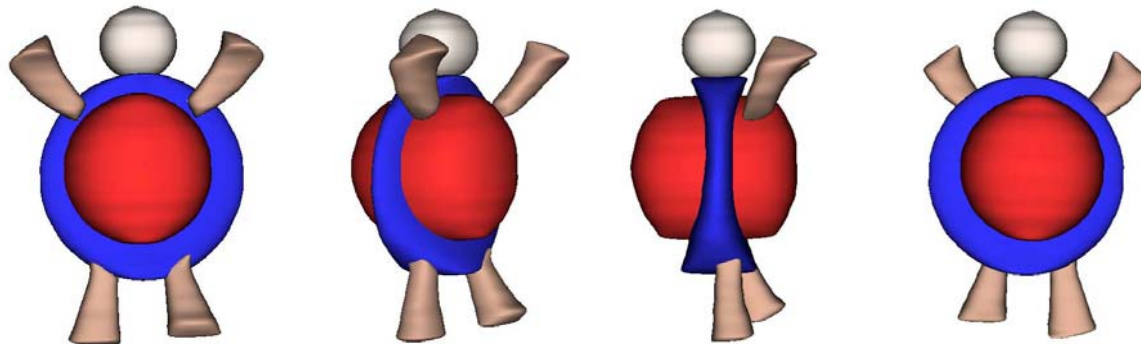
(c) Oblina.



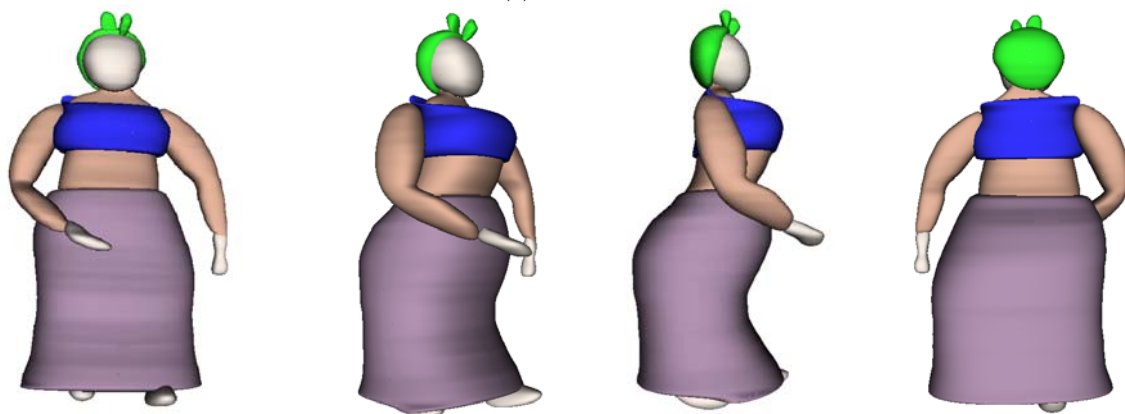
(d) Elfie.



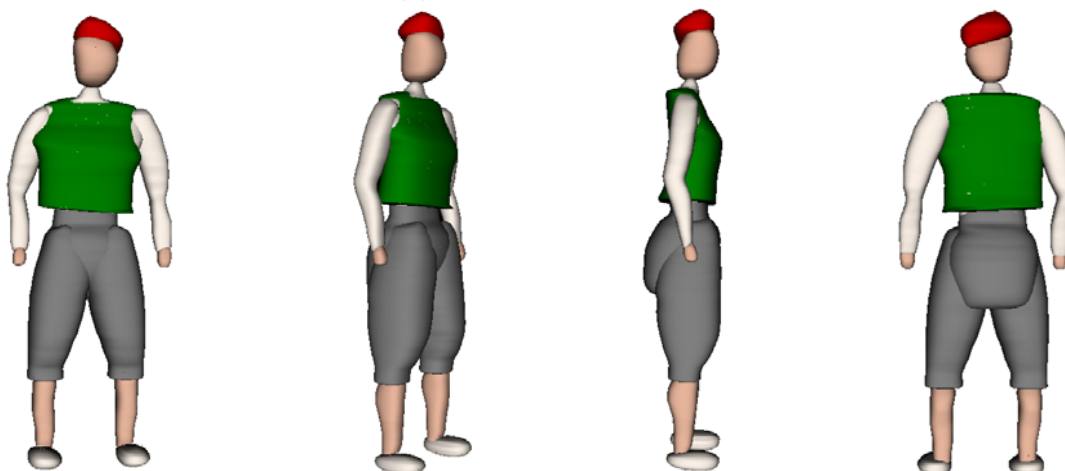
(e) Zozo.



(f) Mahras.



(g) Femme Tunisienne.



(h) Homme Tunisien.

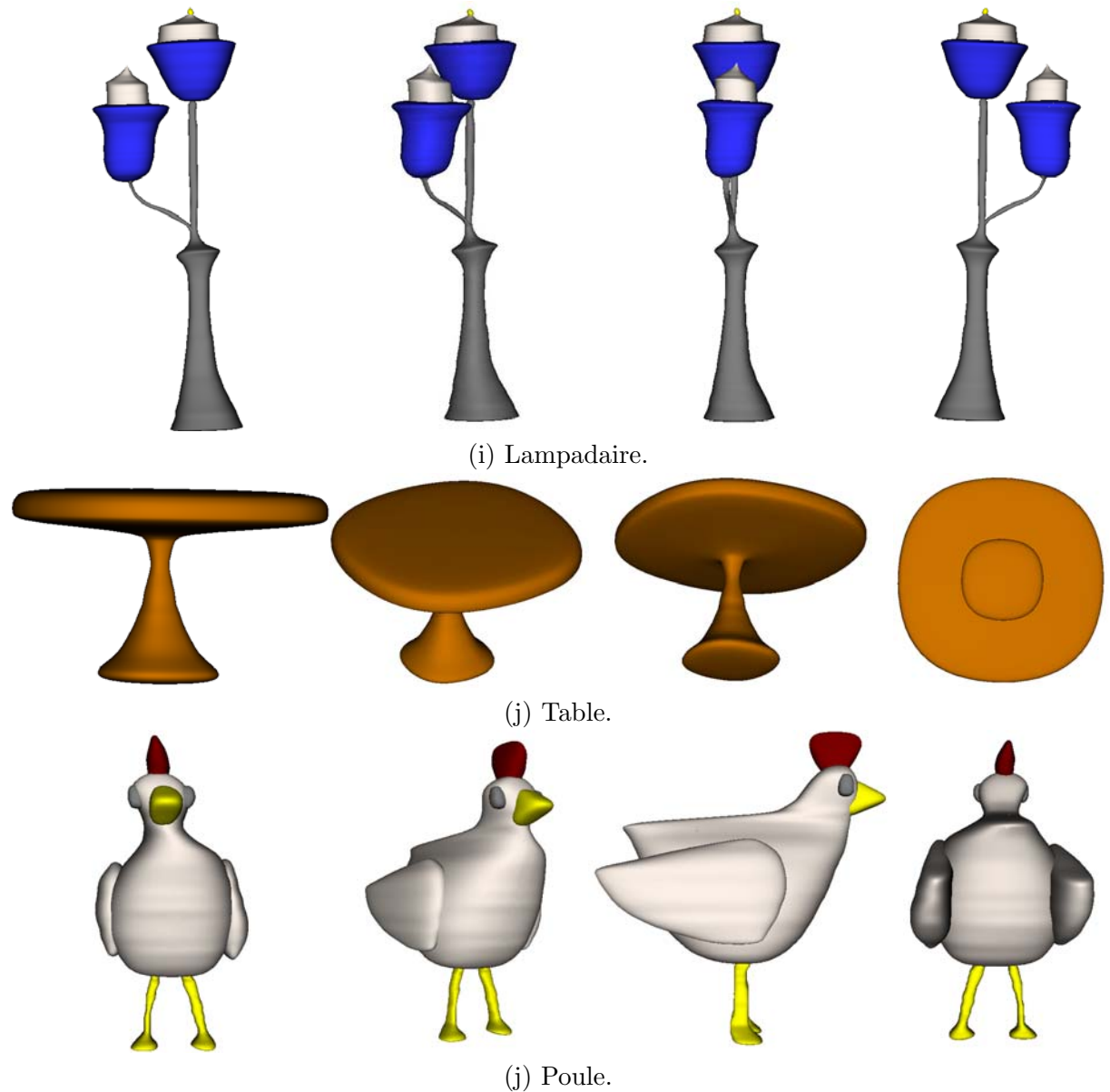


Figure 4.29 – Reconstructions par approximation NURBS : surfaces obtenues avec 3 niveaux de subdivision de Chaikin.

4.4 Génération d'un maillage *seamless*

Dans le cadre de notre application, l'objectif est d'obtenir des modèles pouvant être animés conformément à la norme MPEG-4. Les personnages doivent alors être modélisés par des maillages dits sans couture ou *seamless*, composés d'une unique composante connexe et représentant une surface fermée, sans bords. Les différentes parties reconstruites, modélisées par des surfaces NURBS doivent donc être converties et assemblées en un unique maillage *seamless*.

Une première solution consisterait à appliquer les méthodes de composition (*blending*) de surfaces NURBS, puis à trianguler les surfaces composées ainsi obtenues. Cela mettrait en œuvre des courbes de *trimming*, interconnectées par des surfaces de *blending*.

Toutefois, en raison de la double définition de ces courbes dans des domaines paramétriques différents, les maillages obtenus par ces approches ne sont pas garantis *seamless* [FK90], [KS95], [RHD89] (cf. paragraphe 2.3.5). Il s'agit du problème bien connu du *cracking* [BK97], [WW98], [BNK02].

Une solution à ce problème est proposée dans [KM95] où le maillage *seamless* est généré directement à partir des surfaces *trimmed* NURBS. Pour ce faire, la procédure doit établir une correspondance entre les courbes de trimming. Cela nécessite la résolution du problème complexe d'*inversion de point* [MH03] qui consiste à déterminer les coordonnées paramétriques du point d'une courbe ou d'une surface paramétrique qui est le plus proche d'un point donné de l'espace. La grande complexité de calcul constitue une limitation majeure pour cette procédure de triangulation par mise en correspondance des courbes, d'autant plus que ce calcul doit être appliqué aussi pour les courbes de *trimming*.

Pour cette raison, d'autres approches comme celles décrites dans [BGK03], [KBK02] proposent d'inverser l'ordre des opérations : les composantes des surfaces sont tout d'abord indépendamment triangulées, puis une étape de "couture" (*sewing*) est appliquée pour coller les maillages en un seul modèle *seamless*.

L'approche que nous proposons s'inscrit dans cette filiation. Dans un premier temps, les connexions à effectuer sont spécifiées et déterminées. Chaque élément du modèle est d'abord converti en un maillage en employant une procédure d'échantillonnage régulier exploitant la définition paramétrique des NURBS. Les courbes de trimming sont ensuite déterminées de manière automatique pour chaque connexion. Enfin, une re-triangulation est effectuée au voisinage des courbes de *trimming*, qui inclut, le cas échéant, la génération d'une surface de *blending*.

4.4.1 Critères d'assemblage des parties du modèle

Afin de connecter les différents éléments reconstruits, il est tout d'abord nécessaire de déterminer les relations de connexion à effectuer. Cette étape exploite deux types différents d'information :

- les relations d'adjacence entre les parties segmentées des images 2D du *turn-around*, représentées par une *matrice d'adjacence*. Cette information est automatiquement déterminée à partir des vues 2D ;
- le *squelette d'animation MPEG-4* donnant les relations hiérarchiques entre les différentes parties du modèle reconstruit. Élément clef pour l'animation, le squelette doit être spécifié par l'utilisateur.

La matrice d'adjacence est construite à partir des images 2D du *turnaround* dans un sens maximal : deux objets sont considérés comme connectés si et seulement si leurs images sont connexes dans l'ensemble des vues disponibles.

Afin de ne pas effectuer des connexions erronées à cause des occlusions inévitables entre différentes parties du corps (les bras et la jupe du personnage Elfie par exemple se superposent sur l'ensemble des vues de la Figure 4.30), cette information de connexité est combinée à celle fournie par le squelette de l'objet. Ainsi, si une connexion est présente dans la matrice d'adjacencesans correspondre à une relation hiérarchique parent-enfant sur le squelette, les deux parties restent indépendantes.

Notons que le squelette MPEG-4, créé uniquement à des fins d'animation, ne spécifie pas nécessairement toutes les connexions existantes entre les éléments segmentés. Des détails comme les yeux, le nez ou les oreilles peuvent ne pas figurer dans la hiérarchie du squelette s'ils ne sont pas animés. Les connexions sont alors établies en fonction de relations d'adjacence dérivées des images 2D uniquement, et le critère relatif au squelette est ignoré.

4.4.2 Connexion hiérarchique

Les modélisateurs 3D créent généralement leurs personnages en procédant aussi par parties, puis en assemblant les différents éléments soit manuellement soit à l'aide d'outils de *blending* disponibles dans certains logiciels [MAY], [MAX]. La procédure consiste à dessiner les courbes

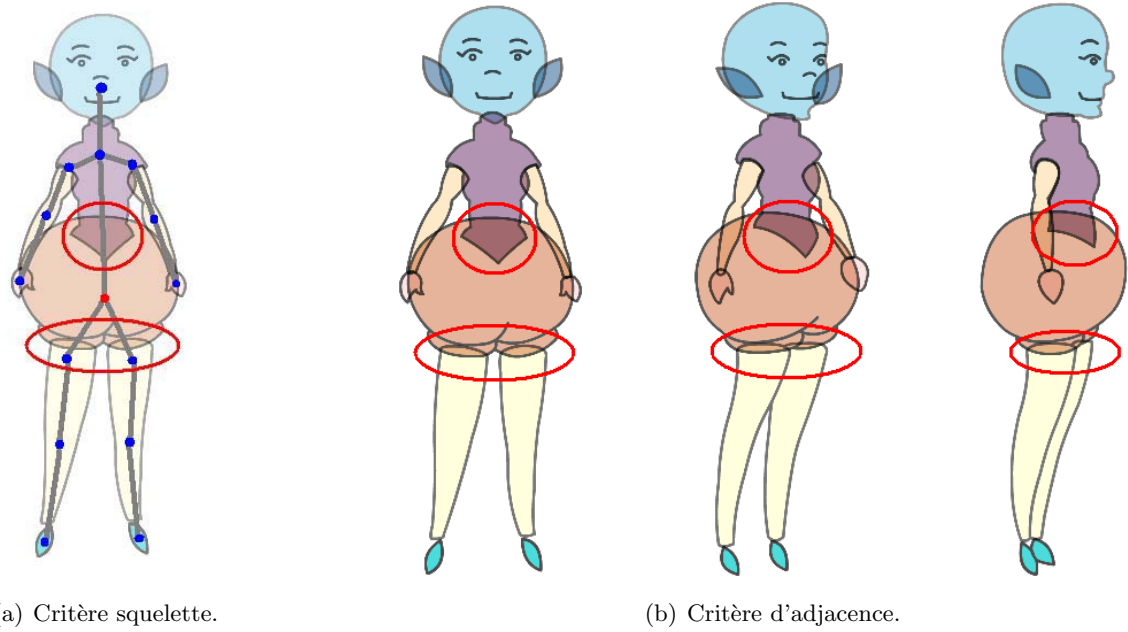


Figure 4.30 – Critère d'assemblage des parties.

de *trimming* sur la surface enfant (par exemple, pour assembler un bras à un torse, la courbe est dessinée sur le bras), puis à définir une direction selon laquelle la partie enfant est déplacée pour atteindre son parent hiérarchique.

Nous nous sommes inspirés de cette procédure pour développer une première approche exploitant les relations hiérarchiques du squelette MPEG-4, qui traite les deux parties à assembler différemment. En effet, l'élément dit *partie enfant* est "extrapolé" afin d'atteindre la *partie parent*.

Les parties reconstruites étant positionnées dans l'espace conformément aux dessins 2D fournis, nous ne pouvons envisager de déplacer la partie enfant comme c'est le cas dans la procédure manuelle 3D, car cela altérerait le modèle 3D final. Nous procédons donc à la création d'une surface de *blending* générée à partir d'un contour de *trimming* calculé de manière automatique.

La procédure de connexion est constituée des étapes suivantes :

1. On associe à la surface enfant un *pôle enfant* (Figure 4.31) défini par son intersection avec le segment de squelette *A* reliant les surfaces parent et enfant.

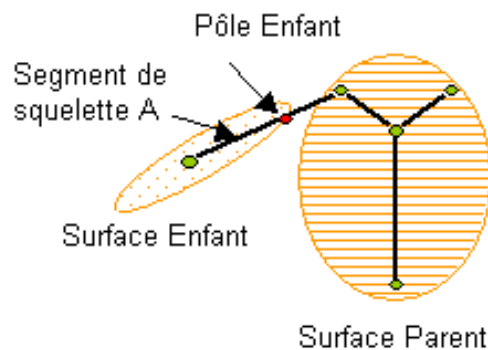


Figure 4.31 – Choix des surfaces Parent et Enfant.

2. En partant de ce pôle, un processus de propagation est réalisé de manière à déterminer un ensemble de triangles adjacents $T = \{t_i\}_i$ vérifiant la condition suivante :

$$\forall t_i \in T, \text{angle}(N_i, n_A) \leq \theta_{max}, \quad (4.29)$$

où n_A est l'orientation du segment de squelette A , N_i est le vecteur normal au plan du triangle t_i , et θ_{max} est un seuil angulaire fixé de manière empirique à 45° .

Notons par C_i le front de propagation défini comme l'ensemble des sommets de bord du patch de surface détecté T (Figure 4.32). Notons que la courbe constituée par ce front est généralement non lisse et présente souvent un aspect en dents de scie.

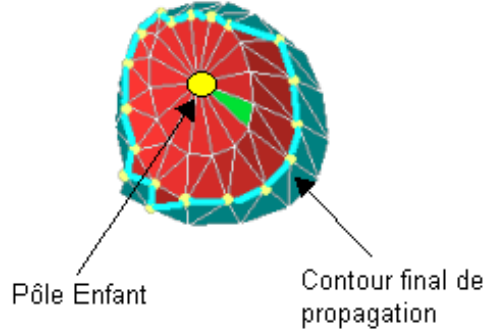


Figure 4.32 – Processus de propagation à partir du pôle enfant.

Soient G le centre de gravité des C_i , et $N_i^{contour}$ les vecteur normaux aux triangles (G, C_i, C_{i+1}) . Afin d'obtenir une courbe acceptable, nous définissons un plan de découpe P passant par le centre de gravité G et dont la normale N^P est donnée par la moyenne des vecteurs $N_i^{contour}$.

3. L'intersection du plan P ainsi défini avec le maillage donne naissance à un contour planaire Γ correspondant à l'ensemble des points I_i d'intersection des arêtes du maillage initial avec le plan P . La surface est ensuite remaillée en intégrant les points I_i et en redéfinissant la connexité au voisinage du contour Γ . Enfin, la partie du maillage intérieure au contour Γ (*i.e.* la partie incluant le pôle) est éliminée.
4. Le contour planaire Γ est projeté sur la surface parent selon la direction du segment de squelette A , définissant ainsi un contour géodésique Γ' sur la surface parent, constitué des points projetés I'_i . Le maillage de la partie parent est alors remaillé de la même manière que la surface enfant dans l'étape précédente, et les triangles intérieurs au contour Γ' sont éliminés.
5. La surface de connexion est enfin définie par l'ensemble des triangles de la forme (I_i, I'_i, I'_{i+1}) et (I_i, I'_{i+1}, I_{i+1}) .

Notons que cette procédure ne peut être appliquée aux parties dont la connexion ne correspond pas à une relation hiérarchique sur le squelette puisque la direction n_A n'est pas définie. Pour gérer cette configuration, un critère de distance est appliqué. Il s'agit de déterminer les deux sommets les plus proches au sens de la distance euclidienne sur les parties parent et enfant. La droite passant par ces deux points remplace alors le segment de droite A , et est utilisée pour définir le critère de propagation ainsi que la direction de projection du contour de découpage.

La Figure 4.33 présente quelques résultats d'assemblage pour des parties du personnage "Elfie". Notons l'aspect disproportionné et peu cohérent avec la forme des parties à connecter des surfaces de *blending* obtenues. Celles-ci étant des éléments cylindriques, elles restent en

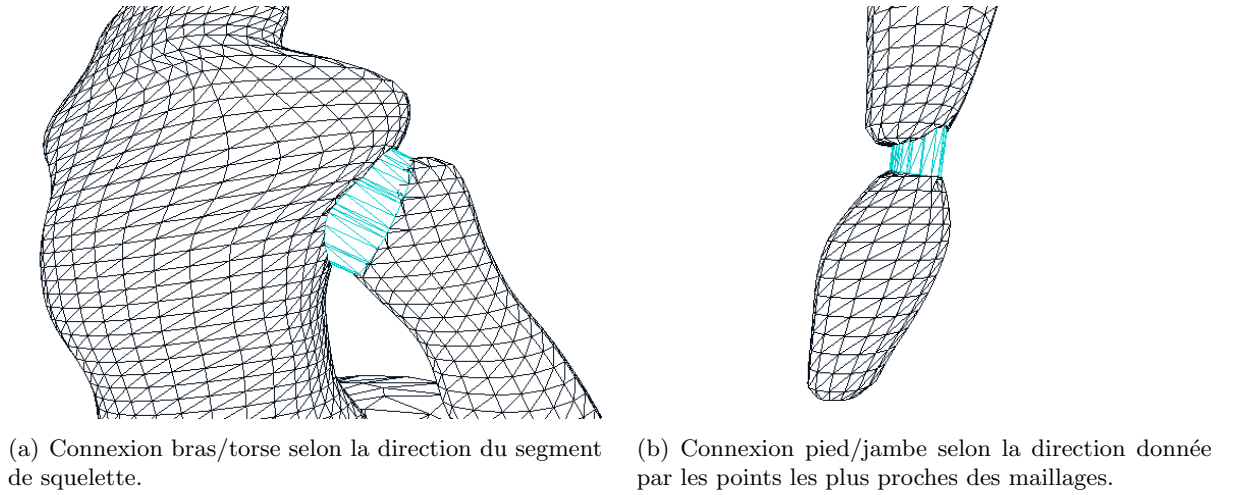


Figure 4.33 – Connexion hiérarchique : quelques résultats obtenus pour le personnage Elfie.

effet trop élémentaires pour prendre en compte des contraintes de continuité de plan tangent et ne peuvent par conséquent conduire à des surfaces lisses. En outre, les erreurs de segmentation, de calibration, ainsi que le caractère artisanal des vues de projection qui ne correspondent pas à un modèle de projection exact peuvent augmenter les écarts entre les parties reconstruites. Cela diminue encore plus la qualité visuelle des assemblages.

Afin de s'affranchir de cet inconvénient, nous proposons une deuxième méthode, qui permet d'assurer une transition plus graduelle entre les parties à assembler. Il s'agit d'une méthode de connexion par *Rolling Ball*.

4.4.3 Connexion par *Rolling Ball*

Le *Rolling Ball* est une technique de *blending* développée initialement dans le contexte des surfaces paramétriques [RR84], [CJ89], qui permet à la fois de déterminer les courbes de *trimming* et la surface de *blending*. L'idée est de simuler le déplacement d'une sphère sur les deux surfaces à assembler. Les courbes de découpage correspondent alors aux courbes de contact de la sphère avec chacune des surfaces, et la surface de *blending* à l'enveloppe surfacique déterminée par le mouvement de la sphère.

