

Thèse

présentée par

Ioana Georgiana CIUCIU

pour obtenir le titre de

DOCTEUR

de l' **Université de Reims Champagne-Ardenne**

Spécialité : **Informatique**

PROPOSITION POUR LA PRISE EN COMPTE DES DÉFORMATIONS MÉTIERS DANS UN SYSTÈME
DE CAO

Date de soutenance : 5 Novembre 2009

Composition du jury :

Sebti FOUFOU	Professeur à l'Université de Bourgogne, Dijon	Rapporteur
Bezhad SHARIAT	Professeur à l'Université Claude Bernard Lyon 1	Rapporteur
Lionel ROUCOULES	Professeur à Arts et Métiers ParisTech - CER d'Aix-en-Provence	Examineur
Yvon GARDAN	Professeur à l'Université de Reims (IFTS)	Directeur de thèse
Estelle PERRIN	Maître de Conférences HDR à l'Université de Reims (IFTS)	Directrice de thèse

Thèse préparée au sein du **ERT Gaspard Monge (CReSTIC)**

IFTS - Institut de Formation Technique Supérieure de Charleville-Mézières

Remerciements

Ici, je voudrais adresser toute ma reconnaissance à tous ceux qui ont contribué à l'aboutissement de mes travaux.

Je commence, naturellement, par remercier mes deux directeurs de thèse, Estelle Perrin et Yvon Gardan, de m'avoir accueillie dans leur équipe. Je leur remercie pour leur confiance, pour leur présence et pour tout ce que j'ai appris chaque jour en travaillant à leur côtés. Je leur suis reconnaissante pour tout ce qu'ils m'ont transmis, tant sur le plan professionnel que sur le plan humain.

Ensuite, je tiens à remercier mes rapporteurs, Monsieur le Professeur Bezhad Shariat de l'Université de Lyon¹ et Monsieur Sebti Foufou, Maître de Conférences HDR à l'Université de Bourgogne, qui m'ont fait l'honneur d'évaluer mon travail. Je leur remercie pour le temps qu'ils ont consacré pour lire mon manuscrit et pour leurs commentaires.

Je suis également reconnaissante à Monsieur le Professeur Lionel Roucoules, de l'ENSAM - CER d'Aix-en-Provence d'avoir accepté d'être membre de mon jury et d'avoir pris le temps de juger mon travail.

Mes travaux de recherche se sont déroulés dans le cadre du projet DIJA de IÉRT Gaspard Monge. Je remercie tous mes collègues pour leurs idées qui ont apporté un plus à mon travail. La collaboration avec DINCCS a été présente au quotidien dans mes travaux. Je remercie Fred, Mike et Nico pour leur aide et leur disponibilité. Je remercie tous les membres de notre équipe pour la bonne ambiance que je n'ai connue nulle part ailleurs. Je remercie également tous les membres de l'administration de l'IFTS pour leur présence et leur serviabilité au quotidien.

Je tiens à remercier le Conseil Général des Ardennes pour avoir financé ma thèse.

Une pensée de reconnaissance à tous ceux que j'ai connus dès mes premiers jours à Charleville et dont l'amitié et la camaraderie ont fait plus supportable mes jours à l'étranger : Etienne, Julien, Mael, Fred H., Nico, Mike, Fred, Eric, Cosmin, Philippe, Siham, Anouar, Slah, Walid,

Mohand, Fadhila, Wafaa, Véronique, Caroline, Tahar, Laurent, Omar, Ben, Romain, Audrey, Seb, Marius, Reli et tout les autres.

Je voudrais maintenant remercier très sincèrement tous ceux qui sont allés au delà des limites de la camaraderie et qui m'on fait partager leur amitié et leur gentillesse pendant ces années.

Ma première pensée est pour Estelle, qui est pour moi un modèle de vie avec Fred et leurs enfants. Je te remercie pour ta confiance et pour que chaque fois que je suis près de toi je me sens une meilleure personne. Tout comme M. Gardan, qui a su bien m'accueillir pas seulement dans son équipe mais aussi dans sa famille. Je lui remercie pour sa gentillesse et sa générosité. Je remercie également Madame Gardan, Marion et Julien. Vous avez toute mon admiration et mon respect.

Merci à Nico, Fred, Alex, Mike et Romain pour leur bon humour et leur gentillesse. Merci pour les soirées pizza-bière-foot, et pour m'avoir accueillie dans vos familles. Merci également à Elodie, Rachel et Mélanie pour leur hospitalité. Quel bonheur de passer des soirées en compagnie de vos enfants ! Mike, ton amitié et ta gaieté m'ont apportée moi aussi autant de bonheur. J'espère que tu vis au mieux ton expérience canadienne.

Un grand merci à Ben et Audrey pour m'avoir toujours traitée comme un membre de leur famille. Je n'oublierai jamais les bons plats ardennais. Merci au papa de Ben pour le délicieux jambon ardennais. Je n'oublie pas bien entendu Romain, Boubounne et leurs amis, ni les mémorables soirées que j'ai passées en leur compagnie. Merci à Seb pour sa gentillesse et pour son aide avec LaTeX (et pas seulement).

Merci à Eric pour son amitié, pour son bon humour et pour sa générosité. Avec Cosmin et Nick, vous m'avez rendu les jours à Charleville plus ensoleillés que s'il y avait du soleil. Et il n'y en a eut pas très souvent. Merci à Cos pour avoir sonné si souvent à ma porte ces trois ans..

Je remercie à mes étudiants qui m'ont souvent aidé à me déconnecter de mes travaux de recherche pour mieux y pouvoir retourner ; je leur remercie aussi pour avoir donné du sens à mes efforts pour leurs transmettre mes enseignements.

A la fin, je remercie mes parents qui ont toujours tout fait pour me permettre de faire des études dans les meilleures des conditions. Je vous suis reconnaissante de m'avoir encouragée et soutenue d'aller plus loin dans mes études. Merci pour la liberté et pour le sens de l'absolu que vous m'avez insufflée dès petite. Je suis honorée d'être votre fille. Ce manuscrit vous est bien sûr dédié.

Je suis reconnaissante à ma sœur Raluca pour son amour, sa gentillesse et ses conseils.

Merci pour m'avoir rendue visite pendant ma thèse, je garderai toujours un beau souvenir de ces jours. merci également à Cos pour sa contribution (majeure). Je remercie Lumi pour ses coups de fils des mercredis après-midi, pour ses encouragements et pour ses péripéties de la jungle amazonienne. Merci d'être venue pour ma soutenance de si loin, je suis fière de t'avoir comme cousine. Je n'oublie pas Mish, qui m'a offert son amitié et son support inconditionnel pendant ces années de thèse. Merci à Ursu qui a su, à sa manière, être à mes cotés dès le début de ma thèse.

Une pensée émue à ma professeure, « doamna diriginta », pour son dévouement et son amour.

Une pensée à mes grands parents qui ne sont plus.. mais qui seront toujours présents pour moi.

Avant de finir, je dois une réponse à Cosmin, aussi chiffrée que son affirmation : EU.

Merci à tous !

Table des matières

Remerciements	i
Liste des tableaux	vii
Table des figures	viii
Introduction	1
1 Etude bibliographique : préliminaires pour la définition d'un système de CAO basé sur les déformations	5
1.1 Introduction	5
1.2 Modèle interne et utilisation	7
1.2.1 Classification des modèles de représentation de surfaces déformables . . .	7
1.2.2 Techniques de modélisation	19
1.3 Etat de l'art des techniques de déformation de surfaces	21
1.3.1 Techniques de déformation basées sur des propriétés physiques des objets	22
1.3.2 La déformation de formes libres (<i>Free Form Deformation</i> (FFD))	23
1.3.3 Déformations soumises à des contraintes de déplacement	25
1.3.4 Fonctions <i>decay</i>	27
1.3.5 FFD avec Surfaces de Subdivision	28
1.3.6 Déformations soumises à des contraintes géométriques	29
1.4 Outils virtuels de déformation de surfaces	30
1.4.1 Techniques d'interaction	30
1.4.2 Outils de déformation	33
1.5 Conclusion	35
2 Contexte : le projet DIJA	38
2.1 Introduction	38
2.2 La philosophie DIJA	39

2.2.1	L'approche synthétique	39
2.2.2	Eléments de dialogue	42
2.3	L'architecture du système DIJA	44
2.3.1	Niveaux d'abstraction	45
2.3.2	Approche modulaire	46
2.3.3	Utilité de l'architecture système dans les déformations	48
2.4	Modélisation par déformation	49
2.4.1	L'aspect métier de l'approche « synthétique »	50
2.4.2	Processus de déformation 2D	51
2.4.3	Outils DIJA orientés métier	53
2.5	Conclusion	58
3	Proposition pour la réalisation des outils de déformation 3D pour la CAO dans un contexte orienté métier	60
3.1	Introduction	60
3.2	Intégration des fonctions <i>decay</i> dans les règles métier	61
3.2.1	Définition des outils 3D à l'aide des fonctions <i>decay</i>	61
3.2.2	Un exemple de processus de déformation 3D	64
3.2.3	Application des outils de déformation DIJA 3D	68
3.3	Un exemple de raffinement de déformation	70
3.4	Une proposition pour l'intégration de la contrainte volumique avec les déformations DIJA 3D	78
3.4.1	Etat de l'art des techniques de déformation de formes libres avec contrainte volumique	78
3.4.2	Intégration d'une contrainte volumique avec les outils 3D de déformation DIJA selon la méthode d'Aubert et Bechmann	87
3.4.3	Proposition d'une méthode pour l'intégration de la contrainte volumique avec les outils de déformation 3D DIJA. Application pour la modélisation de flacons de parfum et de récipients cosmétiques	91
3.5	Intégration d'autres contraintes avec les déformations DIJA. Application pour l'insertion d'un plumier pour une tablette arrière de voiture	94
3.6	Conclusion	96

4	Mise en œuvre informatique d'un modelleur de surfaces discrètes	97
4.1	Introduction	97
4.2	Contexte	97
4.2.1	DIJA	98
4.2.2	Choix techniques	98
4.2.3	Limites du modelleur	99
4.3	Aspects liés à l'implémentation	99
4.3.1	Modèle interne de représentation	100
4.3.2	Représentation interne des outils de déformation	102
4.3.3	Implémentation de déformations	103
4.4	Interface graphique	108
4.4.1	Structure	108
4.4.2	Introduction à VTK	110
4.4.3	Module de visualisation	114
4.5	Synthèse de l'implémentation	117
4.6	Conclusion	119
	Conclusion et Perspectives	121
A	Eléments de géométrie discrète	127
B	Fonctionnalités complémentaires DIJA	132

Liste des tableaux

3.1 Méthodes de FFD avec contrainte volumique.	87
--	----

Table des figures

1.1	Différentes représentations géométriques des surfaces déformables [MDA01]. . . .	8
1.2	Une surface de Bézier [Geo08]. La forme de la surface est modifiée en déplaçant les points de contrôle.	9
1.3	Super quadriques [Bar81].	10
1.4	Exemple de subdivision de surface. Plusieurs niveaux successifs de raffinement (subdivision) [RB04].	11
1.5	Types de <i>p-cellules</i> [OMV01].	14
1.6	Dualité maillages simplexes-triangulations [Toh03] : (a) Maillage triangulaire d'un tore ; (b) Le maillage simplexe dual du maillage (a) ; (c) Un maillage dual triangulaire du maillage simplexe (b) ; les maillages (a) et (c) sont totalement différents même s'ils partagent la même connectivité ; (d) la triangulation du maillage simplexe (b) ; ce maillage n'a pas la même connectivité que (a) et (c).	15
1.7	Discrétisation d'une entité continue en un système masse-ressort.	16
1.8	Une portion d'un système masse-ressort [GM97].	16
1.9	Force d'interaction de type Lenard-Jones entre deux particules en fonction de la distance d qui les sépare.	17
1.10	(a) Eléments de base du modèle (<i>blobs</i>) ; liaison s'étirant et se rompant entre deux atomes ; (b) Image générée utilisant des exposants quadriques [Bli82].	22
1.11	(a) Primitive pliée torsadée ; (b) Primitive pliée torsadée effilée ; (c) Modèle d'une chaise composée de six primitives avec sept opérations de pliage [Bar84].	23
1.12	Le <i>hyper patch</i> de base de la FFD [HHK92] ; les vecteurs s , t et u définissent le système de coordonnées local.	24
1.13	FFD avec un volume de Bézier [Dub].	24
1.14	<i>Patches</i> de contrôle avec une topologie généralisée [Coq90].	24

1.15	Déformation axiale [Dub].	25
1.16	Exemples de styles de fonctions <i>decay</i> [BL94] ; de gauche à droite : bombé, cloche, sommet, cône, plat.	27
1.17	Exemple d'application des fonctions <i>decay</i> en un point [AWW89] ; de gauche à droite : cône, cloche, sommet.	27
1.18	Déformations locale et globale d'une théière [FSJ ⁺ 06] : (a) Maillage de contrôle initial ; (b) Anse modifiée localement ; (c) Corps et bec modifiés globalement, (d) Objet initial ; (e) Déformation locale contrôlée par (b) ; (f) Déformation globale contrôlée par (c).	28
1.19	Interactions par (a) déformations directes et par (b) déformations indirectes. . . .	32
1.20	Région d'influence [AWW89] : (a) En nombre d'arrêtes ; (b) En distance Euclidienne. . .	33
1.21	Opérations « pousser-tirer » appliquées à un maillage triangulaire avec un outil virtuel : (a) Résultat de « pousser » ; (b) Résultat de « tirer ».	35
1.22	Organisation d'un système de modélisation ; interaction entre les différents modules qui le composent.	36
2.1	Conception d'une colonne avec l'approche « synthétique » en appliquant des opérations élémentaires.	40
2.2	Éléments de dialogue sur deux objets de base (parallélépipède et cylindre) : (a) l'objet et ses faces ; (b) l'objet avec les lignes et contours caractéristiques ; (c) l'objet avec ses fibres.	42
2.3	Modèle B-Rep étendu [Dan02].	43
2.4	Les niveaux d'abstraction de DIJA.	45
2.5	Décomposition modulaire de DIJA.	46
2.6	L'architecture cinq niveaux DIJA.	47
2.7	Un exemple de conception d'une cruche par un outil de déformation sur des éléments de dialogue : (a-c) application de l'outil de déformation sur une fibre pour déformer la silhouette générale ; (d-f) application de l'outil de déformation sur une fibre pour déformer l'anse ; (g) déformation locale sur un contour caractéristique pour la création d'un bec ; (h-j) résultats de la déformation dans (g) ; (j) la solution choisie par l'utilisateur.	51
2.8	Modélisation 2D d'objets décoratifs.	53
2.9	Outils de déformation représentés à différents niveaux d'abstraction.	54

2.10	Les composants du module de connaissances.	55
2.11	Paramètre graphonumérique.	56
2.12	Style dérivant de la courbure de la déformation ; la représentation mathématique va des coniques aux polygones.	57
3.1	Trois types de fonctions <i>decay</i> sur la surface d'un cylindre : (a) fonction <i>decay</i> « cloche » ; (b) fonction <i>decay</i> « sommet » ; (c) fonction <i>decay</i> « cône ».	62
3.2	Interférence de deux zones d'influence [AWW89] : (a) le déplacement des points est influencé par les deux points « attachés » ; (b) le déplacement des points est influencé par le plus proche des points « attachés ».	63
3.3	Cas spéciaux de déformations <i>decay</i> [AWW89] : (a) sommets « attachés » ; (b) sommets « ancrés ».	64
3.4	Processus de déformation 3D : (a) l'objet initial ; (b) la zone déformable ; (c) la zone de déformation ; (d) déformation style « bombé » du corps de la bouteille ; (e) déformation de type « sommet » du bouchon de la bouteille ; (f) déformation de type « sommet » de la base de la bouteille.	66
3.5	Opérations de lissage : (a) trois pas de subdivision linéaire d'un maillage repré- sentant la tête d'un mannequin [ZSD ⁺ 00] ; (b) pièce initiale [ZSS96] ; (c) pièce obtenue après six itérations de subdivision Butterfly [ZSS96] ; (d) maillage ini- tial [ZSS96] ; (e) résultat après trois itérations de subdivision Butterfly appliquée au maillage (d)[ZSS96].	67
3.6	Éléments de dialogue d'une déformation 3D : (a) différents types de contours ; (b) point initial et son vecteur de déplacement.	68
3.7	Insertion d'un plumier dans une tablette arrière de voiture avec un outil de dé- formation 3D : (a) tablette initiale (<i>courtesy of FAURECIA Acoustics and Soft Product Line Mouzon R&D Center</i>) ; (b) contour « rectangle » sur la tablette ; (c-d) déformation style « bombé » borné par le contour « rectangle » ; (e) dessin à main levée d'un nouveau contour pour le plumier ; (f-g) le plumier induit par le nouveau contour.	69
3.8	Plumier d'une tablette arrière de voiture.	71
3.9	Premier pas de raffinement : niveau métier.	71
3.10	Deuxième pas de raffinement : niveau commun.	73
3.11	Troisième pas de raffinement : niveau dialogue et géométrique.	74

3.12	Dernier pas de raffinement : niveau géométrique.	76
3.13	Récapitulatif du processus de déformation ; phase de descente des paramètres et de remontée du résultat.	77
3.14	Calcul du volume [HML] : (a) projection d'un triangle dans le plan $\widehat{xOy}$; (b) la variation de volume induite par le déplacement dP_1 du sommet P_1	80
3.15	Déformation d'un tore avec contrainte volumique [HML] : (a) tore initial, tore compressé sans conservation volumique, tore compressé avec contrainte volumique $x - y$; (b) tore initial, tore allongé sans conservation volumique, tore allongé avec contrainte volumique.	81
3.16	Calcul du volume [ZCCD06] ; en bas : déformation de l'abdomen après un impact avec un corps lourd.	82
3.17	Simulation d'une inspiration (partie droite) et d'une expiration (partie gauche) [ZCCD06] ; les flèches bleues indiquent le mouvement de la cage thoracique ; les flèches vertes indiquent le mouvement de l'abdomen.	82
3.18	La variation volumique induite par le déplacement des sommets A vers A' , de B vers B' et de C vers C' est la somme des volumes des tétraèdres (A, B, C, A') , (A, B, C, A') et (A, B, C, C') [AB97].	83
3.19	Déformation DOGME avec une fonction B-spline [BR94] : différentes solutions possibles pour le même objet initial et les mêmes contraintes de déplacement, en fonction du nombre de points de contrôle.	84
3.20	Attraction et répulsion des parties déformables de la sphere pour la même contrainte de déplacement lors d'une déformation avec conservation du volume [BR94]. . . .	85
3.21	Estimation du volume [LCOGL07].	85
3.22	Déformation d'une pièce de lego avec contrainte volumique [LCOGL07] : (a) pièce initiale ; (b-d) pièce déformée sans contrainte volumique ; (c-e) déformation avec correction du volume.	86
3.23	Déformation inattendue sur la surface d'un cylindre représentant une bouteille en forme primaire ; la déformation est générée par l'algorithme de conservation volumique (Algorithme 3) : la partie haute, près du bouchon de la bouteille. . . .	90
3.24	Les configurations d'un carreau dans un cube.	92
3.25	Maillage extrait de 150 coupes IRM utilisant l'algorithme <i>marching cubes</i> [Wik]. .	92

3.26	Déformation d'un flacon de parfum avec contrainte volumique selon différents styles ; le flacon est rétréci pendant la correction du volume : (a) flacon initial ; (b) déformation de type « bombé » ; (c) déformation de type « cloche » ; (d) déformation de type « sommet »	93
3.27	Insertion d'un plumier dans une tablette arrière de voiture par application d'emboutis sur la tablette initiale : (a) tablette initiale (<i>courtesy of FAURECIA Acoustics and Soft Product Line Mouzon R&D Center</i>) ; (b) embouti style « bombé » quand $h \leq h_{max}$; (c) embouti style « bombé » quand $h = h_{max}$; (d) embouti style « bombé » quand $h > h_{max}$; (e) exemple de tablette avec insertion de trois plumiers style « bombé » avec différents contours en respectant les contraintes. . .	95
4.1	Diagramme de classes UML simplifié du modèle B-Rep étendu [Dan02].	100
4.2	Diagramme de classes UML relative à la structure de données « Surface ».	101
4.3	Diagramme de classes UML relative aux structures de données « Surface » et « Zone ».	102
4.4	Diagramme de classes UML relative aux déformations.	103
4.5	Diagramme de classes UML relative au style de déformation.	104
4.6	Schéma des routines de définition de la zone de déformation.	105
4.7	Interface graphique du modeleur.	109
4.8	Le pipeline de visualisation VTK. Les arêtes indiquent le flux de données.	110
4.9	Types de cellules VTK [SML06]. Les nombres désignent l'ordre des points de définition.	112
4.10	Opérations de traitement VTK : (a) Découpage (<i>clipping</i> en anglais) [eIPX] ; (b) Contour (<i>contouring</i> en anglais) [Joh] ; (c) Extraction (<i>extract</i> en anglais) [Hol06].	113
4.11	Différents niveaux d'opacité d'un objet : (a) opacité = 1 ; (b) opacité = 0.5 ; (c) opacité = 0.2.	114
4.12	Différents modalités de visualiser un acteur : (a) visualisation en mode surfacique ; (b) visualisation en mode filaire ; (c) visualisation par points.	115
4.13	Interaction visuelle pour une opération de sélection dans la fenêtre de rendu pour paramétrer la déformation d'une surface plane.	116
4.14	Interface graphique de la première maquette du modeleur DIJA de surfaces discrètes.	117
4.15	Les premiers types de déformation de surface DIJA.	118

4.16	Déformations <i>decay</i> appliquées à un cylindre : en haut à gauche : la zone de déformation sur l'objet initial ; en haut à droite : déformation de type « cône » ; en bas à gauche : déformation de type « sommet » ; en bas à droite : déformation de type « bombé » (Bézier)	118
A.1	Un point discret et le voxel correspondant [JF06] : (a) $P \in \mathbb{Z}^n$; (b) voxel associé à P	128
A.2	Décomposition d'un voxel en cellules de dimensions inférieures [Coe02].	128
A.3	Adjacence dans l'espace 3D [Lac05] : (a) 6-adjacence (par face) ; (b) 18-adjacence (par arête) ; (c) 26-adjacence (par sommet).	129
A.4	Dualité surface discrète-isosurface [Lac05].	129
A.5	Niveaux de continuité des courbes/surfaces [Dub] : continuité de position (haut), de tangence (milieu) et de courbure (bas).	131
A.6	Continuité de tangence [Dub].	131
B.1	Contour caractéristique crée sur la surface d'un cylindre.	132
B.2	Déformation par emboutissage, en appliquant l'outil « embouti ».	134
B.3	Exemple de section du fût d'une colonne architecturale : à gauche, la forme de départ ; à droite, la forme déformée par l'application d'une déformation (angle droit) sur 8 parties du contour.	135

Introduction

La simplicité n'est pas un but
dans l'art, mais on arrive à la
simplicité malgré soi en
s'approchant du sens réel des
choses.

Constantin Brancusi

L'évolution technologique et scientifique dans le monde entier à l'heure actuelle, ainsi que les technologies de pointe développées dans les dernières années se répercutent, en tout ou en partie, sur le domaine de la CAO. On assiste actuellement à un élargissement du panel des domaines où l'empreinte de la conception assistée par ordinateur est de plus en plus présente. Des applications en ingénierie pour la fabrication des pièces mécaniques - dans le domaine du design, pour la création des objets de décoration, la création d'articles utilisables dans la vie quotidienne ou dans des domaines plus spécifiques - font toutes appel à la CAO. Aujourd'hui, les barrières entre les différents domaines tendent à disparaître, grâce à une universalisation des techniques de conception, qui se basent souvent sur les mêmes algorithmes, que ce soit pour l'industrie aéronautique, pour la médecine ou pour d'autres domaines.

Cette pluridisciplinarité naît des différentes vues de la conception, qui ne se fait plus seulement par des ingénieurs, mais par un public très varié, et qui entraîne, à son tour, des différentes techniques d'interaction. Pour répondre à cette diversité, les systèmes de CAO devraient assister l'utilisateur non seulement par des moyens informatiques physiques, mais aussi par des outils virtuels de modélisation plus adaptés, donc plus intuitifs.

En outre, de nos jours, l'ordinateur offre la possibilité de s'ouvrir sur le monde à travers Internet, ce qui se répercute sur les systèmes et sur les méthodologies de travail de la CAO. Ainsi, les projets de CAO ne se restreignent plus à une seule équipe, à une seule entreprise ou à un seul pays. On assiste au phénomène d'entreprise étendue, comme résultat de la mondialisation des technologies de l'information. Cette forme d'entreprise permet de mettre en commun les

connaissances et les compétences des différentes entités situées dans des différents pays du monde. La coordination de leurs activités se fait à travers l'intégration informatique de chacune des entités partenaires, plus précisément à travers des portails collaboratifs.

Les techniques mises en place par des systèmes de CAO commerciaux actuels reposent sur une méthodologie de conception constructive. Bien que puissants d'un point de vue géométrique, ces systèmes ne prennent pas en compte l'aspect fonctionnel du produit conçu. Ceci mène souvent à des temps plus longs de conception, le concepteur étant forcé de revenir en arrière dans le processus de conception pour renforcer certaines fonctions du produit qui n'ont pas été prises en compte dès le début. De plus, à l'heure actuelle, ces systèmes sont encore lourds, basés sur des modèles géométriques complexes et non accessibles à un utilisateur sans une formation préalable. Les interfaces intégrées pas ces systèmes sont complexes et souvent déroutantes pour un néophyte.

La méthodologie modificative est une approche tout à fait différente, qui permet la conception d'un objet à travers des déformations appliquées à la surface de celui-ci. La modélisation géométrique des surfaces constitue un élément de base du domaine de la CAO. Chacun des systèmes de CAO commerciaux intègre un modelleur géométrique qui donne à l'utilisateur la possibilité de créer, représenter et manipuler des formes géométriques. Dans ce sens, les déformations constituent un outil puissant. Souvent, la description fonctionnelle ou géométrique de l'objet modélisé ne peut pas être connue à cause de sa complexité. Il est alors plus facile de commencer le processus de modélisation avec une forme bien connue et de la déformer conformément à des besoins précis pour aboutir à la forme désirée.

Dans l'industrie mécanique, les surfaces fonctionnelles sont utilisées pour donner la fonction de l'objet modélisé, alors que les surfaces de forme libre sont utilisées pour des raisons de design. Les surfaces de forme libre (surfaces *Free Form* en anglais) sont des surfaces utilisées dans le domaine de la CAO pour décrire la peau des objets 3D de manière naturelle. La déformation de surfaces de formes libres (*Free Form Deformation* en anglais, abrégé FFD) se fait selon des techniques spécifiques suivant deux directions : une qui concerne les déformations CAO et une deuxième, celle de la sculpture virtuelle. Dans le premier cas, la déformation de surfaces de forme libres se fait en respectant certaines contraintes qui traduisent les spécificités du modèle. Dans ce cas, la modélisation requiert des connaissances géométriques et une formation technique préalable. Quant à la sculpture virtuelle, elle fournit des outils virtuels pour sculpter l'objet modélisé en émulant des techniques traditionnelles empruntées à la sculpture. Cette technique permet de modéliser un objet à travers des déformations, mais se préoccupe seulement de la forme, sans

respecter les fonctions de l'objet modélisé. Elle est accessible à tous les types d'utilisateurs, car c'est une technique de modélisation intuitive qui ne nécessite pas des connaissances *a priori*.

Une approche de modélisation innovante serait d'intégrer les déformations avec des contraintes métier, quel que soit le niveau de l'utilisateur - expérimenté ou néophyte - à la modélisation de ses objets. Ainsi, le concepteur pourrait se concentrer sur son métier plutôt que sur le modèle géométrique, laissant au système la tâche du passage des paramètres métier aux paramètres géométriques, applicables au modèle. Grâce à cette approche et aux outils de déformation dédiés, la conception assistée par ordinateur serait accessible à tout type d'utilisateurs d'une manière intuitive, permettant de respecter à la fois la forme et les fonctions des objets conçus.

Cette approche est à la base du projet de recherche DIJA, construit autour de plusieurs thématiques de recherche présentes dans l'ERT Gaspard Monge et dans le département DINCCS. DIJA se base sur un concept fédérateur dont l'idée est de créer un système de CAO qui est à la fois fonctionnel, distribué, collaboratif et intuitif. La méthodologie de conception DIJA se fonde sur une approche « synthétique » qui est à la base de notre système et qui renverse le processus de conception. Ainsi, la conception ne se fait plus de manière constructive, classique, mais grâce à une philosophie descendante qui permet de partir avec une forme approximative de l'objet désiré et de la raffiner au fur et à mesure pour aboutir à l'objet final. Le raffinement se réalise à travers des déformations possibles grâce aux outils orientés métier présents dans le système.

L'étude présente se place dans le contexte DIJA, avec le but de créer un modelleur de déformation de surfaces avec un comportement métier bien précis, non pas en utilisant le modèle géométrique sous-jacent, tel que réalisé jusqu'à présent, mais en prenant en compte une triangulation de la surface aussi fine que nécessaire. L'étude permet aussi de démontrer le passage des paramètres métier aux paramètres géométriques et l'interaction avec l'utilisateur à des différents niveaux de l'architecture du système, selon ses compétences.