Nous avons adapté cette méthode de *blending* par *Rolling Ball* au cas des surfaces maillées. Soient M_1 et M_2 deux maillages à assembler. La connexion par *rolling ball* (Figure 4.34) implique les étapes suivantes :

1. **Dilatation des maillages** : chaque partie est dilatée par un facteur R , correspondant au rayon du *rolling ball*. L'opérateur de dilatation est ici effectué en déplaçant chaque sommet $\vec{V}_i = (x_i, y_i, z_i)$ du maillage dans la direction de sa normale à la surface \vec{N}_i par le facteur R :

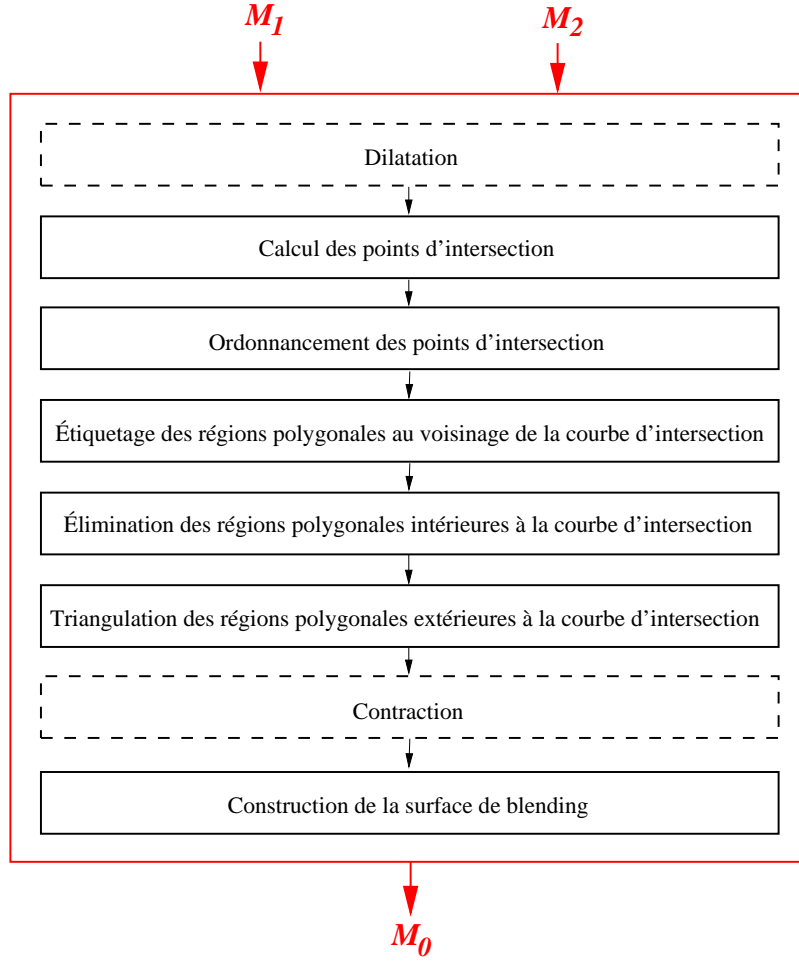
$$\vec{V}'_i = \vec{V}_i + R\vec{N}_i. \quad (4.30)$$

Les normales sont ici supposées pointer vers l'extérieur de la surface.

2. **Calcul de la courbe d'intersection** : la courbe d'intersection des deux parties dilatées est ensuite déterminée en appliquant la procédure décrite dans [LW04].

Les points d'intersection sont tout d'abord déterminés en calculant les intersections entre les triangles de M_1 avec les arêtes de M_2 et *vice versa*.

Une fois les points d'intersection calculés, ils sont ordonnés pour constituer la courbe d'intersection. Cette procédure est effectuée à l'aide d'un système d'étiquetage des triangles contenant chaque point d'intersection, dits *triangles de provenance*. Soit P un

Figure 4.34 – Algorithme de *blending* de maillages par *Rolling Ball*.

point de la courbe donné par l'intersection d'un triangle t_i^1 de M_1 avec une arête e_j^2 de M_2 . Notons par t_k^2 et t_l^2 les triangles de M_2 partageant l'arête e_j^2 (Figure 4.35). On note alors par *triangles de provenance* de P les triangle t_i^1 , t_k^2 et t_l^2 .

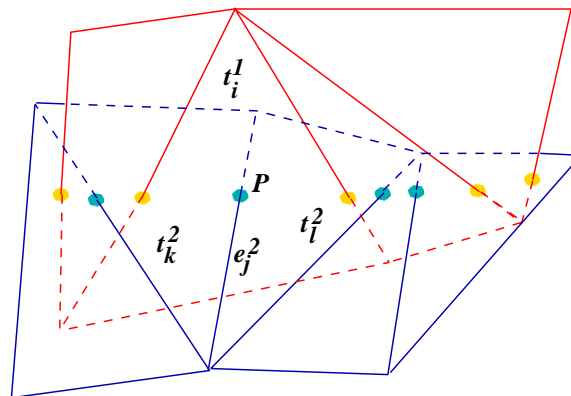


Figure 4.35 – Détermination de la courbe d'intersection.

Deux points sont considérés comme successifs sur la courbe d'intersection si et seulement si ils partagent un triangle sur chacun des maillages M_1 et M_2 . En effet, chaque arête de la courbe d'intersection est constituée de points appartenant à un même triangle sur

un des maillages.

L'ordonnancement des points de la chaîne se fait alors en examinant les triangles contenant chaque point d'intersection, puis en recherchant dans la liste un autre point partageant un triangle de provenance sur chacun des maillages M_1 et M_2 .

3. **Retriangulation au voisinage de la courbe d'intersection :** soient T^1 un triangle du maillage M^1 , et $e^2 = (v_1^2, v_2^2)$ une arête du maillage M^2 joignant les sommets v_1^2 et v_2^2 . Soient P le point d'intersection de e^2 et T^1 , et \vec{n}_{T^1} la normale au triangle T^1 . Les sommets v_i^2 sont respectivement étiquetés comme intérieurs ou extérieurs au maillage M^1 si $\langle \vec{Pv}_i^2, \vec{n} \rangle \leq 0$ ou $\langle \vec{Pv}_i^2, \vec{n} \rangle \geq 0$. Les sommets intérieurs pourront être éliminés lors de l'étape suivante.

Nous obtenons à la fin de cette procédure un ensemble de triangles traversés par la courbe d'intersection, appelés par la suite *triangles d'intersection*. La courbe d'intersection définit sur chacun de ces triangles une partition en régions polygonales. Les régions polygonales intérieures à la courbe d'intersection doivent alors être éliminées du maillage, alors que celles extérieures doivent être gardées et triangulées. Une première étape consiste alors à déterminer si les polygones sont intérieurs ou extérieurs. Pour réaliser cet étiquetage, nous utilisons une procédure fondée sur l'orientation (Figure 4.36).

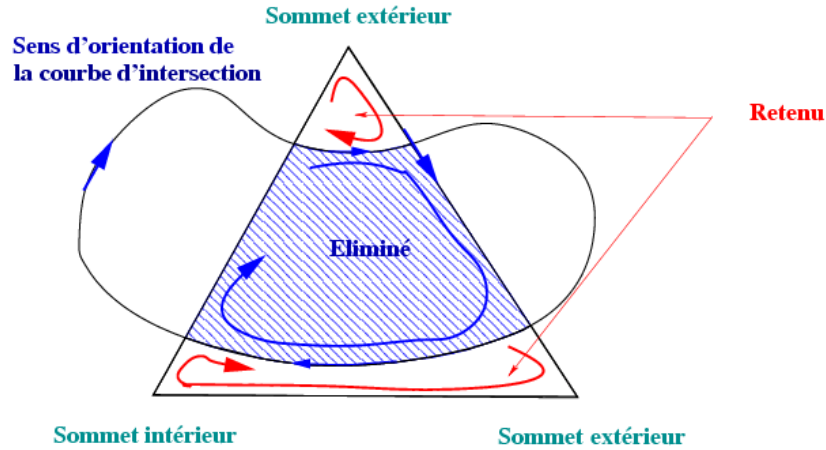


Figure 4.36 – Procédure de classification des polygones générés par l'intersection.

L'ordre des sommets du triangle initial induit un sens de parcours de la courbe d'intersection. On procède en parcourant chacun des polygones obtenus en suivant cet ordre de sommets. Cela donne une orientation à l'ensemble des régions polygonales. Ainsi, seules celles ayant le même sens d'orientation que la courbe de *trimming* sont-elles retenues.

4. Le voisinage des courbes de *blending*, constitué par l'ensemble des triangles traversés par ces courbes et étiquetés comme extérieurs, est alors triangulé (Figure 4.37).
5. Les deux surfaces sont ensuite contractées d'un facteur $(-R)$. La correspondance entre les points de la courbe d'intersection permet de construire la surface de *blending* directement. Notons par P_i un point de la courbe d'intersection entre les surfaces dilatées, et par P_i^1 et P_i^2 les points lui correspondant sur les surfaces contractées M^1 et M^2 . Il est possible de générer un ensemble de points intermédiaires P_i^k situés sur l'arc de cercle défini par P_i^1 et P_i^2 , ($P_i \bar{P}_i^1 = P_i \bar{P}_i^2 = R$), qui définissent un ensemble de courbes intermédiaires C^k . Nous obtenons une surface de *blending* construite en triangulant les bandes définies par chaque couple de courbes consécutives P_i^k . La surface de *blending* ainsi obtenue correspond à une surface générée par une balle roulant au contact des deux éléments assemblés.

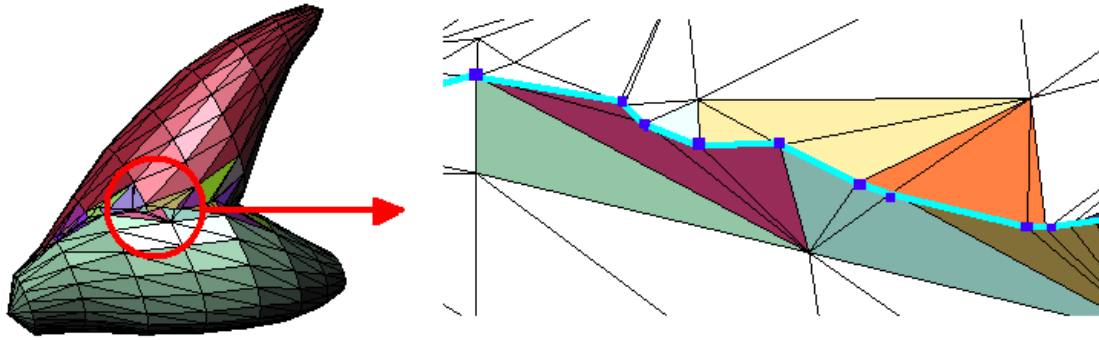


Figure 4.37 – Remaillage des triangles d'intersection.

Notons enfin que cette procédure peut être appliquée sans les étapes de dilatation et contraction (pour un rayon $R = 0$). Les surfaces sont alors simplement re-triangulées de manière à obtenir un maillage *seamless*, mais sans effet de lissage au niveau des jonctions entre les différentes parties du modèle. Cela est utile lorsque l'utilisateur ne veut pas introduire de surface de *blending*, mais uniquement assembler le modèle tel qu'il est reconstruit. La Figure 4.37 illustre un exemple de remaillage au voisinage de l'intersection de deux parties constituant la queue du personnage zozo. Une couleur est attribuée à chaque triangle d'intersection appartenant aux maillages initiaux. La Figure 4.38 illustre les résultats de remaillage obtenus pour le personnage Krumm, sans surfaces de *blending*.

Les Figures 4.39, 4.40 et 4.41 présentent quelques résultats obtenus pour le personnage Elfie pour différentes valeurs du paramètre R . D'une manière générale, cette approche conduit à des résultats visuellement meilleurs que ceux de la méthode de connexion hiérarchique, grâce à la surface de *blending* qui assure une transition plus graduelle entre les parties assemblées.

Notons de plus que le nombre de courbes intermédiaires, ainsi que le rayon du *rolling ball* peuvent être des paramètres modifiables par l'utilisateur en fonction de l'importance du lissage voulu d'une part, et des prérequis d'animation pour chaque articulation d'autre part. La valeur de ces paramètres est déterminante pour la qualité des surfaces de *blending*.

Les temps de calcul nécessaires pour obtenir un unique maillage *seamless* restent importants, notamment lorsqu'il s'agit de maillages en haute résolution avec des milliers de sommets par composante.

Par ailleurs, les surfaces de *blending* apparaissent encore peu naturelles, en raison de l'écart introduit entre les parties reconstruites. En effet, deux objets connexes sur des vues du *turnaround* peuvent être complètement séparés une fois reconstruits indépendamment l'un de l'autre. Ce phénomène est d'autant plus accentué que l'étape de modélisation NURBS implique des procédures de lissage qui ont un effet de rétrécissement. Nous pouvons observer des cas où les modèles volumiques se superposent, alors que les modèles NURBS finaux sont déconnectés. Lorsque les écarts entre parties deviennent trop importants, la seule solution est de déplacer manuellement les images 2D de projection avant d'effectuer la reconstruction, ce qui correspond à une calibration supplémentaire. La figure 4.42(b) illustre ce problème dans le cas des ailes du personnage Zozo. Les surfaces NURBS des ailes sont déconnectées du corps. La reconstruction obtenue à partir d'images translatées manuellement est présentée Figure 4.42(d). Toutefois, cette procédure alourdit le processus de reconstruction.

Enfin, un dernier inconvénient lié à la modélisation par surfaces NURBS est lié à l'uniformisation des vecteurs de nœuds qui constitue une contrainte importante pouvant conduire localement à de fortes densités de points de contrôle.

Afin de s'affranchir de tous ces inconvénients, nous proposons une méthode de reconstruction où l'assemblage est effectué à partir du modèle volumique. Cela permet d'éviter le

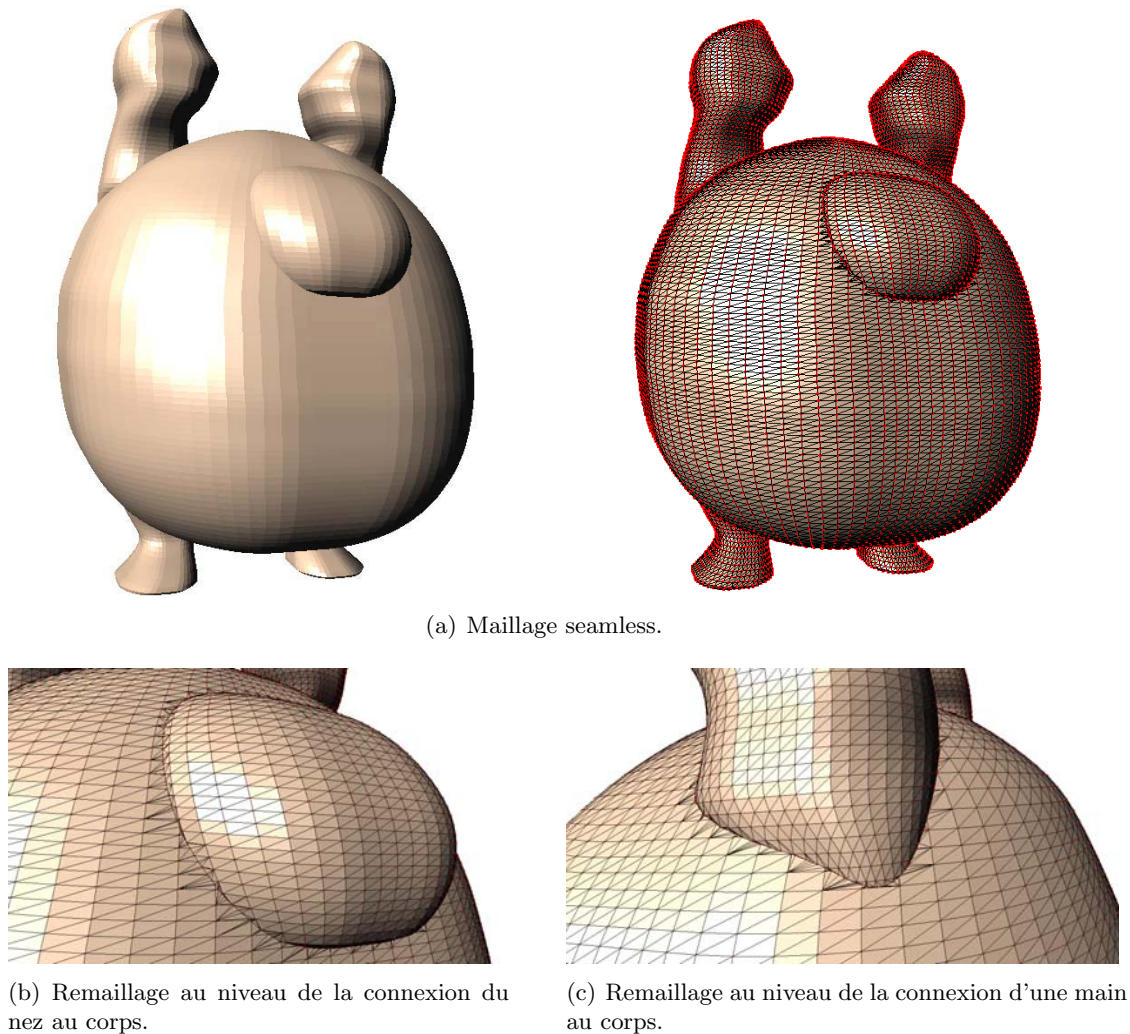


Figure 4.38 – Résultats obtenus pour le personnage Krumm.

problème lié aux surfaces de *blending*. La triangulation est ensuite réalisée directement, sans passer par l'étape de modélisation NURBS, pour générer un unique maillage *seamless*.

4.5 Reconstruction 2D/3D par maillages

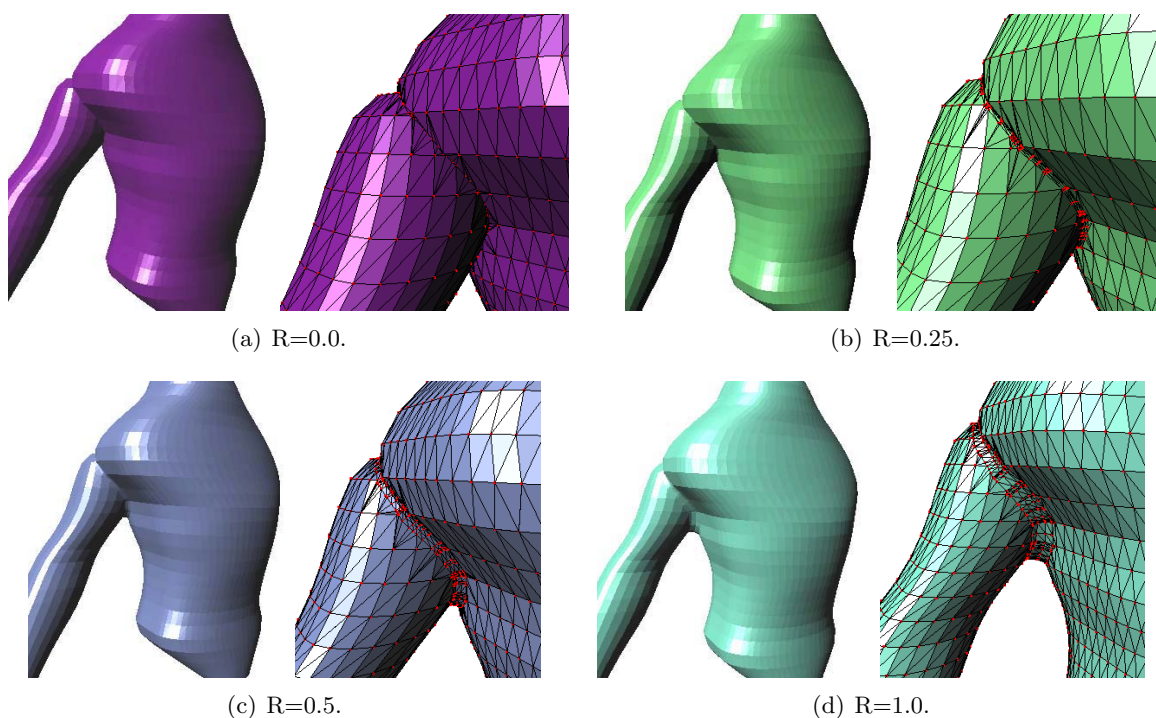
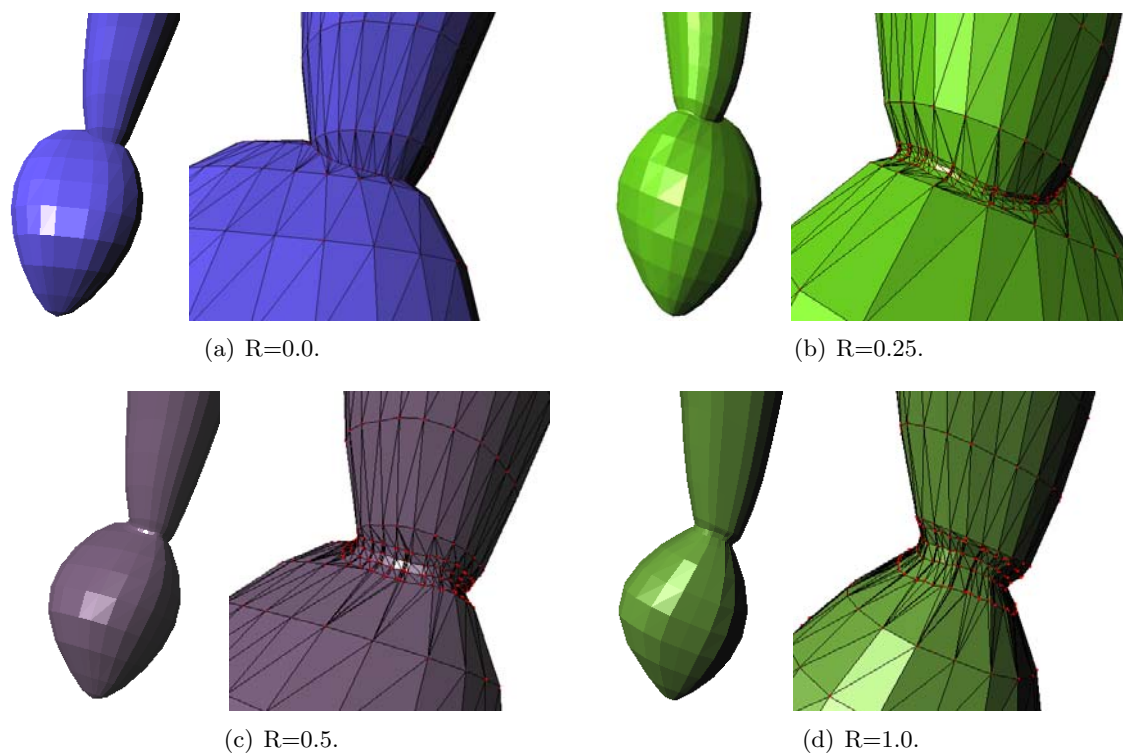
Dans cette approche, l'assemblage des différentes parties d'un même modèle est effectué directement à partir du modèle volumique obtenu via la méthode de reconstruction de l'enveloppe visuelle par coupes.

La triangulation du volume d'intersection exploite conjointement une procédure de subdivision et une triangulation de Delaunay.

La méthode consiste en trois étapes principales (Figure 4.43). Il s'agit de l'assemblage des différents modèles volumiques correspondant aux parties reconstruites, du lissage des coupes, et enfin de la triangulation de la surface.

4.5.1 Assemblage volumique

L'assemblage des parties à partir du modèle volumique a pour objectif de limiter les problèmes de déconnexion ou d'écartement des composantes de l'objet reconstruites séparément. Cela permet de conserver l'information de connexité des images du *turnaround* qui est préservée par le processus d'intersection de volumes. Toutefois, pour s'affranchir d'éventuels problèmes

Figure 4.39 – Connexion du torse et du bras du personnage Elfie par *Rolling Ball*.Figure 4.40 – Connexion du bras et de la main du personnage Elfie par *Rolling Ball*.

d'occlusion, nous maintenons ici la contrainte de connexion conditionnée par les relations hiérarchiques du squelette d'animation. (*cf.* Section 4.4.1).

Les modèles volumiques de chaque partie sont ici représentés par une série de polygones correspondant à des coupes horizontales. L'assemblage du modèle volumique est effectué donc coupe par coupe comme suit : pour chaque coupe et pour chaque ensemble de parties à assem-

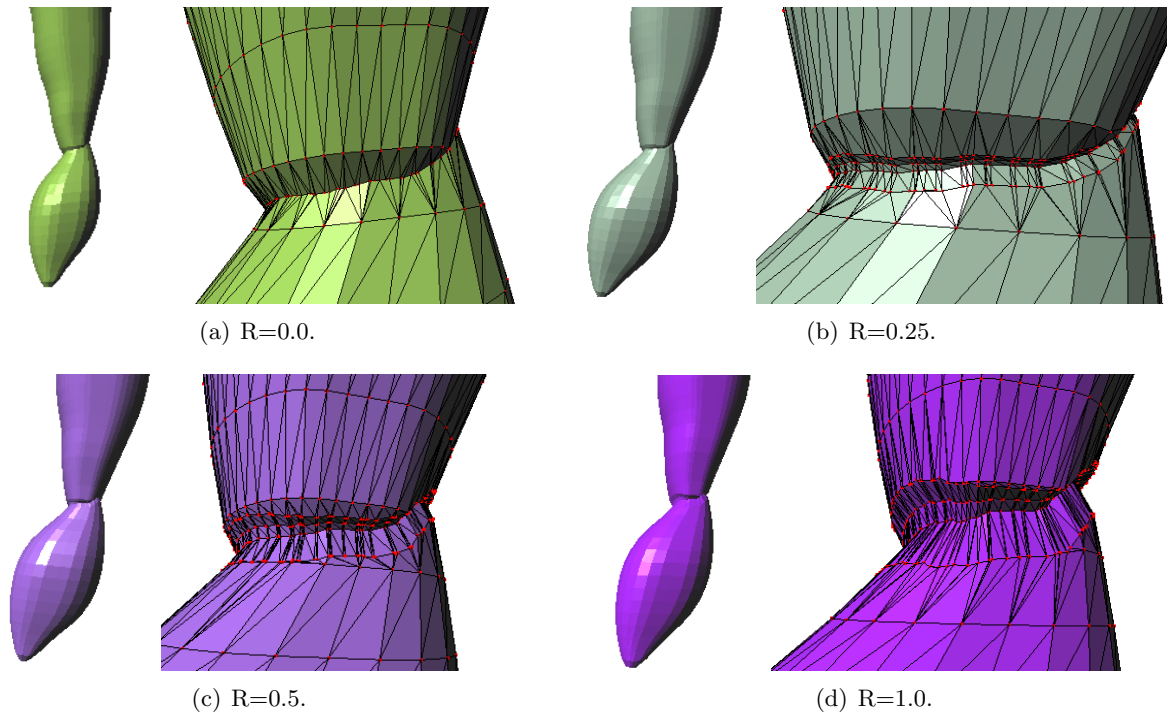


Figure 4.41 – Connexion de la jambe et du pied du personnage Elfie par *Rolling Ball*.

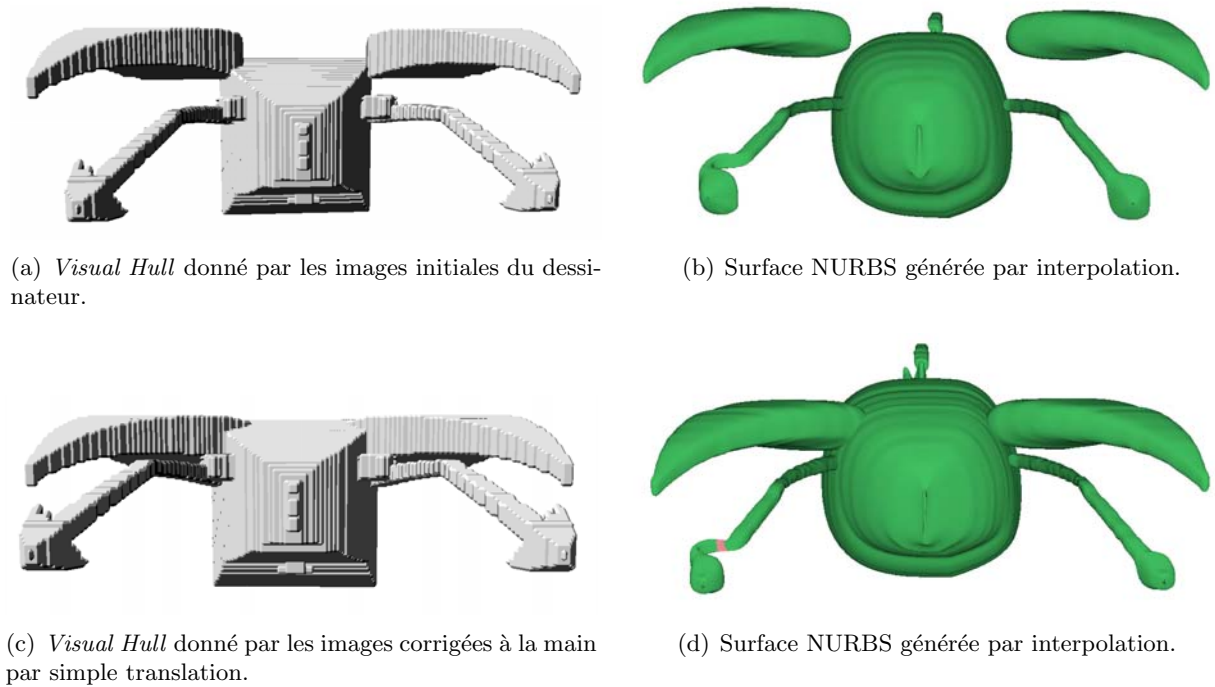


Figure 4.42 – Déconnexion de parties reconstruites en fonction des images d'entrée : bien que les VH des différentes parties sont connexes et conformes aux images projectives de départ (Figure 4.10), le lissage intrinsèque à la modélisation des coupes par des courbes NURBS introduit un vide entre les parties lissées indépendamment.

bler, nous générons le polygone global correspondant à la réunion des polygones représentant chaque partie (Figures 4.44(a) et 4.44(b)). Nous obtenons ainsi une série de coupes contenant un nombre variable de sommets, mais dont la connexité est connue.

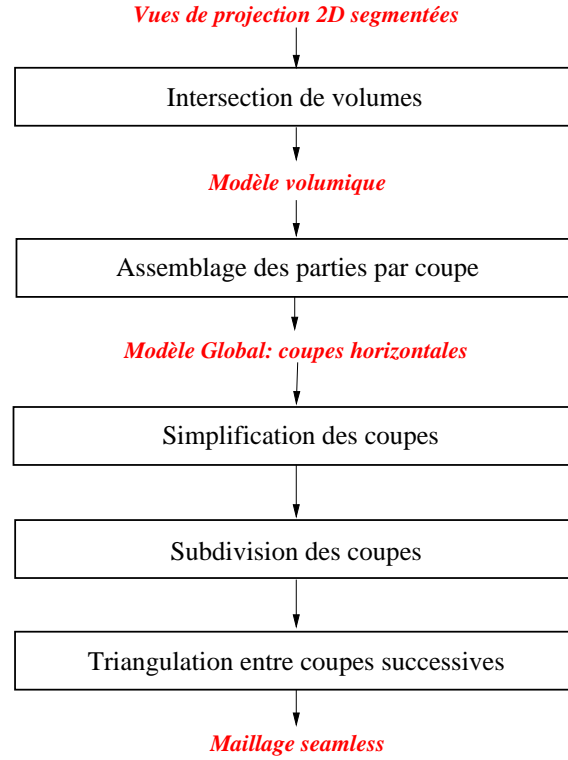


Figure 4.43 – Vue globale de la méthodologie de reconstruction par maillages.

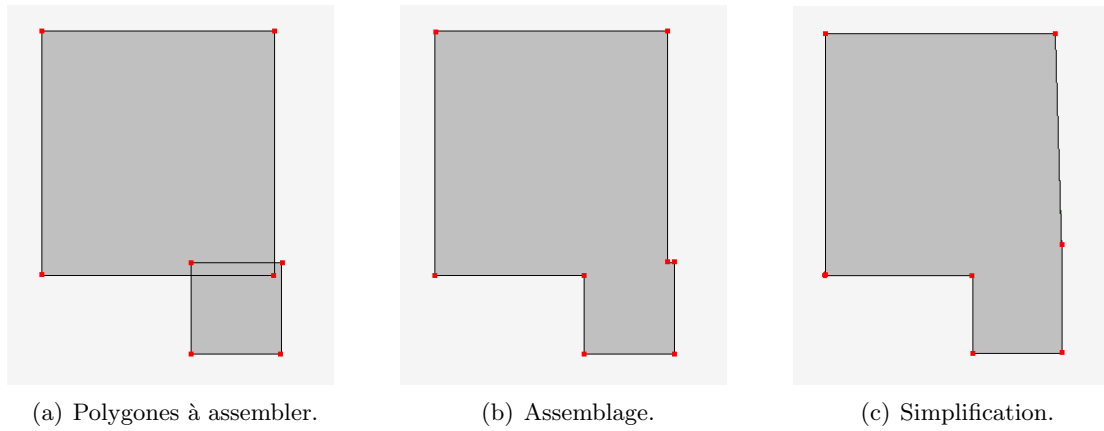


Figure 4.44 – Assemblage des modèles volumiques par coupes.

Une simplification de chaque polygone réunion ainsi obtenu est alors effectuée suivant un critère d'élimination des sommets fondé sur une information de distance relative. Une arête $S_i S_{i+1}$ appartenant à la réunion de deux polygones P_1 et P_2 est alors éliminée si et seulement si :

$$\frac{d(S_i, S_{i+1})}{\sum_{j=0}^{n-1} d(S_j, S_{j+1})} < Err, \quad (4.31)$$

où Err est une erreur maximale donnée et n est le nombre total des sommets du polygone réunion.

Une telle arête est éliminée en supprimant un de ses sommets. Le choix de celui-ci se fonde sur sa non-appartenance à la liste des sommets initiaux des polygones P_1 et P_2 .

En outre, une arête n'est éliminée que si au moins l'une de ses extrémités est générée par l'intersection d'une arête de P_1 avec une arête de P_2 . Cette condition a pour objectif de

conserver les différentes parties du modèle, même si celles-ci ont des dimensions très différentes comme cela peut être le cas pour certains détails tels les yeux ou le nez. La Figure 4.44(c) présente un exemple de simplification du polygone réunion.

Notons que d'autres critères peuvent être également considérés, comme l'estimation de l'influence de la simplification sur l'aire ou la forme du polygone réunion.

4.5.2 Lissage des coupes

Afin d'obtenir un modèle réaliste, nous nous appuyons sur la même hypothèse relative à la forme des personnages : chaque partie est modélisée par une forme simple et arrondie. Cela nous permet de passer d'un volume englobant grossier à un modèle plus lisse. Cette étape met en œuvre une procédure de subdivision de Chaikin, effectuée sur chaque coupe horizontale obtenue après assemblage des parties.

4.5.3 Triangulation

Disposant de l'information de connexité des différentes parties, la triangulation est effectuée en insérant chaque coupe dans la suivante. Cela permet de s'affranchir des méthodes de triangulation de coupes de topologie différentes telles que celle proposée par [SSBT01] qui s'appuie sur des critères de distance. La triangulation est effectuée alors entre chaque couple de coupes successives de la manière suivante :

- Les polygones appartenant à des parties connexes sont tout d'abord identifiés (Figure 4.45(a)).
- Le polygone correspondant à la partie parent sur le squelette d'animation est désigné comme *polygone parent* et constitue le polygone extérieur à trianguler.
- Les polygones de la coupe suivante correspondant à des parties enfant sont insérés dans le polygone parent pour former les contraintes de triangulation (Figure 4.45(b)).
- Une triangulation de Delaunay 2D sous contrainte de contours intérieurs est ensuite effectuée (Figure 4.45(c)).
- Enfin, les sommets des polygones enfant reprennent leur position 3D initiale. La triangulation entre les deux coupes successives est alors complète (Figure 4.45(d)).

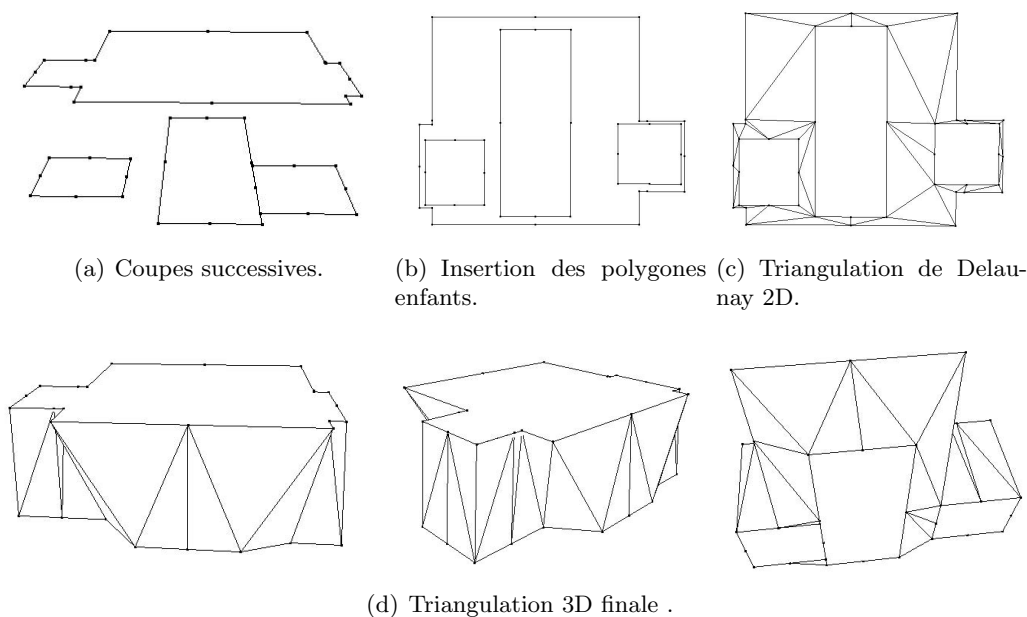


Figure 4.45 – Triangulation Delaunay 2D avec contraintes inter-coupes.

4.5.4 Résultats expérimentaux

Les résultats présentés ici correspondent au personnage Elfie, reconstruit à partir de deux vues projectives segmentées en 11 parties représentées Figure 4.46 par des couleurs différentes. Le modèle est reconstruit en utilisant les coupes horizontales obtenues de l'intersection de volume avec un sous-échantillonnage de pas 5. Les temps de calcul pour chacune des étapes d'intersection de volume et de lissage/triangulation est d'environ 40s pour des images de taille 300×600 . Le modèle final est constitué de 15768 triangles.

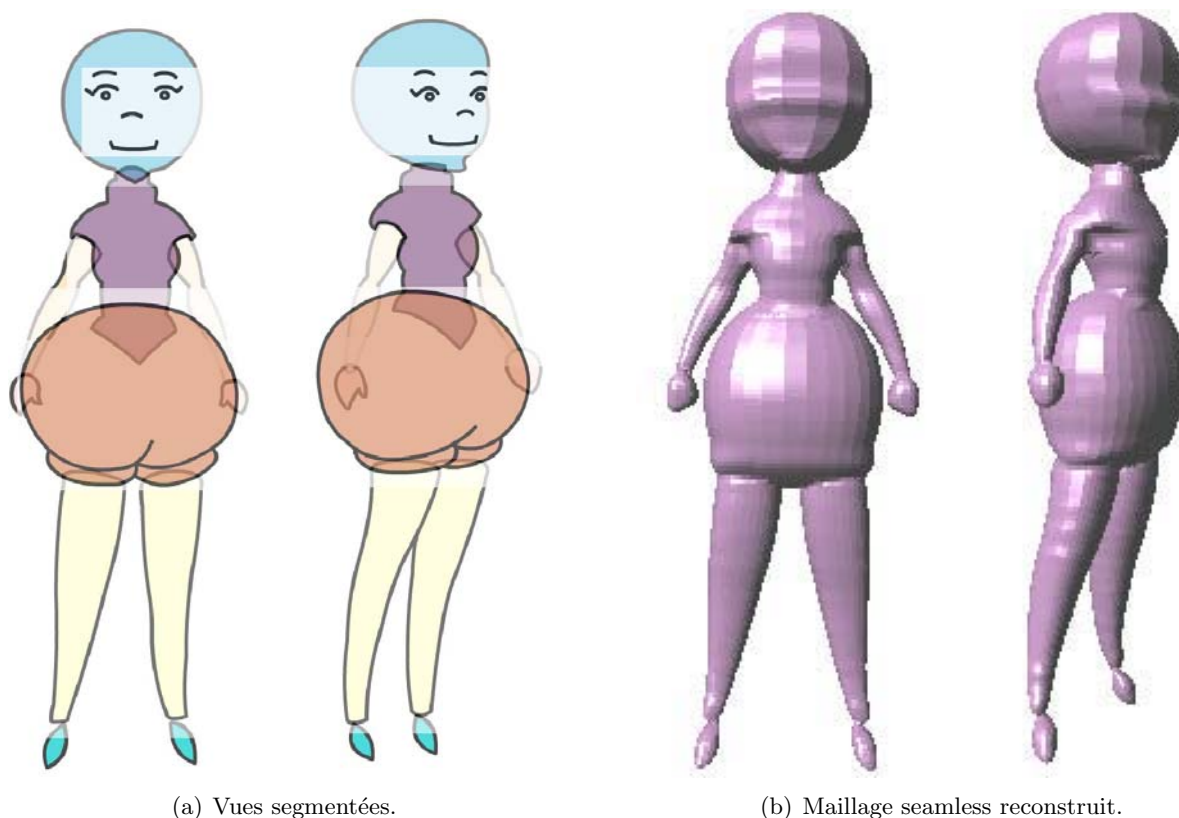
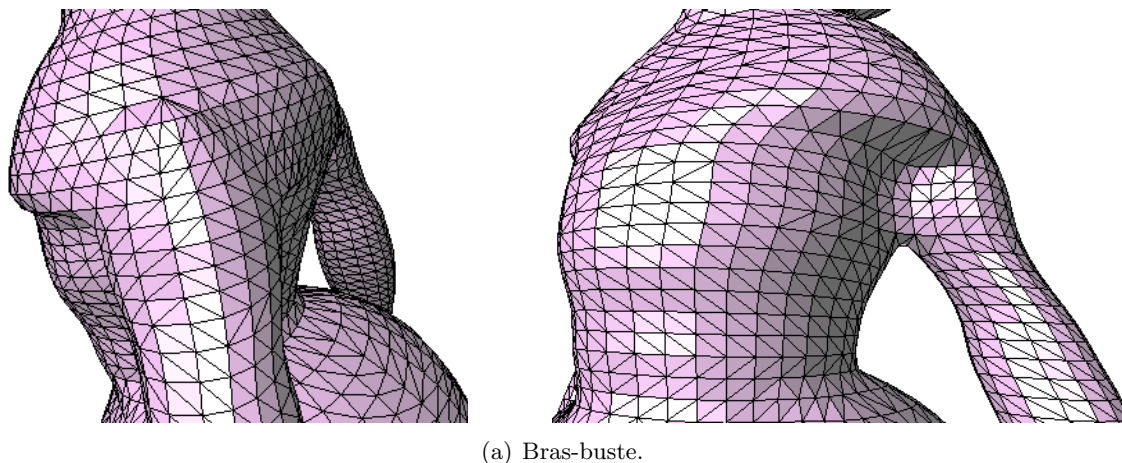
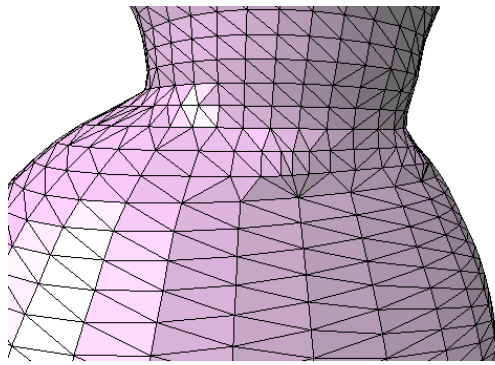


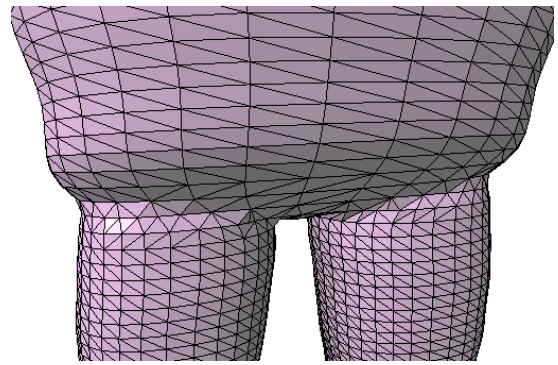
Figure 4.46 – Reconstruction par maillage du personnage Elfie à partir de deux vues.

Nous pouvons constater que la triangulation obtenue est très correcte même dans le cas où les deux parties jointes sont reconstruites avec des niveaux de subdivision différents et donc présentent des triangles de tailles très différentes, comme c'est le cas pour la tête et le buste d'Elfie (Figure 4.47).

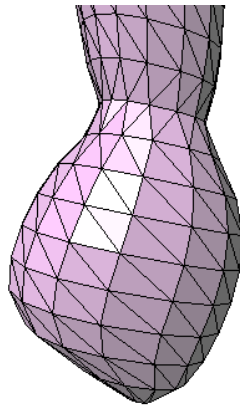




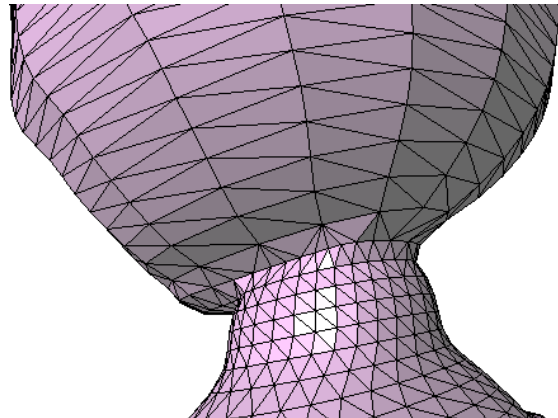
(b) Buste-jupe.



(c) Jupe-jambes.