Le manuscrit est organisé en quatre chapitres, structurés en deux parties : l'état de l'art et les contributions.

Le premier chapitre étudie, d'un point de vue déformations orientées métier, les éléments nécessaires à la définition d'un système de CAO : le modèle, les techniques de modélisation et l'interaction Homme-Machine. L'accent est mis sur les modèles de déformation de surfaces et sur les techniques d'interaction à l'aide des outils virtuels. Les directions de ces travaux de recherche - les techniques de déformation des surfaces et leur adaptation au contexte DIJA - sont annoncées, en motivant, à chaque fois, nos choix par rapport aux éléments présentés.

L'implémentation des principes présentés dans le premier chapitre se fait dans le contexte du projet DIJA, présenté dans le deuxième chapitre. La philosophie DIJA est étudiée ici, avec l'architecture système qui est à la base de la technique de modélisation par déformation. L'intégration des outils de déformation 2D orientés métier et leur application sont illustrées à travers un exemple de modélisation d'une cruche.

L'extension du modèle de déformation DIJA avec le passage de 2D à 3D se fait dans le troisième chapitre, qui propose une méthodologie pour l'intégration des outils de déformation de surfaces orientés métier. Ce chapitre montre la façon dont la liaison entre le modèle de déformation basé sur des fonctions *decay* et les contraintes métier est réalisée. Le déroulement des paramètres métier entre les différents niveaux d'interaction DIJA et leur application au niveau géométrique est démontré avec un exemple d'insertion d'un plumier dans une tablette arrière de voiture. Ce chapitre démontre également l'intégration automatique des autres contraintes métier dans le modèle de déformation de surface qu'on propose.

L'implémentation de la méthode ainsi démontrée se fait à travers un modelleur de surfaces discrètes présenté dans le quatrième chapitre de ce manuscrit. Cette partie montre, après avoir présenté le contexte de l'implémentation ainsi que les choix techniques, la représentation interne du modèle de déformation proposé. L'implémentation des outils de déformation est également étudiée. Un accent spécial est mis sur le module de visualisation à l'aide de VTK, qui facilite l'interaction et qui augmente l'intuitivité de la GUI DIJA. Le chapitre se termine avec une brève synthèse des maquettes réalisées au cours de cette thèse.

Chapitre 1

Etude bibliographique : préliminaires pour la définition d'un système de CAO basé sur les déformations

1.1 Introduction

La conception assistée par ordinateur (CAO) est un des premiers domaines à utiliser les techniques de déformation pour la modélisation d'objets. Ces techniques sont fortement liées aux modèles de représentation et aux outils d'interaction Homme - Machine et sont appliquées pour modifier les données stockées dans le modèle. Les trois éléments - modèle, techniques de modélisation, IHM - constituent les composants fondamentaux, nécessaires et suffisants à la définition d'un système de CAO.

L'objectif de cette thèse est de proposer une technique interactive de modélisation par déformation accessible à n'importe quel type d'utilisateur, néophyte ou expert. Nous nous basons donc, dans notre démarche, sur l'hypothèse que l'utilisateur n'a pas des connaissances mathématiques ou géométriques, mais seulement des connaissances caractéristiques à son métier. L'approche proposée dans cette étude intègre des notions métier dans un modéleur de surfaces en CAO. Ce modéleur met à la disposition de l'utilisateur un panel d'outils de déformation virtuels orientés métier, spécifiques à un certain métier précisé par l'utilisateur au début du processus de modélisation. Pour rendre la modélisation interactive, notre choix a été d'exploiter les possibilités de la géométrie discrète. Dans la géométrie discrète, les espaces de représentation sont discrets, réguliers et dénombrables dans les entiers. Le but est de traiter et analyser des données (pixels, voxels, surfels, etc.) et des structures discrètes (segments, droites, plans, surfaces, etc.) dans des applications d'images et volumes numériques. Or, dans le domaine de la CAO, l'espace discret $\mathbb{Z}^n$ ne convient pas. Le traitement se fait sur les coordonnées des points de l'espace $\mathbb{R}^3$, qui forment

le maillage de l'objet à modéliser. La représentation par des cellules discrètes est un dénominateur commun aux modèles acceptés par notre système. Ce type de représentation nous permet, à partir de n'importe quelle représentation géométrique, de la discrétiser en des cellules simplifiées (triangles, arêtes, points, etc.) afin de pouvoir lui appliquer des algorithmes de déformation simples et performants avec de bons temps de réponse de la part du système.

Ce modeleur s'intègre dans un plus large cadre, le projet DIJA [DDGP02]. DIJA est un concept fédérateur, dont l'idée est de créer un système de CAO innovant qui est à la fois fonctionnel, distribué sur Internet, collaboratif et intuitif. Seuls les aspects fonctionnel [Per06] et intuitif [Dan02] sont pris en compte dans cette étude.

L'aspect fonctionnel est accompli par une décomposition progressive de l'objet à modéliser qui diffère des techniques de modélisation utilisées dans les systèmes de CAO à l'heure actuelle. La précision de la forme finale est atteinte grâce à des déformations qui ne se font plus sur des modèles classiques mais sur des entités visuelles appelées éléments de dialogue (ED). Les éléments de dialogue représentent la silhouette de l'objet et permettent d'obtenir des détails visuels de l'objet. Jusqu'à présent seules les déformations appliquées à des ED 2D ont été considérées dans notre équipe [Dan02, Per06]. C'est pourquoi cette étude est une approche basée sur des déformations appliquées à un ED particulier, la face. Les déformations doivent s'appliquer en respectant une contrainte supplémentaire, celle d'obtenir des réponses satisfaisantes de la part du système. Nous ne considérons que des déformations purement géométriques pour modéliser l'objet afin de lui donner sa forme physique, le modèle n'étant pas induit par des propriétés physiques.

Le système est intuitif car il donne à l'utilisateur la possibilité de réaliser des déformations sans avoir aucune connaissance géométrique, seulement en utilisant son savoir-faire. L'intuitivité est réalisée grâce à des outils orientés métier qui sont censés être identiques à ceux du monde réel et qui aident l'utilisateur à s'immerger dans son environnement de travail.

Ce chapitre a comme objectif de présenter chacun des trois éléments nécessaires à la définition d'un système de CAO et de décrire à chaque fois les avantages, les inconvénients et l'adéquation à nos buts. Après une revue des modèles de représentation de surfaces ainsi que des techniques et des outils de déformation d'un point de vue orienté métier, objectifs visés par cette thèse, nous justifierons le choix des méthodes employées par la suite.

1.2 Modèle interne et utilisation

Cette section présente un état de l’art des modèles de représentation surfaciques ainsi que des techniques de modélisation. Celles-ci sont en forte corrélation avec les techniques d’interaction Homme-Machine [DDGP02] et l’ensemble représente le noyau de l’interaction, puisque les techniques d’interaction donnent à l’utilisateur l’accès aux techniques de modélisation. De plus, ces techniques affectent le processus de modélisation en proposant des outils plus ou moins intuitifs. Le but est donc d’améliorer ces techniques en s’appuyant sur un modèle intuitif et efficace.

1.2.1 Classification des modèles de représentation de surfaces déformables

Un des principaux modèles de représentation inhérents aux systèmes de CAO actuels est le modèle explicite B-Rep (*Boundary representation* en anglais, représentation par frontières), qui peut être couplé à un modèle implicite CSG (*Constructive Solid Geometry* en anglais, conservant un historique de conception). Le modèle B-Rep peut être soit polyédrique, soit basé sur des surfaces gauches, alors que le modèle CSG permet de représenter un objet par compositions, à l’aide d’opérateurs Booléens et d’objets primitifs (sphère, cylindre, etc.) [FvDF⁺00, Gar91].

Le modèle proposé dans cette thèse ne s’inscrit plus dans un cadre classique de représentation de modèles en CAO. Le système de CAO que nous avons créé se base sur un modèle B-Rep étendu proposé par Danesi dans [Dan02]. On accepte, tout de même, n’importe quel modèle de représentation, à condition que le système puisse le discrétiser afin de le transformer dans des structures manipulables par les algorithmes implémentés sous DIJA et que, ensuite, il puisse reconvertir le résultat conformément au modèle de représentation initial. Jusqu’à présent seule l’implémentation des éléments 2D (fibres, contours) a été considérée, même si le concept d’élément de représentation 3D (face) existait déjà dans notre modèle. Cette étude étend les algorithmes sous-jacents au modèle DIJA à l’espace 3D en considérant un élément visuel particulier, la face [CDGP08a]. Nous nous intéressons donc aux modèles surfaciques de représentation, dont une classification possible est donnée dans la figure 1.1 [MDA01].

Deux directions prenant en compte les modifications peuvent être considérées dans le domaine de la modélisation en CAO : une première qui se base sur des modèles classiques continus et une deuxième, plus récente, qui se base sur des modèles discrets.

Nous continuons avec une brève présentation, dans le paragraphe suivant, des modèles continus de représentation surfacique.

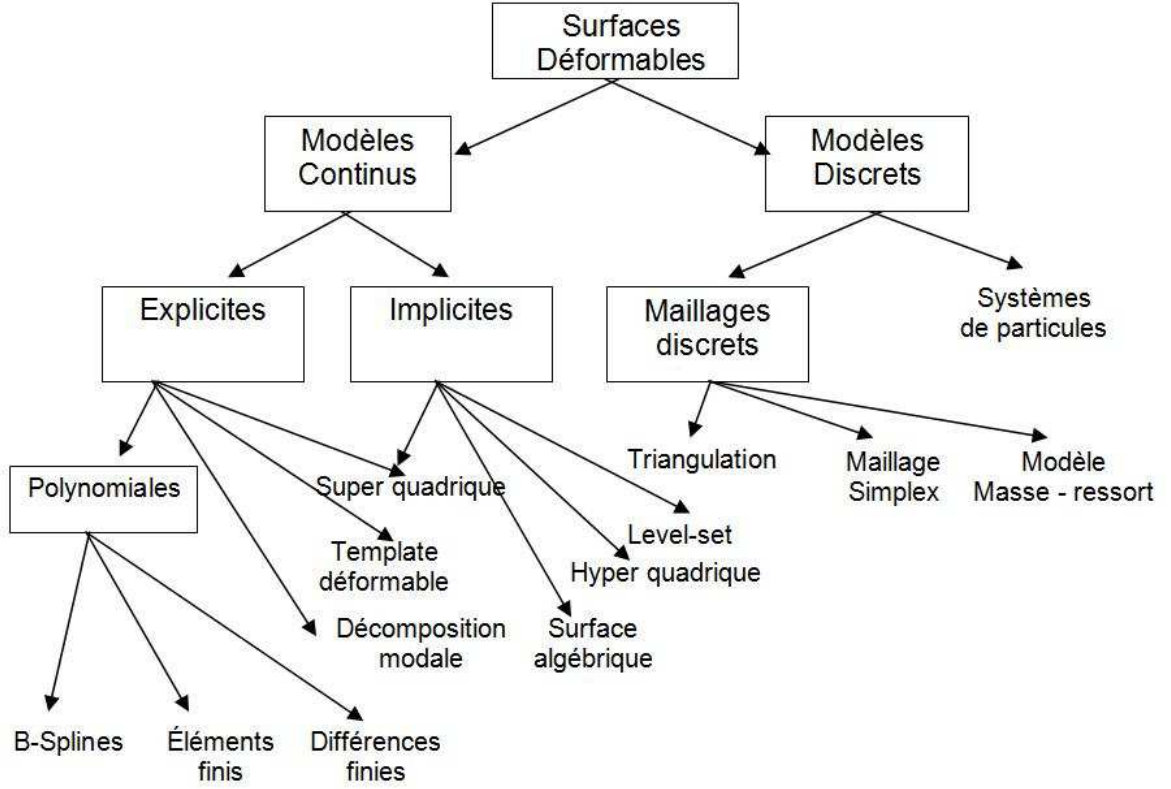


FIG. 1.1: Différentes représentations géométriques des surfaces déformables [MDA01].

A. Modèles continus

Les modèles continus sont des modèles classiques, tels que les surfaces explicites ou implicites. Nous détaillons ces deux types de surfaces dans les paragraphes suivants.

A.1. Les surfaces explicites Une surface continue explicite S_q est représentée par une fonction des coordonnées dépendante d'un vecteur $q = (q_1, \dots, q_{n_q})^T$ de paramètres de forme :

$$S_q : \mathbb{R}^{n_q} \times \Omega \mapsto \mathbb{R}^3, \quad (q_1, \dots, q_{n_q}, r, s) \mapsto p_q(r, s) \quad (1.1)$$

où (r, s) désigne un point du domaine Ω de paramètres de la surface.

La distinction entre les différents modèles explicites se fait en fonction du nombre et de la nature du paramètre q . Le nombre des paramètres influe sur la complexité de la déformation, alors que les paramètres contrôlant les déformations locales influent sur la complexité de calcul. Toutefois, ce sont ces derniers qui permettent d'obtenir des formes topologiquement compliquées. Une classe spéciale de surfaces explicites est représentée par les représentations polynomiales, détaillées dans le paragraphe suivant.

A.1.1. Les représentations polynomiales Elles utilisent des fonctions polynomiales comme support, dont les coefficients correspondent aux paramètres de la forme.

Les représentations polynomiales les plus connues sont les surfaces de Bézier et les B-Splines [BB88, Béz87]. Ces courbes et surfaces associées sont généralement définies par des *points de contrôle* dont le déplacement modifie l'aspect de la courbe/surface de manière intuitive (voir Fig. 1.2 [Geo08]). On peut aussi déplacer directement la courbe ou la surface B-Spline.

Ces modèles ont été créés initialement pour la modélisation dans le domaine de la CAO [Far]. Actuellement, on les applique de plus en plus dans le domaine de l'animation [Ost87] et dans le domaine du graphisme [DTG96, TGM03]. La visualisation des carreaux des surfaces polynomiales se réalise par des évaluations itératives de polynômes [FvDF⁺00].

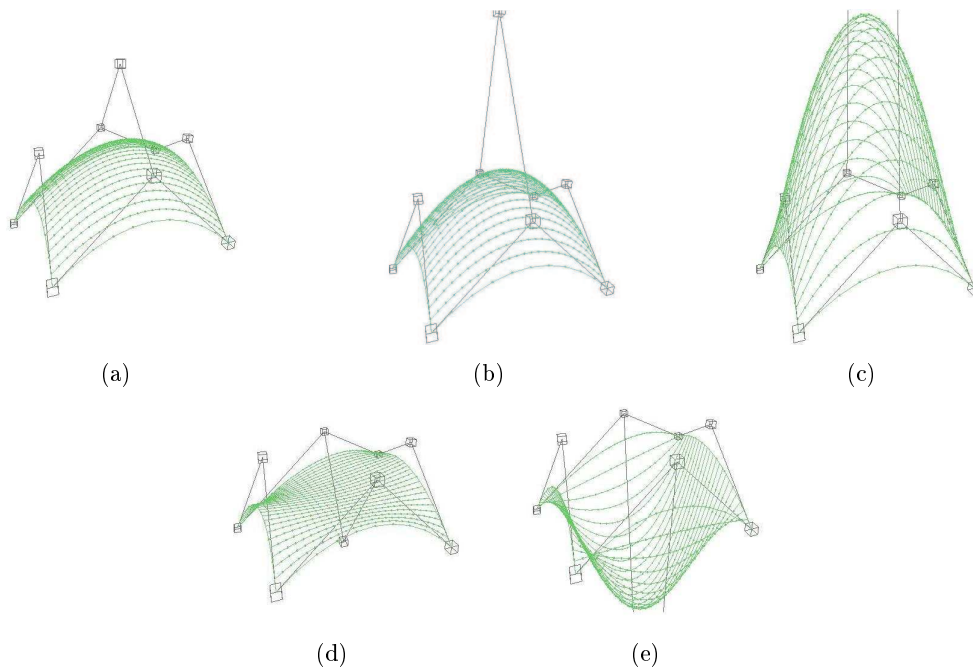


FIG. 1.2: Une surface de Bézier [Geo08]. La forme de la surface est modifiée en déplaçant les points de contrôle.

A.1.2. Les super quadriques Ce type de représentation généralise les quadriques pour représenter des surfaces. Ce sont les surfaces paramétrées les plus utilisées en graphisme et design 3D. La notation de *super quadrique* a été introduite par Barr en 1981 [Bar81]. Dans le domaine de la CAO, les super quadriques sont souvent utilisées comme objets de base pour créer des formes topologiquement sophistiquées. Elles sont employées dans la modélisation de solides car elles sont suffisamment flexibles pour représenter une grande famille de formes [Bar81]. Nean-

moins, l'inconvénient de ce modèle réside dans le fait qu'il permet uniquement la représentation des objets qui possèdent un axe de symétrie (voir Fig. 1.3). Il n'est donc pas approprié pour représenter de formes avec une topologie compliquée.

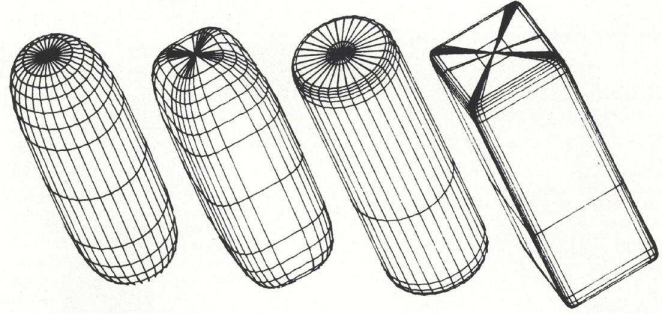


FIG. 1.3: Super quadriques [Bar81].

Les super quadriques peuvent être aussi représentées sous une forme implicite, ce qui permet de déterminer si un point p se trouve à l'extérieur de la surface, à l'intérieur de la surface ou sur la surface même (voir paragraphe suivant).

Représenter des objets de topologie arbitraire à l'aide des modèles paramétriques est une tâche difficile à cause des contraintes de continuité entre les carreaux de surface (voir les annexes A). Pour éviter cela, d'autres modèles surfaciques ont été créés. Les surfaces implicites, par exemple, présentent l'avantage de ne pas restreindre la topologie, ni la géométrie de la surface. Une autre solution pourrait être l'application des surfaces de subdivision avec les algorithmes de lissage sous-jacents. On introduit par la suite ces deux modèles.

A.2. Les surfaces implicites Les modèles surfaciques implicites utilisent une équation implicite afin de localiser les points de la surface. On crée un champ scalaire dans tout l'espace et la surface est définie comme le lieu où ce champ a une valeur donnée. Généralement on définit une surface implicite comme l'ensemble zéro d'une fonction f qui prend des valeurs en $\mathbb{R}$:

$$S_f = \{p \in \mathbb{R}^3 | f(p) = 0\} \quad (1.2)$$

Les champs scalaires définis dérivent généralement de potentiels et l'on parle de *surface équipotentielle* ou d'*isosurface*. Plusieurs surfaces implicites peuvent être combinées très facilement en sommant les potentiels qui les génèrent, ce qui permet de définir et manipuler des objets de topologie variable. La formulation implicite nous permet de savoir si un point de l'espace se

trouve à l'intérieur, à l'extérieur ou sur la surface même en le comparant à l'iso-valeur. Ceci nous permet de détecter facilement les collisions entre les objets.

La visualisation de ce type de modèles est difficile, à cause des singularités présentées par les surfaces implicites. Afin de représenter une surface implicite il faut trouver les points de l'espace 3D appartenant à la surface, puis lisser la représentation graphique de cet ensemble de points, ce qui est une tâche difficile. La représentation des surfaces implicites est toujours un sujet de recherche actuel. Elle pourra être facilitée en utilisant l'algorithme du *marching cubes* [Blo87, LC87], des maillages locaux [DTG96] ou la méthode à base de particules [WH94].

A.3. Les surfaces de subdivision Ce modèle, introduit par Catmull/Clark and Doo/Sabin [CC78, DS78] est une alternative aux surfaces *spline* [HDD⁺94, Sta98] - qui sont aussi contrôlées par des carreaux de contrôle - et aux surfaces implicites.

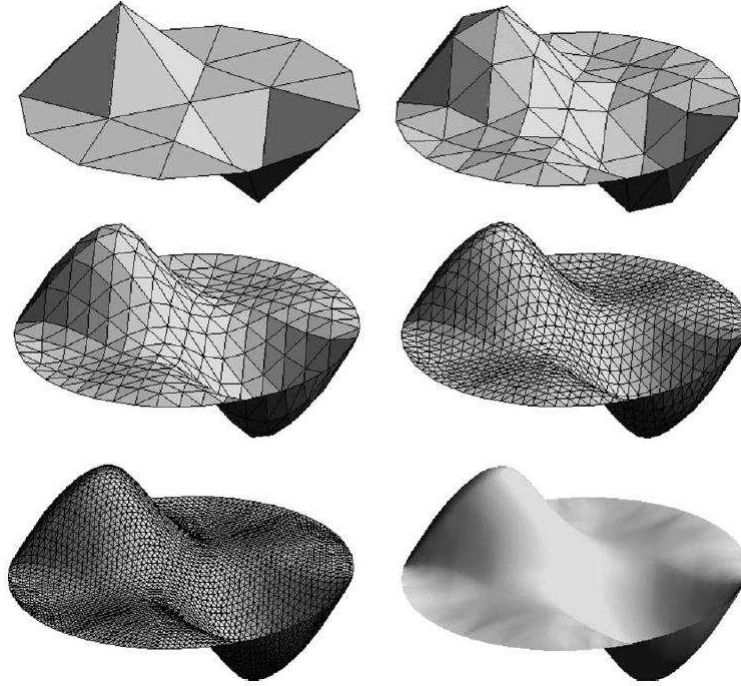


FIG. 1.4: Exemple de subdivision de surface. Plusieurs niveaux successifs de raffinement (subdivision) [RB04].

L'idée de la subdivision [ZSD⁺00] est de définir une surface lisse comme la limite d'une succession de raffinements en partant d'un maillage de contrôle de base et en lui appliquant des lois de subdivision (schémas de subdivision) [DLG90, Loo87, RB04, ZSS96]. A la fin de chaque pas de raffinement topologique, les positions des sommets (anciens et nouveaux) sont ajustées en

fonction de ces lois qui sont locales. La figure 1.4 montre le résultat d'un processus de subdivision d'une surface de base.

Les surfaces de subdivision n'imposent pas de restrictions, ni sur la topologie, ni sur la géométrie, comme les surfaces *spline* et leur structure permet des algorithmes très performants.

Avantages et inconvénients des modèles continus L'avantage des surfaces continues est la continuité géométrique, déterminée par la possibilité de calculer des quantités différentielles (la normale, la courbure, etc.) partout sur la surface. L'inconvénient principal réside dans le fait que les surfaces continues sont difficiles à représenter, à manipuler et visualiser, difficiles à appliquer et coûteuses. C'est la raison pour laquelle les modèles continus ne conviennent pas à notre approche.

Les deux types de représentation - explicite et implicite - sont complémentaires par rapport à l'évaluation (affichage), aux requêtes spatiales ou à la modification surfacique (géométrie *versus* topologie). Les points faibles de la première sont souvent les points forts de la deuxième. C'est donc en fonction de la spécificité du problème géométrique à résoudre que le type de représentation est choisi. Des algorithmes de conversion entre les deux types de représentation s'avèrent, dans ce cas, très utiles. Pour éviter certains de ces inconvénients, notamment à cause de la complexité de représentation et de calcul, il est nécessaire de discrétiser les modèles continus [RB04]. La section suivante introduit donc les représentations surfaciques discrètes.

B. Modèles discrets

La représentation des modèles déformables à l'aide des entités discrètes élimine les problèmes communs des représentations continues (paramétrisation, complexité de calcul, etc.). Pourtant, cette approche induit un nombre large de degrés de liberté sur la surface qui est préjudiciable. Il est donc nécessaire de restreindre l'espace de déformation pour éviter les déformations indésirables. Aussi, les mesures géométriques tel que la normale, la tangente, etc. sont difficiles à estimer. La plupart des modèles surfaciques discrets sont des maillages définis par un ensemble de points avec une relation de connexion. Cela rend leur représentation et leur manipulation plus simples comparé aux surfaces continues. De plus, ces modèles s'adaptent à tout type de topologies.

Quatre modèles discrets existent dans le domaine de la représentation de surfaces : les maillages triangulaires, les maillages simplexes, le modèle masse - ressort et les systèmes à base de particules. Les paragraphes suivants introduisent chacun de ces quatre modèles pour la mo-

délisation de surfaces avec finalement une analyse de l'adaptation à notre approche. Pour une analyse plus détaillée sur les définitions et les notations de géométrie discrète le lecteur est invité à consulter les annexes A.

B.1. Le maillage triangulaire Le maillage triangulaire est un modèle explicite. Il permet de représenter la surface comme un ensemble de sommets avec une relation de voisinage qui induit des contraintes topologiques. Dans le meilleur des cas, le maillage triangulaire est représenté par un ensemble de triangles adjacents. Chaque triangle partage au moins une arête avec un triangle voisin. Chaque triangle est un plan et donc on peut estimer la surface localement, sur une région restreinte [BBN00]. Ceci nous permet d'estimer localement des mesures géométriques, voir la tangente ou la normale [KCY92, PF98].

Un maillage triangulaire consiste en un ensemble d'entités donnant des informations :

- d'ordre géométrique et
- d'ordre topologique.

Dans [BPR⁺], le dernier type de maillage (ensemble d'entités donnant des informations d'ordre topologique) est défini par un ensemble de sommets

$$\nu = \{v_1, \dots, v_V\} \quad (1.3)$$

et par un ensemble de faces triangulaires les connectant

$$\mathcal{F} = \{f_1, \dots, f_F\}, \quad f_i \in \nu \times \nu \times \nu, \quad (1.4)$$

où chaque triangle spécifie ses trois sommets dans ν . La connectivité d'un maillage triangulaire peut également être définie par les arêtes qui le composent :

$$\varepsilon = \{e_1, \dots, e_E\}, \quad e_i \in \nu \times \nu. \quad (1.5)$$

On définit le composant géométrique d'un maillage triangulaire dans $\mathbb{R}^3$ en associant à chaque sommet $v_i \in \nu$ une position 3D p_i :

$$\mathcal{P} = \{p_1, \dots, p_V\}, \quad p_i := p(v_i) = \begin{pmatrix} x(v_i) \\ y(v_i) \\ z(v_i) \end{pmatrix} \in \mathbb{R}^3, \quad (1.6)$$

tel que chaque face $f \in \mathcal{F}$ est un triangle de l'espace 3D défini par les positions de ses trois sommets.

Un autre type de maillage, dont la définition est étroitement liée à celle des triangulations au sens de la topologie, est le maillage simplexe.

B.2. Le maillage simplexe Ce modèle a été introduit par Delingette. Sa définition [Del99] se base sur la définition de *p-cellule* (voir Fig. 1.5). La *p-cellule* est définie récursivement comme un ensemble de points de l'espace tel que :

- Une 0-cellule est un point et une 1-cellule est un segment reliant deux points distincts ;
- Chaque sommet appartenant à une *p-cellule* appartient à $(p - 1)$ cellules distinctes ;
- L'intersection de deux $(p - 1)$ -cellules est soit vide soit une $(p - 2)$ -cellule ;
- Etant donné deux sommets d'une *p-cellule*, il existe un chemin qui les relie.



FIG. 1.5: Types de *p-cellules* [OMV01].

Pour représenter une surface 3D, un maillage 2-simplexe convient. Dans ce cas, chaque sommet est relié à trois voisins. Des paramètres géométriques peuvent être associés à chaque sommet du maillage, ce qui permet le contrôle de la déformation grâce à des contraintes appliquées dans un point. On peut aussi leur associer des paramètres métriques qui définissent la position barycentrique des points par rapport à leurs voisins.

Il existe, pour chaque maillage simplexe, un maillage triangulaire dont le graphe représente le graphe dual du graphe du maillage simplexe (voir Fig. 1.6). La réciproque est aussi vraie, c'est à dire, pour chaque maillage triangulaire il existe un maillage simplexe dont le graphe est le graphe dual du maillage triangulaire. Cette dualité est purement topologique, il n'existe pas une dualité géométrique entre les deux types de maillage [Del94].

Les maillages simplexes sont suffisamment souples pour représenter des surfaces ouvertes ou fermées ayant n'importe quelle topologie. De plus, la géométrie d'un maillage simplexe permet la définition de mesures géométriques discrètes (courbure moyenne, normales) au niveau de chaque sommet. La connectivité constante des sommets permet d'avoir des propriétés intéressantes [Del94]. La visualisation de ce type de modèle nécessite une conversion du maillage simplex en un maillage triangulaire.