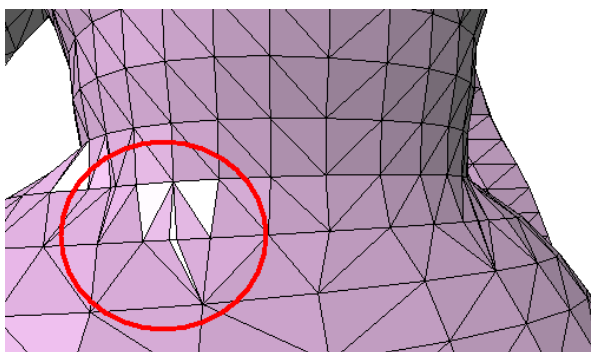
(d) Bras-main.



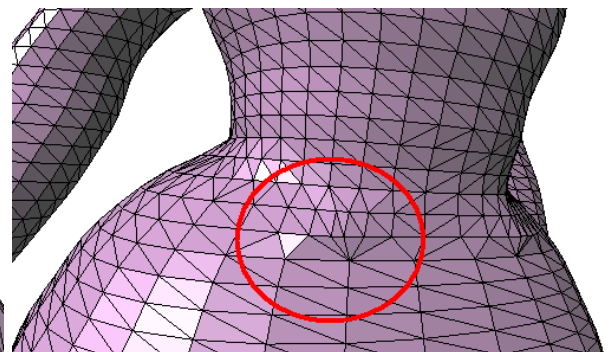
(e) Tête-buste.

Figure 4.47 – Détails des connexions de parties obtenues pour le personnage Elfie.

Enfin, lorsque les composantes jointes sont très proches, cas où l'intersection générerait normalement un maillage présentant une triangulation de mauvaise qualité, la simplification des courbes par coupe tout en prenant en compte les contraintes de reconstruction permet d'obtenir un maillage de meilleure qualité (Figure 4.48).



(a) Sans simplification.



(b) Avec simplification.

Figure 4.48 – Influence de la simplification des polygones sur la qualité de la triangulation.

4.6 Conclusion

Ce chapitre a présenté différentes approches de reconstruction de modèles paramétriques et maillés, ainsi qu'un module d'assemblage des parties modélisées par des maillages triangulaires en un unique maillage *seamless*. Ici, la reconstruction, qui se heurte à un problème de manque d'information crucial du fait du faible nombre d'images dont on dispose et de l'aspect approximatif de la projection, procède en mettant en place des hypothèses relatives à la symétrie et à la forme des objets et personnages dessinés, tout en offrant à l'utilisateur la possibilité de contrôler les formes reconstruites, soit en ajoutant des vues, soit en modifiant les paramètres de lissage longitudinal et horizontal.

Ces différentes méthodes ont été testées sur un ensemble de dessins manuels de *turnaround* d'origines différents :

- deux modèles fournis par un dessinateur professionnel pour le projet TOON (personnages Elfie et Zozo),
- quatre *turnarounds* non dessinés dans le cadre du projet (personnages Krumm, Oblina, Ickis et Titeuf),
- cinq modèles réalisés par des utilisateurs novices (Mahras, Homme et Femme tunisiens et la poule).

Les résultats obtenus prouvent l'efficacité du procédé par segmentation pour générer des modèles articulés complexes tout en restant parfaitement compatible avec les pratiques habituelles des dessinateurs. Toutefois, il est nécessaire d'introduire un module de déformation et de réajustement des modèles reconstruits afin de pouvoir ajouter des détails au modèle ou appliquer des animations non rigides.

Chapitre 5

Déformation 2D/3D de surfaces NURBS

Résumé

Les modèles reconstruits ne sont souvent pas exactement conformes aux dessins manuels fournis. Ces différences s'expliquent d'une part par les erreurs de dessins, comme les incohérences de pose et les erreurs de projection, et d'autre part par le processus de reconstruction qui implique des étapes de lissage et de sous-échantillonnage. Afin de prendre en compte ces différentes sources d'erreur et de les corriger, il est nécessaire d'introduire un module de raffinement et de déformation des modèles reconstruits.

Nous présentons dans ce chapitre une méthode de déformation appliquée aux modèles NURBS en agissant sur les points de contrôle, ce qui offre l'avantage de correspondre à une procédure intuitive dont l'action reste locale.

Ce module de déformation a été testé dans le cadre du raffinement des modèles reconstruits, ainsi que pour ajouter des détails ou déformer les modèles reconstruits selon des dessins projectifs modifiés.

Mots clés

Déformation 2D/3D, Surfaces NURBS, Maillage de contrôle, Projections, Contraintes 2D.

Dans le chapitre précédent, nous avons présenté deux différentes méthodes de reconstruction : la première à l'aide de surfaces NURBS et la seconde par des maillages. Bien que la reconstruction 2D/3D soit une étape pertinente pour le projet TOON, nos objectifs ne se limitent pas à la reconstruction, mais concernent aussi l'animation et la déformation des modèles obtenus, toujours sous la contrainte d'une interaction exclusivement 2D.

Nous présentons ici une méthode de déformation qui s'applique aux modèles NURBS et qui agit sur les points de contrôle tout en garantissant les propriétés de continuité. L'avantage de cette approche est de rester intuitive et locale.

Le module de déformation peut être appliqué soit pour raffiner les modèles reconstruits conformément aux vues projectives de départ, soit pour effectuer de nouvelles déformations en considérant de nouveaux angles de vue.

Ce chapitre propose une approche de déformation de surfaces NURBS 3D à partir de contraintes 2D. La méthode étend l'algorithme proposé dans [HLZ01]. Celui-ci consiste à déformer la surface de manière à ce qu'elle interpole un ensemble de points de coordonnées 3D connues.

Introduisons tout d'abord les notations mathématiques nécessaires. Soit $P(u, v)$ une surface NURBS définie comme :

$$P(u, v) = \sum_{i=0}^n \sum_{j=0}^m P_{i,j} R_{i,j}(u, v). \quad (5.1)$$

Notons par $\hat{P}(u, v)$ la surface modifiée par déplacement des points de contrôle, et supposons que les points de contrôle à déplacer sont les $P_{i,j}$ tels que $i_1 \leq i \leq i_2$ et $j_1 \leq j \leq j_2$. $\hat{P}(u, v)$ s'écrit alors :

$$\begin{aligned} \hat{P}(u, v) &= \sum_{i=0}^n \sum_{j=0}^m \hat{P}_{i,j} R_{i,j}(u, v) \\ &= \sum_{i=0}^n \sum_{j=0}^m (P_{i,j} + \epsilon_{i,j}) R_{i,j}(u, v) \\ &= P(u, v) + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}(u, v), \end{aligned} \quad (5.2)$$

où $\epsilon_{i,j}$ sont les vecteurs de déplacement appliqués aux points de contrôle $P_{i,j}$.

L'objectif est de minimiser ces déplacements, au sens des moindres carrés sous les contraintes considérées. La fonction à minimiser E est définie par :

$$E = \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \|\epsilon_{i,j}\|^2. \quad (5.3)$$

Rappelons tout d'abord la méthodologie 3D/3D telle que décrite dans [HLZ01]. Ici, les contraintes sont des points de \mathbb{R}^3 que la surface déformée doit interpoler.

5.1 Déformation sous contraintes 3D

Dans le cas où la contrainte est un ensemble de points de \mathbb{R}^3 , nous avons adopté la solution proposée par Hu *et al.* dans [HLZ01].

Notons par $(r+1)$ le nombre de points de contrainte, que la surface NURBS doit interpoler. Soient

- $\{T_l\}_{l \in \{0, \dots, r\}}$, les vecteurs de coordonnées des points de contrainte 3D (points cibles),

- $\{S_l = P(u_l, v_l)\}_{l \in \{0, \dots, r\}}$ leurs “projections” sur la surface NURBS initiale (non déformée) P , appelés points sources. S_l est défini comme le point sur la surface P le plus proche, au sens de la distance euclidienne, du point cible T_l .
 - et $\{N_l\}_{l \in \{0, \dots, r\}}$ les normales à la surface P aux points sources S_l .
- La contrainte d’interpolation des points T_l s’exprime par l’équation suivante :

$$\forall l \in \{0, \dots, r\}, \quad T_l = \hat{P}(u_l, v_l) = S_l + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}(u_l, v_l). \quad (5.4)$$

Une deuxième contrainte de conservation des normales à la surface, aux coordonnées paramétriques (u_l, v_l) est aussi imposée. Elle s’exprime par les équations suivantes :

$$\forall l \in \{0, \dots, r\}, \quad \begin{cases} \frac{\partial}{\partial u} \hat{P}(u_l, v_l) \cdot N_l = \left(\frac{\partial}{\partial u} P(u_l, v_l) + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}^u(u_l, v_l) \right) \cdot N_l = 0 \\ \frac{\partial}{\partial v} \hat{P}(u_l, v_l) \cdot N_l = \left(\frac{\partial}{\partial v} P(u_l, v_l) + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}^v(u_l, v_l) \right) \cdot N_l = 0 \end{cases} \quad (5.5)$$

La solution de ce problème d’optimisation sous contraintes est obtenue à l’aide du formalisme lagrangien.

Les déplacements appliqués aux points de contrôle sont alors déterminés en minimisant la fonction lagrangienne définie comme suit :

$$\begin{aligned} L = & \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \|\epsilon_{i,j}\|^2 \\ & + \sum_{l=0}^r \lambda_l \left(T_l - S_l - \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}(u_l, v_l) \right) \\ & + \sum_{l=0}^r \delta_l \left(\left(\frac{\partial}{\partial u} P(u_l, v_l) + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}^u(u_l, v_l) \right) \cdot N_l \right) \\ & + \sum_{l=0}^r \gamma_l \left(\left(\frac{\partial}{\partial v} P(u_l, v_l) + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}^v(u_l, v_l) \right) \cdot N_l \right) \end{aligned} \quad (5.6)$$

où $\lambda_l = [\lambda_l^x, \lambda_l^y, \lambda_l^z]^T$, $\delta_l \in \mathbb{R}$ et $\gamma_l \in \mathbb{R}$, $l \in \{0, \dots, r\}$, sont les multiplicateurs lagrangiens, et $\|\cdot\|$ est la norme euclidienne.

Afin de déterminer le minimum du lagrangien, nous recherchons les zéros de ses dérivées partielles par rapport aux multiplicateurs lagrangiens et aux déplacements $\epsilon_{i,j}$:

$$\begin{cases} \forall l \in \{0, \dots, r\}, & \frac{\partial}{\partial \lambda_l}(L) = 0, \\ \forall l \in \{0, \dots, r\}, & \frac{\partial}{\partial \delta_l}(L) = 0, \\ \forall l \in \{0, \dots, r\}, & \frac{\partial}{\partial \gamma_l}(L) = 0, \\ \forall i \in \{i_1, \dots, i_2\}, \forall j \in \{j_1, \dots, j_2\}, & \frac{\partial}{\partial \epsilon_{i,j}}(L) = 0. \end{cases} \quad (5.7)$$

Cela conduit au système d’équations suivant :

$$\left\{ \begin{array}{l} \forall l \in \{0, \dots, r\}, \\ \forall l \in \{0, \dots, r\}, \\ \forall l \in \{0, \dots, r\}, \\ \forall i \in \{i_1, \dots, i_2\}, \forall j \in \{j_1, \dots, j_2\}, \\ 2\epsilon_{i,j} - \sum_{l=0}^r (\lambda_l R_{i,j}(u_l, v_l) - \delta_l R_{i,j}^u(u_l, v_l).N_l - \gamma_l R_{i,j}^v(u_l, v_l).N_l) = 0. \end{array} \right. \quad \begin{array}{l} T_l = S_l + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}(u_l, v_l), \\ \left(\frac{\partial}{\partial u} P(u_l, v_l) + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}^u(u_l, v_l) \right) . N_l = 0, \\ \left(\frac{\partial}{\partial v} P(u_l, v_l) + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}^v(u_l, v_l) \right) . N_l = 0, \end{array} \quad (5.8)$$

La contrainte de conservation des normales peut être relâchée, surtout lorsque les déformations souhaitées sont relativement faibles. Cela permet de diminuer la taille de la matrice à inverser et d'accélérer ainsi significativement le temps de calcul. On obtient alors le système linéaire suivant :

$$\left\{ \begin{array}{l} \forall l \in \{0, \dots, r\}, \\ \forall i \in \{i_1, \dots, i_2\}, \forall j \in \{j_1, \dots, j_2\}, \end{array} \right. \quad \begin{array}{l} T_l = S_l + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}(u_l, v_l), \\ 2\epsilon_{i,j} = \sum_{l=0}^r \lambda_l R_{i,j}(u_l, v_l). \end{array} \quad (5.9)$$

Ce système s'écrit sous la forme matricielle ci-dessous :

$$\begin{bmatrix} R & 0_{r+1} \\ -2I_{K+1} & R^t \end{bmatrix} \cdot \begin{bmatrix} \epsilon_0 \\ \vdots \\ \epsilon_K \\ \lambda_0 \\ \vdots \\ \lambda_r \end{bmatrix} = \begin{bmatrix} T_0 - S_0 \\ \vdots \\ T_r - S_r \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (5.10)$$

où $K = (i_2 - i_1)(j_2 - j_1)$ est le nombre de points de contrôle à déplacer, 0_{r+1} et I_{K+1} sont respectivement la matrice nulle et la matrice identité de taille $(r+1) \times (r+1)$ et $(K+1) \times (K+1)$, et :

$$R = \begin{bmatrix} R_0(u_0, v_0) & \dots & R_K(u_0, v_0) \\ \vdots & & \vdots \\ R_0(u_r, v_r) & \dots & R_K(u_r, v_r) \end{bmatrix}. \quad (5.11)$$

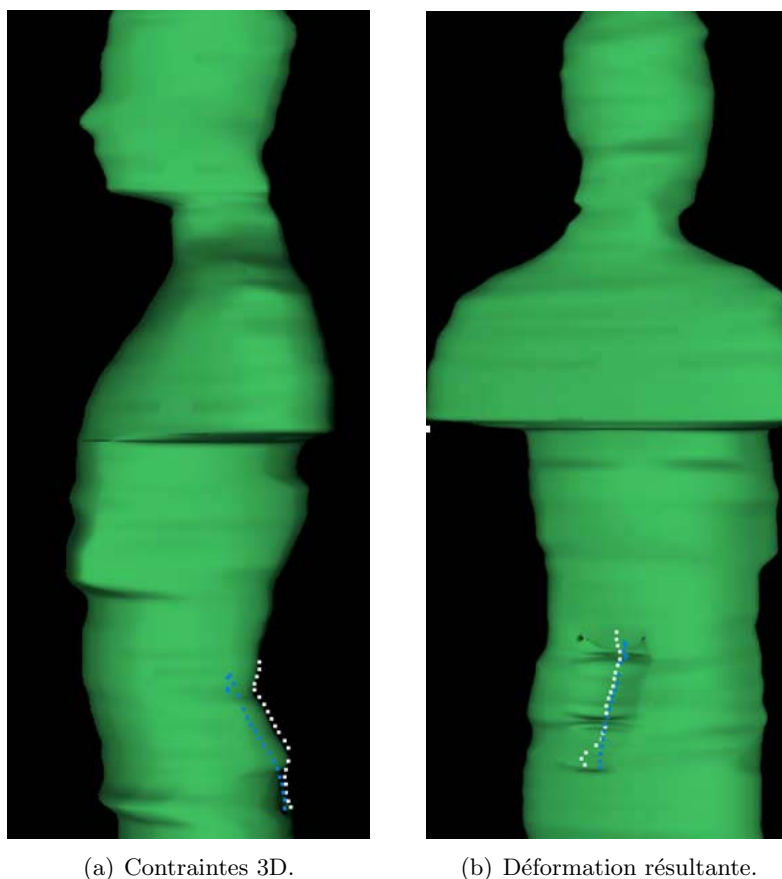
Pour des raisons de stabilité, le système 5.10 est résolu par SVD¹ [PTVF92]. La surface déformée est alors obtenue en appliquant les déplacements ϵ_k aux points de contrôle P_k , tout en conservant les valeurs initiales des coefficients R_k .

La Figure 5.1 présente un premier résultat obtenu en utilisant deux points de contrainte, pour créer un nez à un modèle d'humanoïde. Un deuxième résultat est présenté Figure 5.2, où un creux est appliqué au dos de l'humanoïde à l'aide d'un ensemble de 16 points de contrainte.

¹Singular Value Decomposition



Figure 5.1 – Déformation sous contraintes 3D du nez d'un humanoïde.



(a) Contraintes 3D.

(b) Déformation résultante.

Figure 5.2 – Déformation sous contraintes 3D du dos d'un humanoïde.

Cette approche de déformation nécessite la spécification manuelle des contraintes 3D. Cela limite son application directe dans le cadre du projet TOON.

Afin de s'affranchir de cet inconvénient, nous proposons une extension de la méthode de Hu qui permet de déformer la surface NURBS 3D à partir de contraintes 2D, correspondant à des contours fournis par les dessinateurs.

5.2 Déformation sous contraintes 2D

Le principe consiste à remplacer les contraintes ponctuelles 3D par des contours dessinés sur les images de projection. Considérons le cas d'une image de projection, correspondant à l'angle de vue θ . L'algorithme procède en effectuant une rotation d'un angle θ du modèle NURBS 3D autour de l'axe z . Les contraintes sont ensuite déterminées et les déformations effectuées. Enfin, une rotation du modèle déformé de $-\theta$ est réalisée. Ainsi est-il possible de déformer successivement le modèle sous différents angles de vue.

Soit A le *contour apparent* de la surface P . Ce contour est obtenu en projetant le *contour générateur* G , géodésique à la surface P , défini par l'ensemble des points de P où la normale à la surface est orthogonale à la direction de vue.

L'objectif est alors de déformer la surface de manière à ce que le contour apparent coïncide avec le contour cible :

$$A = \Pi_x(G) = C, \quad (5.12)$$

où Π_x est l'opérateur de projection selon l'axe horizontal x (Figure 5.3).

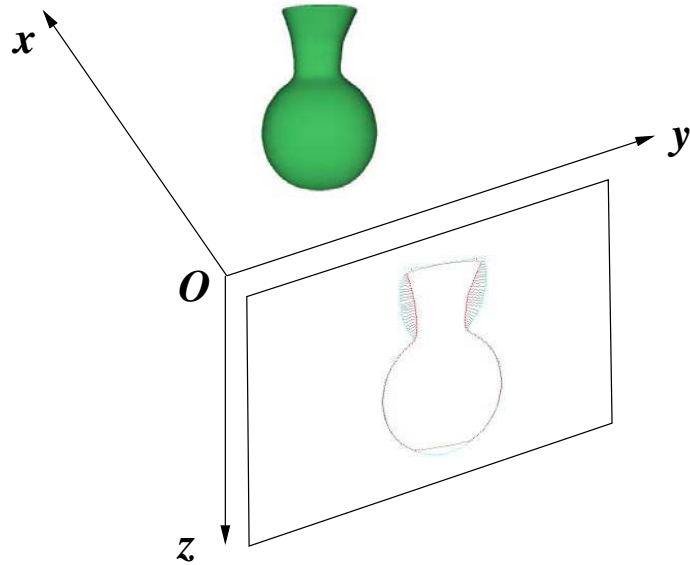


Figure 5.3 – Contrainte 2D de projection.

Afin de réduire le problème au formalisme présenté dans la section 5.1, nous réalisons un échantillonnage des contours cible et générateur pour déterminer un ensemble de couples de points $\{(T_l, S_l)\}_{l \in \{0, \dots, r\}}$ où :

- $(r + 1)$ est le nombre d'échantillons considérés,
- T_l sont des points du contour projectif 2D cible,
- S_l sont les points sources correspondants sur le contour générateur, dont la détermination est détaillée dans le paragraphe 5.2.1.

La contrainte de déformation s'écrit alors :

$$\begin{aligned} T_l &= \Pi_x(\hat{P}(u_l, v_l)) \\ &= \Pi_x(S_l + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}(u_l, v_l)) \end{aligned} \quad (5.13)$$

Les déplacements $\epsilon_{i,j}$ des points de contrôle sont alors déterminés en minimisant le lagrangien suivant :

$$\begin{aligned}
L &= \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \|\epsilon_{i,j}\|^2 \\
&+ \sum_{k=0}^K \left(\sum_{l=0}^r \lambda_l \left(T_l - \Pi_x \left(S_l + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}(u_l, v_l) \right) \right) \right) \\
&+ \sum_{l=0}^r \delta_l \left(\left(\frac{\partial}{\partial u} P(u_l, v_l) + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}^u(u_l, v_l) \right) N_l \right) \\
&+ \sum_{l=0}^r \gamma_l \left(\left(\frac{\partial}{\partial v} P(u_l, v_l) + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}^v(u_l, v_l) \right) N_l \right)
\end{aligned} \tag{5.14}$$

où les $\lambda_l \in \mathbb{R}^2$, $\delta_l \in \mathbb{R}^2$ et $\gamma_l \in \mathbb{R}^2$ sont les multiplicateurs lagrangiens.

Pour des raisons de simplicité de calcul, nous considérons dans ce qui suit uniquement la contrainte de projection, ce qui nous donne le lagrangien simplifié suivant :

$$L = \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \|\epsilon_{i,j}\|^2 + \sum_{l=0}^r \lambda_l \left(T_l - \Pi_x \left(S_l + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j} R_{i,j}(u_l, v_l) \right) \right), \tag{5.15}$$

En développant, nous obtenons :

$$L = \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \|\epsilon_{i,j}\|^2 + \sum_{l=0}^r \lambda_l \left(T_l - \Pi_x(S_l) + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \Pi_x(\epsilon_{i,j}) R_{i,j}(u_l, v_l) \right), \tag{5.16}$$

La minimisation du lagrangien est effectuée en résolvant les équations aux dérivées partielles :

$$\begin{cases} \forall l \in 0, \dots, r, & \frac{\partial}{\partial \lambda_l}(L) = 0, \\ \forall i \in i_1, \dots, i_2, \forall j \in j_1, \dots, j_2, & \frac{\partial}{\partial \epsilon_{i,j}}(L) = 0. \end{cases} \tag{5.17}$$

Notons que les dérivées partielles par rapport au multiplicateurs lagrangiens λ_l donnent la contrainte 5.13.