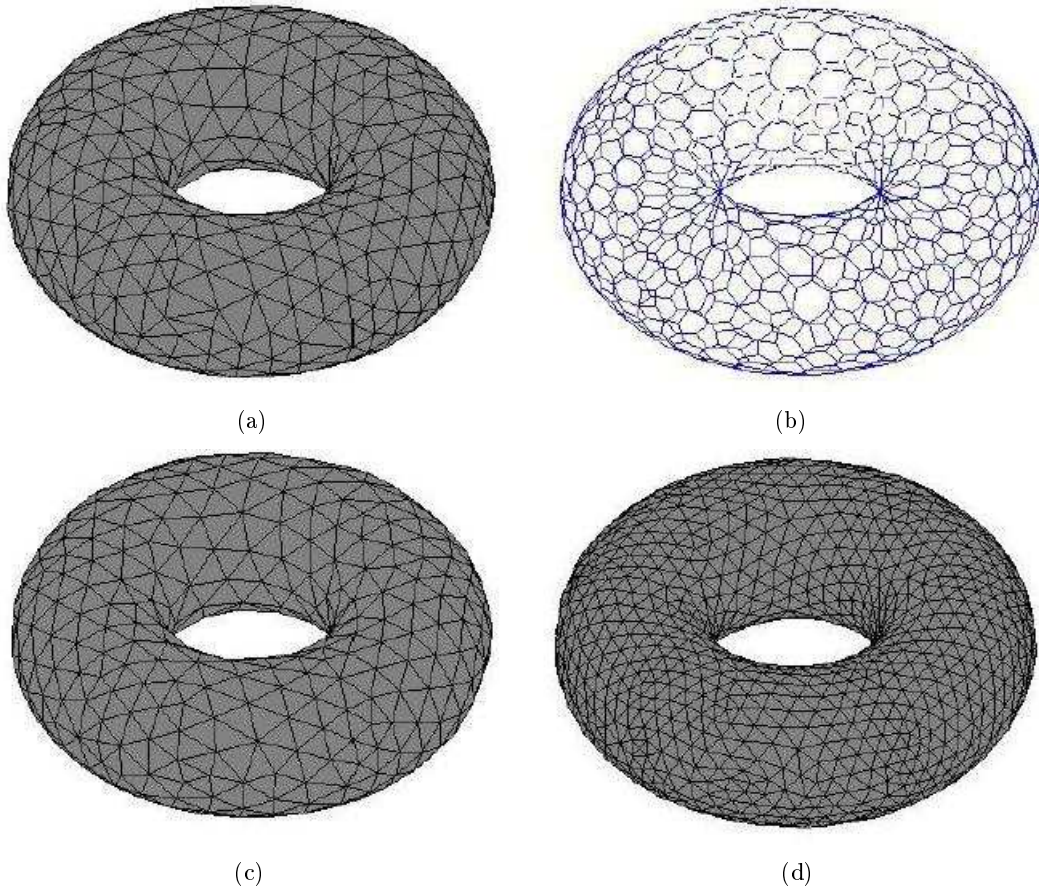


FIG. 1.6: Dualité maillages simplexes-triangulations [Toh03] : (a) Maillage triangulaire d'un tore ; (b) Le maillage simplexe dual du maillage (a) ; (c) Un maillage dual triangulaire du maillage simplexe (b) ; les maillages (a) et (c) sont totalement différents même s'ils partagent la même connectivité ; (d) la triangulation du maillage simplexe (b) ; ce maillage n'a pas la même connectivité que (a) et (c).

B.3. Le modèle masse-ressort Ce modèle se base sur des principes de la physique afin de modéliser des objets déformables. Il est souvent utilisé pour la simulation des processus physiques complexes qui seraient impossibles à modéliser uniquement avec des méthodes géométriques. L'objet à modéliser est discrétisé (voir Fig. 1.7) pour obtenir une représentation sous la forme d'une grille rectangulaire organisée en carrés avec des ressorts diagonaux.

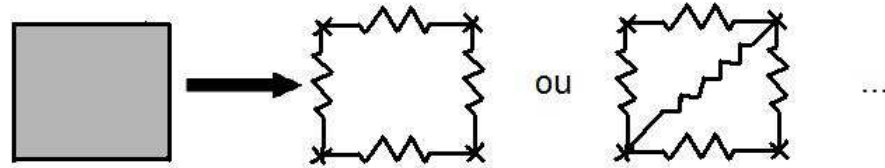


FIG. 1.7: Discrétisation d'une entité continue en un système masse-ressort.

Chaque point de la grille représente une masse et chaque arête un ressort (voir Fig. 1.8). Pour décrire le comportement de l'objet, on exprime le bilan des forces appliquées à chaque point.

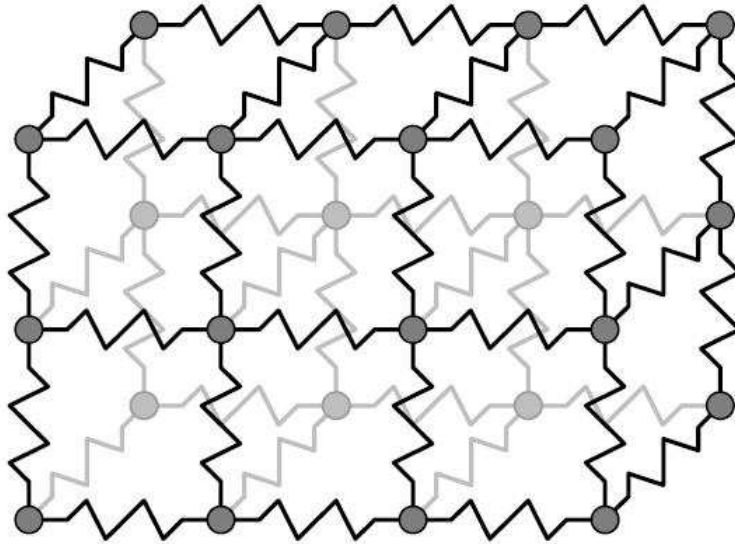


FIG. 1.8: Une portion d'un système masse-ressort [GM97].

Ceci se fait à l'aide d'une méthode de résolution (algorithme) itérative. A chaque pas, on calcule le bilan de forces pour l'ensemble des points du réseau. La valeur des déplacements est ensuite calculée en fonction de l'intervalle de temps. Les points sont ensuite déplacés. La stabilité

du nouveau système est testée ; si le système n'a pas acquis un état stable, le processus est répété jusqu'à la satisfaction de cette contrainte.

De tels modèles ont été appliqués initialement par Platt et Badler [PB81] comme premier modèle d'expression faciale. Chadwick [CHP89] les a employés dans le domaine de l'animation. Aujourd'hui, les systèmes masses-ressorts sont utilisés notamment pour modéliser des tissus, comme par exemple les vêtements humains chez Volino [VMT97]. Ces modèles sont aussi appliqués pour modéliser des volumes d'organes dans des applications médicales.

Le principal avantage de cette méthode réside dans sa simplicité et facilité de mise en œuvre. Elle peut être efficace si on prend soin de limiter la rigidité du système. Plus le système est rigide, plus le pas de temps d'intégration doit être faible, ce qui entraîne un grand nombre d'itérations. Le phénomène de raideur des objets a été étudié par Baraff et Witkin [BW92]. Un autre inconvénient est que ce modèle offre des possibilités limitées concernant les changements de topologie. Cela pourrait être résolu à l'aide d'une extension de ce modèle, les systèmes de particules, que nous abordons dans la section suivante.

B.4. Le système de particules Appliqué pour la première fois dans le domaine de la synthèse d'images, le système de particules généralise le système masses-ressorts. L'extension consiste à donner une dimension géométrique aux masses ponctuelles et aussi à ajouter la possibilité d'interaction entre les particules d'un même objet ou des objets différents. Cette interaction est généralement restreinte à un rayon maximal d'influence, pour limiter les calculs.

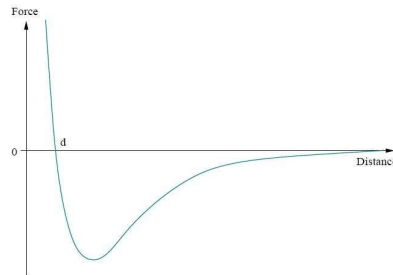


FIG. 1.9: Force d'interaction de type Lenard-Jones entre deux particules en fonction de la distance d qui les sépare.

Des systèmes de particules construits par de particules qui n'interagissent pas les unes avec les autres ont été appliqués pour modéliser des phénomènes naturels, tels que le feu, la fumée [Ree83] ou les cascades [Sim90]. Reynolds [Rey97] a simulé le comportement global de bancs de poissons

ou de vols d'oiseaux, les particules réagissant que selon des lois simples. Récemment, les systèmes de particules se basent sur des principes de la dynamique moléculaire pour modéliser des liquides et des solides [Mil89, TPF89].

Un système de particules est composé d'un ensemble d'entités élémentaires appelées particules. Chaque particule est définie par sa localisation, son accélération, sa masse et d'autres paramètres possibles, spécifiques à l'application à laquelle ils s'appliquent. Les relations entre les masses ponctuelles ne sont plus préétablies, mais évoluent dynamiquement. Les forces inter-particules sont souvent des lois d'attraction-répulsion. Les particules ont tendance à s'arranger sur une latte quand les forces externes s'épuisent. La figure 1.9 représente la force exercée par une particule i sur une particule j en fonction de la distance d qui les sépare.

Comme pour les masses-ressorts, la méthode de résolution est explicite. Pour connaître le comportement de l'objet modélisé, on exprime le bilan des forces appliquées à chaque particule et on intègre ensuite les résultats au système afin d'obtenir les nouvelles positions des particules dans l'espace. Le processus s'arrête une fois atteint un état stable. L'algorithme est décrit par le pseudo code suivant :

Algorithme 1 Le pseudocode de l'algorithme Particules

```
1: Départ
2: Tant que non(Etat Stable)
    Pour chaque particule  $i$ 
        Pour chaque particule  $j$ 
            Calculer la force exercée par la particule  $j$  sur la particule  $i$  ;
            Mettre à jour la somme des forces appliquées sur la particule  $i$  ;
            Calculer la force exercée par la particule  $i$  sur la particule  $j$  ;
            Mettre à jour la somme des forces appliquées sur la particule  $j$  ;
        Fin Pour
    Fin Pour
    Mise à jour des positions des particules en fonction des forces qui leurs sont appliquées
3: Fin Tant que
4: Arrêt
```

La méthode est facile à utiliser et autorise des changements de topologie en fonction du choix de l'utilisateur. A l'aide d'une loi de comportement simple au niveau atomique, on obtient un comportement global fixe. L'affichage est simple, il se fait à l'aide des réseaux masses-ressorts à topologie fixe. Toutefois, ce modèle soulève plusieurs problèmes dont les plus importants sont les résultats non-conformes à la réalité et la complexité de calcul qui peut empêcher le traitement en temps réel. Un autre problème est le fait qu'il est difficile, avec ce type de représentation, d'avoir

un contrôle exact sur la forme de la surface [ST92]. La représentation géométrique de systèmes de particules n'est pas compatible avec d'autres représentations géométriques.

Un bilan des modèles discrets est présenté dans le paragraphe suivant.

Avantages et inconvénients des modèles discrets de représentation de surfaces Les représentations surfaciques discrètes présentent comme avantages majeurs la simplicité de représentation (par rapport aux modèles continus), la facilité de manipulation et de visualisation. Généralement, ces modèles permettent tout type de topologies. En choisissant correctement le type de représentation, la complexité de calcul est fortement réduite. Dans le domaine de la CAO, les modèles masses-ressorts et les systèmes de particules sont importants car ils permettent l'utilisation des lois physiques mécaniques qui aident à la simulation. Néanmoins, leurs inconvénients résident dans un trop grand degré de liberté et une impossibilité à réaliser des mesures géométriques sur la surface.

Rapporté à notre approche, les deux derniers modèles présentés - modèle masses-ressorts et système de particules - ne sont pas adaptés car ce sont des modèles physiques. Nous sommes particulièrement intéressés par les maillages triangulaires, à cause de leur généralité, de leur souplesse de représentation, de leur simplicité de mise en œuvre et de leur rapidité en temps de calcul et rendu visuel. Les maillages triangulaires ne sont pas forcément le modèle au cœur du système DIJA. Ils sont nécessaires d'une part dans le cadre de la visualisation, d'autre part dans le cadre des déformations aux vues de leur caractéristiques. DIJA est capable d'accepter tout type de représentation, grâce à son architecture multi-niveaux, présentée dans le deuxième chapitre.

Les algorithmes qui passent les choix de l'utilisateur au modèle interne de représentation correspondent à des différentes techniques de modélisation dans le domaine de la CAO, présentées dans la section suivante.

1.2.2 Techniques de modélisation

Les techniques de modélisation géométrique correspondent aux algorithmes intrinsèques aux systèmes de CAO utilisés pour répercuter les actions de l'utilisateur sur le modèle. Ces techniques correspondent aux outils de modélisation proposés à l'utilisateur par le système. Danesi [Dan02] propose une classification de ces méthodes en se basant sur une approche interne. Cette approche représente la méthodologie de modélisation et est considérée comme une doctrine fédératrice des outils de modélisation. Elle peut être qualifiée comme :

- constructive,
- descriptive ou
- modificative.

Les trois techniques de modélisation sont détaillées dans les paragraphes suivants.

L’approche constructive Elle représente l’approche classique utilisée dans les systèmes actuels de CAO. Elle correspond à une approche *bottom-up* (ascendante) qui permet de concevoir l’objet étape par étape. Cette méthode utilise des outils de construction (qui permettent de construire un objet en partant de rien [Gai96]) et des outils de combinaison (qui permettent de construire un objet à partir de plusieurs éléments primitifs [Gar91, GP96, FvDF⁺00] (cylindre, sphère, etc.)).

L’approche descriptive Cette approche correspond à la modélisation implicite. L’utilisateur décrit sa forme soit géométriquement (par exemple par esquisses [BCF94, EHBE97]) soit fonctionnellement. C’est ensuite au système d’en déduire la forme.

L’approche modificative Ceci est une approche de modélisation par déformations qui utilise des outils de modification. Ces outils permettent de perturber la forme d’un objet existant en altérant sa surface localement [Par77] ou globalement [Bar84, SP86]. Les outils de déformation sont des outils de modélisation puissants qui permettent la modification des formes à un niveau haut ; ils seront introduits dans la dernière section de ce chapitre.

Avantages et inconvénients La technique constructive est grandement utilisée, bien qu’elle soit intrinsèquement non-intuitive. Cette méthode requiert la connaissance du modèle interne et son fonctionnement de la part de l’utilisateur. Ce dernier doit maîtriser les outils que le système lui propose et doit être familier avec les éléments primitifs et leur applicabilité. L’utilisateur doit réaliser un processus complet de pré-modélisation en même temps que la planification de la tâche de modélisation (outils, actions, etc.). Cette technique ne s’adapte pas à un utilisateur novice et impose une formation préalable. En outre, la tâche lourde de la spécification géométrique exacte constitue un obstacle pour la productivité de l’utilisateur et ne l’amène pas aux étapes essentielles de design du cycle de vie du produit. D’un autre côté, l’approche descriptive gère le processus de planification par l’interprétation d’une description synthétique de la forme désirée. Ces techniques sont encore conceptuelles et pas assez opérationnelles pour constituer la base d’un

système de CAO. Finalement, seule l'approche modificative offre des aspects intuitifs du point de vue de la technique de modélisation. Cette technique utilise une perspective synthétique comme base du processus de modélisation. Utiliser le raisonnement synthétique de l'utilisateur [KP00] plutôt que ses connaissances transgresse la difficulté du processus de modélisation. Avec cette technique, les outils de déformation sont beaucoup utilisés dans des contextes comme le graphisme et l'animation [SF98].

L'approche présentée dans cette étude se fonde sur une méthodologie nouvelle qu'on appelle « synthétique » et qui utilise des outils de modification dans un contexte CAO pour raffiner une forme initiale basique. Cette méthode a été développée dans le cadre du projet DIJA et sera présentée dans le deuxième chapitre.

Un état de l'art des techniques de déformation de surfaces est présenté dans la section suivante. On rappelle que notre objectif est de trouver une méthode qui applique des déformations purement géométriques localement sur des surfaces discrètes en intégrant des contraintes métier en temps réel.

1.3 Etat de l'art des techniques de déformation de surfaces

La modélisation géométrique des surfaces constitue un élément de base du domaine de la CAO. Chacun des systèmes de CAO commerciaux a un modéleur géométrique intégré, une implémentation des techniques de modélisation qui donne à l'utilisateur la possibilité de créer, représenter et manipuler des formes géométriques. En ce sens, les déformations constituent un outil puissant. Souvent, la description fonctionnelle ou géométrique de l'objet modélisé ne peut pas être connue. Il est alors plus facile d'approcher le processus de modélisation en commençant par une forme de base et de la déformer conformément aux besoins du concepteur pour aboutir à la forme finale désirée. Des nombreuses méthodes de déformation de surfaces ont été développées pendant les dernières décades. Le choix de la technique à appliquer se fait selon une évaluation qui comporte plusieurs critères :

- l'efficacité en termes de stockage et coût ;
- l'intuitivité, atteinte par la simplicité d'utilisation grâce à des outils de haut niveau et par l'indépendance par rapport à la représentation de l'objet ;
- l'habilité à déformer la surface localement et globalement ;
- le nombre de degrés de liberté : une prolifération de degrés de liberté diminue la clarté et la simplicité d'utilisation ;

- la correction : qui est assurée si on évite les auto-intersections ou les silhouettes irrégulières manquant des caractéristiques de forme dans le cas des maillages polygonaux soumis à des déformations répétées.

Nous commençons le survol des techniques de déformation historiquement, en commençant par les techniques de déformation qui emploient les propriétés physiques des objets dans le paragraphe suivant.

1.3.1 Techniques de déformation basées sur des propriétés physiques des objets

Les premières études [BZWB89, Bli82] dans le domaine de la déformation de surfaces ont été réalisées par Barr et Blinn. Ils proposent des approches qui se basent sur des simulations physiques pour obtenir des formes et des mouvements réalistes [BZWB89] ou sur la description mathématique des objets [Bli82]. C’est en contrôlant la physique de l’objet qu’on obtient d’une façon mathématique les déformations désirées. Blinn représente son modèle implicitement à l’aide de la mécanique quantique. La surface est donc représentée par un ensemble d’atomes (voir Fig. 1.10) pour lesquels la fonction de densité est égale à une valeur seuil donnée :

$$F(x, y, z) = D(x, y, z) - T \quad (1.7)$$

où tous les points de la surface ont des densités d’électrons supérieures à T .

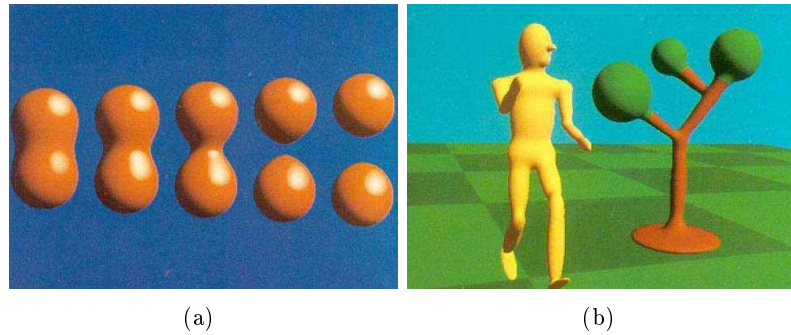


FIG. 1.10: (a) Eléments de base du modèle (*blobbs*) ; liaison s’étirant et se rompant entre deux atomes ; (b) Image générée utilisant des exposants quadriques [Bli82].

Ces techniques ne correspondent pas à notre approche car elles sont coûteuses (impossibilité d’avoir des résultats en temps réel) et peu intuitives (le contrôle sur la forme désirée est difficile à atteindre). La représentation des objets doit s’adapter aux simulations utilisées, donc elle ne

peut pas être arbitraire. C'est pour ces raisons que nous nous concentrons, dans cette étude, sur des déformations purement géométriques.

Une deuxième approche pour la déformation des surfaces est la déformation de formes libres, présentée dans le paragraphe suivant.

1.3.2 La déformation de formes libres (*Free Form Deformation* (FFD))

Une des premières méthodes pour la déformation spatiale a été proposée par Barr [Bar84]. Cette technique applique localement ou globalement des déformations à l'aide des opérations particulières que l'auteur propose (voir Fig. 1.11) : « plier » (*bent* en anglais), « torsader » (*twisted* en anglais), « effiler » (*tapered* en anglais). Ces opérations sont censées étendre les opérations conventionnelles de rotation, intersection et différence.

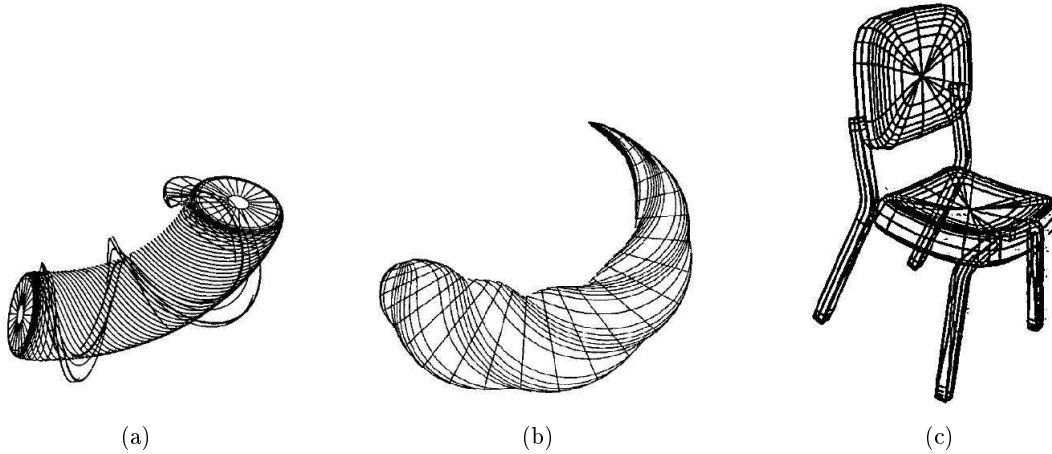


FIG. 1.11: (a) Primitive pliée torsadée; (b) Primitive pliée torsadée effilée; (c) Modèle d'une chaise composée de six primitives avec sept opérations de pliage [Bar84].

Toutefois, cette technique ne s'adresse pas aux formes arbitraires et elle ne précise pas la façon dont l'utilisateur spécifie les déformations et la façon dont les déformations locales sont réalisées.

La méthode plus connue pour la déformation spatiale, ébauchée par Bézier [Béz78], a été introduite par Sederberg et Parry [SP86] et porte le nom de déformation des formes libres (*Free Form Deformation* en anglais). Cette technique consiste à englober l'objet à déformer dans un volume ayant la forme d'un parallélépipède (volume de Bézier tri varié, voir Fig. 1.12) et d'imposer des coordonnées locales à chaque point de l'objet. Une grille de points de contrôle est ensuite définie sur le parallélépipède. L'objet est déformé en déplaçant les points de contrôle.

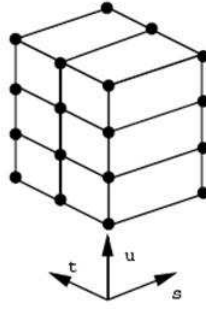


FIG. 1.12: Le *hyper patch* de base de la FFD [HHK92] ; les vecteurs s , t et u définissent le système de coordonnées local.



FIG. 1.13: FFD avec un volume de Bézier [Dub].

Le modèle de Bézier peut être remplacé par une spline [GP89] ou par une NURBS [LW94] pour améliorer la flexibilité et le contrôle sur la déformation. Le réseau de contrôle proposé par Sederberg (voir Fig. 1.13) a été étendu à des réseaux plus complexes avec des topologies généralisées [Coq90, MJ96] (voir Fig. 1.14).

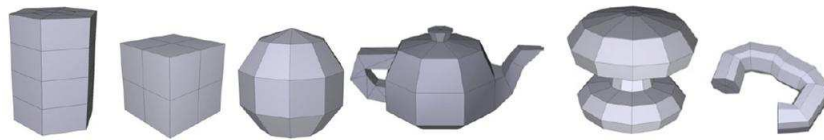


FIG. 1.14: *Patches* de contrôle avec une topologie généralisée [Coq90].

Les inconvénients majeurs de ces techniques sont le manque d'intuitivité - dû au fait qu'on n'a pas le contrôle exact sur le placement des points de l'objet - et la difficulté, voire l'impossibilité de réaliser des déformations locales. Même si une partie de l'objet est prise et englobée dans un espace afin que l'objet soit modifié localement, il est difficile de réaliser un lissage au moment de la réintégration dans l'objet global.

Lazarus [LCJ94] introduit la déformation axiale. Avec cette technique, l'utilisateur définit

un axe 3D qui est associé à l'objet à déformer (à l'intérieur ou à l'extérieur), les déformations appliquées à l'axe se répercutant sur l'objet comme le montre la figure 1.15. Les auteurs laissent à l'utilisateur la possibilité de restreindre la déformation à une partie de l'objet à l'aide de deux paramètres $ZImin$ et $ZImax$ qui démarquent la zone d'influence de la déformation. Les inconvénients de cette méthode résident dans le fait que celle-ci pourrait produire des courbures inattendues de l'objet. De plus, le coût algorithmique de cette méthode est élevé. Des problèmes de continuité géométrique entre les parties déformées et les parties non-déformées peuvent apparaître.



FIG. 1.15: Déformation axiale [Dub].

Les techniques FFD permettent une grande variété de déformations ; pourtant, elles manquent de naturel et s'avèrent souvent encombrantes, surtout si le modèle contient un grand nombre des points, car c'est la tâche de l'utilisateur de définir des points de contrôle autour de la zone à déformer. De plus, avec ces techniques, le coût est souvent trop grand pour permettre des applications interactives. Voilà pourquoi ces méthodes ne sont pas adaptées à notre projet. Pour éliminer ces inconvénients, Borrel et Bechmann [BB91] ont proposé une nouvelle technique de déformation qui remplace les réseaux de contrôle avec la manipulation directe, se basant sur des contraintes de déplacement. Cette technique, qui permet de déformer localement un objet, peut s'adapter à DIJA si on arrive à associer les contraintes de déplacement à des contraintes métier en temps réel.

1.3.3 Déformations soumises à des contraintes de déplacement

La philosophie de cette technique consiste à associer des contraintes de déplacement aux points de l'objet et ensuite à déplacer les points et la zone qui les entourent tout en respectant les contraintes et la continuité géométrique. Le modèle peut accepter n'importe quel nombre de contraintes spécifiées par l'utilisateur. Si deux contraintes sont associées à un même point dans l'espace, le système calcule la meilleure approximation de la solution. Chaque contrainte est définie comme un segment orienté qui relie deux points :

- l’origine de la contrainte (U).
- la position du point initial après la déformation (ΔU).

A chaque contrainte on peut associer une région d’influence sur la surface. Cette région est ensuite déformée selon une ou plusieurs fonctions mathématiques. La région d’influence pourrait être une sphère, un parallélépipède, un polyèdre ou un volume de *voxels*. Ce modèle est caractérisé par la fonction de déformation suivante :

$$\mathcal{D} : \mathbb{R}^n \mapsto \mathbb{R}^n, \quad \mathcal{D}(X) = X + M \cdot \mathcal{F}(X) = \tilde{X} \quad (1.8)$$

où $\mathcal{F} : \mathbb{R}^n \mapsto \mathbb{R}^m$ est une fonction qui spécifie comment la déformation se propage autour de chaque contrainte et M est une matrice de taille $n \times m$ (n représente la dimension de l’espace et m le nombre de contraintes) qui résume une transformation linéaire combinant l’interaction de toutes les contraintes. Le modèle peut être instancié de plusieurs façons. Pour des déformations de l’espace 3D, $n = 3$.

Ce modèle porte le nom de DOGME dont plusieurs extensions existent [BD93, BR94]. DOGME est facile à utiliser, les interactions de l’utilisateur étant minimales. La possibilité de déformer la surface localement à l’aide d’un rayon d’influence présente un intérêt particulier pour notre approche. DOGME réalise aussi des déformations globales à l’aide des fonctions polynomiales. La forme de la déformation dans la zone d’influence est déterminée soit par des B-splines, soit par d’autres fonctions mathématiques. Pourtant, la nature de la déformation n’est pas liée au fait qu’elle soit définie par des contraintes ; les contraintes sont seulement utilisées pour calculer les paramètres d’une fonction de déformation. Par conséquent, la forme de la déformation n’est pas corrélée avec les contraintes. Elles sont seulement respectées. Pendant le processus de déformation, le système laisse à l’utilisateur le choix d’une solution qui obéit aux contraintes. Ceci n’est pourtant pas suffisant pour que les déformations réalisées avec DOGME s’adaptent à nos buts car nous désirons contrôler au mieux les déformations. Un autre inconvénient est le fait que la continuité n’est pas toujours atteinte entre la zone déformée et le reste de la surface. Une solution pourrait être l’application des schémas de subdivision pour lisser la surface, mais ceci dépend de l’application. Cette méthode est souvent coûteuse, ce qui empêche l’interaction en temps réel.

Les fonctions mathématiques appliquées couramment sur les points de la zone à déformer sont des fonctions *decay* (ou fonctions d’altération). Ce type de fonctions est introduit dans la section suivante.

1.3.4 Fonctions *decay*

Le concept de fonctions *decay* a été introduit par Parent [Par77]. L'idée consiste à déformer une partie de la surface en appliquant une fonction d'altération sur les points qui y appartiennent. Une fonction d'altération est donc caractérisée par un *style* (voir Fig. 1.16), une *action* et une *portée*.



FIG. 1.16: Exemples de styles de fonctions *decay* [BL94] ; de gauche à droite : bombé, cloche, sommet, cône, plat.

Le *style* détermine l'aspect de la déformation, l'*action* consiste à déformer localement la surface et la *portée* spécifie le nombre de sommets voisins de la triangulation affectés par la déformation. L'amplitude de la déformation résulte donc de l'interpolation de la position des points par rapport aux frontières de la zone à déformer (voir Fig. 1.17).

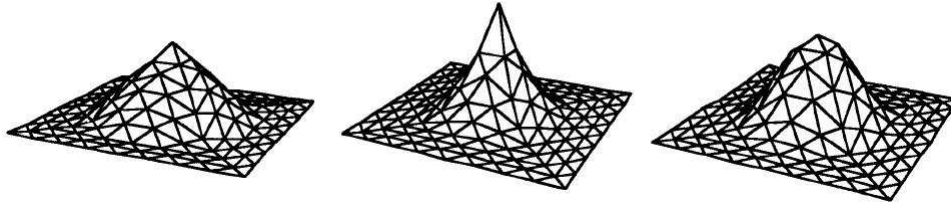


FIG. 1.17: Exemple d'application des fonctions *decay* en un point [AWW89] ; de gauche à droite : cône, cloche, sommet.

Une extension de ce modèle est vue dans [AWW89, BL94]. Les fonctions d'altération s'adaptent bien à des maillages polygonaux triangulaires. Cette technique est très rapide et simple à mettre en œuvre. Sa généralité permet d'obtenir diverses formes de déformation, en corrélation avec les fonctions mathématiques sous-jacentes. De plus, les déformations appliquées sont locales et peuvent se propager sur la surface au bon vouloir de l'utilisateur. Ceci fait que les fonctions *decay* sont un très bon candidat pour le développement des outils de déformation lorsqu'elles sont corrélées avec des règles métier précises. Leur utilisation dans un système de CAO orienté métier est décrite dans le troisième chapitre.

Un des problèmes courants soulevés par les techniques de déformation locales est la discontinuité entre la zone déformée et le reste de la surface. Une solution corrigeant cet inconvénient peut être l'application des surfaces de subdivision. Une nouvelle méthode de déformation de formes libres, qui applique le principe de surface de subdivision, est introduite dans la section suivante.

1.3.5 FFD avec Surfaces de Subdivision

Cette technique de déformation, introduite par Jieqing Feng [FSJ⁺06], propose une surface de subdivision avec une topologie arbitraire comme outil de déformation (voir Fig. 1.18). C'est à dire que, dans ce cas, le volume de Bézier est remplacé par une surface de subdivision. Les auteurs motivent ce choix par le fait que les surfaces de subdivision représentent un pont entre les maillages discrets et les surfaces continues. L'outil de déformation est un maillage polygonal et appartient à la classe des méthodes de déformation avec des outils 2D.

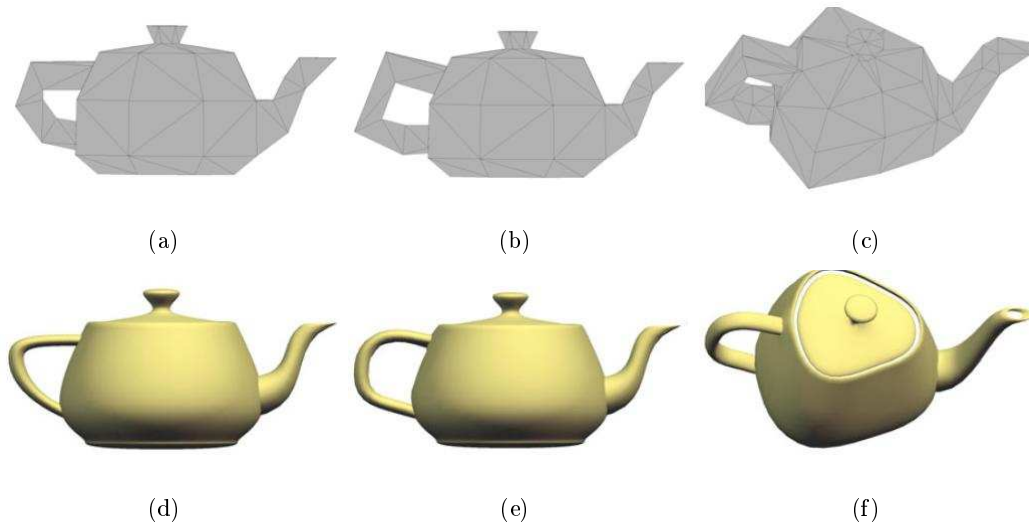


FIG. 1.18: Déformations locale et globale d'une théière [FSJ⁺06] : (a) Maillage de contrôle initial ; (b) Anse modifiée localement ; (c) Corps et bec modifiés globalement, (d) Objet initial ; (e) Déformation locale contrôlée par (b) ; (f) Déformation globale contrôlée par (c).

L'objet à déformer est englobé dans l'espace de déformation selon la règle de plus proches voisins. La propriété intrinsèque de multi résolution des surfaces de subdivision se répercute sur la méthode de déformation proposée.

Cette méthode est simple à utiliser, assez intuitive, et la complexité algorithmique est réduite par rapport aux autres techniques FFD. Elle requiert des représentations polygonales triangu-

laire de la surface à déformer, donc le modèle stocké n'a pas une représentation complexe et lourde. Des déformations globales et locales sont possibles avec cette méthode, les dernières se réalisant en appliquant des schémas de subdivision [DLG90, Loo87].

Le principe de raffinement de surfaces de subdivision pourrait être une solution intéressante pour notre approche d'un point de vue lissage. Cette technique donne des résultats agréables visuellement, pourtant, le problème de l'intégration des règles métier reste toujours ouvert. On a déjà vu que les contraintes de déplacement ne suffisent pas pour obtenir un contrôle total sur les déformations obtenues. Il nous faudrait un outil plus puissant pour imposer des règles orientées métier au processus de déformation. Des contraintes géométriques pourraient être imposées à condition que celles-ci correspondent bien à des règles métier.

1.3.6 Déformations soumises à des contraintes géométriques

Les contraintes géométriques ont été grandement appliquées dans des domaines comme la conception et la modélisation géométrique assistée par ordinateur et l'animation, où des déformations contraintes s'imposent. Les contraintes géométriques se classifient en deux grandes classes. La première correspond à la classe des contraintes linéaires [BB89, Fow92, FB93, Gle92, WW92] - comme la position, la tangence, l'orthogonalité et la symétrie - permettant la manipulation interactive des surfaces *free form*. Ce type de contraintes offre l'avantage d'être efficace du point de vue calcul. D'un autre côté, les contraintes non-linéaires - l'aire [Elb01], le volume [RSB96], la convexité [KS95, PE98], les contraintes de courbure [CG91, Gre96, PGL⁺02] - sont complexes du point de vue de calcul, ce qui fait que leur utilisation pour la manipulation interactive des formes est limitée. La contrainte volumique est importante pour des applications industrielles mais pas seulement. Lassiter [Las44] l'applique pour augmenter le réalisme en déformant les personnages dans des animations. Plusieurs études ont été réalisées pour améliorer la déformation avec conservation du volume. Cette contrainte a été appliquée pour des solides *free form* [RSB96], pour la FFD [HML], pour la déformation spatiale [vFTS06], pour les maillages [LCOGL07] et pour les maillages multi résolution [BK, SHB07, SHBE08]. Une méthode pour la déformation avec conservation volumique est proposée, dans le quatrième chapitre, pour modéliser des flacons de parfum et des tablettes arrière de voiture. Les deux applications sont présentées dans le troisième chapitre et montrent les principes des outils orientés métier.

Pour interagir avec la surface et la déformer, l'utilisateur a besoin d'une interface Homme-Machine bien adaptée à ses buts. Il faut tout d'abord qu'il puisse visualiser le modèle et ensuite

qu'il ait la possibilité d'y appliquer des déformations de manière successive, localement ou globalement, à la manière d'un sculpteur qui modèle un objet en argile, pour aboutir au résultat désiré. Ceci se fait à l'aide des outils d'interaction qu'on appelle outils virtuels et qui sont présentés de manière générale dans la section suivante.

1.4 Outils virtuels de déformation de surfaces

Les techniques de modélisation par déformation ont à la base des outils de modification qui permettent de changer la forme d'un objet en induisant des déformations sur sa surface, soit localement, soit globalement. Cette approche est similaire à une technique de sculpture virtuelle (*Computer Sculpting* en anglais) qui émule les techniques traditionnelles de la sculpture : l'utilisateur modélise son objet étape par étape, par déformation locale ou globale de sa surface, à la manière d'un sculpteur. De nombreux systèmes ont adopté cette approche en créant des outils virtuels de sculpture spécifiques [BDHO92, Cla76, GH91, GLS95, Kam97, LG94, MN94]. Nous étudions dans cette section les outils qui rendent possible la modification surfacique. Selon Bill [BL94], un système de modélisation qui implémente la sculpture virtuelle doit posséder les caractéristiques suivantes :

- la modélisation se fait sur des surfaces de formes libres ;
- le système est intuitif, grâce à la mise en œuvre des techniques qui émulent les techniques traditionnelles de sculpture ;
- les réponses aux interactions se font en temps réel ;
- le modèle est manipulé et modifié en utilisant des outils virtuels.

On verra par la suite les techniques d'interaction qui permettent l'application des outils virtuels.

1.4.1 Techniques d'interaction

Dans un système de CAO, une surface a couramment une double représentation physique :

- une représentation géométrique (habituellement un B-Rep avec un historique CSG) et
- une représentation explicite, approximative de la représentation géométrique (le plus souvent un maillage triangulaire), utilisée pour la visualisation de l'objet.

La déformation de la surface peut ainsi être appliquée soit sur le modèle exact, soit sur l'approximation de la surface. Dans le premier cas, les déformations sont appelées « normalisées » (appliquées sur le modèle inhérent) ; dans le deuxième cas, les déformations sont appelées « non-

normalisées » (appliquées directement sur les points de la triangulation). Ces deux types de déformation sont explicités dans les sections suivantes.

Déformations normalisées

Ces techniques manipulent directement le modèle de représentation de la surface et peuvent être soit explicites, soit implicites. La déformation, dans ce cas, se fait en manipulant les degrés de liberté de la représentation mathématique inhérente, telle que la position des points de contrôle d'une surface NURBS.

Ces techniques ont une propriété essentielle pour un système de CAO : lorsqu'une surface vérifie une propriété liée à son modèle mathématique, celle-ci sera toujours valide après l'application d'une telle déformation. Cette propriété soulève, pourtant, le principal inconvénient de ces méthodes : elles demandent des connaissances mathématiques, comme, par exemple, le résultat d'un déplacement d'un point de contrôle, etc. Malgré le fait qu'on peut associer un équivalent visuel à un paramètre mathématique, l'intuitivité de ces techniques n'est pas atteinte puisqu'elles nécessitent d'importantes connaissances mathématiques. Par conséquent, ces méthodes s'inscrivent dans la même démarche que l'approche constructive, qui demande des connaissances pré-requises.

Déformations non-normalisées

Elles correspondent aux techniques de la sculpture en proposant des outils intuitifs dont l'activation conduit à des résultats identiques à la réalité. Ce type de manipulation s'appuie sur une approximation polyédrique (maillage triangulaire) de la surface de l'objet à déformer. Les premiers systèmes basés sur des maillages polygonaux ont été ceux de Parent [Par77], Allan [AWW89], Leblanc [LKMTT91] et Elson [Els90b, Els90a]. De nombreuses études ont été faites sur les outils de manipulation directe. Le concept de sculpture virtuelle appliquée à des maillages polygonaux a été introduit par Parent [Par77]. Cette technique est devenue classique dans le domaine du graphisme et la modélisation, y compris la CAO. A l'heure actuelle, elle est très développée et utilise de nouvelles technologies de dialogue (tel que le retour haptique) [Gai96, MCQ05].

Bien que la sculpture virtuelle se fasse habituellement directement, sur les points d'une triangulation qui approche bien la surface, il existe, tout de même, des techniques non-normalisées de déformation qui sont indirectes. Ce type de techniques utilise un outil de déformation comme la base d'interaction et qui englobe l'objet à déformer. L'utilisateur manipule l'outil et les modifications se répercutent sur l'objet englobé (voir Fig. 1.19).

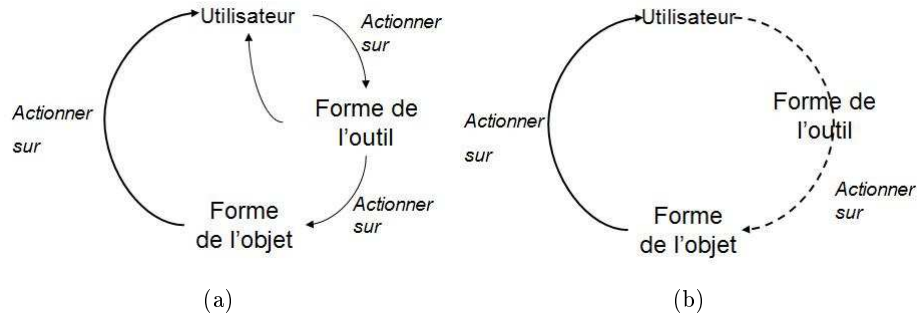


FIG. 1.19: Interactions par (a) déformations directes et par (b) déformations indirectes.

Ce type d'outils se classifie en quatre classes, en fonction de leur degré :

- outils 3D : ce sont les volumes de contrôle spécifiques à la FFD ;
- outils 2D : ce sont représentés par des surfaces [FSJ⁺06] ;
- outils 1D ou axiaux : ce sont représentés par une ligne ou une courbe. Ce type d'outils a été introduit par Barr [Bar84] et étendu par les techniques *AxDF* [LCJ94] et *Wires* [SF98] ;
- outils directs : l'outil est indirectement déformé en conformité avec une déformation locale directe du maillage. La déformation reste toujours indirecte puisque la modification globale de l'objet est déduite de la déformation de l'outil. Dans ce cas l'outil est transparent à l'utilisateur, qui actionne seulement sur la surface. Ce type d'outil peut être n -dimensionnel [BB91].

Ce type de techniques présente des inconvénients majeurs. Premièrement, la complexité algorithmique est élevée, car il est nécessaire de calculer les coordonnées des points de la surface par rapport aux réseaux de contrôle. Ces méthodes soulèvent aussi des problèmes dus à des considérations mathématiques. La spécification des déformations nécessite une bonne compréhension des B-splines. Des problèmes de continuité peuvent apparaître dans le modèle et la validité de la surface n'est pas sûre, ce qui est inacceptable d'un point de vue CAO.

En conclusion, les deux types de déformation ne sont pas adéquats pour une approche orientée métier. D'un côté, les déformations normalisées nécessitant des connaissances mathématiques solides, ne permettent pas une approche intuitive. De l'autre côté, les déformations non-normalisées, plus intuitives, conduisent à des divergences dans la géométrie de l'objet modélisé.

L'idée de notre approche est d'offrir à l'utilisateur une interface indépendante du modèle utilisé. Après avoir appliqué une déformation, le système DIJA propose à l'utilisateur plusieurs solutions. Le deuxième chapitre explique comment les outils virtuels de déformation sont définis et

appliqués sous DIJA de manière générale, en s'appuyant sur des outils 2D. Le troisième chapitre présente une extension de ce qui a été fait avant, un nouveau type des outils virtuels orientés métier en trois dimensions qui s'applique sur des maillages polygonaux triangulaires.

La section suivante introduit le concept d'outil virtuel de déformation.

1.4.2 Outils de déformation

Qu'est ce que c'est qu'un outil virtuel de déformation ?

L'outil virtuel de déformation est un objet géométrique englobant une fonction mathématique.

Il se caractérise par trois éléments :

- la forme de l'outil,
- l'action de l'outil et
- sa portée ou son rayon d'influence.

La forme de l'outil virtuel s'exprime à l'aide des fonctions mathématiques, telles que, par exemple, les fonctions super quadriques. L'action concerne la façon dont l'outil perturbe la surface de l'objet ; cet aspect sera discuté par la suite. Quant à la portée de l'outil, celle-ci peut s'exprimer de plusieurs façons - en distance euclidienne ou en nombre d'arêtes à partir du point à déplacer (voir Fig. 1.20) - et représente le nombre des points voisins de la surface affectés par l'application de l'outil.

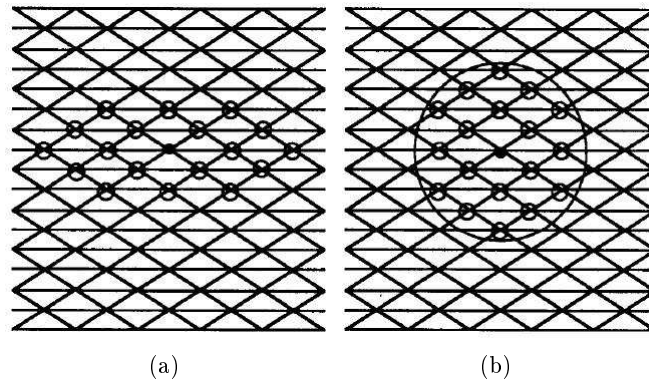


FIG. 1.20: Région d'influence [AWW89] : (a) En nombre d'arêtes ; (b) En distance Euclidienne.

L'application d'un outil virtuel

L'utilisateur applique un outil virtuel pour déformer un maillage polygonal. Les outils virtuels sont stockés dans des bases de connaissances des systèmes de modélisation. Ils peuvent être soit basiques, soit complexes, en fonction des besoins de modélisation. Les utilisateurs expérimentés

ont la possibilité de créer de nouveaux outils à partir des outils préexistants et les enregistrer dans les bases de connaissance correspondantes. Kil propose une librairie d'outils 3D appelés « brosses déformantes » (*warp brushes* en anglais) [KRK⁺06] et donne la possibilité à l'utilisateur de l'étendre à des outils plus compliqués simplement en transformant les maillages polygonaux déformés par une « brosse déformante » en des « brosses déformantes » plus complexes.

Les outils virtuels assurent l'intuitivité par un haut niveau d'abstraction. Les outils doivent s'adapter au mieux au domaine et aux connaissances de la personne qui les manipulent. L'utilisateur est guidé dans ce processus seulement par son savoir-faire, tous les détails mathématiques inhérents à la modélisation lui sont transparents. Ils permettent l'application intuitive de plusieurs primitives de façon simultanée. Des outils virtuels qui réalisent des opérations booléennes ont été créés pour la modélisation des solides [GH91, Par77]. Les premières opérations *pull-push* sur un point de la surface ont été proposées par Brewer [BA77]. Cette méthode utilise un outil planaire pour déplacer les points de contrôle d'une surface paramétrique. Szeliski [ST92] propose des outils pour appliquer et annuler des forces dans un système de particules. D'autres outils combinent leurs fonctions intrinsèques avec des dispositifs d'entrée. Les outils proposés par Gaylean [GH91] sont des objets volumiques définis dans le même espace que le modèle et incluent : un outil de soustraction « routage » (*routing* en anglais), qui enlève de la matière, un outil d'addition « tube de dentifrice » (*toothpaste tube* en anglais) pour ajouter de la matière et un outil « pistolet à chaleur » (*heat gun* en anglais) qui fait fondre la matière. Même si les outils proposés par ce système sont intéressants, ce dernier reste limité à cause de difficultés de contrôle et d'intuitivité créées par les dispositifs d'entrée 3D utilisés. Le système SAM-IAM de Bill et Lodha [BL94] propose des outils qui englobent des fonctions *decay* pour sculpter des maillages polygonaux. Les actions d'un tel outil sur la surface sont les suivantes :

- « définir zone » : cette action assure la localité de la déformation en délimitant la zone à déformer du reste de la surface ; l'utilisateur indique une valeur qui correspond au rayon d'influence de la déformation ;
- « pousser » : c'est l'action qui pousse un point à l'intérieur de la surface, en entraînant avec lui les points qui l'entourent dans la région d'influence ;
- « tirer » : cette action tire le point initial et les points qui l'entourent à l'extérieur de la surface.

La figure 1.21 montre les opérations « pousser - tirer » appliquées à un maillage triangulaire.

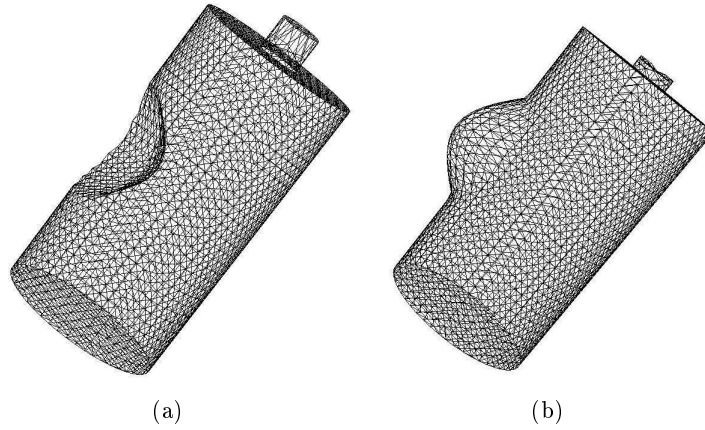


FIG. 1.21: Opérations « pousser-tirer » appliquées à un maillage triangulaire avec un outil virtuel : (a) Résultat de « pousser » ; (b) Résultat de « tirer ».

Les actions « pousser - tirer » (*push - pull* en anglais) représentent un moyen simple et intuitif pour déformer une surface. Si en plus des fonctions mathématiques et des contraintes géométriques spécifiques sont associés, des outils puissants sont obtenus, qui pourraient correspondre à la philosophie DIJA. Le troisième chapitre explique l'intérêt de coupler ce type d'outils avec des règles métier bien précises pour la modélisation des surfaces en 3D. DIJA propose sa propre librairie d'outils virtuels, qui ont un degré élevé d'abstraction et qui sont censés donner à l'utilisateur la liberté d'exprimer son savoir-faire dans un environnement métier intuitif en appliquant des déformations orientées métier sur la surface de l'objet. Jusqu'à présent, seuls des outils appliqués à des éléments 2D de la surface ont été implémentés sous DIJA. La création et l'application des outils inhérents à notre système, qui sont à la base de la modélisation dite « synthétique », sont détaillés dans le deuxième chapitre.

1.5 Conclusion

Un système de CAO est complètement défini quand on connaît ses trois entités de base : le modèle, le modéleur et l'interface Homme - Machine (voir Fig. 1.22). L'objectif de cette étude est de proposer des outils pour améliorer le couple modélisation - interaction en s'appuyant sur un modèle simple et facile à utiliser. La finalité est de fournir à n'importe quel utilisateur - expérimenté ou novice - la possibilité de choisir, utiliser et manipuler interactivement les techniques de modélisation proposées par le système d'une manière simple, rapide et intuitive.

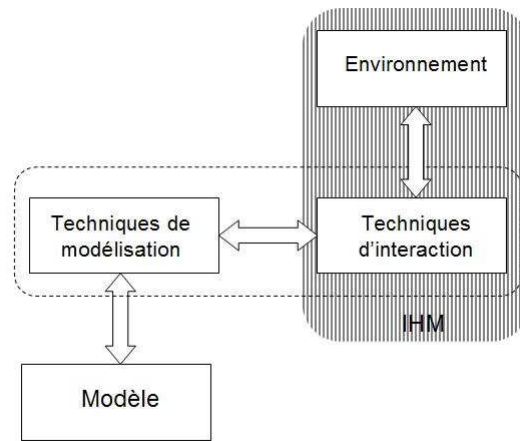


FIG. 1.22: Organisation d'un système de modélisation ; interaction entre les différents modules qui le composent.

La première partie de ce chapitre est dédiée à l'étude de différents modèles de représentation de surfaces et aux techniques de modélisation. Après une évaluation de ceux-ci du point de vue performance, intuitivité et correction, nous avons opté pour la modélisation par déformation des surfaces représentées par des maillages polygonaux triangulaires. Le système de modélisation proposé, DIJA, accepte n'importe quel modèle à condition que celui-ci s'adapte à notre approche, et plus précisément, à condition qu'on puisse obtenir une triangulation de la surface qu'on modélise par déformations successives et qu'à la fin du processus on puisse reconvertir le résultat en modèle initial. Les maillages polygonaux représentent un bon compromis entre la complexité algorithmique et le stockage ; de plus, ceux-ci sont bien adaptés pour la sculpture virtuelle et permettent à l'utilisateur de se concentrer sur les outils de déformation plutôt que sur les propriétés de la surface.

La modélisation par déformation est une technique très puissante de modélisation qui correspond bien à l'approche « synthétique » de DIJA. Un état de l'art de ce type de techniques est fait dans la deuxième partie de ce chapitre. L'étude qui a été faite dans ce premier chapitre montre qu'il est possible de s'appuyer sur des fonctions *decay* dans notre démarche à condition de les coupler adéquatement avec des contraintes géométriques.

Les contraintes géométriques et les fonctions *decay* sont intégrées dans des outils virtuels de déformation pour passer les actions de l'utilisateur au modèle géométrique. Les outils virtuels de déformation et les techniques d'interaction sont étudiés dans la troisième partie de ce chapitre.

Ainsi, nous terminons ce chapitre en proposant la mise en pratique des principes étudiés dans

un modéleur de surfaces discrètes orienté métier, en gardant à l'esprit les idées qui sont à la base de l'approche « synthétique » DIJA. Nous continuons ce manuscrit en présentant le contexte et les principes nécessaires à la compréhension de ce travail dans le deuxième chapitre.

Chapitre 2

Contexte : le projet DIJA

2.1 Introduction

Actuellement, les innovations dans le domaine de la CAO visent des technologies révolutionnaires qui apportent des avancées significatives en ce qui concerne la vitesse, le graphisme et l'utilisation caractéristiques à un système de CAO [MYS99, LX01, RK03, CB07, RLR07, TF08, MCL08, FUN⁺08]. Pourtant, les systèmes de CAO commerciaux actuels nécessitent encore des améliorations. Ils restent toujours monolithiques et sont basés sur des modèles géométriques complexes. Ces systèmes proposent des interfaces Homme-Machine complexes qui représentent souvent un obstacle entre le processus de modélisation et les utilisateurs non-expérimentés.

En se basant sur la définition donnée dans le chapitre précédent, nous avons créé DIJA [Dan02, Den03, Gar05, Hes05, Hil08] - un système de CAO innovant, dont l'objectif est de fournir à l'utilisateur - qu'il soit néophyte ou expert - un système de CAO distribué sur Internet, collaboratif, fonctionnel et intuitif à la fois. Seuls les aspects fonctionnel et intuitif sont pris en compte dans cette étude. Nous nous basons, dans cette démarche, sur une approche de modélisation descendante (*top-down* en anglais) qu'on appelle « synthétique », qui est à l'opposé des techniques utilisées à l'heure actuelle. Cette approche, présentée dans la section 2.2, suit la façon de réfléchir de l'utilisateur, en allant d'un raisonnement global à un raisonnement local dans le processus de conception. Cette façon de modéliser lui permet d'obtenir la forme finale de l'objet désiré en lui appliquant des raffinements successifs. Ceci conduit à plusieurs niveaux de réflexion qui se traduisent, sous DIJA, en une architecture multi-niveaux qui est détaillée dans la section 2.3. Le raffinement de la forme se fait en induisant des déformations à l'aide des outils dédiés. La section 2.4 introduit la technique de modélisation par déformation et les outils 2D sous-jacents, développés sous DIJA.

2.2 La philosophie DIJA

2.2.1 L'approche synthétique

L'approche dite « synthétique » permet de recréer l'environnement naturel habituel de l'utilisateur grâce à une interface intuitive et à une technique de modélisation descendante. L'accent est mis sur le raisonnement, le savoir-faire et les gestes de l'utilisateur qui sont censés être similaires à ceux qui pourraient être effectués dans le monde réel. L'utilisateur ne doit plus se soucier des aspects mathématiques inhérents à la modélisation. Il doit simplement avoir une représentation mentale de ce qu'il veut obtenir et, en partant d'une forme initiale qui approche plus ou moins bien l'objet désiré, lui appliquer successivement des outils dédiés afin d'aboutir au résultat final. Le choix de la forme de départ est le premier pas dans le processus de conception qui détermine la convergence vers le résultat désiré. Nous détaillons cet aspect dans le paragraphe suivant.

Le choix de la forme initiale

La forme initiale peut résulter soit des fonctions de l'objet, soit de sa géométrie. Dans le premier cas, l'utilisateur spécifie les fonctions de l'objet à concevoir, le système étant capable de lui proposer un certain nombre de formes prédéfinies qui correspondent aux fonctions spécifiées. Le passage de la fonction à la forme est possible grâce à des scripts enregistrés dans la base de connaissance dans l'architecture du système DIJA (voir section 2.3). Dans le deuxième cas, l'utilisateur choisit une forme de départ dans une base de données. Ce choix peut être fait de plusieurs façons :

- l'utilisateur débute avec une forme initiale choisie dans une base de données qui contient des objets de base (sphère, cylindre, boîte, etc.), mais aussi des objets plus complexes (pièces mécaniques prédéfinies, etc.) ; cette base de données est adaptée au domaine d'application de l'utilisateur ;
- l'utilisateur a la possibilité de choisir dans cette base de formes un englobant de la pièce qu'il veut modéliser (sphère, cylindre, boîte, etc.) ;
- l'utilisateur peut choisir une des pièces déjà modélisées et enregistrées dans la base de données.