Calculons à présent les dérivées partielles par rapport aux déplacements $\epsilon_{i,j}$.

$$\frac{\partial}{\partial \epsilon_{p,q}}(L) = 2\epsilon_{p,q} - \frac{\partial}{\partial \epsilon_{p,q}} \left(\sum_{l=0}^r \lambda_l \cdot \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \Pi_x(\epsilon_{i,j}) R_{i,j}(u_l, v_l) \right) \tag{5.18}$$

$$= 2\epsilon_{p,q} - \frac{\partial}{\partial \epsilon_{p,q}} \left(\sum_{l=0}^r \lambda_l \cdot \Pi_x(\epsilon_{p,q}) R_{p,q}(u_l, v_l) \right) \tag{5.19}$$

$$= 2\epsilon_{p,q} - \frac{\partial}{\partial \epsilon_{p,q}} \left(\sum_{l=0}^r \lambda_l \cdot \Pi_x(\epsilon_{p,q}) \right) R_{p,q}(u_l, v_l). \tag{5.20}$$

$$\tag{5.21}$$

Cette équation, exprimée pour chacune des coordonnées x , y et z , se traduit alors par le

système :

$$\begin{cases} \frac{\partial}{\partial \epsilon_{p,q}^x}(L) &= 2\epsilon_{p,q}^x, \\ \frac{\partial}{\partial \epsilon_{p,q}^y}(L) &= 2\epsilon_{p,q}^y - \sum_{l=0}^r \lambda_l^y R_{p,q}(u_l, v_l), \\ \frac{\partial}{\partial \epsilon_{p,q}^z}(L) &= 2\epsilon_{p,q}^z + \sum_{l=0}^r \lambda_l^z R_{p,q}(u_l, v_l). \end{cases} \quad (5.22)$$

Notons que la solution triviale de la première équation est $\epsilon_{p,q}^x = 0$, ce qui implique que les points de contrôle sont toujours déplacés parallèlement au plan de l'image de contrainte. Les déplacements $\epsilon_{i,j} = [0, \epsilon_{p,q}^y, \epsilon_{p,q}^z]^T$ sont alors obtenus en résolvant les deux systèmes d'équations

– Système linéaire en y :

$$\begin{cases} T_l^y &= S_l^y + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j}^y R_{i,j}(u_l^{\theta_k}, v_l^{\theta_k}), \\ 0 &= 2\epsilon_{i,j}^y - \sum_{l=0}^r \lambda_l^y R_{i,j}(u_l, v_l). \end{cases} \quad (5.23)$$

– Système linéaire en z :

$$\begin{cases} T_l^z &= S_l^z + \sum_{i=i_1}^{i_2} \sum_{j=j_1}^{j_2} \epsilon_{i,j}^z R_{i,j}(u_l, v_l), \\ 0 &= 2\epsilon_{i,j}^z + \sum_{l=0}^r \lambda_l^z R_{i,j}(u_l, v_l). \end{cases} \quad (5.24)$$

Afin de résoudre ce système, nous devons d'abord déterminer les couples de points cibles et sources.

5.2.1 Détermination des points sources 3D

Rappelons que le contour générateur correspond à l'ensemble de points du modèle 3D où la direction de vue est tangente à la surface du modèle. Cette tangence se traduit par un changement de signe (passage par 0) du produit scalaire entre la direction de vue D_v et le vecteur normal à la surface N_S . Afin de déterminer le contour générateur de la surface S_l , nous optons alors pour une méthode itérative pour rechercher le zéro de la fonction

$$f(u) = \langle N_S(u), D_v \rangle. \quad (5.25)$$

L'ensemble des points correspondant aux coordonnées relatives u vérifiant l'équation $f(u) = 0$ définit une courbe (ou un ensemble de courbes) sur la surface 3D, qui génère le contour projectif ou contour apparent.

Afin d'obtenir un ensemble fini de points de contraintes, nous effectuons tout d'abord un échantillonnage longitudinal de la surface, selon un ensemble de coordonnées $\{u_j\}_j$. Pour chaque u_j , on applique alors les étapes suivantes :

1. Détection des intervalles de nœuds $[u_i, u_{i+1}]$ correspondant à un changement de signe de la fonction $f(u)$ (Figure 5.4(a)).
2. Initialisation des k valeurs recherchées pour chaque intervalle $[u_i, u_{i+1}]$ à une suite $\{u^{(n)}\}_n$ convergeant vers le zéro de la fonction f :

$$u^{(0)} = \frac{u_i + u_{i+1}}{2}. \quad (5.26)$$

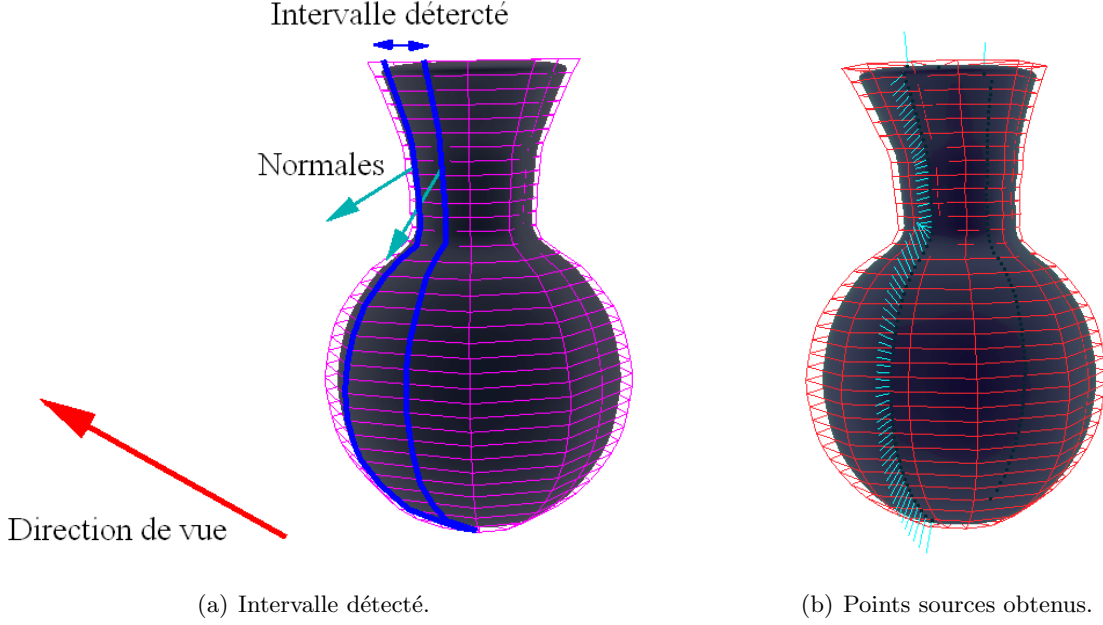


Figure 5.4 – Détermination des points sources sur le contour générateur.

3. Itérations de Newton pour déterminer le zéro de la fonction f . A chaque étape, la valeur $u^{(n)}$ est actualisée comme suit :

$$u^{(n)} = u^{(n-1)} - \frac{f(u^{(n-1)})}{\frac{\partial}{\partial u} f'(u^{(n-1)})}. \quad (5.27)$$

où f' désigne la dérivée de la fonction f .

La Figure 5.4(b) illustre les points de contrainte S_l obtenus pour le vase.

Une fois les points sources déterminés, il reste à calculer les points cibles correspondants sur les contours 2D de projection.

5.2.2 Détermination des contraintes de projection

Les contraintes de projection T_l sont obtenues de la manière suivante :

1. Les points du contour apparent S_l sont tout d'abord projetés selon la direction de vue considérée D_v . Soient S_l^{2d} leurs projections.
2. Les points de contrainte T_l sont définis par les projections des S_l^{2d} sur la courbe de contrainte fournie par le dessinateur, modélisée par une courbe NURBS 2D. La projection sur une courbe NURBS étant impossible à résoudre directement (de par la définition itérative des courbes et surfaces NURBS), nous utilisons une méthode itérative similaire à celle mise en oeuvre pour déterminer les points du contour apparent S_l . L'initialisation du système est cette fois donnée à l'aide des coordonnées relatives $u_l^{(0)}$ des points sources sur la courbe polygonale de projection $\{S_0^{2d}, \dots, S_{N_c}^{2d}\}$, où N_c est le nombre de points de contrainte :

$$u_l^{(0)} = \frac{\sum_{i=1}^l \|S_i - S_{i-1}\|}{\sum_{i=1}^{N_c} \|S_i - S_{i-1}\|} \quad (5.28)$$

Notons que, afin d'éviter tout décalage entre les coordonnées relatives des deux courbes de projection et de contrainte, celles-ci sont orientées dans le même sens, en commençant par le point haut gauche.

3. L'algorithme itératif de Newton approche alors u_l suivant la contrainte

$$f(u_l) = \langle T_C(u_l), S_l - C(u_l) \rangle = 0, \quad (5.29)$$

où $C(u_l)$ représente les coordonnées 2D correspondant à la coordonnée relative u_l sur la courbe de contrainte C , $T_C(u_l)$ est le vecteur tangent à la courbe de contrainte C en la coordonnée relative u_l , et S_l est le point source 2D considéré.

Toutefois, lorsque les contours de projection et cible sont localement éloignés (forte déformation), l'initialisation des coordonnées relatives cibles introduit un décalage sur les correspondances. Afin d'éviter ce problème, nous avons opté pour une initialisation fondée sur un échantillonnage régulier de la courbe NURBS de contrainte en N_e échantillons $\{c_0, \dots, c_{N_e-1}\}$, et sur une initialisation par la coordonnée relative $u(c_{j_{min}})$ de l'échantillon $c_{j_{min}}$ le plus proche en terme de distance euclidienne du point source considéré :

$$j_{min} = \arg \min_{j \in \{0, \dots, N_e-1\}} (\|S_l - c_j\|), \quad (5.30)$$

$$u_l^{(0)} = u(c_{j_{min}}). \quad (5.31)$$

$$(5.32)$$

La Figure 5.5 illustre les contraintes calculées pour le modèle de la tête de Titeuf. Ici, les contours projectifs cibles sont représentés en bleu, et les contours source correspondant à la projection de la surface sont représentés en rouge. Les contraintes 2D utilisées sont représentées par des segments de droite reliant des points sur les deux courbes. Notons que ce code de représentation est repris pour l'ensemble des figures de déformation suivantes.

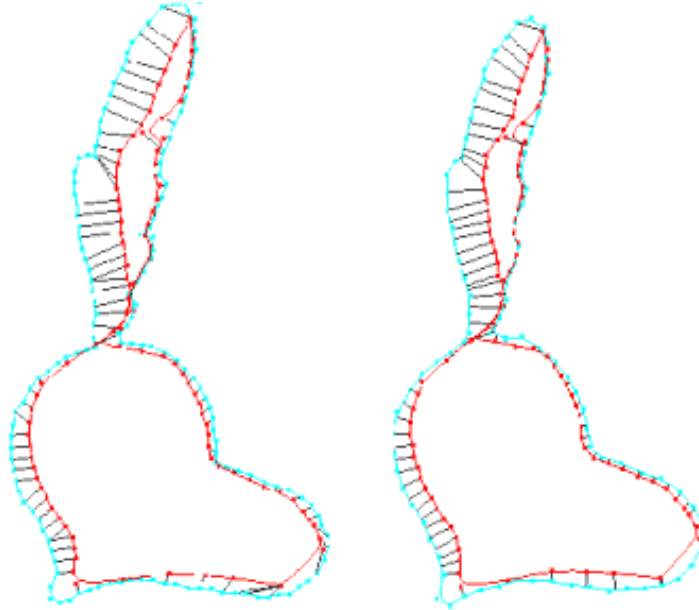


Figure 5.5 – Contraintes de déformation obtenues pour le modèle de la tête de Titeuf reconstruit et le contour de départ : initialisation des contraintes par correspondance des coordonnées relatives (à gauche), puis correspondances après application de l'algorithme itératif de Newton (à droite).

5.2.3 Résultats expérimentaux

Une première application de la méthode de déformation sous contraintes 2D consiste à corriger les modèles reconstruits conformément aux contours initiaux. En effet, les méthodes de construction utilisées introduisent des étapes de lissage. De plus, l'intersection de volumes conduit en cas d'incohérences entre les coupes et d'erreurs de projection à un modèle volumique qui ne correspond déjà plus aux courbes produites par les dessinateurs. Appliquer une déformation peut alors aider à mieux approcher ces contours.

Les Figures 5.6, 5.7 et 5.8 illustrent respectivement des exemples de correction dans le cas de la tête, du pull, du pied et de la main du personnage Titeuf selon des contraintes projectives dérivées d'une ou d'un ensemble de vues utilisées pour la reconstruction. Ici, les modèles sont reconstruits à l'aide de la méthode d'interpolation de courbes NURBS horizontales, puis lissés et sous-échantillonnés, ce qui a induit une erreur entre les contours utilisés et les re-projections du modèle 3D obtenu.

Notons que dans le cas du buste et du pied, les contraintes utilisées correspondent à une unique vue et sont satisfaites comme le montrent les re-projections Figures 5.6(d) et 5.7(d).

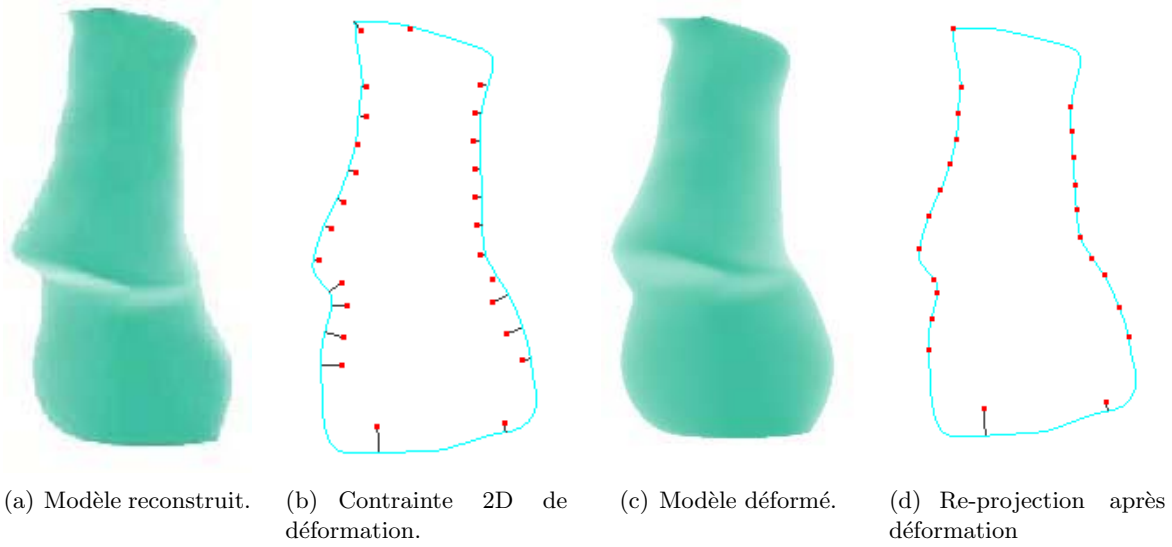


Figure 5.6 – Réajustement du pull du personnage Titeuf selon la vue $\theta = 90^\circ$.

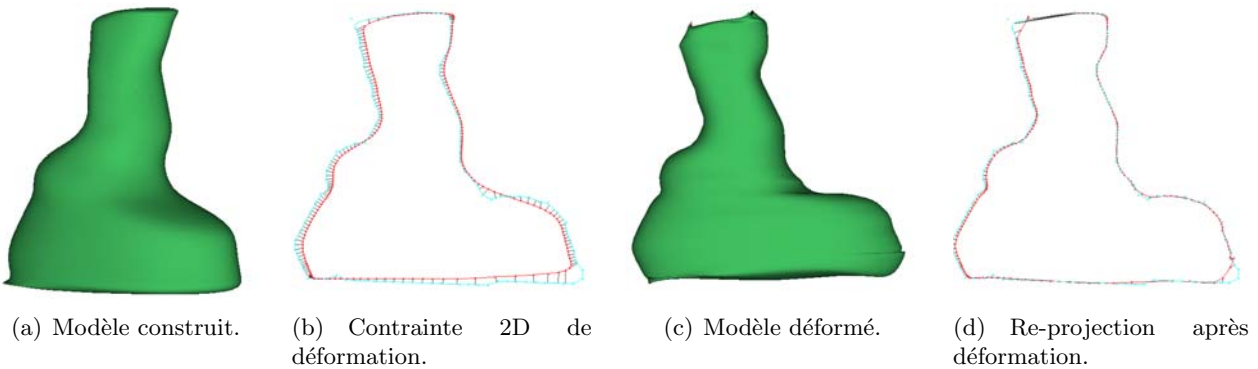


Figure 5.7 – Réajustement d'une jambe du personnage Titeuf selon la vue $\theta = 90^\circ$.

En revanche, pour l'exemple de la main, les corrections sont effectuées conformément à l'ensemble des vues utilisées pour la reconstruction (face et profil). On observe que les contraintes ne sont pas pleinement satisfaites. Cela est justifié par le fait que la méthode de déformation procède par minimisation de l'énergie de déplacement des points de contrôle. Cette étape est

concrétisée dans ce cas par la résolution d'un système linéaire où le nombre d'équations est supérieur au nombre de variables. Ainsi, lorsqu'un point de contrôle est sollicité par plusieurs contraintes correspondant à des vues différentes, la résolution ne peut-elle pas nécessairement satisfaire l'ensemble des contraintes.

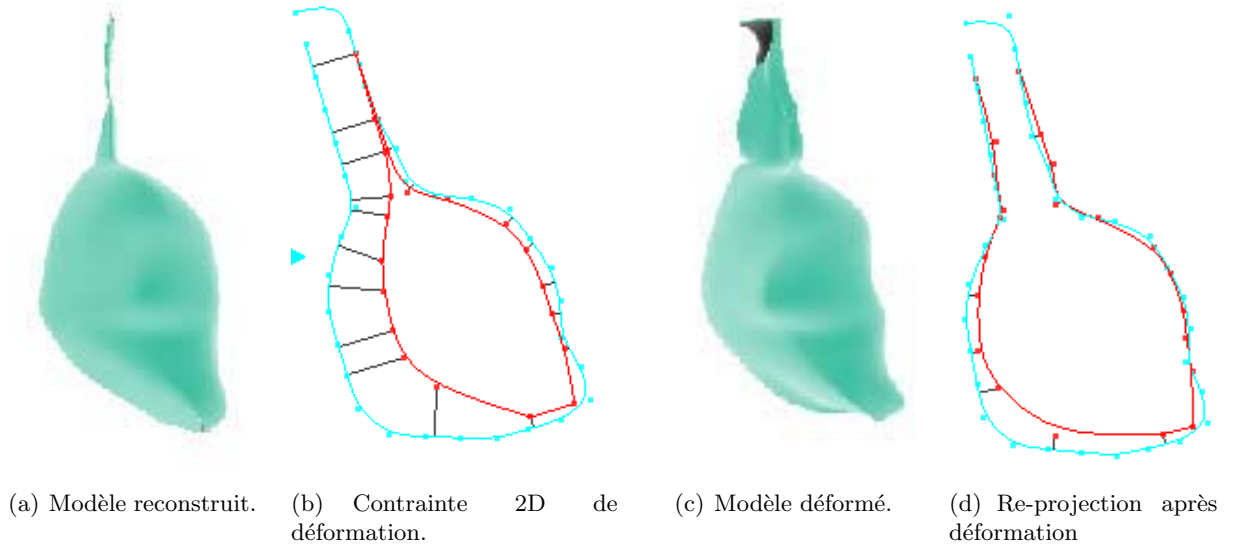


Figure 5.8 – Réajustement de la main du personnage Titeuf selon les vues $\theta = 0^\circ$ et $\theta = 90^\circ$.

Les figures 5.9 et 5.10 présentent les résultats des déformations obtenues respectivement pour la tête et le torse du personnage Elfie, réalisées à partir de la vue de projection à $\theta = 30^\circ$ non utilisée pour la reconstruction.

Soulignons, dans le cas du buste, l'apparition de la forme creuse au niveau de l'épaule du personnage grâce à cette étape de déformation. Cela met en valeur l'importance de cette étape qui permet de modéliser des formes ne pouvant pas être reconstruites directement. En effet, la reconstruction exploite l'enveloppe visuelle qui est par définition convexe pour chaque coupe.

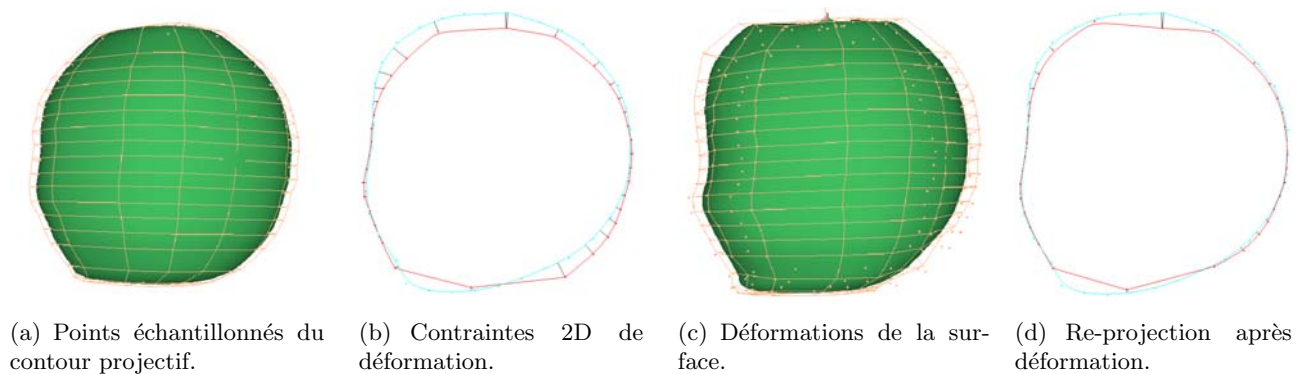


Figure 5.9 – Déformation de la tête d'Elfie selon la vue $\theta = 30^\circ$.

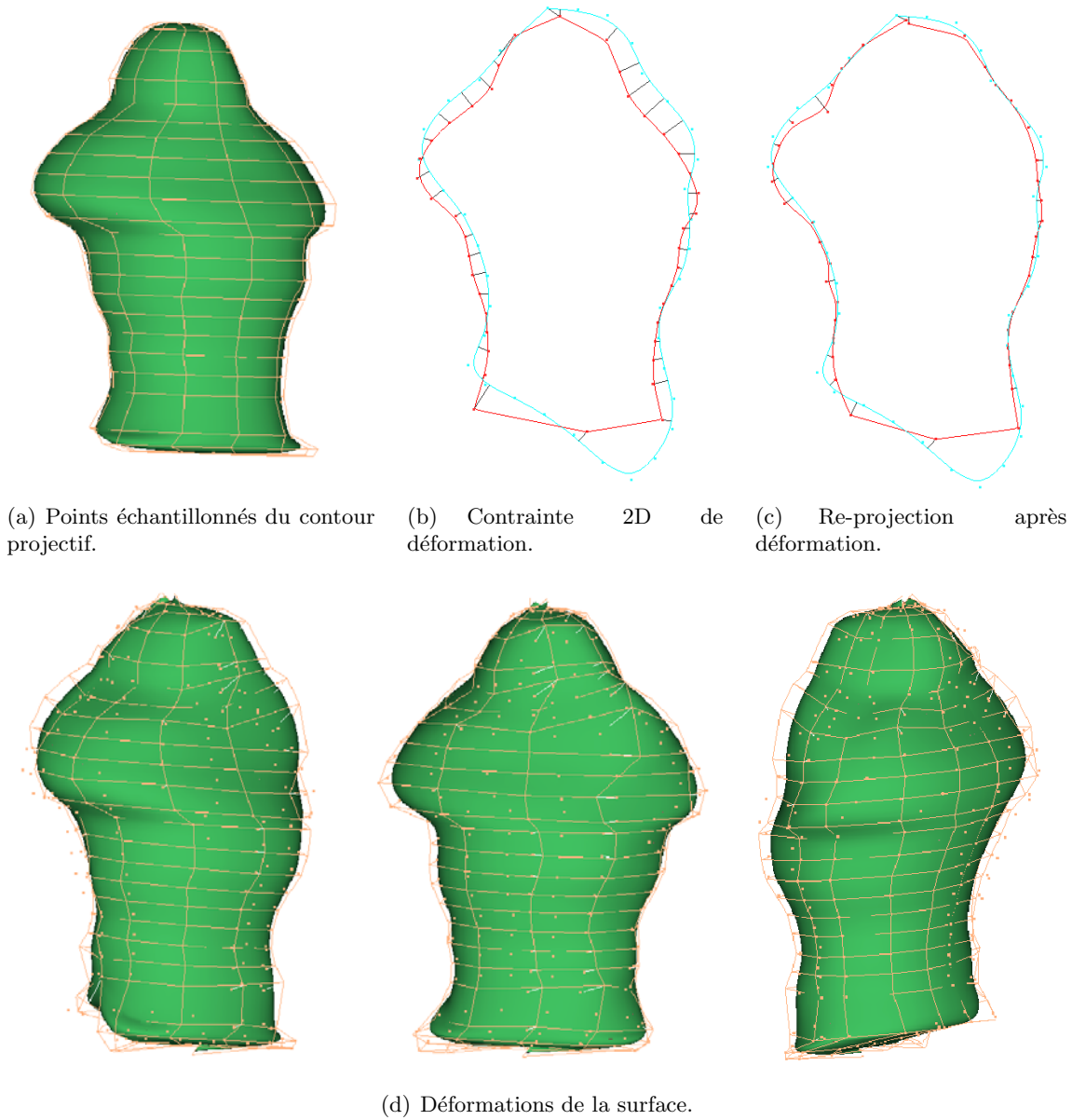


Figure 5.10 – Déformation du buste D'Elfie selon la vue $\theta = 30^\circ$.