En fonction de l'expérience de l'utilisateur, cette forme peut être plus ou moins proche de l'objet final, ce qui entraîne un nombre plus ou moins grand de raffinements. Enrichir le contenu de la base de formes de départ peut réduire considérablement le nombre d'étapes lors de la conception. Le raffinement de la forme se réalise en appliquant des outils qui englobent, entre

autre, des opérations élémentaires, opérations que nous décrivons dans le paragraphe suivant.

Opérations élémentaires

Les opérations élémentaires sont en nombre de trois :

- la **division** : elle permet de diviser l'objet en plusieurs parties ; pour cela, l'utilisateur définit un plan de coupe associé à l'objet et dessine un ou plusieurs contours dans ce plan ; ceci peut aussi se faire directement sur la surface de l'objet ; chaque nouveau contour définit une nouvelle face sur laquelle l'utilisateur peut interagir localement ;
- la **déformation** : elle représente le cœur de l'interaction sous DIJA et permet de modifier la forme de l'objet globalement ou localement ;
- la **transmutation** : elle permet de changer la topologie d'un objet ou d'une partie d'un objet (transformer un cylindre en boîte (Fig. 2.1), etc.). La forme résultante est déduite des paramètres de la forme initiale ; dans le cas où cette opération s'applique uniquement sur une partie de l'objet, le système doit considérer des problèmes liés aux raccordements de surface.

Les trois opérations élémentaires DIJA représentent les fonctions des outils de base de la modélisation « synthétique ». Leur combinaison conduit à des outils complexes, d'où la puissance de la modélisation « synthétique ».

Nous illustrons, par un exemple simple, les étapes de la modélisation synthétique et l'application des opérations élémentaires dans le paragraphe suivant.

Un exemple de modélisation « synthétique »

Un exemple simplifié de modélisation « synthétique » d'une colonne dans le domaine de l'architecture est illustré dans la figure 2.1. La forme de base est représentée par un cylindre auquel l'utilisateur applique les trois opérations élémentaires afin d'obtenir une colonne :

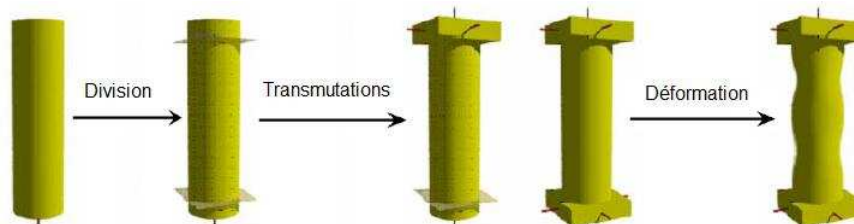


FIG. 2.1: Conception d'une colonne avec l'approche « synthétique » en appliquant des opérations élémentaires.

- le cylindre est divisé en trois parties qui correspondent aux différentes fonctions de la colonne : le chapiteau (la partie supérieure), le fût (la partie centrale) et la base (la partie inférieure) ;
- les cylindres qui correspondent au chapiteau et à la base sont transmutés en parallélépipèdes ;
- le fût est déformé par esquisse pour lui induire un design désiré.

Le paragraphe suivant présente les avantages de la méthode « synthétique » comparé à la méthode constructive.

Comparaison avec la méthode classique constructive

L'exemple présenté dans le paragraphe précédent est très illustratif pour montrer l'intérêt de la méthode « synthétique » par rapport à l'approche constructive classique en quelques remarques :

- l'approche synthétique se base sur des déformations et non pas sur des opérations Booléennes, ce qui favorise la cohérence et la facilité d'utilisation surtout pour un utilisateur novice ;
- les contraintes de placement (sur l'exemple, les contraintes qui imposent que les trois parties soient construites autour du même axe) sont automatiquement prises en compte avec l'approche synthétique, alors que dans la méthode constructive l'utilisateur doit les définir lui même ;
- le design du fût peut se faire à l'aide des outils métier qui intègrent des paramètres liés à la construction ; dans l'approche constructive, la déformation du fût ne peut se faire que selon sa définition, en modifiant le profil, ce qui est une tâche difficile pour un utilisateur non-expérimenté.

On a déjà vu que, une fois la forme de départ définie, l'utilisateur peut la modéliser, de proche en proche, pour obtenir finalement la forme désirée. Pour cela, il a à sa disposition un panel d'outils de déformation qui correspondent à son métier et qui sont une combinaison de trois outils de base DIJA (division, déformation et transmutation). Grâce à ces outils, l'utilisateur immerge dans son environnement de travail et applique directement son savoir-faire sur l'objet modélisé. Pourtant, le problème n'est qu'à moitié résolu. L'utilisateur dispose des outils dédiés, il a son savoir faire, mais il faut aussi lui suggérer comment les appliquer à l'objet modélisé.

Classiquement, dans la modélisation constructive, l'utilisateur doit agir, visuellement ou mathématiquement (par exemple, par des opérations Booléennes), sur des parties de l'objet, sur des

caractéristiques de formes ou directement sur les composants de base et/ou sur les paramètres de ces éléments : les points, les cotations, les points de contrôle, etc. Dans l’approche « synthétique » DIJA, tous ces éléments sont transparents à l’utilisateur, sauf si la modélisation est faite par un expert et qu’il désire avoir accès à ces éléments. Pour que l’intuitivité soit atteinte, on doit tenir compte du fait que l’utilisateur ne connaît que ce qu’il voit. Nous nous basons donc, dans notre approche, sur des entités visibles appelées « éléments de dialogue ». Ce sont des éléments qui retracent la silhouette de l’objet, facilement identifiables par un utilisateur expérimenté aussi bien que par un utilisateur novice. Les éléments de dialogue sont présentés dans la section suivante.

2.2.2 Éléments de dialogue

Les éléments de dialogue représentent, comme on l’a déjà annoncé dans le premier chapitre, une nouvelle approche pour représenter l’objet modélisé et pour agir sur ce dernier. La représentation à l’aide d’éléments de dialogue est détaillée dans cette section. Les outils d’interaction qui s’appliquent aux éléments de dialogue sont introduits dans la section 2.4.

Il existe, sous DIJA, trois types d’éléments de dialogue (voir Fig. 2.2) qui permettent d’avoir des détails visuels sur l’objet modélisé :

- la **face** (3D) ;
- la **ligne caractéristique** et le **contour caractéristique** (2D) ;
- la **fibre** (2D).

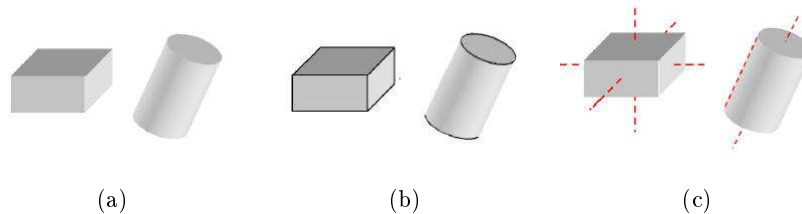


FIG. 2.2: Éléments de dialogue sur deux objets de base (parallélépipède et cylindre) : (a) l’objet et ses faces ; (b) l’objet avec les lignes et contours caractéristiques ; (c) l’objet avec ses fibres.

L’élément de dialogue « face » représente la totalité ou une partie d’une surface homogène. C’est un terme générique qui peut être instancié soit par une surface qui représente la peau de l’objet, soit par une face d’un modèle B-Rep, soit, dans le cas le plus simple, par un point. L’utilisateur peut agir, pour raffiner sa pièce, sur la totalité de l’objet ou bien très localement, en définissant des zones d’interaction sur la surface de celui-ci. L’utilisateur a aussi la possibilité d’interagir directement sur les points de la surface et de redéfinir leurs coordonnées. Pour

modifier localement la surface, l'utilisateur a besoin de délimiter la zone à perturber du reste de la surface. Cette séparation se fait à l'aide de lignes et de contours (ensembles fermés de lignes) caractéristiques. Ces entités représentent des lignes visibles sur la surface de l'objet. L'utilisateur peut donc les utiliser pour faire la jonction entre deux faces, mais aussi pour déformer l'objet. Dans le dernier cas, les perturbations appliquées à des lignes ou contours caractéristiques se répercutent sur l'ensemble ou sur une partie de l'objet.

Les fibres représentent des caractéristiques globales de l'objet et forment de manière générale la silhouette de l'objet.

Le modèle géométrique sous-jacent à la représentation par éléments de dialogue est un modèle B-Rep étendu [Dan02] qui ajoute au modèle classique B-Rep des concepts supplémentaires (voir Fig. 2.3) pour représenter un objet :

- le concept de face : présent dans le modèle B-Rep classique, elle peut être définie par un ensemble de triangles ;
- le concept de ligne caractéristique : c'est un ensemble d'arêtes, parmi celles stockées dans le modèle de représentation, ou un ensemble de lignes caractéristiques ;
- le concept de contour caractéristique : c'est un ensemble de lignes caractéristiques formant un contour fermé ;
- le concept de fibre.

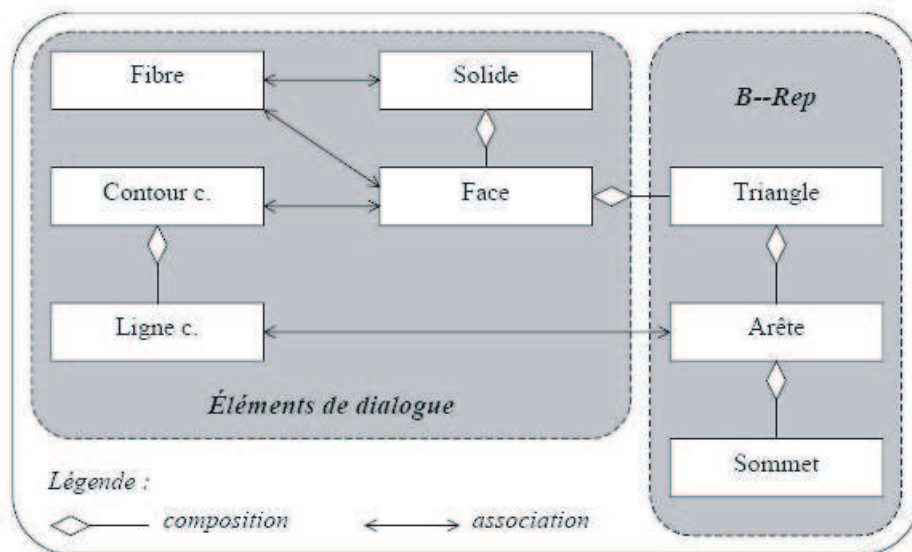


FIG. 2.3: Modèle B-Rep étendu [Dan02].

Ces éléments complètent au niveau topologique les informations contenues dans le modèle, par l'ajout des relations Triangles - Faces et Faces - Faces à un graphe B-Rep classique restreint à : Faces - Arêtes - Sommets. Ce modèle est explicite et consiste en une triangulation de la surface de l'objet à modéliser. Il contient à la fois le modèle de visualisation et le modèle à la base de l'interaction Homme - Machine.

Jusqu'à présent l'approche « synthétique » a été implémentée uniquement pour des déformations réalisées sur des éléments de dialogue 2D. Le concept de face existe dans notre système, mais, jusqu'à présent, les algorithmes sous-jacents aux déformations 3D n'ont pas été développés. Les travaux présentés dans ce manuscrit étendent le panel d'outils de déformation en ajoutant une troisième dimension. Les déformations appliquées sur l'élément de dialogue face sont introduites dans le troisième chapitre.

La modélisation par éléments de dialogue n'est qu'une des possibilités d'interaction mises à la disposition de l'utilisateur par le système DIJA. L'interaction s'adapte à chaque utilisateur, pouvant se réaliser, sous DIJA, à plusieurs niveaux, grâce à son architecture multicouches. Cette architecture est présentée dans la section suivante.

2.3 L'architecture du système DIJA

On a déjà vu que DIJA se base sur une approche descendante (section 2.2.1) qui permet de décomposer le raisonnement de l'utilisateur. Celui-ci part avec une idée globale de l'objet et focalise son attention sur des parties de plus en plus fines. Ceci conduit - en fonction des connaissances et du métier de l'utilisateur - à des outils et à un vocabulaire progressivement focalisés, et naturellement à plusieurs niveaux d'abstraction [DGGP03]. Ceci permet au système de décrire les connaissances à un niveau métier pour qu'elles soient accessibles à l'utilisateur et de les décomposer successivement en passant par les niveaux d'abstraction pour arriver à les décrire à un niveau géométrique. Ainsi, l'utilisateur peut se concentrer sur son métier à chaque phase de la conception. Les niveaux d'abstraction sont présentés dans la section 2.3.1. L'objet est déformé au fur et à mesure, par des raffinements successifs entre les niveaux d'abstraction. Le raffinement, la visualisation et la communication entre les niveaux se font grâce à une hiérarchie de modules qui est présentée dans la section 2.3.2. Le processus de déformation basé sur des raffinements entre les cinq niveaux est décrit dans la section 2.3.3.

Nous continuons avec la présentation de l'architecture multi-niveaux dans la section suivante.

2.3.1 Niveaux d'abstraction

Les niveaux d'abstraction sont au nombre de cinq (voir Fig. 2.4) et diffèrent en fonction des tâches et de la sémantique :

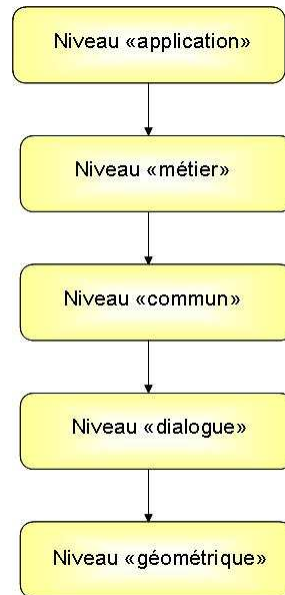


FIG. 2.4: Les niveaux d'abstraction de DIJA.

Du plus abstrait au plus concret, le rôle de chacun des cinq niveaux est le suivant :

- **le niveau application** : est le niveau qui permet de paramétrer l'application en fonction du métier de l'utilisateur et qui emploie le vocabulaire le plus abstrait. Il permet de sélectionner le métier souhaité, le scénario de conception, le panel des règles métier, etc. et d'adapter l'interface utilisateur en conséquence ;
- **le niveau métier** : concerne le vocabulaire et les outils propres à un métier particulier. Les mêmes notions ont des significations différentes selon le métier concerné et donnent naissance à des conceptions différentes. Par exemple, créer un trou dans une pièce induit des approches différentes pour le domaine de l'usinage et pour le domaine de la plasturgie ;
- **le niveau commun** : est un niveau intermédiaire qui regroupe des concepts globaux, propres à plusieurs métiers (mais sans être des termes du niveau « métier »), évitant ainsi les définitions redondantes ;
- **le niveau dialogue** : à ce niveau, chaque mot représente une caractéristique visible de l'objet modélisé. Des outils de déformations sont disponibles, qui permettent à l'utilisateur de manipuler intuitivement la géométrie de l'objet par l'intermédiaire des éléments de dialogue.

Les entités de haut niveau sont traduites ici en termes d'éléments de dialogue. Par exemple l'opération « créer embouti » est décomposée en une suite d'opérations qui permettent de définir le contour, les fibres et la face qui correspondent à l'embouti ;

- **le niveau géométrique** : le vocabulaire correspond aux primitives géométriques (sommets, segments, facettes, etc.). Un utilisateur expérimenté peut interagir à ce niveau sur les entités géométriques du modèle. Conformément à ses buts, il peut modifier les coordonnées des points, etc. Pour aboutir à des résultats satisfaisants, il doit avoir des connaissances géométriques car les déformations à ce niveau ne sont pas intuitives. C'est à ce niveau qu'on retrouve soit un B-Rep, soit des éléments finis, pour pouvoir visualiser ou simuler l'objet conçu. Il est donc possible d'avoir plusieurs vues d'un même modèle sous-jacent.

Le système DIJA doit être capable de décomposer la connaissance à chaque niveau d'abstraction et aussi de gérer les aspects liés à l'interaction et à la visualisation. Ceci se réalise par une décomposition modulaire de chaque niveau d'abstraction présentée dans la section suivante.

2.3.2 Approche modulaire

L'approche modulaire permet de décomposer et de descendre les connaissances entre les cinq niveaux d'abstraction, du plus haut (niveau application) au plus bas (niveau géométrique) et de remonter le résultat pour visualiser l'objet final. Ce processus, qui porte le nom de raffinement, se réalise par une hiérarchie de modules [DGGP03]. Chaque niveau d'abstraction est composé de cinq modules regroupés sous le nom de « environnement » (voir Fig. 2.5 et Fig. 2.6).

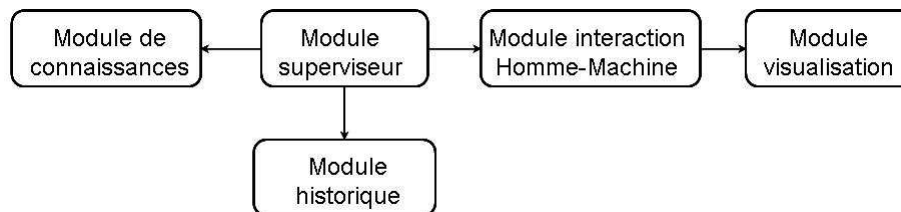


FIG. 2.5: Décomposition modulaire de DIJA.

Chaque environnement peut communiquer avec l'environnement qui lui est directement inférieur pour lui transmettre une instruction à raffiner. L'environnement immédiatement inférieur traduit l'instruction qui lui est transmise en une suite d'instructions de son niveau et retransmet le résultat au niveau supérieur qui a lancé la demande. Si l'instruction ne peut pas être traduite par le niveau inférieur, elle est transmise au niveau en-dessous de lui et le processus

continue de la même façon. Ainsi, le système DIJA peut raffiner les spécifications de l'utilisateur jusqu'aux primitives du modèle géométrique avant de remonter le résultat vers le niveau qui a lancé l'opération.

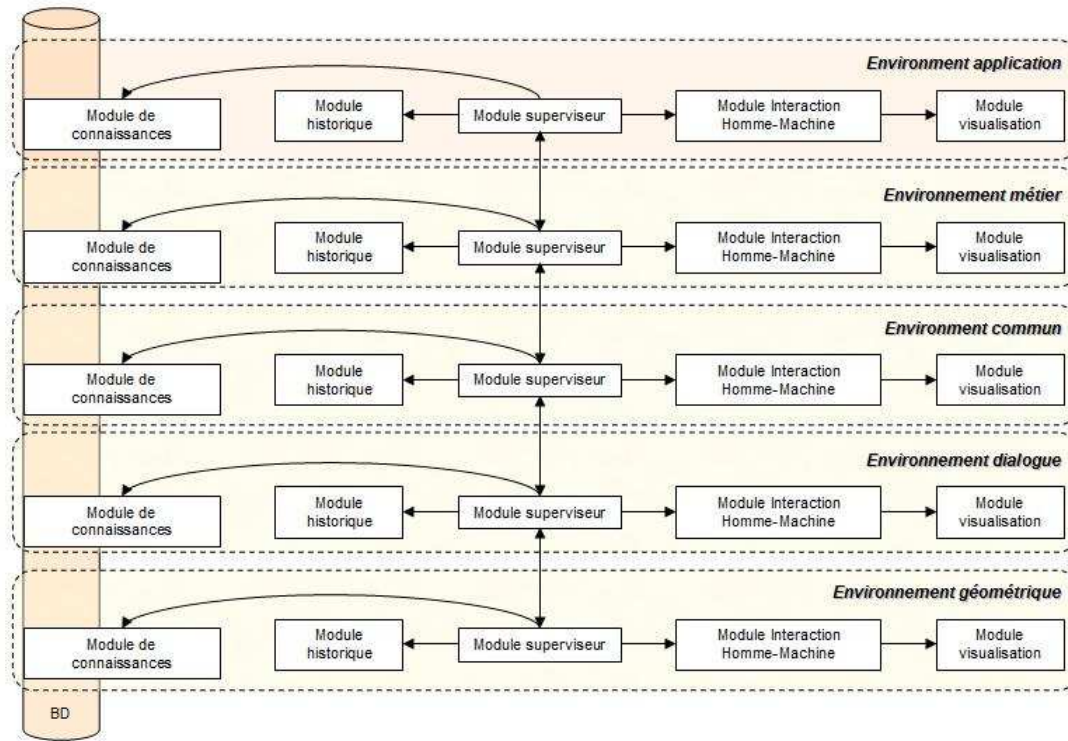


FIG. 2.6: L'architecture cinq niveaux DIJA.

Quelque soit le niveau d'abstraction, les cinq modules ont chacun un rôle bien précis :

- **le module d'interaction Homme-Machine (MIHM)** : permet de gérer les interactions entre l'utilisateur et le système de CAO. Il donne à l'utilisateur la possibilité de paramétrer les déformations et il crée les interfaces utilisateur selon le contexte métier ;
- **le module de visualisation(MV)** : est géré par le MIHM. Il permet à l'utilisateur de visualiser le résultat de l'objet modélisé ainsi que les données sauvegardées par le MH ;
- **le module d'historique (MH)** : est le module qui enregistre les actions de l'utilisateur lors de la conception et les données de conception au niveau de la base de connaissances, afin de pouvoir revenir en arrière. Il communique avec les autres modules de manière indirecte, en passant par le MS ;
- **le module superviseur (MS)** : assure le flux de données entre les différents modules. Il reçoit des instructions à exécuter du niveau supérieur et les transmet au module MH

pour être sauvegardées. S'il ne comprend pas une instruction, il la transmet au module MDC. Sinon, il l'exécute ou l'envoie au module approprié (soit au MS du niveau inférieur si l'instruction nécessite être raffinée, soit au MIHM du niveau courant). L'algorithme qui implémente le traitement d'une instruction par le MS est donnée dans Algorithme 2 ;

- **le module de connaissances (MDC)** : contient les données relatives à la conception. Lorsqu'il reçoit une instruction provenant du MS, il retourne un scénario pour réaliser l'instruction. Les connaissances - représentées par des scénarii de collaboration, des règles métier, des scripts, des contraintes géométriques, topologiques ou métier - sont interprétées comme des scripts [GG03].

L'ensemble de l'architecture DIJA est présenté dans la figure 2.6. Elle permet une structuration hiérarchique d'un système de CAO, en le décomposant en plusieurs environnements. Ceci assure l'aspect fonctionnel et intuitif du système de CAO. Nous présentons, dans la section suivante, la façon dont cette architecture sert à la réalisation des déformations sous DIJA.

2.3.3 Utilité de l'architecture système dans les déformations

Le processus de déformation se présente, à chaque niveau, comme un scénario composé d'instructions. Plus le niveau est haut, plus le niveau sémantique est haut et vice versa. Ces instructions sont exécutées par le « module superviseur » du niveau courant qui dirige leur traitement vers un module adapté (voir Algorithme 2), dépendant de leur type. Si pendant l'exécution une instruction n'est pas compréhensible par le « module de connaissances » du niveau concerné, le « module superviseur » l'envoie au niveau inférieur. Le processus est repris de la même façon au niveau inférieur et peut descendre ainsi jusqu'au niveau géométrique. Une fois le processus accompli, le résultat est remonté par le « module superviseur » et éventuellement envoyé au « module visualisation » du niveau qui a lancé la demande. A chaque niveau, l'utilisateur interagit avec le système par l'intermédiaire du « module IHM » pour paramétrer la déformation. En fonction du niveau d'abstraction, les paramètres sont différents ; ainsi, au niveau « métier » ou « commun » des termes de haut niveau sémantique leur sont associés, au niveau « dialogue » les paramètres correspondent à des éléments de dialogue identifiables sur la surface de l'objet, alors qu'au niveau « géométrique » les paramètres correspondent à des primitives géométriques associées à l'objet modélisé.

Algorithme 2 Le pseudocode de l'algorithme de traitement d'une instruction i par le superviseur

```

1: procédure traitement( $i$ )
2: Si niveau( $i$ ) = niveauCourant Alors
    Envoyer  $i$  au MH ;
3: FinSi
4: Si interactif( $i$ ) Alors
    Envoyer  $i$  au MIHM ;
    Attendre le résultat  $r$  ;
    traitement( $r$ ) ;
5: SinonSi exécutable( $i$ ) Alors
    executer( $i$ ) ;
6: Sinon
    Demander au MDC de décomposer  $i$  en un scénario  $s$ 
    Pour tout  $i' \in s$  Faire
        traitement( $i'$ )
    Fin Pour
7: FinSi
8: Fin procédure

```

Le paramétrage dépend aussi de l'outil appliqué ; si on débute avec un outil qui intègre un nombre plus grand de paramètres, l'intervention de l'utilisateur aux niveaux plus bas est plus réduite, car les paramètres sont déduits par les paramètres de début. Un exemple explicite de raffinement DIJA est étudié dans le troisième chapitre pour la création d'un plumier sur une tablette arrière de voiture.

L'architecture que nous venons de présenter est à la base de la modélisation par déformation. Plusieurs types de déformation sont possibles selon les spécificités de l'utilisateur et grâce aux plusieurs niveaux sémantiques et aux outils présents à chaque niveau :

- déformation de niveau sémantique élevé (orientées métier) ;
- déformation à l'aide des caractéristiques visibles (éléments de dialogue) ;
- déformation de niveau bas (purent géométrique).

Nous présentons dans la section suivante les mécanismes mis en œuvre pour réaliser les déformations DIJA 2D et les outils qui rendent leur application possible d'une manière intuitive.

2.4 Modélisation par déformation

Les déformations constituent le noyau d'interaction Homme-Machine DIJA. Grâce aux déformations que nous proposons, l'utilisateur a la possibilité de concevoir sa pièce de manière simple et intuitive, en s'appuyant sur une approche orientée métier et sur des outils dédiés. Cette section introduit le concept de déformation orientée métier et continue par un exemple concret qui sert

à approfondir l'approche avant de présenter les outils de déformation proposés par DIJA.

2.4.1 L'aspect métier de l'approche « synthétique »

Les déformations que DIJA propose sont des déformations orientées métier dans le cadre d'une approche dont l'idée est de fournir à n'importe quel utilisateur une interface innovante le plaçant dans son contexte naturel. Le but est de recréer son environnement de travail habituel au niveau des interactions par le biais des outils dédiés. L'environnement physique de l'utilisateur n'est pas recréé de manière virtuelle comme par exemple dans [MCQ05] où les auteurs utilisent des lunettes 3D, un dispositif d'entrée haptique à six degrés de liberté et un outil « chauve-souris » (*bat* en anglais) pour s'immerger dans l'espace de travail. Nous utilisons simplement la souris comme dispositif d'entrée pour permettre une interaction entre l'utilisateur et le système la plus intuitive possible. Le terme « intuitif » est utilisé pour décrire une interface qui opère de manière familière, habituelle. Pourtant, les utilisateurs novices n'ont pas l'habitude d'interagir avec un système de CAO de la même façon que les utilisateurs experts. C'est à ce niveau que l'approche orientée métier intervient.

Prenons l'exemple d'un potier créant ses objets en argile. Dans l'environnement réel, l'artisan commence avec une forme grossière qu'il déforme progressivement avec ses mains pour obtenir l'objet désiré. Le même processus peut être reproduit sous DIJA grâce à l'approche « synthétique » : le potier commence par sélectionner un métier - poterie dans ce cas - et une forme de base choisie dans une base de formes initiales disponible dans le système. Ensuite, conformément à ce qu'il veut obtenir, il choisit un élément de dialogue associé à l'objet modélisé et lui applique une déformation plus ou moins sophistiquée à l'aide des outils DIJA. Les outils employés englobent des fonctions mathématiques et des paramètres qui transmettent les actions de l'utilisateur au modèle géométrique grâce à l'architecture multiniveaux. C'est la tâche du système de raffiner ces paramètres, de réaliser la déformation au niveau géométrique et de remonter le résultat à un niveau sémantique haut (tel que le « niveau métier ») pour le visualiser en temps réel. Nous basons donc notre IHM moins sur l'environnement physique et plus sur les techniques d'interaction. Nous nous dirigeons donc naturellement dans cette approche vers les outils dédiés qui facilitent l'interaction Homme-Machine.

Un outil dédié est un composant de l'interface. Par l'intermédiaire de paramètres et appliqué aux caractéristiques visibles de l'objet modélisé, son activation conduit à des déformations virtuelles qui sont identiques à celles du monde réel. La corrélation avec les gestes de l'utilisateur

dépend de l'activation des paramètres métier. Ces paramètres sont propres au métier de l'utilisateur et correspondent au vocabulaire employé par l'utilisateur. Ceci signifie que l'utilisateur n'a pas besoin de comprendre le modèle, il utilise simplement son savoir faire pour concevoir son objet. La déformation se fait sur les données internes en concordance avec le métier, les outils, les paramètres et le contexte courant de l'application. L'intégration des paramètres avec les outils métier est discutée dans la section 2.4.3, après avoir présenté un exemple de conception par déformation dans la section suivante.

2.4.2 Processus de déformation 2D

Prenons l'exemple de la conception d'une cruche.