Les figures 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.17, 5.18, 5.19, 5.20, 5.21 et 5.22 illustrent des exemples de déformation pour différents modèles reconstruits à l'aide de deux vues de projection, où certaines d'entre elles ont été par la suite modifiées.

Nous observons globalement une conformité très correcte des reprojections après déformation avec les contours de contraintes.

Notons que le nombre de contraintes utilisées est paramétrable, ce qui permet de réaliser un compromis entre les temps de calcul et la précision de l'échantillonnage des contours générateur et de contrainte. Ici, les temps de calculs varient entre 30s et 1mn en fonction de la taille du système déterminée à la fois par le nombre de contraintes, le maillage de contrôle et le degré de la NURBS.

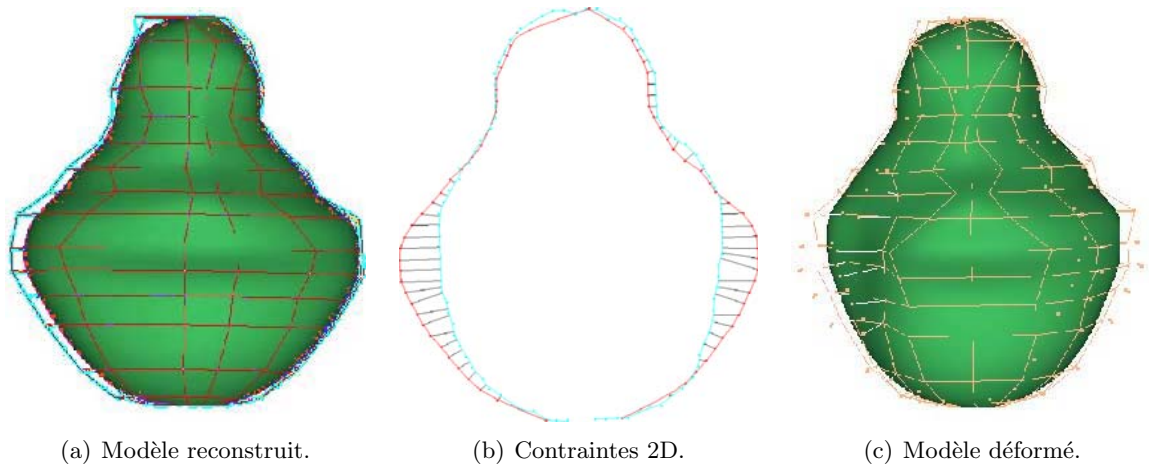


Figure 5.11 – Déformation de la tête du personnage Al selon la vue $\theta = 0^\circ$ pour obtenir un effet d'amincissement.

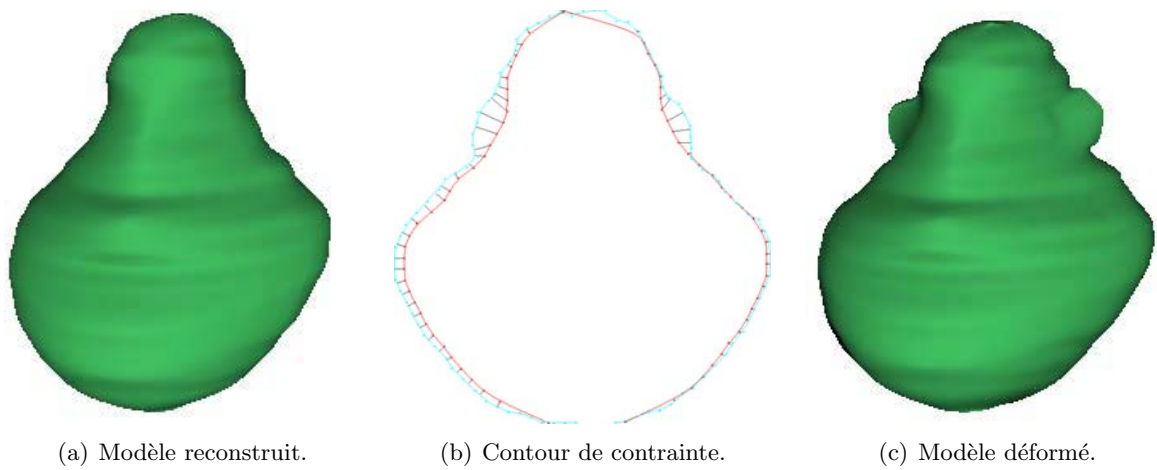


Figure 5.12 – Déformation par contrainte projective : ajout d'oreilles à la tête du personnage Al en modifiant une unique vue projective.

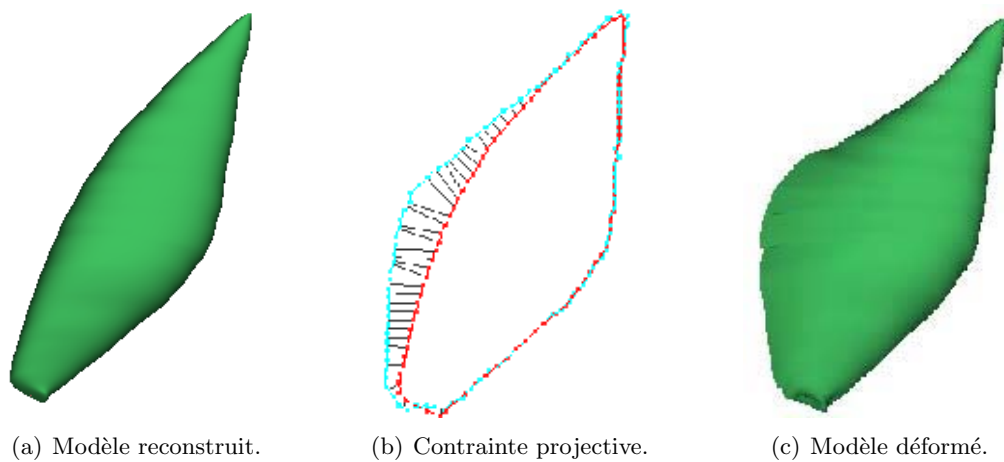


Figure 5.13 – Déformation de l'oreille du personnage Ickis selon la vue $\theta = 0^\circ$ pour lui donner plus d'importance.

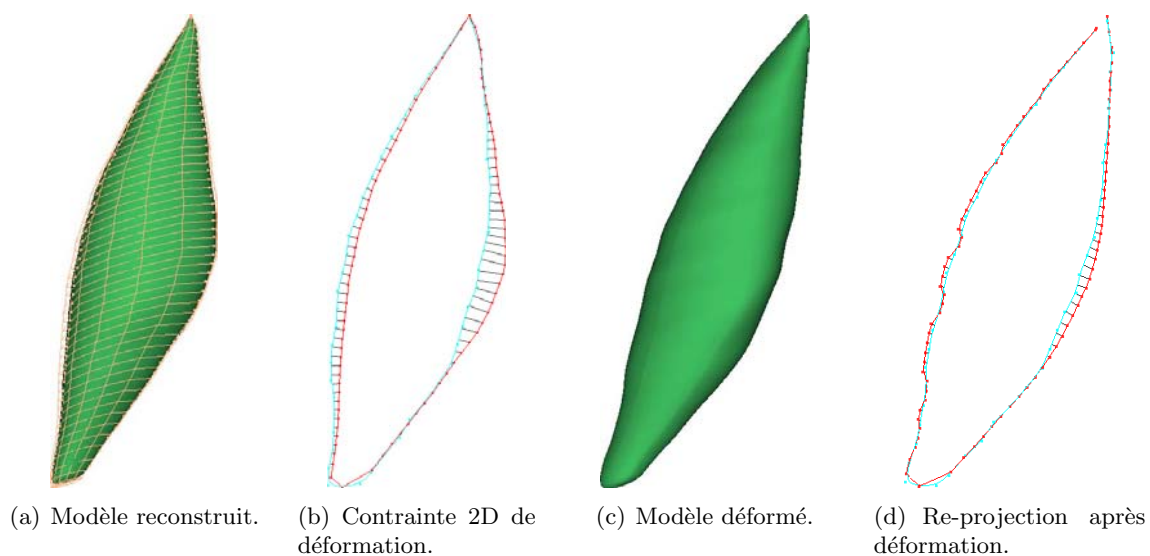


Figure 5.14 – Déformation de l'oreille du personnage Ickis selon la vue $\theta = 0^\circ$ pour l'amincir.

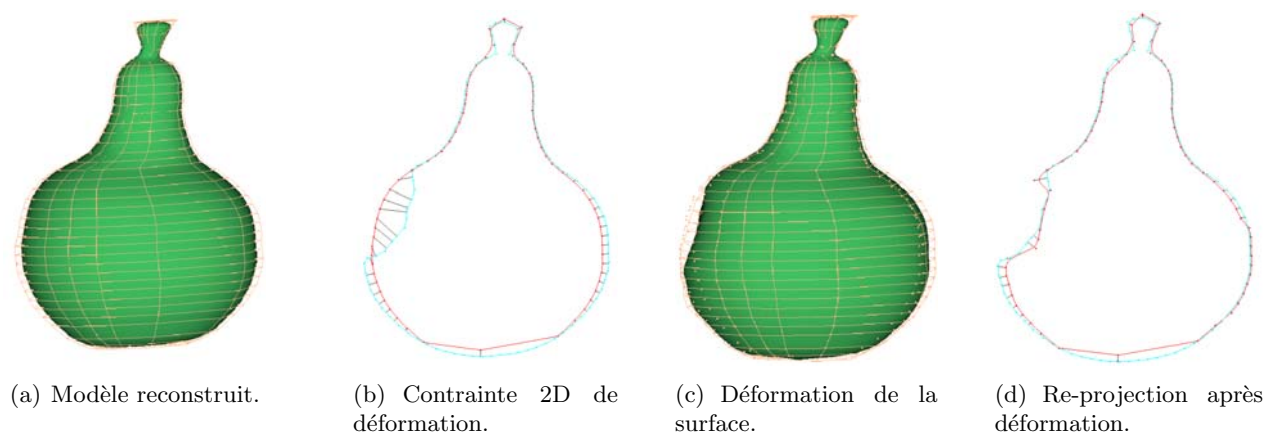


Figure 5.15 – Déformation de la poire (croquée) selon la vue $\theta = 0^\circ$.

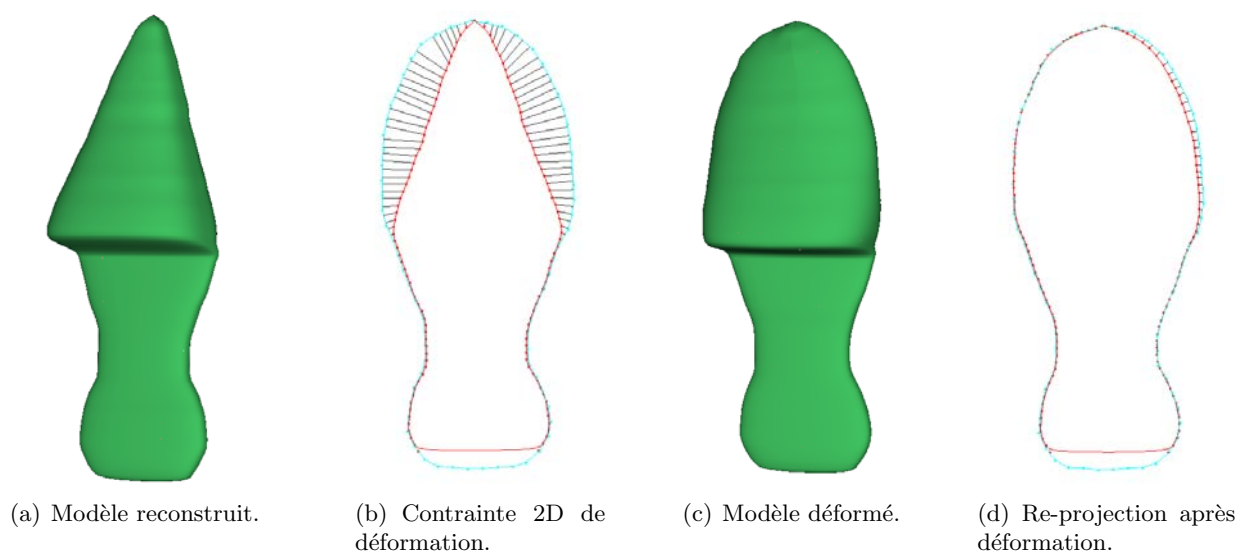


Figure 5.16 – Déformation d'une chaussure selon la vue $\theta = 90^\circ$.

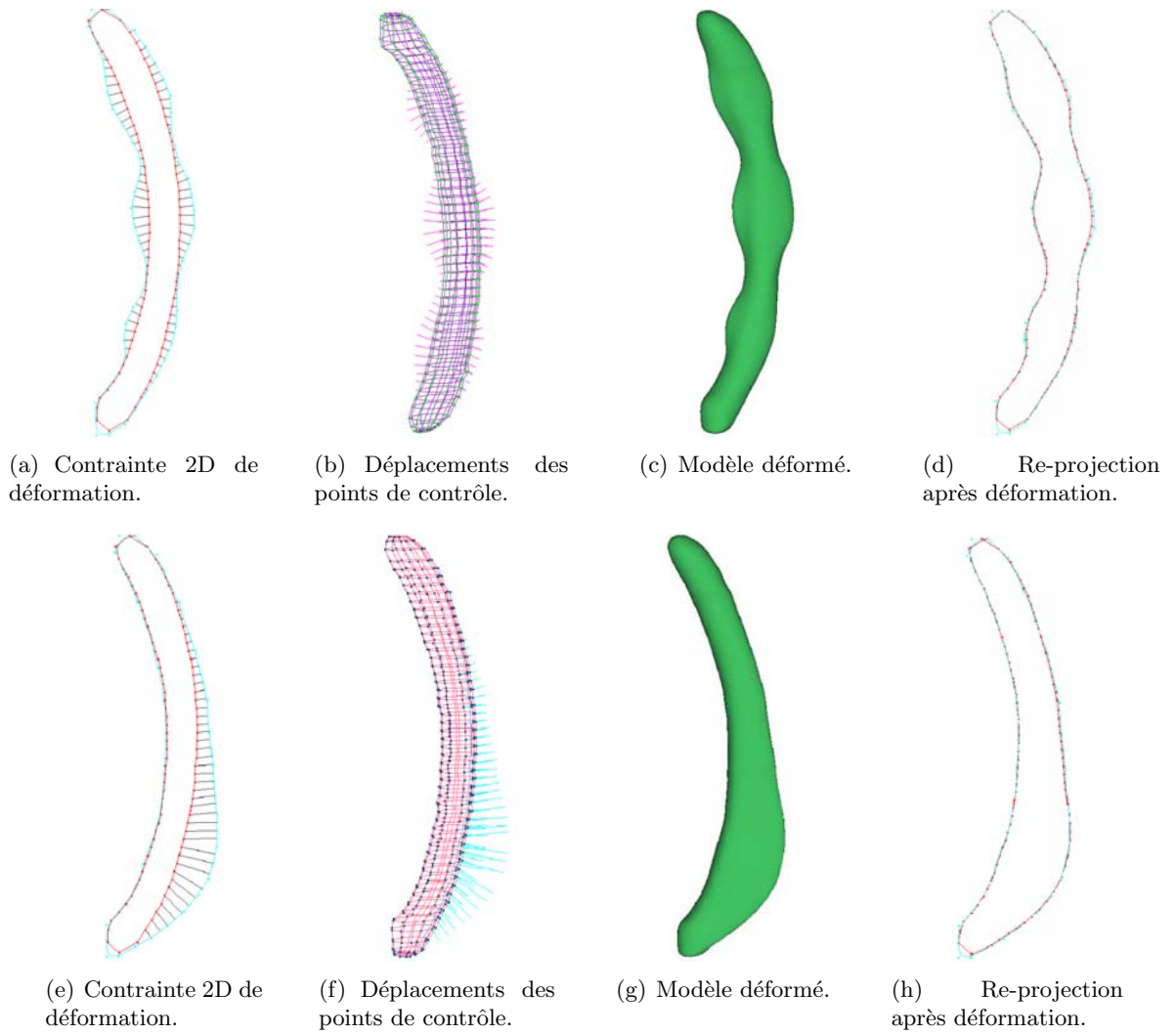


Figure 5.17 – Déformation du corps du personnage Oblina selon la vue $\theta = 90^\circ$.

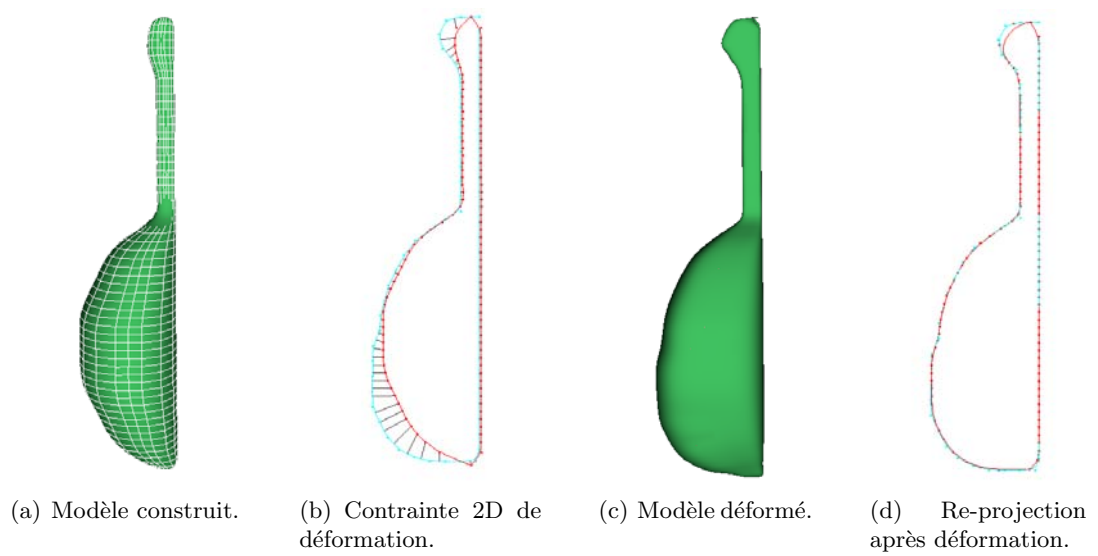


Figure 5.18 – Déformation d'une mandoline en un luth selon la vue $\theta = 90^\circ$.

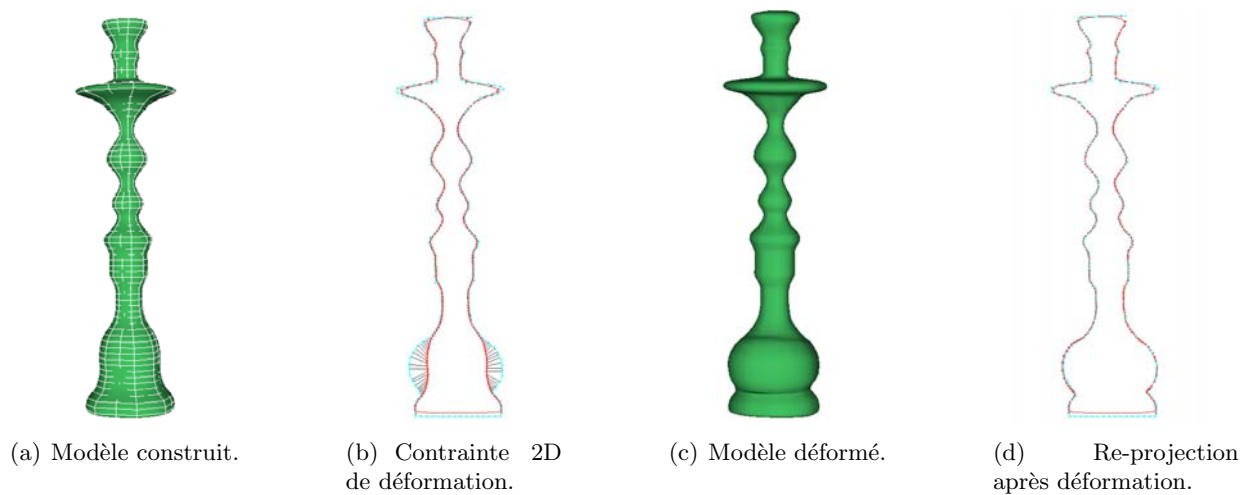


Figure 5.19 – Déformation d'un narguile selon les vues 0 45, 90° et 135.

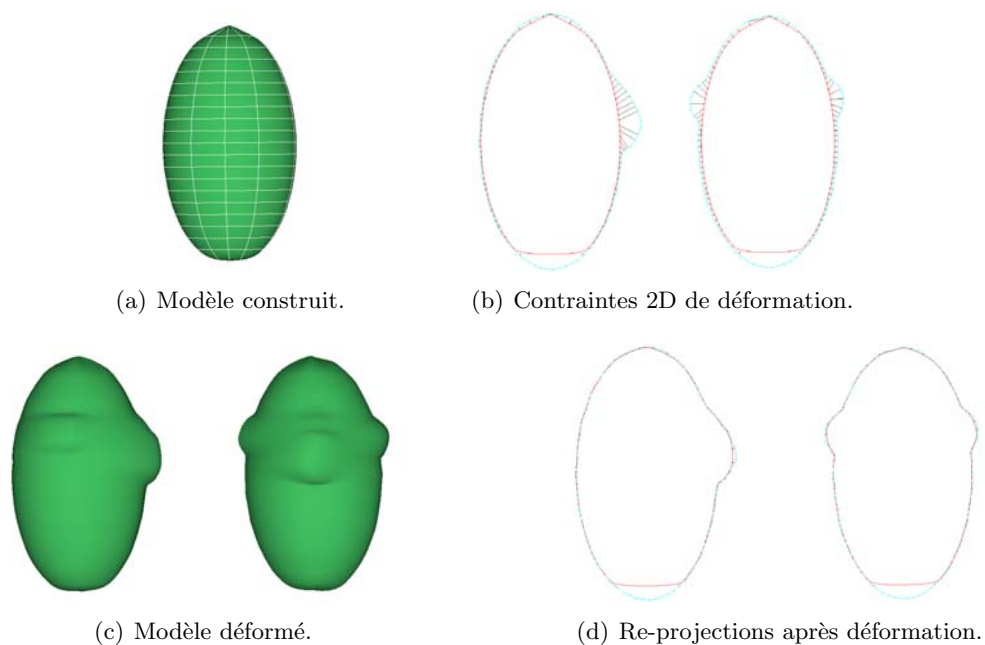


Figure 5.20 – Déformation d'une ellipsoïde en tête selon les vues 0 et 90°.