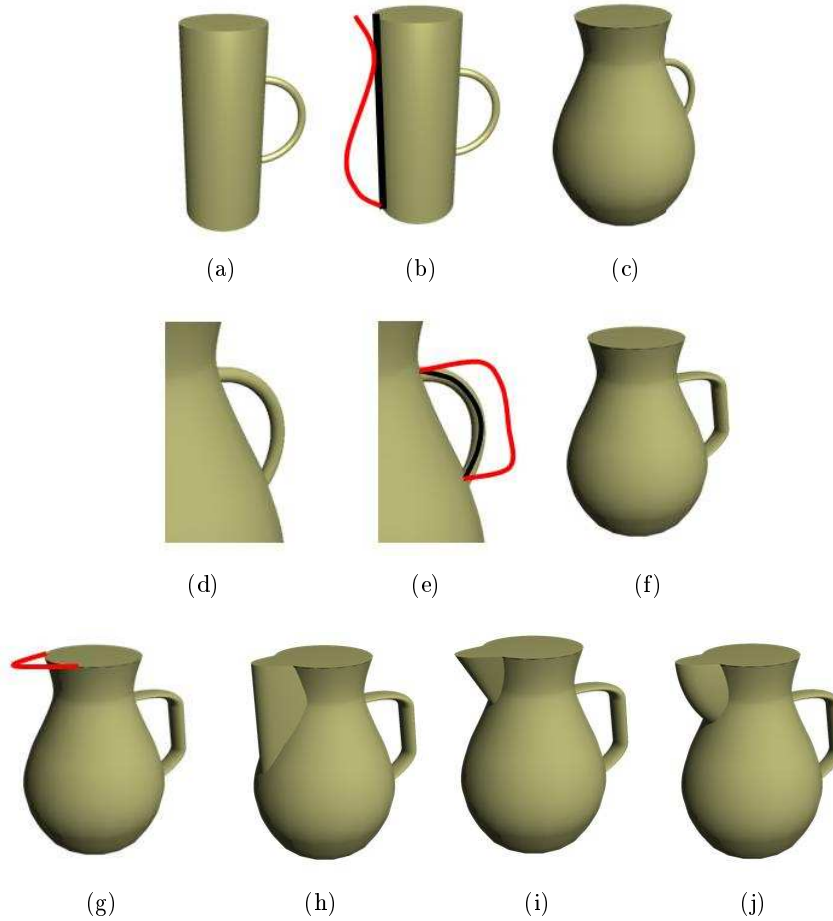


FIG. 2.7: Un exemple de conception d'une cruche par un outil de déformation sur des éléments de dialogue : (a-c) application de l'outil de déformation sur une fibre pour déformer la silhouette générale ; (d-f) application de l'outil de déformation sur une fibre pour déformer l'anse ; (g) déformation locale sur un contour caractéristique pour la création d'un bec ; (h-j) résultats de la déformation dans (g) ; (j) la solution choisie par l'utilisateur.

Le scénario est le suivant :

- l'utilisateur spécifie le cahier de charges de la conception : « un récipient de liquide avec une anse » ;
- le système lui met à la disposition une librairie de formes qui correspondent aux spécifications, à condition qu'un script existe dans le « module de connaissances » pour relier la spécification à la forme ;
- l'utilisateur choisit parmi ces formes une forme initiale :
 - un cylindre avec une anse, qui correspond topologiquement à l'objet final (voir Fig. 2.7 (a)) ;
- l'utilisateur raffine la forme initiale :
 - la forme globale du cylindre est modifiée par une déformation d'une fibre qui lui est associée : la fibre est redéfinie, ce qui induit la modification de la révolution du cylindre (voir Fig. 2.7 (a-c)) ;
 - la forme globale de l'anse est modifiée de la même façon, en déformant la fibre qui lui est associée (voir Fig. 2.7 (d-f)) ;
 - la forme est déformée localement : un contour caractéristique est défini pour le bec ; l'utilisateur redessine la forme du contour ce qui détermine une modification des surfaces des faces adjacentes à ce contour (voir Fig. 2.7 (g-j)) ; le système propose plusieurs possibilités de pichet montrées dans la figure 2.7 (h-j) ;
- l'objet final choisi par l'utilisateur est celui montré en figure 2.7 (j).

L'exemple présenté montre la façon dont se déroule un processus de déformation sous DIJA. Plus précisément, les déformations présentées ont été appliquées sur des éléments de dialogue 2D pour déformer globalement la silhouette de la cruche (déformation sur une fibre) et ensuite localement une partie de la cruche pour l'insertion d'un bec (déformation sur un contour caractéristique). Tout au long de la déformation l'utilisateur interagit intuitivement avec le système pour déformer l'objet grâce aux caractéristiques visuelles sur la surface de celui-ci. La figure 2.8 montre d'autres exemples d'objets décoratifs modélisés de la même façon.

Ces déformations sont le résultat de l'application d'un outil de déformation sur les éléments de dialogue définis sur la surface de l'objet, l'interaction ayant donc lieu à un niveau dialogue. La définition des outils, l'interaction avec l'utilisateur et leur application à différents niveaux sont présentés dans la section suivante.

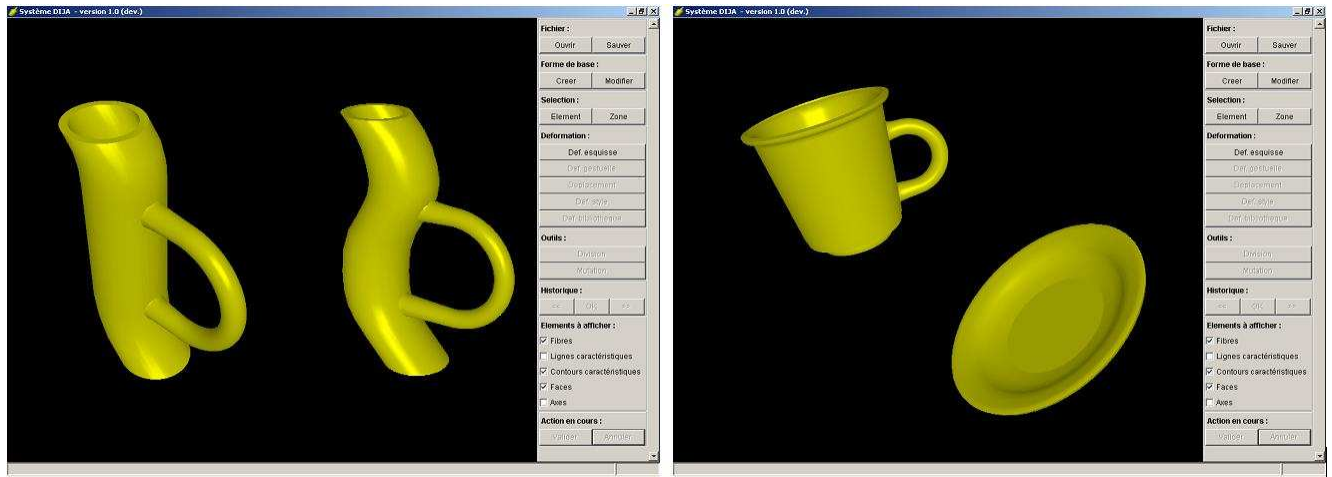


FIG. 2.8: Mod lisation 2D d'objets d coratifs.

2.4.3 Outils DIJA orient s m tier

Nous proposons, dans notre approche, un type g n rique de d formation qui convient   n'importe quelle application dans le domaine de la CAO, mais qui est capable de s'adapter au contexte sp cifique de l'utilisateur. La technique d'interaction doit  tre unique pour tous les types de d formation m me si le contexte peut changer dans le cadre de la m me session de conception. De plus, le syst me est capable de fournir, au cours du processus de conception, plusieurs types de d formations et il peut sugg rer plusieurs solutions au concepteur. Afin de s parer la m thode de mod lisation DIJA du mod le de d formation, l'interface utilisateur propos e se base sur des outils virtuels de d formation qui unifient les interactions et qui se comportent de mani re similaire   ceux du monde r el. En cons quence, pour appliquer une d formation au mod le, l'utilisateur doit manipuler un outil int gr  avec le syst me DIJA.

Nous avons vu que la d formation de l'objet mod lis  peut se r aliser   diff rents niveaux de l'architecture DIJA gr ce   des outils qui y sont pr sents. Ainsi, nous proposons des outils de haut niveau s mantique aux niveaux « application » qui sont raffin s en des outils de niveau « m tier » et « commun », en des outils qui op rent sur les caract ristiques visibles au niveau « dialogue » et finalement en des outils de d formation de niveau bas qui s'appliquent sur des primitives g om triques du mod le au niveau « g om trique » (voir Fig. 2.9 et Annexe B).

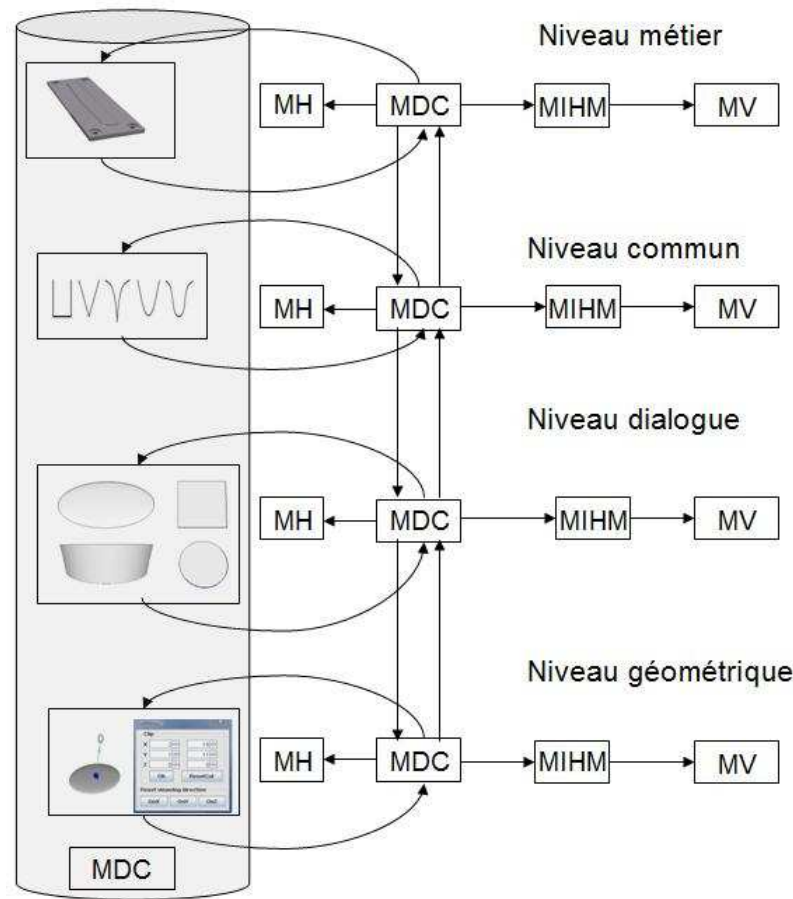


FIG. 2.9: Outils de déformation représentés à différents niveaux d'abstraction.

Suivant les spécifications du projet DIJA, l'utilisateur peut être le plus loin possible des déformations purement géométriques s'il le désire. Pourtant, des outils applicables aux primitives géométriques existent dans notre système. Ceci dérive de la rigueur architecturale du système qui fait que à chaque niveau on retrouve les mêmes modules mais avec une sémantique différente à chaque niveau. Le module de connaissances est transversal à chaque niveau et stocke la connaissance sous la forme de scripts, de règles, des contraintes et des entités géométriques (voir Fig. 2.10). L'environnement géométrique contient ainsi la connaissance représentée au niveau le plus bas des outils spécifiques pour la manipuler, utilisables par un utilisateur expérimenté.

De nombreux outils de déformation ont été proposés pour la sculpture virtuelle [BL94, GH91, KRK⁺06, Par77], mais sans prendre en compte le respect des règles métier. Les déformations DIJA ne sont pas faites d'un point de vue purement design, l'utilisateur doit avoir un contrôle précis sur le processus et sur le résultat qui doivent correspondre à son savoir-faire. Le contrôle des déformations et le respect des contraintes métier se réalise grâce à des paramètres métier qui

renforcent les règles métier tout au long du processus de déformation.

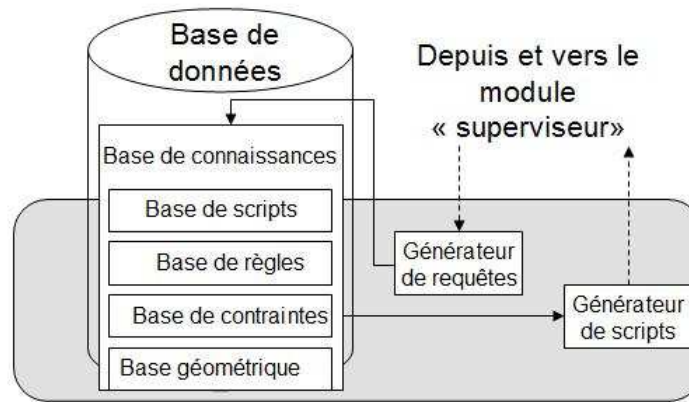


FIG. 2.10: Les composants du module de connaissances.

Les paramètres métier se propagent, eux aussi, de haut vers le bas à travers l'architecture du système. Supposons que l'utilisateur choisit de déformer la surface de son objet en appliquant un outil de haut niveau sémantique au niveau « métier », comme nous le montrons dans l'exemple de raffinement détaillé dans le troisième chapitre pour l'insertion d'un plumier pour une tablette arrière de voiture. Dans ce cas, l'outil du niveau métier est paramétré par l'utilisateur avec des paramètres de haut niveau sémantique. Ces paramètres sont descendus au niveau « commun » et traduits en des paramètres propres au vocabulaire de ce niveau lors du raffinement. Quand le niveau « commun » est traduit en niveau « dialogue », les paramètres du niveau commun sont traduits en des termes spécifiques aux éléments de dialogue. Les paramètres du niveau « dialogue » sont ensuite reliés aux paramètres géométriques lors du passage du niveau « dialogue » au niveau « géométrique ».

Nous détaillons dans les paragraphes suivants les mécanismes d'intégration du savoir-faire avec les outils de déformation et les paramètres sous-jacents.

Conformément à la philosophie DIJA, l'utilisateur modélise en utilisant son savoir-faire. Le savoir-faire est représenté, comme le montre la figure 2.10, par des contraintes et des règles de production, à un niveau métier mais aussi à un niveau géométrique. Le système DIJA associe la géométrie à une action ou à une règle métier. Le paramétrage du modèle est ainsi orienté métier. Les actions que l'utilisateur entreprend se traduisent, dans notre système, en des scénarios de conception que l'utilisateur utilise de manière intuitive et qui définissent chaque tâche à accomplir pour atteindre la solution désirée. Les scénarios de conception sont regroupés dans une base de

scénarii. Les scénarii représentent la connaissance sous une forme procédurale. Ils englobent des contraintes et des règles de conception grâce à des paramètres graphonomériques et à un langage spécifique, proche du langage naturel (voir Annexe B). Trois classes de vocabulaire sont définies dans [Gar05] :

- la classe **action** : qui permet de définir l'action à exécuter par le système de CAO, grâce à un verbe ;
- la classe **objet** : qui permet d'indiquer un objet dans le MDC ;
- la classe **contrainte** : qui contient les contraintes qui permettent de contraindre les objets par rapport à d'autres.

La signification des termes diffère selon leur environnement d'appartenance et en fonction du domaine concerné.

C'est en se basant sur ce vocabulaire et sur un type spécifique de paramètres que les paramètres graphonomériques sont définis [Gar05]. Un paramètre graphonomérique représente un n-uplet $\langle \text{action} \rangle \langle \text{objet} \rangle \langle \text{contrainte}(s) \rangle \langle \text{objet} \rangle \langle \text{contrainte}(s) \rangle \dots$ (voir Fig. 2.11). Ce type de paramètre permet de définir des paramètres qui peuvent être liés à des contraintes ou à des règles. Ces paramètres sont définis soit au début du processus de conception, soit pendant la conception.

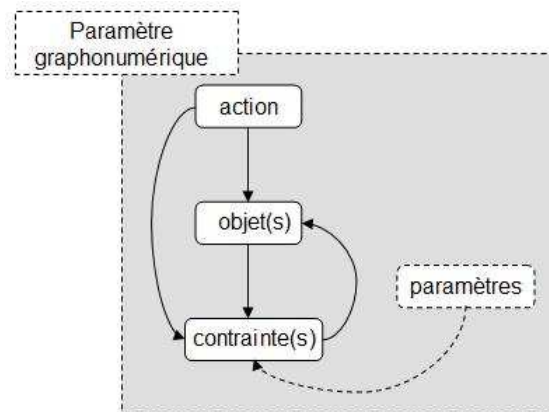


FIG. 2.11: Paramètre graphonomérique.

En s'appuyant sur les paramètres graphonomériques, quatre niveaux de paramétrage sont identifiés dans [Rei09] pour faciliter la conception :

- les paramètres fonctionnels : ce sont les paramètres imposés au concepteur dès le début à la fin du processus de conception ;

- les paramètres d’optimisation : ils permettent à l’utilisateur de tester des solutions alternatives de sa conception et d’optimiser ainsi certaines zones de la pièce ;
- les paramètres de simplification : ce sont des booléens qui permettent de simplifier le modèle lors de la conception ;
- les paramètres métier : ce sont les paramètres liés au métier et donc au savoir-faire.

Ces derniers paramètres sont ceux pris en compte dans notre étude.

En s’appuyant sur la représentation du savoir-faire détaillée précédemment, nous restons au niveau dialogue pour expliquer le concept d’outil de déformation DIJA.

Sous DIJA, un outil de déformation est défini par sa géométrie et par son comportement. La géométrie de l’outil est représentée par les éléments de dialogue dont la forme peut s’exprimer :

- soit explicitement, par une forme libre, qui correspond à une esquisse à main levée,
- soit mathématiquement, par une équation explicite ou implicite,
- ou par une forme choisie dans une librairie de formes qui comprend deux sous parties. La première, la partie procédurale, contient un ensemble des outils à utiliser pour générer une déformation. La seconde partie c’est la partie morphologique, qui propose un ensemble de styles de déformation imprimés à la déformation par l’outil choisi. La librairie de formes fournit, pour chaque forme implicite qu’elle contient, la définition explicite ou la méthode de déformation explicite de l’élément de dialogue considéré.

Sous DIJA, le terme du vocabulaire qui définit un outil est corrélé à la représentation mathématique de l’outil. Par exemple, le degré de courbure d’une représentation mathématique est associée à un style qualifié comme « bombé » dans la bibliothèque de formes (voir Fig. 2.12). Les fonctions mathématiques sont intéressantes à utiliser pour définir la forme de la déformation, car elles sont de modèles faciles à gérer (coniques, polygones, etc.) et simples du point de vue calcul.

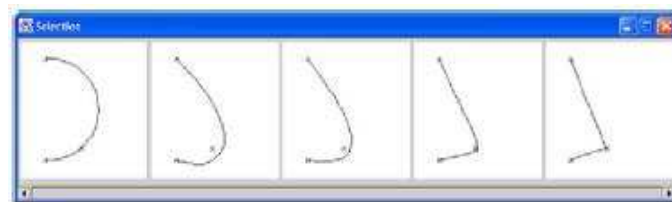


FIG. 2.12: Style dérivant de la courbure de la déformation ; la représentation mathématique va des coniques aux polygones.

L’utilisateur dispose de plusieurs possibilités pour définir la forme de l’outil qu’il manipule. Après avoir choisi la forme, qui correspond à un style précis, il applique des quantificateurs (du

genre « plus », « moins », etc.) qui sont proches de son langage naturel (voir Fig. 2.12). Il peut ainsi tester plusieurs solutions et choisir celle qui convient mieux à ses buts. Une autre modalité est de définir les paramètres de l'outil plus directement, en modifiant leurs valeurs via des boîtes de dialogue ou d'ascenseurs. Les quantificateurs et les boîtes de dialogue sont complémentaires car ils permettent de définir un outil soit de manière vague soit précise ou les deux (un utilisateur peut définir son outil de manière vague au début et ensuite appliquer une définition précise pour le finaliser).

Quant au comportement d'un outil de déformation, celui-ci consiste à déformer l'élément de dialogue choisi. L'utilisateur dispose de deux manières pour induire une action à un outil de déformation. La première consiste à spécifier la déformation par esquisses, en redéfinissant une partie ou la globalité de l'élément de dialogue sur lequel la déformation s'applique. Le système interprète l'esquisse et propose plusieurs choix [DGMP99, DGMP00a, DGMP00b]. La deuxième possibilité spécifie la déformation de manière « textuelle », en utilisant un mot du vocabulaire qui correspond à son métier pour préciser un style ou un outil de déformation explicitement défini parmi les paramètres du système [DDGP02].

Un outil de déformation est complètement défini quand les éléments de dialogue à déformer sont connus. L'outil et les éléments de dialogue sur lesquels il s'applique doivent avoir la même dimension : pour des faces on applique des outils 3D et pour des lignes, contours et fibres on applique des outils 2D. L'utilisateur a donc à sa disposition une méthode intuitive pour modéliser son objet en utilisant des mots d'un vocabulaire qu'il comprend pour appliquer des déformations sur des éléments qu'il identifie visuellement sur la surface de son objet. Il sélectionne ou construit un outil et l'applique sur plusieurs éléments de dialogue. La géométrie de l'objet est déduite de la géométrie des éléments de dialogue. Les utilisateurs experts ont ainsi la possibilité de définir des nouveaux outils géométriquement et fonctionnellement, pendant que les autres utilisateurs peuvent les utiliser. Les nouveaux outils sont enregistrés dans la base de connaissances du système (voir Fig. 2.10 et Annexe B).

2.5 Conclusion

Nous avons présenté, dans ce deuxième chapitre, l'approche « synthétique » du projet DIJA, qui est à la base de la modélisation par déformation. Ainsi, nous avons vu dans une première partie, la philosophie de l'approche descendante qui suit la façon synthétique de réfléchir de l'utilisateur en allant d'une vue globale à une vue plus fine de l'objet modélisé. Grâce à cette

approche, l'utilisateur a la possibilité de modéliser en agissant sur des éléments visibles qu'il identifie sur la surface de l'objet à déformer. Les déformations sont orientées métier et possibles grâce à une architecture multi-niveaux que notre système propose. L'architecture hiérarchique se base sur cinq environnements qui décomposent le savoir faire du niveau le plus haut (application) au niveau le plus bas (géométrique), permettant à l'utilisateur de se concentrer sur son métier et non pas sur les détails mathématiques qui peuvent s'avérer rapidement encombrants. Les déformations sont possibles grâce à des outils de déformation dédiés qui sont présents à chaque niveau grâce à une base de connaissances qui traverse les cinq niveaux de l'architecture du système DIJA. Jusqu'à présent seulement des outils de déformation 2D ont été implémentés, bien que le concept d'outil de déformation 3D existe dès le début dans notre approche. On conclut en présentant les avantages majeurs apportés par l'approche « synthétique » :

- réutilisabilité : une fois un outil défini, il peut servir à déformer n'importe quel objet par un utilisateur concerné ;
- modularité : les modèles de déformation peuvent être vus comme des *plug ins* pour l'utilisateur ;
- homogénéité : DIJA propose une interface unifiée ;
- consistance : un objet modélisé en utilisant un outil de déformation peut devenir à son tour un outil pour déformer d'autres objets ;
- fonctionnalité : le système DIJA offre des outils dédiés qui conservent l'intention de conception au long du processus de modélisation.

Le chapitre suivant propose des outils de déformation 3D pour la CAO dans le contexte du projet DIJA.

Chapitre 3

Proposition pour la réalisation des outils de déformation 3D pour la CAO dans un contexte orienté métier

3.1 Introduction

Ce chapitre représente une extension des travaux de recherche effectués par notre équipe de recherche autour du projet DIJA. Plus précisément, en s'appuyant sur les principes de la méthode de modélisation dite « synthétique », énoncés dans le chapitre précédent et sur certaines techniques de déformation présentées dans le premier chapitre adaptées à notre approche, nous proposons des outils de déformation 3D qui s'appliquent sur une portion de surface ou sur la face de l'objet modélisé. Le raccordement de la surface déformée avec le reste de la surface doit être pris en compte par les outils de déformation ou par d'autres outils spécifiques. Dans l'espace 3D, les règles métier s'appliquent directement sur la totalité de la face à déformer à l'aide des paramètres de déformation et de contraintes géométriques. Au niveau géométrique ceci se répercute en appliquant discrètement la déformation (paramétrée en préalable) dans chaque point 3D de la surface.

Un outil 3D de déformation est construit en utilisant les fonctions *decay* pour imposer la forme et la quantification de la déformation sur l'ensemble des points de la face à déformer et les contraintes géométriques pour induire à l'outil de déformation une action spécifique (métier). Ces aspects sont présentés dans la section 3.2. L'application d'un outil 3D sur une face est introduite dans la section 3.2.3. La section 3.3 montre, de manière concrète, comment se déroulent les paramètres de déformation de haut vers le bas dans l'architecture du système DIJA avec un exemple de raffinement de déformation 3D pour l'insertion d'un plumier dans une tablette arrière de voiture. Le concept d'outil 3D de déformation est finalement illustré sur un exemple de

déformation avec contrainte volumique et ses applications dans la section 3.4, avant de conclure ce chapitre.

3.2 Intégration des fonctions *decay* dans les règles métier

Parmi les techniques de déformation étudiées dans le premier chapitre, les fonctions *decay* représentent la technique qui s'identifie mieux avec nos intentions de modélisation. Tout d'abord, car elle permet de déformer localement une partie de la surface (ensemble fini des triangles appartenant au maillage). La déformation se réalise de manière simple et rapide, à travers un algorithme qui modifie les coordonnées géométriques des points ; l'amplitude de la déformation est en fonction de la position des points par rapport au bord de la partie à déformer. Ensuite, les fonctions *decay* ont été considérées par ce que les déformations qui en résultent sont en corrélation avec des fonctions mathématiques, ce qui permet un panel varié de styles de déformation. Les déformations sont ainsi intuitives et peuvent être corrélées avec des règles métier grâce à leur géométrie. Ces aspects sont illustrés dans la section suivante, au niveau sémantique DIJA le plus bas, le « niveau géométrique ».

3.2.1 Définition des outils 3D à l'aide des fonctions *decay*

La définition d'un outil de déformation 3D, bien que appliqué sur une face de l'objet, suit le même formalisme que celui montré dans le chapitre précédent : l'outil doit posséder une géométrie et une action. Nous avons déjà vu, pour les outils 2D, que la forme de la déformation peut être induite par une fonction mathématique qui donne le style de la déformation, le comportement étant spécifié par l'utilisateur à l'aide des esquisses à main levée ou d'un terme du vocabulaire métier. Les fonctions *decay* proposent un mécanisme qui définit en même temps la forme et le comportement de la déformation. Nous détaillons ces aspects par la suite.

La forme d'une fonction *decay* correspond à sa définition mathématique. On associe à chaque forme un terme intuitif qui suggère à l'utilisateur la forme qu'il peut obtenir en l'appliquant : « cloche », « cylindre », « bombé », « cône », « sommet », etc. Ceci permet d'obtenir un spectre large de déformations avec de différentes formes qui correspondent à n'importe quelle fonction mathématique $f : [0, 1] \mapsto [0, 1]$, tel que, par exemple :

- Cloche : $f(x) = \cos((1 - x)\frac{\pi}{2})$;
- Sommet : $f(x) = x^2$;
- Cône : $f(x) = x$.

A savoir que la forme de la fonction *decay* s'ajoute à la forme de la surface déformée. Un exemple de trois types de fonctions *decay* appliquées localement à un objet cylindrique sont illustrés dans la figure 3.1.

L'action d'un outil de déformation englobant une fonction *decay* est quantifiée à l'aide d'un rayon d'influence, qui détermine la localité de la déformation, et des contraintes de déplacement des points. Ainsi, le rayon d'influence détermine la propagation de la déformation sur les points entourant le point déplacé. Le déplacement des points dans le voisinage du point initial dépend de la distance qui les sépare du point initial, mais aussi du vecteur de déplacement du point initial.

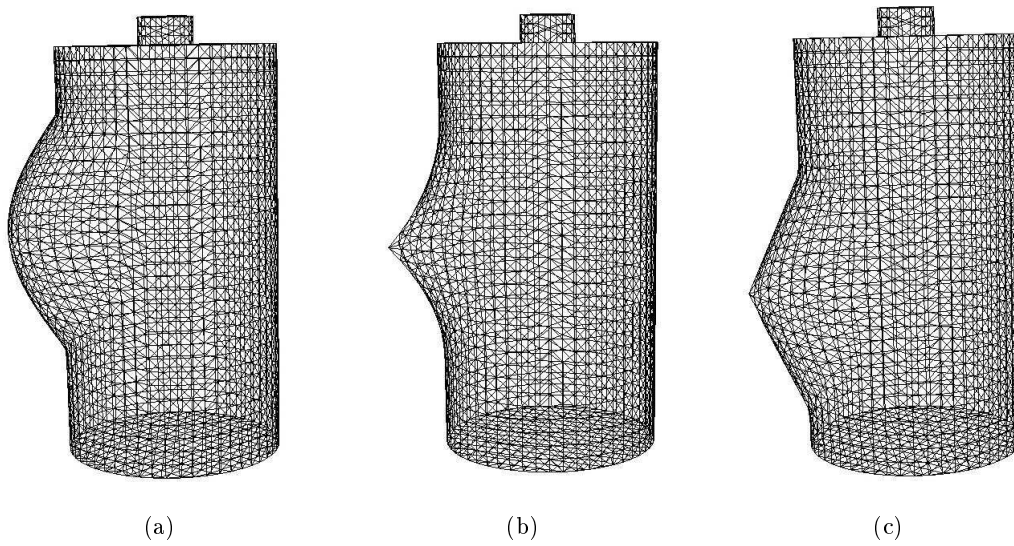


FIG. 3.1: Trois types de fonctions *decay* sur la surface d'un cylindre : (a) fonction *decay* « cloche » ; (b) fonction *decay* « sommet » ; (c) fonction *decay* « cône ».