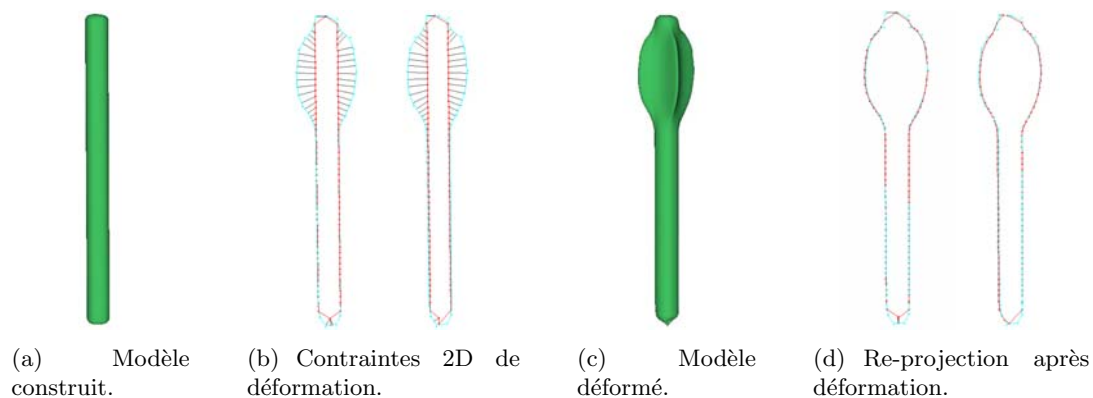


Figure 5.21 – Construction d'une flèche à l'aide de déformations selon les vues 0 et 90°.

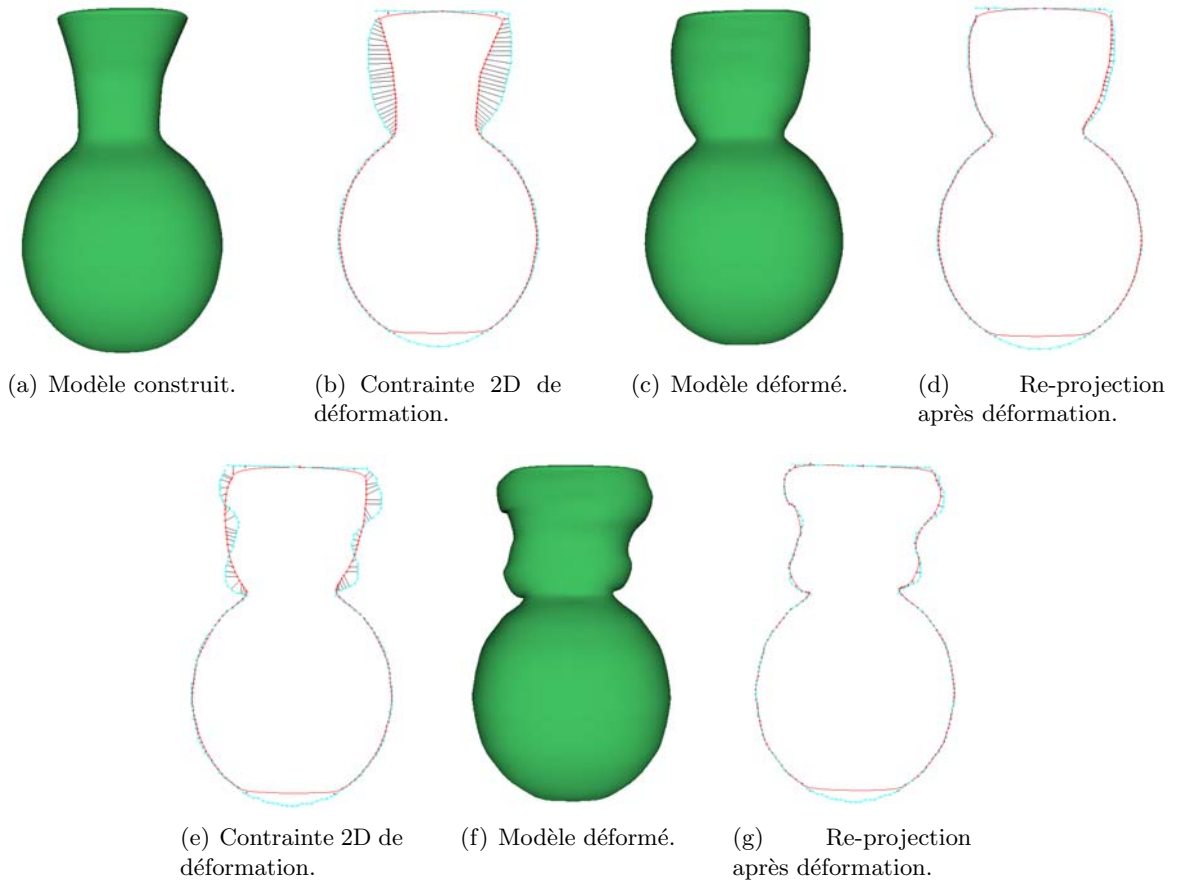


Figure 5.22 – Déformations d'un vase selon la vue $\theta = 90^\circ$: résultats obtenus pour un modèle reconstruit sans subdivision de Chakin et avec 3 niveaux de subdivision.

Notons qu'il est possible d'étendre la procédure de calcul des contraintes de manière à réaliser des déformations plus importantes. Pour cela, il suffit de définir des déformations locales où seul un segment de la courbe de projection est redessiné. Le segment de courbe de contrainte remplacerait alors le segment de courbe de projection et les contraintes 2D correspondraient alors à des couples de points de même coordonnée curviligne (Figure 5.23).

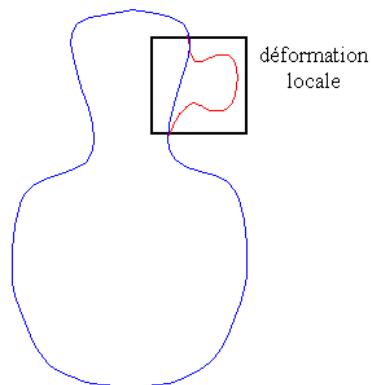


Figure 5.23 – Procédure de déformation locale.

Enfin, la figure 5.24 montre une comparaison de la déformation obtenue pour le modèle du vase à partir du même contour de contrainte 2D, appliquée à des modèles reconstruits par approximation avec différents paramètres de subdivision. Cela met en évidence l'impact du niveau de subdivision du maillage de contrôle sur la déformation obtenue : plus ce maillage est fin, plus les déformations sont locales et plus il est grossier, plus les déformations sont globales.

Cette propriété pourrait être exploitée pour introduire une représentation avec différents niveaux de détail du maillage de contrôle. Cela permettrait de générer des déformations sur différents niveaux de résolution. Dans ce schéma, le niveau de subdivision du maillage de contrôle auquel on appliquerait la déformation serait alors en fonction d'une zone d'influence spécifiée par l'utilisateur. Cela permettrait alors de réaliser des déformations de différentes amplitudes sur un même modèle.

De manière générale, les résultats obtenus montrent les possibilités d'application de ce module de déformation pour :

- obtenir une plus grande variété de formes à partir d'un même modèle de base : les dessinateurs peuvent construire une bibliothèque de formes (tête, meuble, arbre, ...) et les utiliser comme base de création de nouveaux personnages ou éléments de décor. Cette bibliothèque s'avère en particulier utile pour représenter plusieurs éléments dans une même scène tout en introduisant une faible variabilité (une forêt ou une foule de personnes par exemple).
- animer un personnage qui change de forme dans une séquence donnée : les déformations telles que l'exagération et l'aplatissement sont très souvent utilisées dans le cadre des dessins animés [Las87]. Réaliser ces déformations à partir d'une reprojection du modèle 3D allège le processus d'animation puisque l'animateur ne doit plus redessiner le personnage en entier mais seulement spécifier les endroits déformés.
- modéliser de manière incrémentale un personnage ou une forme complexe qui ne peut être spécifiée par deux vues de projections uniquement : l'utilisation des reprojections selon de nouveaux angles permet alors d'accélérer le processus de création du personnage. En outre, il devient possible d'ajouter des détails en exploitant la finesse du maillage de contrôle subdivisé (exemples de la tête figure 5.20 et de la flèche figure 5.21).

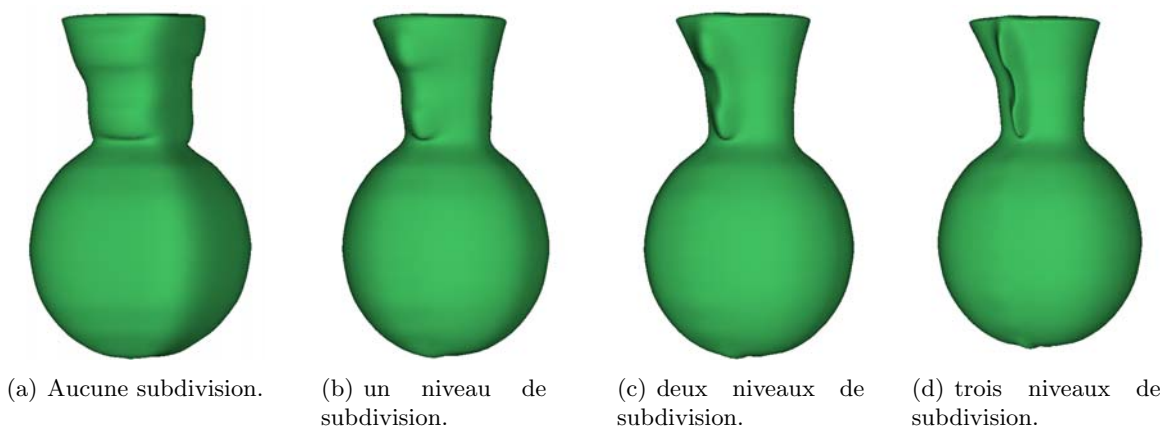


Figure 5.24 – Déformation du vase pour différents niveaux de subdivision de reconstruction.

5.3 Conclusion

Ce chapitre a présenté une approche de déformation de modèles NURBS 3D sous deux types de contraintes : l'interpolation d'un ensemble de points 3D de coordonnées connues et

la correspondance à des contours projectifs 2D modifiés ou nouveaux. Notons en particulier que la procédure de déformation sous contraintes 2D permet d'alléger le travail du dessinateur qui peut alors utiliser les contours de projection obtenus à partir d'une reconstruction avec deux vues uniquement. Il suffit ensuite de modifier localement les contours projetés selon de nouveaux angles de vue sans avoir à redessiner le personnage en entier. Les méthodes présentées ainsi que les résultats obtenus démontrent la faisabilité et l'efficacité du processus de reconstruction 2D/3D à l'aide de surfaces NURBS, non seulement pour un objectif de re-projection et de visualisation uniquement 2D, mais aussi dans le cadre d'applications 3D où l'interaction 2D reste plus intuitive.

Enfin, nous avons mis en évidence l'intérêt d'une représentation multi-résolution dans le cadre de la modélisation par NURBS. En particulier, un tel mode de représentation permettrait de réaliser des déformations en fonction d'une zone d'influence spécifiée par l'utilisateur. Cela permettrait de s'affranchir de la dépendance des déformations vis-à-vis de la finesse du maillage de contrôle et de réaliser ainsi des déformations de différentes amplitudes sur une même surface NURBS.

Conclusion et perspectives

Les systèmes de reconstruction interactifs exploitant des dessins manuels représentent un domaine de recherche très récent pour des applications telles que le *design*, les productions d'animation, ou la modélisation et simulation scientifique.

Nos recherches s'inscrivent dans ce contexte de reconstruction de formes libres. Plus précisément, elles concernent la production de dessins animés 2D. Ce domaine qui a conservé un mode de travail traditionnel très faiblement automatisé contraste avec celui des films 3D bénéficiant de l'essor des technologies 3D.

Afin d'analyser les raisons du fossé qui existe entre les productions d'animation 2D et 3D, le premier chapitre présente une étude comparée des deux chaînes de fabrication. Il en ressort un manque d'outils permettant aux animateurs 2D d'exploiter les techniques 3D tout en conservant leur mode de travail, notamment *via* une interactivité exclusivement 2D.

Les deuxième et troisième chapitres brossent un état de l'art des modèles de représentation de surfaces 3D, ainsi que des techniques d'animation 3D. Cette étude synthétique et critique nous a conduit à retenir les surfaces maillées pour représentation finale, même si nous avons exploité la modélisation par NURBS pour reconstruire en 3D un objet en l'associant à un module de déformation sous contraintes 2D. En terme d'animation, nous avons opté pour l'approche descriptive, par squelette morphologique, qui permet de réaliser des animations par pose clef, de manière cohérente avec les animations traditionnelles 2D. Ces choix, compatibles avec la norme MPEG-4/AFX, permettent de se placer dans un contexte multi-plateformes où les équipes d'animateurs sont délocalisées et pour lequel la conformité à une norme facilite les échanges de données et garantit l'interopérabilité des applications.

Nos contributions ont donc porté sur le développement :

- d'un module de reconstruction 2D/3D (*cf.* Chapitre 4), intégrant plusieurs approches avec ou sans modélisation intermédiaire par surfaces NURBS,
- d'un module de déformation de surfaces NURBS 3D sous contraintes de projection 2D (*cf.* Chapitre 5).

Bien que structuralement et conceptuellement différentes, les différentes approches de reconstruction développées exploitent une première approximation volumique de l'objet définie comme l'enveloppe visuelle et obtenue à partir des dessins du *turnaround* à l'aide d'une procédure d'intersection de volumes. Afin d'intégrer dans le processus un maximum d'information, la reconstruction de l'objet est réalisée par parties. Cela implique une segmentation des images réalisée par les dessinateurs selon un protocole d'acquisition multi-couches mis au point dans le cadre du projet TOON.

Une première catégorie d'approches de reconstruction exploite une modélisation intermédiaire par surfaces NURBS. Ici, chaque partie de l'objet est approchée par une surface NURBS lisse et adaptée à des objectifs de manipulation interactive et de contrôle local. La surface est déterminée en employant une approche de construction par sections. Dans ce cadre, deux stratégies différentes ont été adoptées et développées :

- La première méthode consiste à approcher chaque coupe horizontale de l'enveloppe visuelle par une courbe NURBS, puis à interpoler l'ensemble des courbes ainsi définies par une surface NURBS,
- La deuxième technique introduit tout d'abord une étape de subdivision de Chaikin qui

visé à lisser les coupes horizontales de l'enveloppe visuelle. Une procédure d'insertion de sommets est ensuite utilisée afin d'homogénéiser le nombre des sommets pour l'ensemble des coupes. Une surface NURBS est alors définie dont le maillage de contrôle est constitué par l'ensemble des sommets ainsi obtenus.

Dans les deux cas, les différentes surfaces NURBS d'un même modèle sont ensuite triangulées, puis assemblées en un unique maillage sans couture (*seamless*). Des surfaces de *blending* entre les différentes parties de l'objet sont alors générées de manière à garantir des transitions lisses entre les éléments assemblés. Pour déterminer les différentes parties à assembler, la procédure d'assemblage exploite conjointement les relations d'adjacence entre les composantes de l'objet dans les images du *turnaround* et un squelette d'animation spécifié par l'utilisateur.

La principale difficulté de ce type d'approches est liée notamment à la phase d'assemblage des parties qui nécessite la mise en œuvre de techniques de *blending* et de re-triangulation lourdes en temps de calcul.

Une deuxième approche plus directe de reconstruction 2D/3D est alors proposée. Le principe consiste à abandonner la phase de modélisation NURBS et à générer directement un maillage *seamless* à partir du modèle volumique. Une procédure de subdivision de Chaïkin, réalisée indépendamment pour chaque coupe garantit d'obtenir des modèles lisses. Le maillage final est construit en projetant chaque paire de coupe successives dans un même plan et en appliquant une triangulation de Delaunay 2D avec contraintes.

Pour offrir la possibilité de corriger/raffiner les modèles 3D obtenues, la méthode de reconstruction par modélisation NURBS a été enrichie par un module de déformation 2D/3D qui permet d'ajuster/raffiner interactivement la surface reconstruite à l'aide d'un ensemble de contraintes de projection que la surface doit interpoler. Ces contraintes sont définies comme un ensemble de courbes 2D spécifiées par l'utilisateur sur les images de projection. Un mécanisme de déformation par minimisation d'énergie est alors mis en œuvre pour garantir que le contour apparent de la surface approche le contour cible.

Les méthodes développées ont été validées sur un grand nombre de modèles de personnages virtuels de complexités et de formes différentes, humanoïdes ou non. En conclusion, après avoir mis en évidence la nécessité de créer des outils visant à automatiser la chaîne de fabrication des dessins animés 2D tout en restant conformes aux pratiques traditionnelles de travail des dessinateurs professionnels, nous avons proposé un ensemble de méthodes de reconstruction, modélisation et déformation 2D/3D de personnages virtuels. Le caractère opérationnel des outils développés a été démontré dans le cadre d'une application industrielle grandeur réelle, liée au projet TOON. Leur intégration dans un premier prototype de la plate-forme TOON montre un gain en temps de fabrication de 20% sur l'ensemble de la chaîne de production. En outre, cela permet de créer une passerelle entre deux types d'animation, 2D et 3D, jusqu'à présent indépendants, tout en garantissant une complète interopérabilité des applications via le standard MPEG-4.

Les perspectives de recherche sont multiples et concernent principalement :

- La reconstruction par surfaces de subdivision : la méthode de reconstruction directe de maillages pourrait être modifiée afin de générer des maillages adaptés à la subdivision. Cela permettrait de disposer de différents niveaux de détail pour les modèles reconstruits.
- L'amélioration du module de déformation des surfaces NURBS : les déformations réalisées dépendent fortement des degrés et du maillage de contrôle de la surface. En particulier, il serait intéressant d'étudier la possibilité d'introduire la notion d'une *zone d'influence* paramétrable et indépendante du modèle de surface à déformer.
- Le développement d'un outil de déformation de surfaces maillées sous contraintes 2D, de manière similaire à celui réalisé pour les surfaces NURBS.
- La généralisation des déformations à des contours non générateurs, de manière à pouvoir exploiter les éléments de dessin représentant souvent des creux ou des plis qui nécessitent la mise en œuvre de techniques de déformation différentes.

Bibliographie

- [ACWK04] A. Angelidis, M.P. Cani, G. Wyvill, and S. King. Swirling-sweepers : Constant volume modeling. In *Pacific Graphics'2004*, pages 10–15, Korea, oct 2004.
- [AGB04] A. Alexe, V. Gaildrat, and L. Barthe. Interactive modelling from sketches using spherical implicit functions. In *AFRIGRAPH '04 : Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 25–34, Stellenbosch, South Africa, 2004. ACM Press.
- [AJ03] B. De Araujo and J. Jorge. Blobmaker : Free-form modelling with variational implicit surfaces. In *12th Portuguese Computer Graphics Meeting*, pages 17–26, Porto, Portugal, October 2003.
- [Ale02] M. Alexa. Recent advances in mesh morphing. In *Computer Graphics Forum*, volume 21, pages 173–196, June 2002.
- [Bae01] P. Baerlocher. *Inverse kinematics techniques of the interactive posture control of articulated figures*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Lasanne, Suisse, 2001.
- [Baj94] C. Bajaj. Polynomial surface patch representations. Technical Report 94-038, Purdue University, Computer Sciences, 1994.
- [Bar84] A.H. Barr. Global and local deformations of solid primitives. In *SIGGRAPH Computer Graphics*, volume 18, pages 21–30, 1984.
- [BCCD04] D. Bourguignon, M.P. Cani, R. Chaine, and G. Drettakis. Relief : A modeling by drawing tool. In *First Eurographics Workshop on Sketch-Based Interfaces and Modeling (SBM'04)*, pages 151–160, Grenoble, France, September 2004.
- [BCD01] D. Bourguignon, M.P. Cani, and G. Drettakis. Drawing for illustration and annotation in 3d. In *Eurographics Computer Graphics Forum*, volume 20, pages 114–122, September 2001.
- [BGA04] A. Barbier, E. Galin, and S. Akkouche. A framework for modeling, animating and morphing taxtures implicit models. Technical report, LIRIS, CNRS, Université Claude Bernard Lyon I, 2004.
- [BGK03] A. Balázs, M. Guthe, and R. Klein. Fat borders : Gap filling for efficient view-dependent lod rendering. Technical Report CG-2003-2, Universität Bonn, Germany, June 2003.
- [Bir] J. Birn. www.3drender.com. Tutorial : NURBS Head Modeling.
- [BK97] Gill Barequet and Subodh Kumar. Repairing cad models. In *VIS '97 : Proceedings of the 8th conference on Visualization '97*, page 363, Phoenix, Arizona, United States, 1997. IEEE Computer Society Press.
- [Bli82] J.F. Blinn. A generelization of algebraic surface drawing. In *ACM Transactions on Graphics*, volume 1, pages 235–256, 1982.
- [BNK02] P. Borodin, M. Novotni, and R. Klein. Progressive gap closing for mesh repairing. In *Advances in Modelling, Animation and Rendering*, pages 201–213. Springer Verlag, 2002.

- [Boo87] C. De Boor. Cutting corners always works. In *Computer Aided Geometric Design*, volume 4, pages 125–132, 1987.
- [BS91] J. Bloomenthal and K. Shoemake. Convolution surfaces. In *Computer Graphics*, volume 25, pages 251–255, 1991.
- [BW76] N. Burtnyk and M. Wein. Interactive skeleton techniques for enhancing motion dynamics in key frame animation. In *Proceedings of the 2nd annual conference on Computer graphics and interactive techniques*, volume 19, pages 78–80, Bowling Green, Ohio, June 1976.
- [BW97] D. Baraff and A. Witkin. Physically based modeling : principles and practice. Siggraph course notes, 1997.
- [BWW94] B. Boufama, D. Weinshall, and M. Werman. Shape from motion algorithms : a comparative analysis of scaled orthography and perspective. In *Proceedings of the 3rd European Conference on Computer Vision*, pages 199–204, Stockholm, Sweden, May 1994.
- [Cas63] P. De Casteljaeu. Courbes et surfaces à pôles. Technical report, André Citroën Automobiles, S. A., Paris, 1963.
- [Cas87] M.S. Casale. Free-form solid modeling with trimmed surface patches. In *IEEE Computer Graphics and Applications*, volume 7, pages 33–43, 1987.
- [CC78] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. In *Computer-Aided Design*, volume 10, pages 350–355, 1978.
- [Cha74] G.M. Chaikin. An algorithm for heigh speed curve generation. In *Computer Graphics and Image Processing*, volume 3, pages 346–349, 1974.
- [CHP89] J.E. Chadwick, D.R. Haumann, and R.E. Parent. Layered construction for deformable animated characters. In *SIGGRAPH '89 : Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 243–252, New York, NY, USA, 1989. ACM Press.
- [CJ89] B.K. Choi and S.Y. Ju. Constant radius blending in surface modelling. In *Computer Aided Design*, volume 21, pages 213–220, 1989.
- [CKB04] P. Chaudhuri, P. Kalra, and S. Banerjee. A system for view dependent animation. volume 23, pages 411–420. Blackwell Publishers, 2004.
- [Coq90] S. Coquillart. Extended free-form deformation : a sculpturing tool for 3d geometric modeling. In *SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 187–196, Dallas, TX, USA, 1990. ACM Press.
- [Cot01] O. Cotte. *Il était une fois le dessin animé et le cinéma d'animation*. Paris : Dreamland, 2001.
- [CSSJ05] J. J. Cherlin, F. Samavati, M. C. Sousa, and J. A. Jorge. Sketch-based modeling with few strokes. In *21st Spring Conference on Computer Graphics*, Budmerice, Slovak Republic, 2005.
- [DKL98] E. Dam, M. Koch, and M. Lillholm. Quaternions interpolation and animation. Technical Report DIKU 98/5, University of Copenhagen, Denmark, 1998.
- [DM95] A. Dolenc and I. Malenka. Optimised triangulation of parametric surfaces : The mathematics of surfaces. In Adrian Bowyer, editor, *Computer Aided Surface Geometry and Design*, volume 4, pages 169–183, 1995.
- [Doo78] D. Doo. A subdivision algorithm for smoothing down irregularly shaped polyhedrons. In *Int. Conf. Ineractive Techniques in Computer Aided Design, Bologna, Italy, IEEE Computer Soc*, pages 157–165, 1978.
- [Dov00] B. Dovnikovic. *La technique du dessin animé*. Paris : Dreamland, 2000.