Ces valeurs sont soit précisées par l'utilisateur à l'aide des boîtes de dialogue, de slideurs, de la souris, ou elles sont inférées par le système à partir de la position cible du point initial de déplacement et du contour caractéristique qui délimite la zone à déformer du reste de la surface. La localisation de la déformation s'exprime soit en nombre d'arêtes, soit en distance Euclidienne. Le premier cas est peut être plus facile à comprendre pour l'utilisateur mais moins prévisible, alors que le deuxième peut sembler plus abstrait mais la forme de la déformation est indépendante de la géométrie du maillage. Une attention particulière doit être accordée au maillage de l'objet déformé par des fonctions *decay*. Afin d'obtenir des résultats intuitifs, la distribution des sommets dans le maillage triangulaire doit être le plus homogène possible. Pour

uniformiser le maillage, des outils de subdivision peuvent être appliqués sur la zone d'influence préalablement à la déformation.

Ainsi, si N représente le rayon d'influence en terme de nombre d'arêtes, tous les points qui se trouvent à une distance inférieure ou égale à N sont déplacés ensemble avec le point initial P_i . Si le point initial $P_i(x, y, z)$ se déplace le long de l'axe Ox conformément à une fonction *decay* f avec d_x et si D représente la distance en nombre d'arêtes entre P_i et un point P dans la zone d'influence, la valeur du déplacement du point P au long de l'axe Ox se calcule selon la formule :

$$d_x^D = f\left(1 - \frac{D}{N+1}\right)d_x \quad (3.1)$$

Une fois ces éléments définis, l'outil de déformation est paramétré et prêt à être appliqué. Le même outil peut être réutilisé si on marque sur la surface de l'objet un ou plusieurs points « attachés » (*bound* en anglais) au point initial [AWW89]. Quand le point initial est déplacé, tous les points qui lui sont attachés subissent la même translation. Si un rayon d'influence est défini, une région de la même taille autour des points « attachés » est affectée. Cette opération est très utile lorsqu'on veut appliquer simultanément des déformations identiques sur la surface du même objet. Si deux zones d'influence interfèrent lors d'un tel processus, le déplacement de certains points est influencé par plus d'un point initial. Ce conflit peut être résolu de plusieurs manières : les points sont soit influencés par chacun des points auxquels il est attaché, soit par le plus proche de ces points (voir Fig. 3.2).

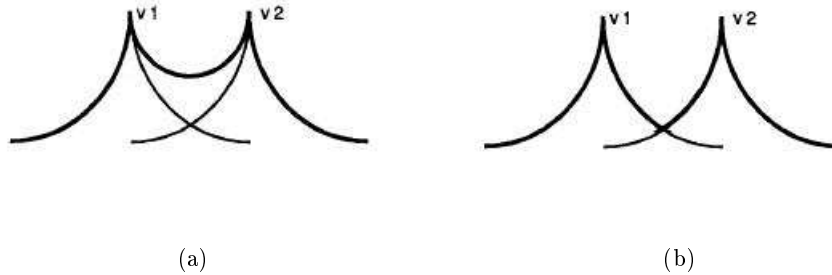


FIG. 3.2: Interférence de deux zones d'influence [AWW89] : (a) le déplacement des points est influencé par les deux points « attachés » ; (b) le déplacement des points est influencé par le plus proche des points « attachés ».

Une autre opération peut être implémentée pour arrêter brusquement la propagation de la déformation *decay*. Il suffit de marquer dans la zone d'influence un ou plusieurs points dits « ancrés » (*anchored* en anglais) qui ne peuvent pas être affectés par la déformation et qui

arrêtent sa propagation vers le bord de la zone d'influence. La Fig. 3.3 montre deux exemples de déformation *decay* avec des sommets « attachés » et « ancrés ».

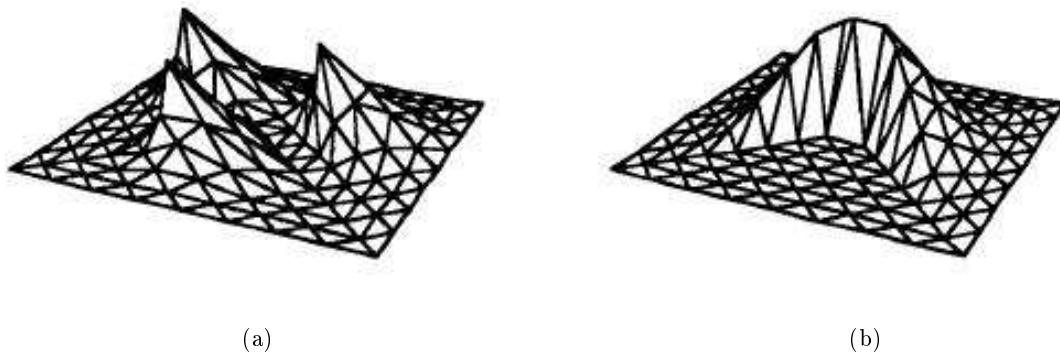


FIG. 3.3: Cas spéciaux de déformations *decay* [AWW89] : (a) sommets « attachés » ; (b) sommets « ancrés ».

On étudie, dans la section suivante, l'implémentation de ces principes avec un exemple simple de déformation de surfaces à l'aide des outils DIJA 3D.

3.2.2 Un exemple de processus de déformation 3D

Nous illustrons l'application des fonctions *decay* par la conception d'une bouteille en appliquant des déformations 3D simples. Le processus de déformation se déroule de la manière suivante :

- l'utilisateur spécifie le cahier de charges de la conception : « un récipient de liquide posable avec un bouchon » ;
- le système lui propose plusieurs formes qui correspondent à sa spécification ;
- l'utilisateur choisit une forme de départ simple qui correspond à ses buts :
 - un cylindre avec un bouchon (cylindre plus petit) qui respecte la topologie de l'objet spécifié ;
- le système demande à l'utilisateur d'indiquer la zone déformable, afin de respecter la contrainte du cahier de charges qui demande que l'objet soit posable ;
- l'utilisateur définit la zone déformable en excluant la base de la bouteille (voir Fig. 3.4 (b)) ;
- l'utilisateur raffine l'objet initial :
 - le corps du cylindre est déformé localement en lui appliquant simultanément des déformations du style « bombé » sur plusieurs points « attachés » :

- l'utilisateur définit un contour sur la surface du cylindre, qui représente le bord de la déformation et le point initial de déplacement ; le contour et le point initial peuvent être définis de plusieurs façons :
 - soit en indiquant le point initial avec la souris directement sur la surface du cylindre et une valeur qui représente le rayon d'influence autour du point initial à l'aide d'une boîte de dialogue ;
 - soit par une esquisse à main levée ou choisie parmi plusieurs contours proposés par le système, dans quel cas le système infère le point initial (le centre du contour) et le rayon d'influence ;
- la zone de déformation est calculée par le système à partir du contour défini par l'utilisateur (voir Fig. 3.4 (c)) ;
- l'utilisateur définit plusieurs points « attachés » sur la surface du cylindre, pour indiquer l'endroit où les déformations doivent s'effectuer ;
- il indique ensuite avec la souris ou à l'aide d'une boîte de dialogue la position cible du point initial ;
- le système propose plusieurs styles de déformation parmi lesquels l'utilisateur choisit « bombé » ;
- la déformation est paramétrée, l'utilisateur peut donc lancer l'opération de déformation ;
- le système récupère les paramètres, les transmet aux algorithmes du niveau géométrique, il récupère ensuite le résultat et l'affiche (voir Fig. 3.4 (d)) ;
- le bouchon du cylindre est déformé localement de la même façon, en lui appliquant des déformations du style « sommet » (voir Fig. 3.4 (e)) ;
- l'utilisateur revient sur le corps de la bouteille, près de la base, pour ajouter des éléments décoratifs (voir 3.4 (f)) ;
- l'objet final est montré en Fig.3.4 (f) ;

Une fois l'objet déformé, l'utilisateur a la possibilité de lui appliquer, globalement ou localement, un outil de lissage implémenté à l'aide des schémas de subdivision. Si le lissage s'applique localement, l'utilisateur définit, comme pour l'outil de déformation, une zone effective de lissage.

Deux outils de lissage par subdivision sont proposés par le système DIJA :

- un outil de subdivision linéaire (voir Fig. 3.5 (a)) ;
- un outil de subdivision Butterfly [ZSS96, ZSD⁺00] (voir Fig. 3.5 (b)-(e)).

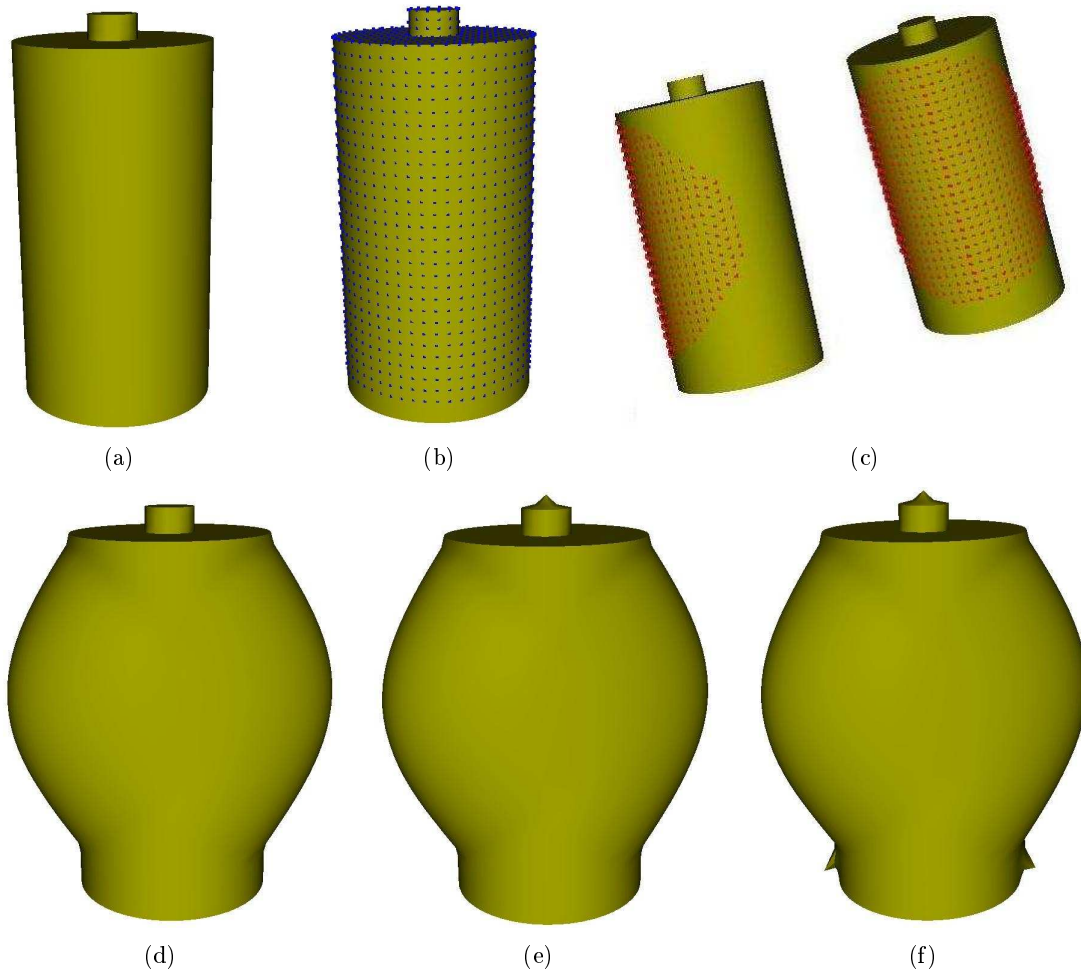
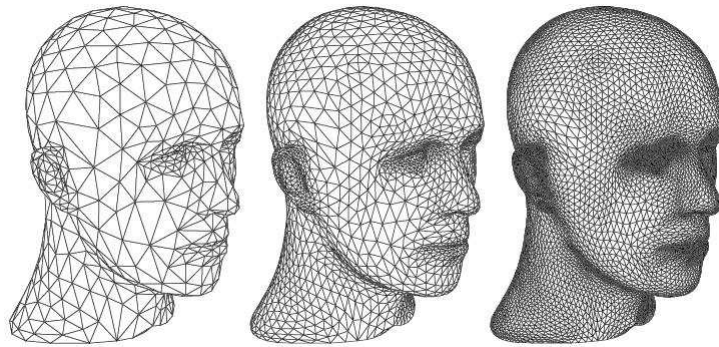


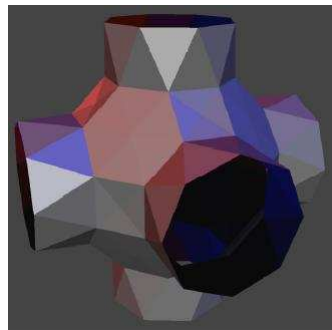
FIG. 3.4: Processus de déformation 3D : (a) l'objet initial ; (b) la zone déformable ; (c) la zone de déformation ; (d) déformation style « bombé » du corps de la bouteille ; (e) déformation de type « sommets » du bouchon de la bouteille ; (f) déformation de type « sommets » de la base de la bouteille.

Même si l'application d'une opération de lissage peut s'avérer utile dans certains cas, la qualité du résultat dépend beaucoup du maillage initial. Il est donc préférable de débiter la modélisation avec un maillage assez homogène, auquel cas les outils de déformation suffisent, l'opération de lissage pouvant être ainsi évitée. L'outil de lissage est présent dans notre système surtout pour raffiner une partie de la surface en un maillage plus fin avant de la déformer.

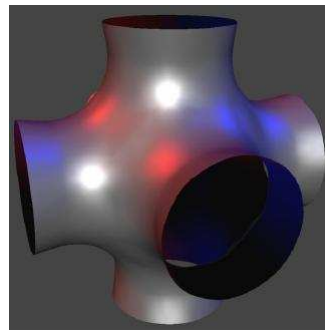
L'exemple que nous avons présenté dans ce paragraphe nous donne une idée globale d'un processus de déformation DIJA 3D. Nous continuons par illustrer, dans la section suivante, les détails de l'application des outils 3D au niveau « dialogue ».



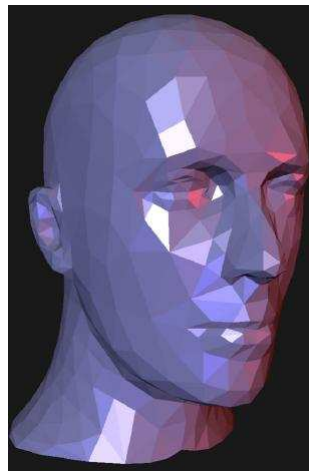
(a)



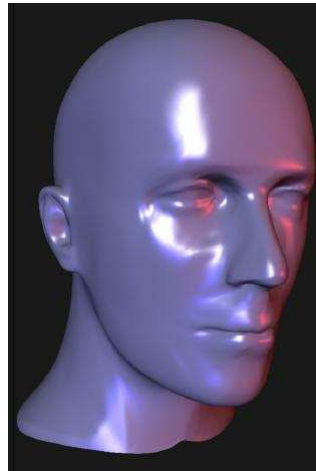
(b)



(c)



(d)



(e)

FIG. 3.5: Opérations de lissage : (a) trois pas de subdivision linéaire d'un maillage représentant la tête d'un mannequin [ZSD⁺00] ; (b) pièce initiale [ZSS96] ; (c) pièce obtenue après six itérations de subdivision Butterfly [ZSS96] ; (d) maillage initial [ZSS96] ; (e) résultat après trois itérations de subdivision Butterfly appliquée au maillage (d)[ZSS96].

3.2.3 Application des outils de déformation DIJA 3D

L'outil de déformation 3D DIJA s'applique localement sur une portion de la surface. Au niveau « dialogue », ceci revient à définir à l'aide des éléments visuels la délimitation entre la surface à déformer et le reste de la surface, ainsi que la forme, la taille et l'emplacement de la surface à déformer. Cette délimitation se fait, comme nous l'avons déjà vu, soit à l'aide des contours caractéristiques sur la face de l'objet soit en indiquant une valeur pour le rayon d'influence. La première possibilité est plus intuitive, car elle laisse à l'utilisateur la possibilité d'agir de manière visuelle sur la surface de l'objet. Cette méthode est détaillée par la suite.

Le processus commence par un dialogue entre l'utilisateur et le système afin de sélectionner le contour de la déformation. Après avoir choisi un contour proposé par le système (voir Fig. 3.6 (a)), l'utilisateur le positionne sur la surface de l'objet et le redimensionne si besoin. Le redimensionnement se fait soit à l'aide d'ascenseurs, soit en utilisant des termes de la logique floue (« plus », « moins », etc.). Le point initial de déplacement est ensuite indiqué par l'utilisateur à l'intérieur du contour par une opération de *picking* (récupérer les coordonnées d'un point de l'espace avec la souris) sur la surface de l'objet (voir Fig. 3.6 (b)). L'utilisateur a aussi le choix de faire le *picking* du point initial en 2D mais il faudrait après trouver l'équivalent en 3D. La position cible du point de déplacement est ensuite indiquée par un *picking* ou à l'aide de boîtes de dialogue (méthode non intuitive, accessible seulement aux utilisateurs expérimentés). La surface est ensuite déformée conformément au style de l'outil choisi, à la forme du contour de déformation (voir Fig. 3.6 (a)) et au vecteur de déplacement du point initial. Si l'utilisateur le désire, il a toujours la possibilité de changer la forme du contour après la déformation, et la zone à l'intérieur va suivre la forme et l'extension du nouveau contour (voir Fig. 3.7 (e-f)), en gardant les paramètres de déformation.

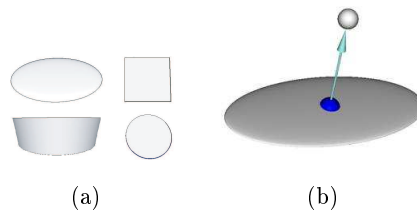


FIG. 3.6: Eléments de dialogue d'une déformation 3D : (a) différents types de contours ; (b) point initial et son vecteur de déplacement.

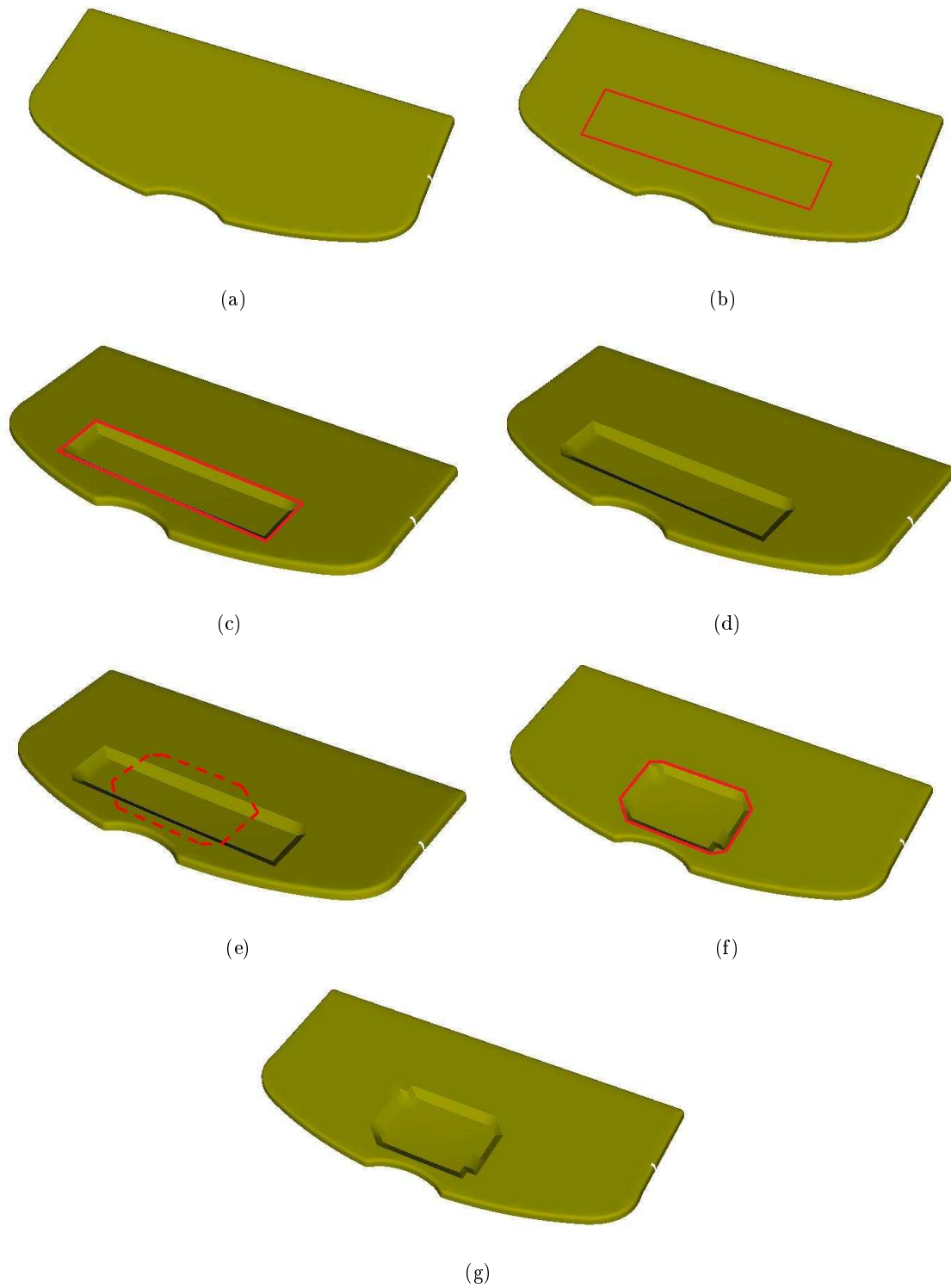


FIG. 3.7: Insertion d'un plumier dans une tablette arrière de voiture avec un outil de déformation 3D : (a) tablette initiale (*courtesy of FAURECIA Acoustics and Soft Product Line Mouzon R&D Center*) ; (b) contour « rectangle » sur la tablette ; (c-d) déformation style « bombé » borné par le contour « rectangle » ; (e) dessin à main levée d'un nouveau contour pour le plumier ; (f-g) le plumier induit par le nouveau contour.

La figure 3.7 montre un exemple de déformation d'une tablette arrière de voiture pour l'insertion d'un plumier en partant d'un contour « rectangle » et un style de déformation « bombé ». Le résultat, montré dans Fig. 3.7 (c-d), est modifié en changeant la forme du contour du plumier par un dessin à main levée (voir Fig. 3.7 (e)).

Le paramétrage présenté ici n'est qu'une des étapes d'un processus de déformation DIJA 3D qui se déroule à travers les cinq niveaux de l'architecture du système. Les différentes formes que les outils 3D peuvent prendre à chaque niveau et le vocabulaire correspondant sont étudiés dans la section suivante.

3.3 Un exemple de raffinement de déformation

Cette section présente une déformation de surface appliquée à une tablette arrière de voiture pour l'insertion d'un plumier. L'exemple montre le raffinement de la connaissance entre plusieurs niveaux d'abstraction, ce qui permet d'automatiser certaines tâches et de séparer les différentes définitions du modèle de CAO.

Le processus de déformation nécessite une série de paramètres demandés à l'utilisateur ou inférés par le système à différentes étapes de la déformation à différents niveaux d'abstraction. Pour que la déformation soit effective, les paramètres doivent être descendus jusqu'au niveau « géométrique » où ils seront passés à des différents algorithmes de déformation. Les paramètres nécessaires à l'insertion d'un plumier dans une tablette arrière de voiture sont :

- l'outil de déformation ;
- l'objet à déformer ;
- la zone à déformer ;
- le point initial de déplacement et sa position cible ;
- le style de déformation ;
- le volume et la hauteur maximaux de la déformation (pour ce cas précis).

Nous allons voir par la suite comment ces paramètres sont initialisés et descendus au niveau « géométrique ».

Le raffinement se réalise comme suit :

Acteur : l'utilisateur. Au niveau « métier », l'utilisateur sélectionne l'outil de déformation qu'il va utiliser pour réaliser une opération de thermoformage pour l'insertion d'un plumier (voir Fig. 3.8) sur une tablette arrière de voiture, comme illustré dans Fig. 3.9.

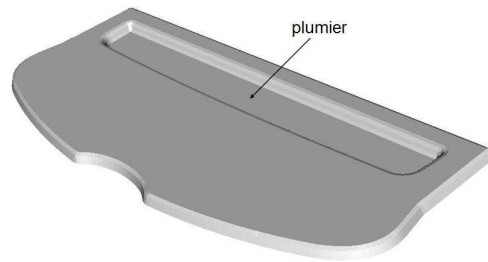


FIG. 3.8: Plumier d'une tablette arrière de voiture.

Le choix et le paramétrage de l'outil de déformation au niveau « métier » correspond à un script fourni par le module « de connaissances » et exécuté par le module « superviseur » :

Niveau métier	
MDC	MS
Définir l'outil de déformation(forme)	Définir l'outil de déformation(forme) comme thermoformage d'un plumier d'une hauteur maximale
= script {	= - outil = ?
- outil	demander à l'utilisateur → «Thermoformage»
- paramètres de l'outil	- plumier_max_height = ?
}	demander à l'utilisateur → hmax

Ceci se réalise au niveau du module IHM grâce à une interface orientée métier (voir Fig. 3.9) .

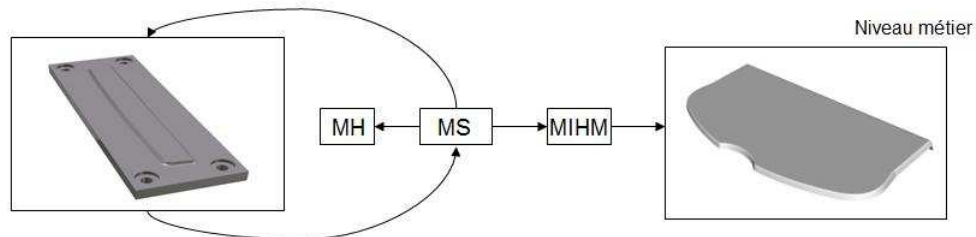


FIG. 3.9: Premier pas de raffinement : niveau métier.

Acteur : le système. Le module « superviseur » récupère le choix de l'utilisateur et le transmet au module « de connaissances » pour interprétation. Le module « de connaissances » renvoie le

script suivant :

Niveau métier

MDC

Appliquer thermoformage d'un plumier (forme)

Ce script ne peut pas être interprété à ce niveau. Par conséquent, le module « superviseur » demande au module « de connaissances » du niveau inférieur (niveau « commun ») de raffiner l'instruction « thermoformage d'un plumier ». Le module « de connaissances » du niveau « commun » retourne « embouti ». Le script est raffiné, au niveau « commun », comme suit :

Niveau commun

MDC	MS
Appliquer thermoformage d'un plumier(forme)	Appliquer thermoformage d'un plumier(forme)
= script {	= Choisir les paramètres de l'embouti
Choisir les paramètres de l'embouti	- forme = ?
- forme	demander à l'utilisateur → «Cloche»
- choisir le volume maximal de l'embouti	- embouti_max_vol = ?
- choisir la hauteur maximale de l'embouti	demander à l'utilisateur → vmax
Déformer(forme)	- embouti_max_height = ?
}	déduit du niveau «métier» → hmax
	Déformer(forme)

Afin de choisir une forme pour l'embouti, le module « superviseur » consulte le module « de connaissance » et exécute le script retourné par celui-ci. Les choix se font par l'utilisateur grâce au module IHM (voir Fig. 3.10). Le résultat de l'exécution des instructions devrait conduire à l'insertion d'un embouti sur la forme, en respectant la hauteur maximale et le volume maximal fournis.

Acteur : l'utilisateur. L'utilisateur choisit la forme de l'embouti (voir Fig. 3.10). Ensuite, grâce à l'interface fournie par le module IHM, il initialise les paramètres propres à l'embouti

(hauteur et volume maximaux).

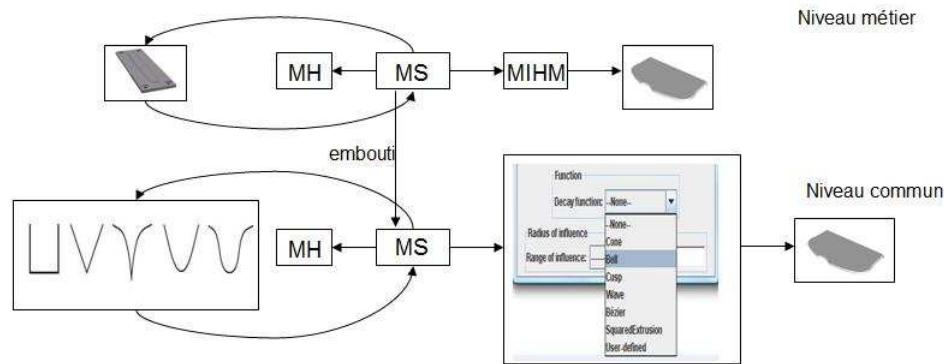


FIG. 3.10: Deuxième pas de raffinement : niveau commun.