- [DS78] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. In *Computer-Aided Design*, volume 10, pages 356–360, 1978.
- [EHBE97] L. Eggli, C. Hsu, B. Bauderlin, and G. Elber. Inferring 3d models from freehand sketches and constraints. In *Computer Aided Design*, volume 29, pages 101–112, 1997.
- [Eva03] T. Evans. Dead or alive : is 2d animation still wanted ? Technical report, CUSTODIEV Market Watch, 2003.
- [Far96] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press Professional, Inc., San Diego, CA, USA, 1996.
- [FBC⁺95] J.D. Fekete, E. Bizouarn, E. Cournarie, T. Galas, and F. Tallefer. Tictactoon : a paperless system for professional 2d animation. In *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 79–90, New York, NY, USA, 1995. ACM Press.
- [FF01] N. Foster and R. Fedkiw. Practical animation of liquids. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23–30, New York, NY, USA, 2001. ACM Press.
- [Fil89] D.J. Filip. Blending parametric surfaces. *ACM Trans. Graph.*, 8(3) :164–173, 1989.
- [FK90] D.R. Forsey and R.V. Klassen. An adaptive subdivision algorithm for crack prevention in the display of parametric surfaces. In *Proceedings of Graphics interface '90*, pages 1–8, Halifax, Nova Scotia, 1990.
- [fla] <http://www.macromedia.com/>.
- [Flo97] M.S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3) :231–250, 1997.
- [FR02] F. Di Fiore and F. Van Reeth. Employing approximate 3d models to enrich traditional computer assisted animation. In *Computer Animation*, pages 183–192, Geneva, Switzerland, 2002. IEEE Computer Society.
- [FSER01] F. Di Fiore, P. Schaeken, K. Elens, and F. Van Reeth. Automatic inbetweening in computer assisted animation by exploiting 2.5d modelling techniques. In *Computer Animation CA'01*, pages 192–200, Seoul, Korea, 2001.
- [Fuc03] P. Fuchs. *Le traité de la réalité virtuelle*. Presses de l'École des Mines, 2003.
- [FvDFH97] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics : Principles and Practice*. Addison Wesley, 1997.
- [FvdPT97] P. Faloutsos, M. van de Panne, and D. Terzopoulos. Dynamic free form deformations for animation synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 3(3) :201–214, 1997.
- [Gar93] J. Garding. Direct estimation of shape from texture. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 15, pages 1202–1208, 1993.
- [GGS03] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3d meshes. *ACM Trans. Graph.*, 22(3) :358–363, 2003.
- [Gla89] A. Glassner. *An Introduction to Ray Tracing*. Academic Press Ltd., London, UK, 1989.
- [Gle00] M. Gleicher. Animation from observation : Motion capture and motion editing. *SIGGRAPH Comput. Graph.*, 33(4) :51–54, 2000.
- [Gén03] B. Génin. *Cinéma d'animation : dessin animé, marionnettes, images de synthèse*. Paris : Cahiers du cinéma, 2003.

- [Gri99] C. Grimm. Implicit generalized cylinders using profile curves. In *Proceedings of Implicit Surfaces'99*, pages 33–41, Bordeaux, France, 1999. Eurographics Association.
- [HAC03] S. Hornus, A. Angelidis, and M.P. Cani. Implicit modeling using subdivision curves. In *The Visual Computer*, volume 19, pages 94–104, 2003.
- [HAN] <http://h-anim.org/>.
- [HDD⁺94] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 295–302, New York, NY, USA, 1994. ACM Press.
- [HL94] S.C. Hsu and I.H.H. Lee. Drawing and animation using skeletal strokes. In *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 109–118, New York, NY, USA, 1994. ACM Press.
- [HLZ01] S.M. Hu, Y.F. Li, and J.X. Zhu. Modifying the shape of nurbs surfaces with geometric constraints. In *Computer Aided Design*, volume 33, pages 903–912, 2001.
- [HNC02] D. Hinsinger, F. Neyret, and M.P. Cani. Interactive animation of ocean waves. In *SCA '02 : Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 161–166, San Antonio, Texas, 2002. ACM Press.
- [HQ04] J. Hua and H. Qin. Haptics-based dynamic implicit solid modeling. *IEEE Transactions on Visualization and Computer Graphics*, 10(5) :574–586, 2004.
- [IITS04] T. Ijiri, T. Igarashi, S. Takahashi, and E. Shibayama. Sketch interface for 3d modeling of flowers. Technical Sketch at ACM SIGGRAPH, August 2004.
- [IMT99] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy : a sketching interface for 3d freeform design. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [JLD82] J.P. J.M. Lo Duca. *Le Dessin animé, histoire, esthétique, technique*. Paris : Prisma, 1982.
- [Kar01] O. Karpenko. Interactively generating 3d models from contour drawings. Master's thesis, Brown University, Providence, Rhode Island, USA, 2001.
- [KBK02] F. Kahlesz, A. Balazs, and R. Klein. Multiresolution rendering by sewing trimmed nurbs surfaces. In *SMA '02 : Proceedings of the seventh ACM symposium on Solid modeling and applications*, pages 281–288, Saarbrücken, Germany, 2002. ACM Press.
- [KE95] K. Kim and G. Elber. A symbolic approach to freeform surface blends. *Journal of Visualization and Computer Animation*, 8 :69–80, 1995.
- [Ker04a] I.v. Kerlow. *The art of 3D computer animation and effects*. Wiley, Hoboken, New Jersey, 2004.
- [Ker04b] I.v. Kerlow. *The art of 3D computer animation and effects, chpitre 11*. Wiley, Hoboken, New Jersey, 2004.
- [KHR02] O. Karpenko, J.F. Hughes, and R. Raskar. Free-form sketching with variational implicit surfaces. *Computer Graphics Forum*, 21(3), 2002.
- [KHR04] O. Karpenko, J.F. Hughes, and R. Raskar. Epipolar methods for multi-view sketching. In *Eurographics Workshop on Sketch-Based Interfaces*, pages 167–174, Grenoble, France, 2004.
- [Kis02] S. Kiss. Computer animation for articulated 3d characters. Technical Report TR-CTIT-02-45, University of Twente, The Netherlands, 2002.

- [KM95] S. Kumar and D. Manocha. Efficient rendering of trimmed nurbs surfaces. In *Computer-Aided Design*, volume 27, pages 509–521, 1995.
- [Kor86] J.U. Korein. *A geometric investigation of reach*. MIT Press, Cambridge, MA, USA, 1986.
- [Kor02] A. Kort. Computer aided inbetweening. In *NPAR '02 : Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 125–132, Annecy, France, 2002. ACM Press.
- [KPA03] M. Kazakov, A. Pasko, and V. Adzhiev. Interactive metamorphosis and carving in a multi-volume scene. In *GRAPHITE '03 : Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 103–ff, Melbourne, Australia, 2003. ACM Press.
- [Kre91] E. Kreyszig. *Differential Geometry*. Dover Publications, New York, 1991.
- [KS95] R. Klein and W. Straber. Large mesh generation from boundary models with parametric face representation. In *SMA '95 : Proceedings of the third ACM symposium on Solid modeling and applications*, pages 431–440, Salt Lake City, Utah, United States, 1995. ACM Press.
- [Kuk04] Z. Kukelova. A user interface for freeform modeling based on convolution surfaces from sketched silhouette curves. In *Central European Seminar on Computer Graphics, Budmerice, SR*, 2004.
- [Las87] J. Lasseter. Principles of traditional animation applied to 3d computer animation. In *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 35–44, New York, NY, USA, 1987. ACM Press.
- [Lau99] A. Laurentini. The visual hull of curved objects. In *International Conference on Computer Vision*, pages 356–361, Kerkyra, Greece, 1999.
- [LC87] W.E. Lorensen and H.E. Cline. Marching cubes : A high resolution 3d surface construction algorithm. In *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, 1987. ACM Press.
- [LCJ94] F. Lazarus, S. Coquillart, and P. Jancène. Axial deformations : an intuitive deformation technique. *Computer-Aided Design*, 26(8) :607–613, 1994.
- [LCR⁺02] J. Lee, J. Chai, P.S.A. Reitsma, J.K. Hodgins, and N.S. Pollard. Interactive control of avatars animated with human motion data. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 491–500, San Antonio, Texas, 2002. ACM Press.
- [LGXS03] Y. Li, M. Gleicher, Y.Q. Xu, and H.Y. Shum. Stylizing motion with drawings. In *SCA '03 : Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 309–319, San Diego, California, 2003. Eurographics Association.
- [LKG⁺03] R. Lau, T. Kunii, B. Guo, N. Magnenat-Thalmann, D. Thalmann, F. Li, B. Zhang, S. Kshirsagar, and M. Gutierrez. Emerging web graphics standards and technologies. volume 23, pages 66–75, 2003.
- [Loo87] C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of mathematics, 1987.
- [lor] <http://www.lordoftherings.net/>.
- [LV98] F. Lazarus and A. Verroust. 3d metamorphosis : A survey. *The Visual Computer*, 14 :373–389, 1998.

- [LW04] S.H. Lo and W.X. Wang. A fast robust algorithm for intersection of triangulated surfaces. In *Engineering with Computers*, volume 20, pages 11–21, 2004.
- [MAX] www4.discreet.com/3dsmax/.
- [MAY] www.aliaswavefront.com/.
- [mel] <http://www.sli.unimelb.edu.au/>.
- [MFCD99] F. Multon, L. France, M.P. Cani, and G. Debunne. Computer animation of human walking : a survey. *The Journal of Visualization and Computer Animation*, 10 :39–54, 1999.
- [MH03] Y.L. Ma and W.T. Hewitt. Point inversion and projection for nurbs curve and surface : control polygon approach. *Comput. Aided Geom. Des.*, 20(2) :79–99, 2003.
- [MPE] <http://www.mpeg.org/>.
- [MR97] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *International Journal of Computer Vision*, 23 :149–168, 1997.
- [MSK02] J. Mitani, H. Suzuki, and F. Kimura. 3d sketch : sketch-based model reconstruction and rendering. *From geometric modeling to shape modeling*, pages 85–98, 2002.
- [MTSC04] K. Melikhov, F. Tian, H.S. Seath, and Q. Chen. Frame skeleton based auto inbetweening in computer aided cel animation. In *International Conference on Cyberworlds*, pages 216–223, Tokyo, Japan, 2004.
- [NHK⁺85] H. Nishimura, M. Hirai, T. Kawai, T. Kawata, I. Shirakawa, and K. Omura. Object modeling by distribution function and a method of image generation. In *The Transactions of the Institute of Electronics and Communication Engineers of Japan*, volume J68-D, pages 718–725, 1985.
- [OCNF02] Y. Ono, B.Y. Chen, T. Nishita, and J. Feng. Free form deformation with automatically generated multiresolution lattices. In *IEEE Cyberworlds 2002 Conference Proceedings*, pages 472–479, Tokyo, Japan, 2002.
- [ONNI03] S. Owada, F. Nielsen, K. Nakazawa, and T. Igarashi. A sketching interface for modeling the internal structures of 3d shapes. In *Smart Graphics 2003, Lecture Notes in Computer Science (LNCS)*, volume 2733, pages 49–57, 2003.
- [Par03] R. Parent. *Animatique : Algorithmes et techniques*. Paris : Vuibert, 2003.
- [PC01] F. Perbet and M.P. Cani. Animating prairies in real-time. In *SI3D '01 : Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 103–110, New York, NY, USA, 2001. ACM Press.
- [peg] www.mediapegs.com.
- [Ple88] D. Pletincks. *The Use of Quaternions for Animation, Modelling and Rendering*. Springer, Heidelberg, 1988.
- [Pon98] M.T Poncet. *Dessin animé*. Paris : Voiron, 1998.
- [PP04] M. Preda and F. Prêteux. Virtual character within mpeg-4 animation framework extension. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(7) :975–988, July 2004.
- [Pre02] M. Preda. *Système d'animation d'objets virtuels : de la modélisation à la normalisation MPEG-4*. PhD thesis, Université Paris V - René Descartes, France, December 2002.
- [PT95] L. Piegl and W. Tiller. *The NURBS book*. Springer-Verlag, London, UK, 1995.

- [PT98] L.A. Piegl and W. Tiller. Geometry based triangulation of trimmed nurbs surfaces. In *Computer Aided Design*, volume 30, pages 11–18, 1998.
- [PTVF92] W. H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical recipes in C : art of scientific computing*. Cambridge University Press, 1992.
- [Rad99] P. Rademacher. View-dependent geometry. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 439–446, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [REA] <http://www.realsoft.com/>.
- [Ree81] W.T. Reeves. Inbetweening for computer animation utilizing moving point constraints. In *SIGGRAPH '81 : Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, pages 263–269, Dallas, Texas, United States, 1981. ACM Press.
- [Ree96] F. Van Reeth. Integrating 2.5d computer animation techniques for supporting traditional animation. In *Computer Animation '96*, pages 118–125. IEEE Computer Society Press, 1996.
- [Rey82] C.W. Reynolds. Computer animation with scripts and actors. In *SIGGRAPH '82 : Proceedings of the 9th annual conference on Computer graphics and interactive techniques*, pages 289–296, Boston, Massachusetts, United States, 1982. ACM Press.
- [Rey87] CW. Reynolds. Flocks, herds, and schools : a distributed behavioral model. In *SIGGRAPH 87, Computer Graphics*, volume 4, pages 25–34, 1987.
- [RGL05] S. Redon, N. Galoppo, and M.C. Lin. Adaptive dynamics of articulated bodies. *ACM Trans. Graph.*, 24(3) :936–945, 2005.
- [RHD89] A. Rockwood, K. Heaton, and T. Davis. Real-time rendering of trimmed surfaces. In *SIGGRAPH '89 : Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 107–116, New York, NY, USA, 1989. ACM Press.
- [Rie75] R. Riesenfeld. On chaikin's algorithm. In *IEEE Computer Graphics and Applications*, volume 4, pages 304–310, 1975.
- [RR84] J.R. Rossignac and A.A.G. Requicha. Constant radius blending in solid modeling. In *Computers in Mechanical Engineering*, volume 3, pages 65–73, 1984.
- [Sar90] R.F. Sarraga. Computer modeling of surfaces with arbitrary shapes. In *Computer Graphics and Applications*, pages 67–77. IEEE Computer Society Press, 1990.
- [SCMS01] G. Slabaugh, B. Culbertson, T. Malzbender, and R. Schafer. A survey of methods for volumetric scene reconstruction from photographs. In *the Joint IEEE TCVG and Eurographics Workshop (VolumeGraphics-01)*, pages 81–100, June 2001.
- [SF98] K. Singh and E. Fiume. Wires : a geometric deformation technique. In *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 405–414, New York, NY, USA, 1998. ACM Press.
- [SG92] T. W. Sederberg and E. Greenwood. A physically based approach to 2d shape blending. In *SIGGRAPH '92 : Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 25–34, New York, NY, USA, 1992. ACM Press.
- [SGWM93] T. W. Sederberg, P. Gao, G. Wang, and H. Mu. Shape blending : An intrinsic approach to the vertex path problem. In *Computer Graphics Conference Series*, pages 15–18, 1993.

- [She99] A. Sherstyuk. Interactive shape design with convolution surfaces. In *SMI '99 : Proceedings of the International Conference on Shape Modeling and Applications*, page 56, Washington, DC, USA, 1999. IEEE Computer Society.
- [Sho85] K. Shoemake. Animating rotation with quaternion curves. In *SIGGRAPH '85 : Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, New York, NY, USA, 1985. ACM Press.
- [Sli92] G. Slinker. Inbetweening using a physically based model and nonlinear path interpolation. Master's thesis, Brigham Young University, Computer Science Department, 1992.
- [SP86] T.W. Sederberg and S.R. Parry. Free-form deformation of solid geometric models. In *SIGGRAPH '86 : Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, volume 20, pages 151–160, New York, NY, USA, 1986. ACM Press.
- [SSBT01] T. Surazhsky, V. Surazhsky, G. Barequet, and A. Tal. Blending polygonal shapes with different topologies. *Computers & Graphics*, 25 :29–39, 2001.
- [Sug02] J. Sugier. Triangulation of nurbs surfaces through adaptive refinement. *MGEV*, 11(1) :37–52, 2002.
- [Tay96] R. Taylor. *The encyclopedia of animation techniques*. Philadelphia, PA : Running Press Book Publishers, 1996.
- [TDM99] O. Tolba, J. Dorsey, and L. McMillan. Sketching with projective 2d strokes. In *ACM symposium on User Interface Software and Technology*, pages 149–157, 1999.
- [TG95] N. Tsingos and M.P. Gascuel. Un modeleur interactif d'objets définis par des surfaces implicites. *Revue Internationale de CFAO et d'Informatique Graphique*, 10(4) :355–366, octobre 1995.
- [Tha96] D. Thalmann. Physical, behavioral and sensor-based animation. In *Graphicon 96, St Petersburg, Russia*, pages 214–221, 1996.
- [the] <http://www.digimation.com>.
- [Tho60] B. Thomas. *Walt Disney : l'art du dessin animé*. Simon et Schuster, New York, 1960.
- [TO99] G. Turk and J. O'Brian. Shape transformation using variational implicit surfaces. In *Proceedings of SIGGRAPH Computer Graphics*, pages 335–342, 1999.
- [tooa] <http://www.toonboom.com/>.
- [toob] www.toonz.com/.
- [TSC96] D. Thalmann, J. Shen, and E. Chauvineau. Fast human body deformations for animation and vr applications. In *Computer Graphics International*, pages 166–174. IEEE Computer Society Press, 1996.
- [TT96] N. Magnenat Thalmann and D. Thalmann. Computer animation. *ACM Comput. Surv.*, 28(1) :161–163, 1996.
- [Tur95] R. Turner. Leman : A system for constructing and animating layered elastic characters. pages 185–203, Leeds, England, June 1995. Academic Press, San Diego, CA.
- [TZF03] C.L. Tai, H.X. Zhang, and C.K. Fong. Prototype modeling from sketched silhouettes based on convolution surfaces. In *Computer Graphics Forum*, 2003.
- [Vic01] L. Vicci. Quaternions and rotations in 3-space : The algebra and its geometric interpretation. Technical report, Chapel Hill, NC, USA, 2001.

- [VRM] The virtual reality modeling language. International Standard ISO/IEC 14772-1 :1997.
- [WMW86] G. Wyvill, C. McPheeters, and B. Wyvill. Data structure for soft objects. In *The visual computer*, volume 2, pages 227–234, 1986.
- [WW] A. Watt and M. Watt. *Advanced Rendering and Animation Techniques : Theory and Practice*. Reading, MA.
- [WW98] K. Weihe and T. Willhalm. Why cad data repair requires discrete algorithmic techniques. In *Proceedings of 2nd International Workshop, WAE '92, Algorithm Engineering*, pages 1–12, Saarbrücken, Germany, August 1998.
- [XSI] <http://www.softimage.com/products/xsi/>.
- [YBS02] S. Yoshizawa, A.G. Belyaev, and H.P. Seidel. A simple approach to interactive free-form shape deformations. In *10 th Pacific Conference on Computer Graphics and Applications*, pages 471–474, Beijing, China, 2002. IEEE Computer Society Press.
- [YS03] A.J. Yezzi and S. Soatto. Structure from motion for scenes without features. In *CVPR'03*, volume 1, pages 171–178, 2003.
- [ZG98] S. Zelinka and M. Garland. Wires : a geometric deformation technique. In *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 405–414, New York, NY, USA, 1998. ACM Press.
- [ZH02] V.B. Zordan and J.K. Hodgins. Motion capture-driven simulations that hit and react. In *SCA 02 : Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 89–96, San Antonio, Texas, 2002. ACM Press.
- [ZHH96] R.C. Zeleznik, K.P. Herndon, and J.F. Hugues. Sketch : An interface for sketching 3d scenes. In *SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 163–170, New York, NY, USA, 1996. ACM Press.
- [ZS00] D. Zorin and P. Schröder. *Subdivision for Modeling and Animation*. 2000. SIGGRAPH course notes.
- [ZS03] R. Zenka and P. Slavik. New dimension for sketches. In *SCCG '03 : Proceedings of the 19th spring conference on Computer graphics*, pages 157–163, New York, NY, USA, 2003. ACM Press.

Résumé

La production de dessins animés 2D qui suit actuellement un schéma mis en place dans les années 1920 fait intervenir un très grand nombre de compétences humaines. Par opposition à ce mode de travail traditionnel, la production de films 3D, en exploitant les technologies et outils les plus récents de modélisation et d'animation 3D, s'affranchit en bonne partie de cette composante artisanale et vient concurrencer l'industrie du dessin animé 2D en termes de temps et coûts de fabrication.

Les enjeux à relever par l'industrie du dessin animé 2D se posent donc en termes de :

1. Réutilisation des contenus selon le célèbre paradigme du “*Create once, render many*”,
2. Facilité d'échange et transmission des contenus ce qui nécessite de disposer d'un unique format de représentation,
3. Production efficace et économique des contenus requérant alors une animation automatisée par ordinateur.

Dans ce contexte compétitif, cette thèse, réalisée dans le cadre du projet industriel TOON financé par la société Quadraxis avec le support de l'Agence Nationale pour la Valorisation de la Recherche (ANVAR), a pour objectif de contribuer au développement d'une plate-forme de reconstruction, déformation et animation de modèles 3D pour les dessins animés 2D.

Un état de l'art des méthodes et outils contribuant à la reconstruction de modèles 3D et à leur animation est présenté et discuté au regard des contraintes spécifiques des règles de création des dessins animés 2D et de la chaîne de fabrication 2D.

Ayant identifié les verrous technologiques à lever, nos contributions ont porté sur :

- L'élaboration d'une de reconstruction des personnages virtuels 3D à partir de dessins 2D,
- La mise au point d'une méthode de modélisation 3D par surfaces maillées, compatible avec le standard MPEG-4/AFX,

Les développements réalisés, intégrés dans un premier prototype de la plate-forme TOON, montrent un gain en temps de 20% dans la chaîne de production et une complète interopérabilité des applications *via* le standard MPEG-4.