Acteur : le système. Une fois l'embouti paramétré, le système essaie d'interpréter l'instruction « Déformer ». Pour ce faire, l'instruction est transmise au niveau inférieur (niveau « dialogue ») où elle est raffinée comme montré dans le troisième script (niveaux « dialogue » et « géométrique »).

Pour définir le contour de la déformation, le module « superviseur » consulte le module « de connaissances » du niveau « dialogue » et propose plusieurs choix à l'utilisateur via l'interface graphique.

Acteur : l'utilisateur. L'utilisateur choisit un contour « ovale » parmi les plusieurs contours disponibles et indique le positionnement du contour sur la surface de la tablette. Il indique en même temps le point initial de déplacement à l'intérieur du contour, comme le montre Fig. 3.11.

Acteur : le système. Le système récupère ces paramètres et le fait descendre au niveau « géométrique » pour récupérer les positions des points de la surface qui correspondent au contour et au point initial de déplacement. Ceci se réalise par le biais d'algorithmes que nous avons implémentés, qui récupèrent les positions 3D sur la surface de l'objet qui sont les plus proches des coordonnées 2D fournies en entrée.

Le système renvoie les points 3D résultats au niveau « dialogue » pour être visualisés. Ceci est illustré dans la figure 3.11.

Niveaux dialogue & géométrique

MDC	MS
Déformer(zone)	Déformer(forme)
= script {	= - contour = ?
Définir contour	demander à l'utilisateur → «ovale»
Définir point initial	- PI = ?
Appliquer contour et point initial sur la	demander à l'utilisateur → pos2D
forme initiale	Appliquer contour et point initial sur la forme
Définir la position cible du point initial	initiale
Déformer(forme)	algorithme → calculer les coordonnées
}	des points du contour et du point initial
	au niveau géométrique
	- PC = ?
	trouverPos(PI, valeur_slideur)
	Déformer(forme)

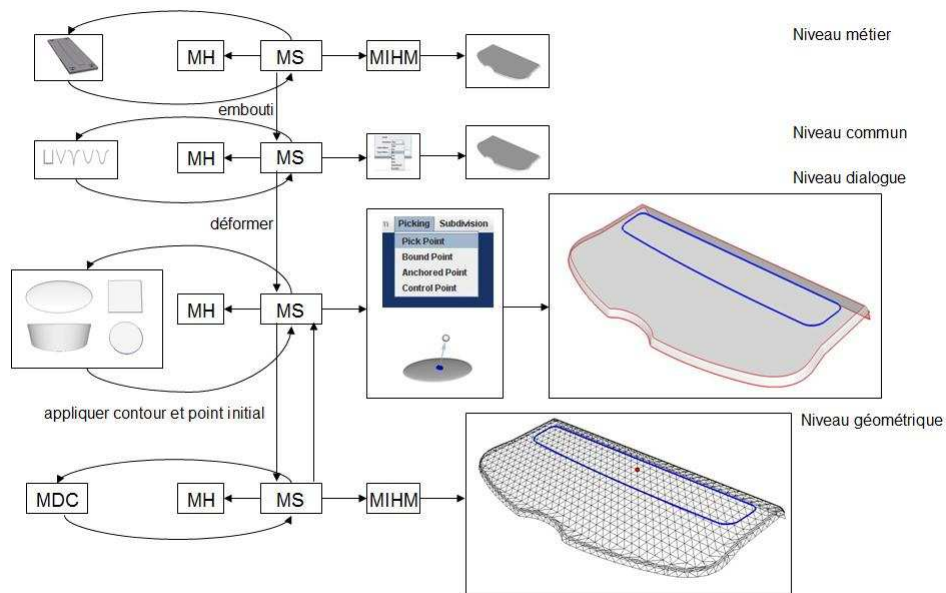


FIG. 3.11: Troisième pas de raffinement : niveau dialogue et géométrique.

Acteur : l'utilisateur. Une fois les éléments de dialogue affichés sur la surface de la tablette, l'utilisateur peut commencer à les déformer. Il applique un outil (slideur) pour indiquer la quan-

tité de déformation, i.e. la distance entre le point initial de déplacement et sa position cible.

Acteur : le système. Le système est maintenant capable d'initialiser la position cible du point de déplacement au niveau « géométrique ». Les coordonnées du point cible sont renvoyées au niveau « dialogue ». Tous les paramètres étant maintenant connus, le pas de déformation peut s'appliquer. L'instruction « Déformer » ne peut pas s'exécuter au niveau « dialogue », elle est donc descendue au niveau « géométrique » où le script devient :

Niveau géométrique

MDC	MS
Déformer(forme)	Déformer(forme)
= script {	= Définir les paramètres de déformation
Définir les paramètres de déformation	- zone = ?
- zone	algorithme → trouverPointsInterieurs (surface, contour)
- posInitiale	- posInitiale = ?
- posCible	algorithme → trouverPos3D(surface, PI)
- styleDeformation	- posCible = ?
- zoneMaxVolume	algorithme → trouverPos3D(surface, PC)
- zoneMaxHeight	- styleDeformation = ?
}	déduit du niveau supérieur → fctDecayCloche
	- zoneMaxVolume = ?
	déduit du niveau supérieur → vmax
	- zoneMaxHeight = ?
	déduite du niveau supérieur → hmax

Un algorithme est exécuté pour le déplacement de tous les points à l'intérieur du contour avec une quantité qui dépend de leur distance par rapport au point initial de déplacement. Le résultat peut être visualisé dans la figure 3.12.

La forme de la déformation est déterminée par la forme de l'embouti qui correspond, au niveau « géométrique », à une fonction mathématique. Les contraintes qui imposent la hauteur et le volume maximaux sont respectées grâce à un algorithme interactif qui propage la déformation sur la surface à l'extérieur du contour si les quantités maximales sont dépassées.

Les nouveaux éléments de dialogue sur la surface déformée sont aussi calculés par cet algorithme. La tablette résultante est transmise au niveau « métier » pour être visualisée (voir Fig. 3.13).

L'exemple que nous avons étudié représente un exemple complet de processus de déformation,

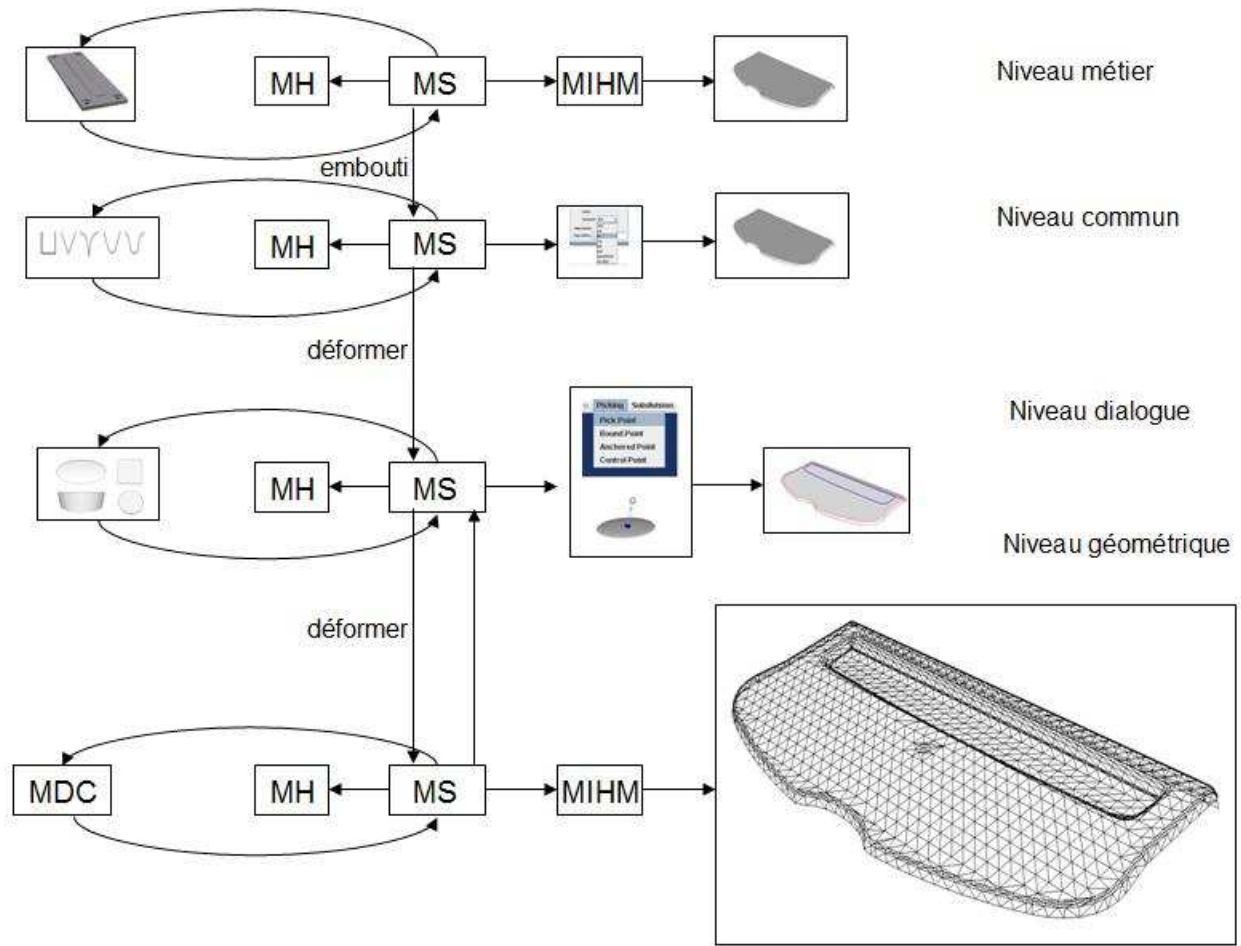


FIG. 3.12: Dernier pas de raffinement : niveau géométrique.

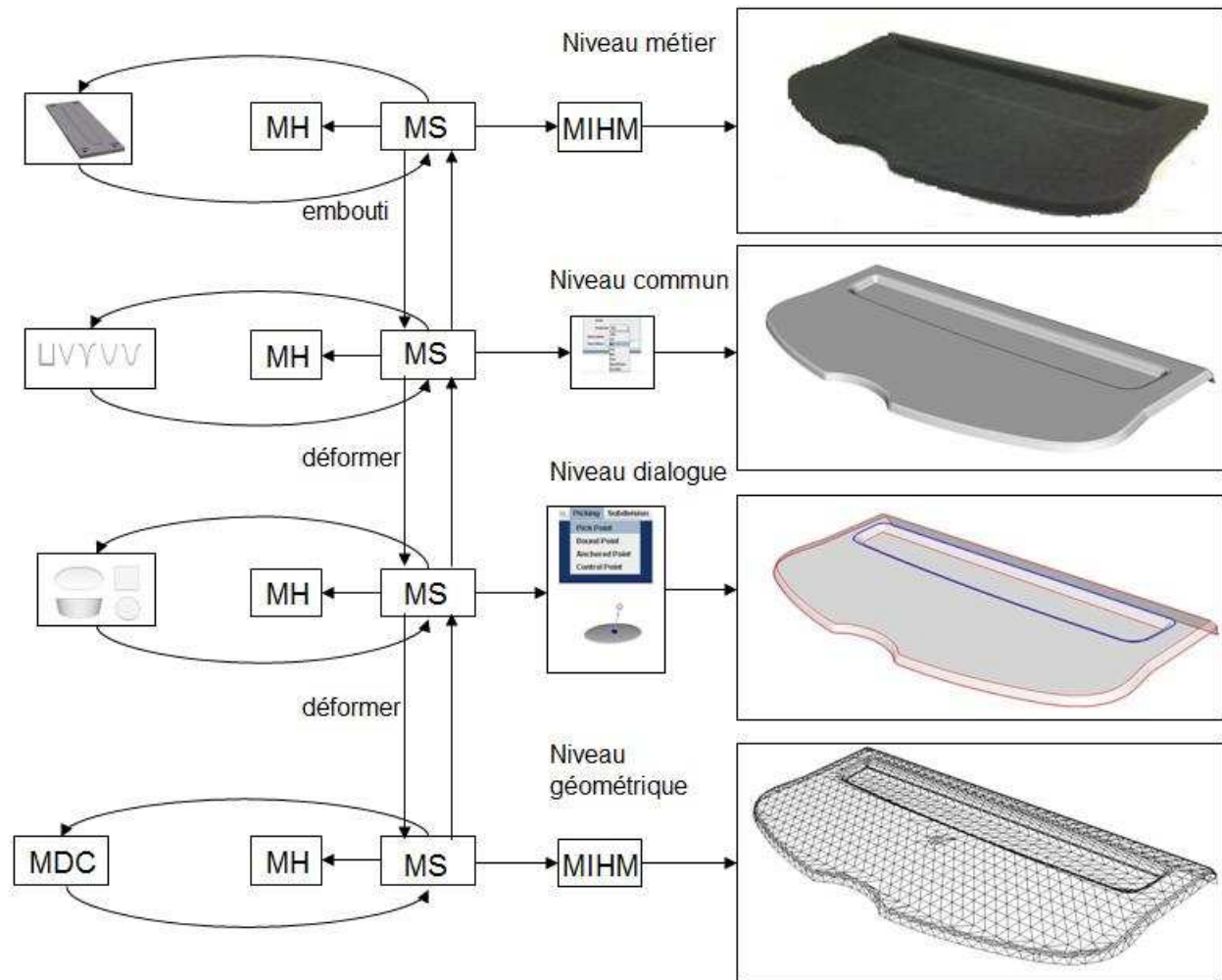


FIG. 3.13: Récapitulatif du processus de déformation ; phase de descente des paramètres et de remontée du résultat.

qui comprend la phase de descente des paramètres et la phase de remontée du modèle déformé. Le système demande, pendant la phase de descente, l'interaction de l'utilisateur pour définir des valeurs pour certains paramètres, étant capable de les inférer tout seul s'ils sont déjà définis à un niveau supérieur. Si on avait appliqué un outil intégrant un nombre plus grand de paramètres dès le début du processus, le système aurait demandé moins d'interaction de la part de l'utilisateur aux niveaux d'abstraction inférieurs au niveau de départ.

L'insertion d'un plumier dans une tablette arrière de voiture se base sur des outils de déformation avec une contrainte volumique. Cet aspect est détaillé dans la section suivante.

3.4 Une proposition pour l'intégration de la contrainte volumique avec les déformations DIJA 3D

La contrainte volumique s'intègre dans les contraintes géométriques non-linéaires évoquées dans le premier chapitre. En dépit du fait que c'est une contrainte dont le respect entraîne des calculs lents, son application a été intensément étudiée dans le domaine de la CAO et dans l'animation. Dans le domaine du design industriel et particulièrement pour le système DIJA, l'intérêt de cette contrainte réside dans le fait qu'elle est respectée automatiquement, sans limiter la créativité et l'application du savoir-faire du concepteur [CDGP08b]. Plusieurs méthodes existent dans ce sens pour la déformation de formes libres.

Un état de l'art des déformations de formes libres avec contrainte volumique, qui constitue le préambule pour notre proposition d'intégration de la contrainte volumique, dans la section 3.4.2, avec les outils de déformation DIJA 3D, est présenté dans la section suivante.

3.4.1 Etat de l'art des techniques de déformation de formes libres avec contrainte volumique

Dans cette section, les techniques de déformation avec contrainte volumique sont classifiées en deux classes. La première classe comprend des méthodes où la correction volumique est réalisée à travers d'algorithmes d'optimisation qui utilisent les propriétés physiques du modèle de déformation. La deuxième classe prend en compte uniquement les propriétés géométriques du modèle. Les méthodes propres aux deux classes sont étudiées dans les paragraphes suivants, en mettant l'accent sur les éléments importants pour notre approche : la représentation du modèle, la rapidité des algorithmes et l'intuitivité lors de leur utilisation.

Conservation volumique en tenant compte des propriétés physiques du modèle de déformation

Rappoport [RSB96] propose une méthode s'appliquant sur un modèle géométrique défini par des solides de Bézier auxquels s'ajoutent des propriétés physiques tel que le volume et l'énergie interne de déformation. Le solide de Bézier se définit en utilisant un ensemble de points de contrôle $P_{ijk} \in \mathbb{R}^3$. L'image d'un point paramétrique dans le cube unité est définie par

$$F(u, \nu, w) = \sum_{i=0}^{n_u} \sum_{j=0}^{n_\nu} \sum_{k=0}^{n_w} B_i^{n_u}(u) B_j^{n_\nu}(\nu) B_k^{n_w}(w) P_{ijk} \quad (3.2)$$

où $B_i^n(t)$ est le polynôme de Bernstein défini par

$$B_n^i(t) = \binom{n}{i} t^i (1-t)^{n-i}. \quad (3.3)$$

Le volume est calculé par une intégrale triple définie sur le cube unité

$$\int_0^1 \int_0^1 \int_0^1 J_F du dv dw \quad (3.4)$$

où J_F est le déterminant de la matrice Jacobienne de F :

$$J_F = \det\left(\frac{\partial F_i}{\partial x_j}\right), \quad i = x, y, z \quad x_j = u, \nu, w. \quad (3.5)$$

Le problème de préservation volumique de l'objet initial se traduit en un problème d'optimisation qui consiste à :

- minimiser la distance entre la configuration initiale P et celle finale Q des points de contrôle ; cette distance peut être calculée soit en utilisant une approche qui se base sur l'énergie interne nécessaire pour passer d'un état à l'autre, soit en utilisant la méthode des moindres carrés ;
- minimiser la variation ΔV entre le volume initial V_i et le volume final V_f ;
- et à respecter les contraintes linéaires imposés aux points de contrôle.

L'algorithme de préservation de volume se base sur l'algorithme d'optimisation non-linéaire d'Uzawa [AHU58, Mau03]. Cet algorithme est complexe du point de vue du calcul, et ne satisfait pas interactivement la contrainte volumique et les contraintes linéaires simultanément. La modélisation de l'objet à l'aide des points de contrôle ainsi que le manque d'une fonction de déformation indépendante diminue considérablement l'intuitivité, ce qui fait que cette approche ne s'adapte pas à nos buts.

Une autre méthode qui emploie le calcul de l'énergie interne avec une approximation lagrangienne pour préserver globalement le volume est celle proposée par Hirota et al. dans [HML]. Cette méthode s'applique à n'importe quel modèle, à condition que les maillages polygonaux multi-résolution puissent en être déduits. Le volume est calculé de manière explicite, en sommant les contributions de chaque triangle du maillage polygonal. Le volume induit par chaque triangle est calculé en multipliant l'aire A_{xy} de la projection du triangle dans le plan $\widehat{xOy}$ (voir Fig. 3.14 (a)) par la hauteur moyenne du triangle (voir Fig. 3.14 (b)) :

$$V = A_{xy}(P_{1,z} + P_{2,z} + \frac{P_{3,z}}{3}), \quad A_{xy} = (\frac{1}{2}(P_2 - P_1) \times (P_3 - P_1))_z \quad (3.6)$$

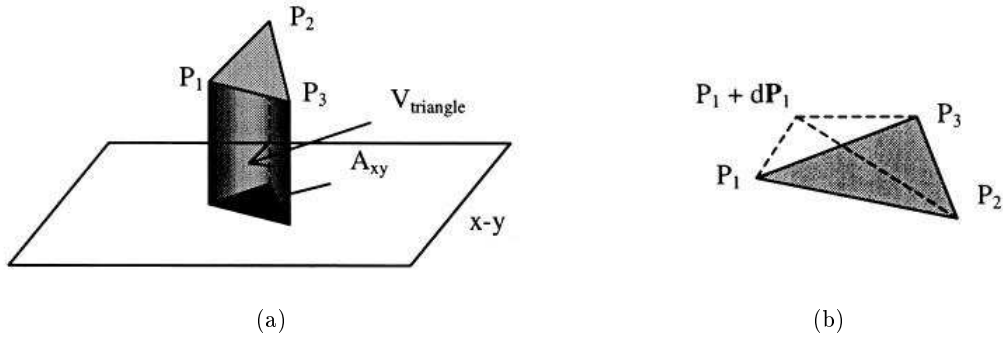


FIG. 3.14: Calcul du volume [HML] : (a) projection d'un triangle dans le plan $\widehat{xOy}$; (b) la variation de volume induite par le déplacement dP_1 du sommet P_1 .

La préservation du volume se réalise avec un algorithme itératif se basant sur une méthode d'optimisation Lagrangienne et sur une fonction d'énergie d'un réseau masse-ressorts :

$$E_{spring}(X) = \sum_j \frac{k}{2} (\sqrt{(X_{e_j} - X_{s_j})^2} - L_j)^2 \quad (3.7)$$

où X est une masse, j est son indice, e_j et s_j sont les indices des masses voisines et L_j la longueur naturelle du ressort.

A chaque itération, la variation du volume est calculée en se basant sur une approximation triangulaire de l'objet, ce qui requiert un maillage fin pour obtenir une bonne précision. Pour des raisons de coût algorithmique, les auteurs utilisent une représentation multi-résolution du maillage.

L'algorithme proposé est interactif et donne des résultats en temps réel (voir Fig. 3.15), grâce au raffinement progressif du maillage. Néanmoins, le modèle de déformation se base sur des lois physiques, ce qui implique qu'il ne soit pas approprié pour notre approche.

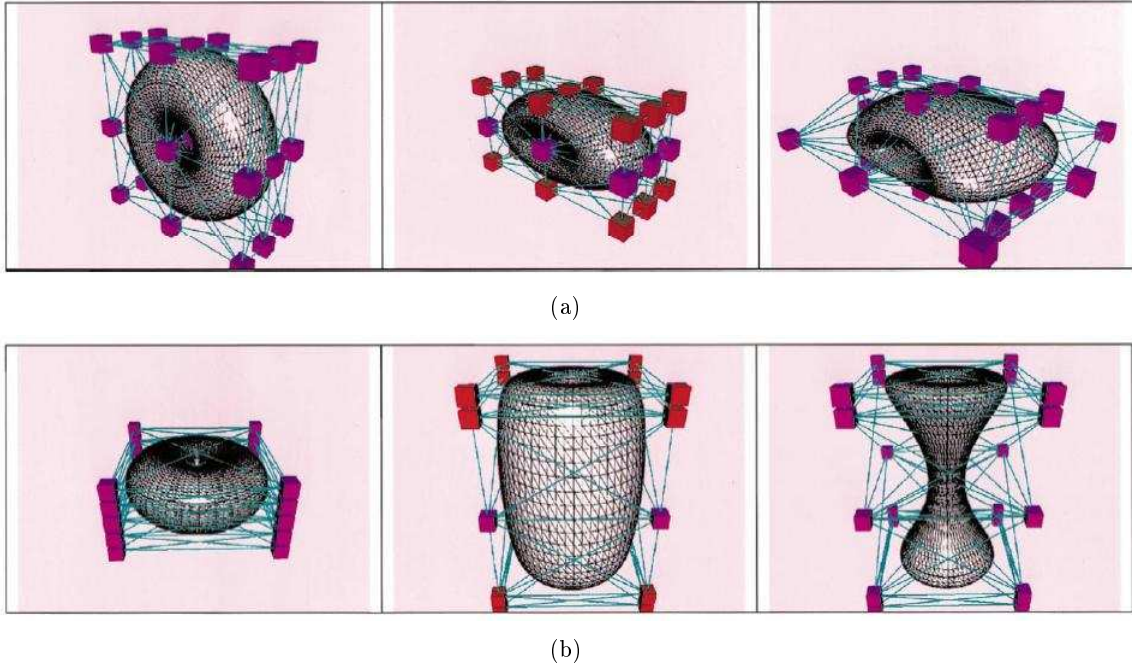


FIG. 3.15: Déformation d'un tore avec contrainte volumique [HML] : (a) tore initial, tore comprimé sans conservation volumique, tore comprimé avec contrainte volumique $x - y$; (b) tore initial, tore allongé sans conservation volumique, tore allongé avec contrainte volumique.

Un modèle anathomique basé sur des masses-ressorts pour modéliser la respiration humaine a été proposé par Zordan et al. [ZCCD06]. Un algorithme pour la conservation volumique des éléments incompressibles dans les corps déformables lui est associé pour approximer le mouvement de l'abdomen relativement à la respiration. La respiration est simulée en utilisant des forces qui tirent et qui déforment la totalité des corps impliqués dans le processus (muscles, organes internes, os) et des forces de pression pour préserver le volume des parties déformables (tel que les poumons).

L'estimation du volume est faite en sommant les volumes pyramidaux définis entre le centre de masse de l'objet et chaque face de sa surface (voir Fig. 3.16).

L'algorithme qui induit la préservation volumique se base sur les forces de pression, en faisant varier le volume linéairement avec la pression. Un exemple de simulation d'une respiration et d'une expiration complète est montré dans la figure 3.17.

Cette méthode est fortement physique, l'interaction avec l'utilisateur n'étant pas présente. Par conséquent, elle se positionne loin par rapport à nos buts de modélisation.

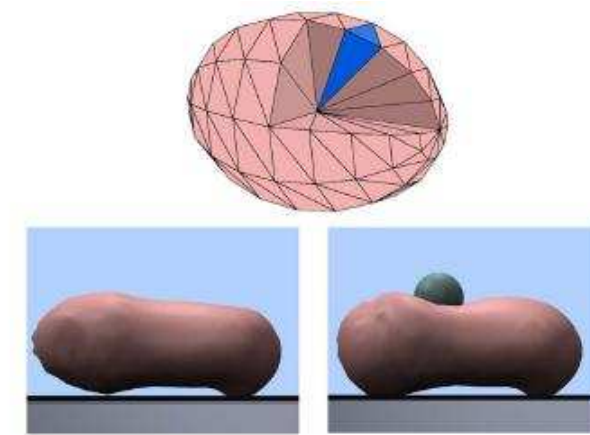


FIG. 3.16: Calcul du volume [ZCCD06] ; en bas : déformation de l'abdomen après un impact avec un corps lourd.

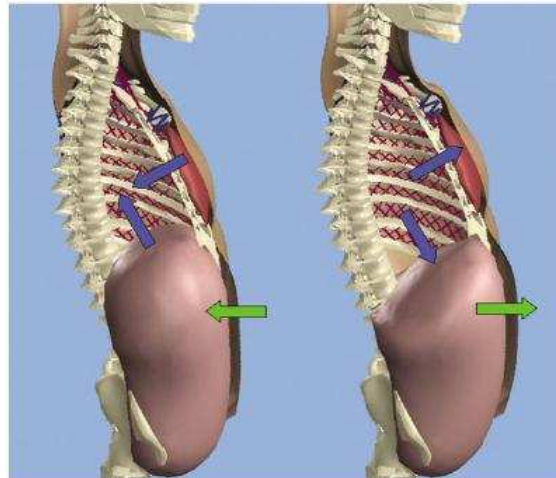


FIG. 3.17: Simulation d'une inspiration (partie droite) et d'une expiration (partie gauche) [ZCCD06] ; les flèches bleues indiquent le mouvement de la cage thoracique ; les flèches vertes indiquent le mouvement de l'abdomen.

Une deuxième approche pour la déformation avec contrainte volumique est constituée des méthodes de correction volumique globale ou locale basées sur des contraintes géométriques, tel que les contraintes de déplacement ou les contraintes de rotation. Elles sont détaillées dans le paragraphe suivant.

Conservation volumique en tenant compte des propriétés géométriques du modèle de déformation

Une première méthode est celle proposée par Aubert et Bechmann [AB97]. Cette méthode se base sur le modèle de déformation DOGME [BD93, BR94] présenté dans le premier chapitre. On rappelle que l'idée de DOGME est de manipuler directement les points de l'espace en leur imposant des contraintes linéaires. Le résultat d'une déformation sous DOGME n'est pas unique, plusieurs solutions pouvant exister pour les mêmes contraintes de déplacement imposées par l'utilisateur (voir Fig. 3.19). Dans l'espace 3D, le déplacement d'un point U en U' se réalise par la fonction suivante :

$$d(U)_{3 \times 1} = M_{3 \times m} f(U)_{m \times 1} \quad (3.8)$$

où f représente la fonction « d'extrusion » qui diffère selon le cas (plusieurs expressions de f peuvent être vues dans [BR94]) et M est la matrice de projection. La fonction f transforme un point de $\mathbb{R}^3$ en un point de $\mathbb{R}^m (\forall m)$, et M transforme un point de $\mathbb{R}^m$ en un point de $\mathbb{R}^3$.

La variation de volume entre l'objet initial et l'objet déformé est calculée en sommant les variations volumiques déterminées par chaque contrainte de déplacement (voir Fig. 3.18) et s'exprime par la formule suivante :

$$\Delta V(P) = \frac{1}{6} \sum_{i=1}^n \alpha_i dU_i \quad (3.9)$$

où P est l'objet initial, $U_i, i \in [1, n]$ sont les sommets de P , dU_i sont les vecteurs de déplacement des sommets de P et α_i représente la somme des produits vectoriels des facteurs de dU_i .

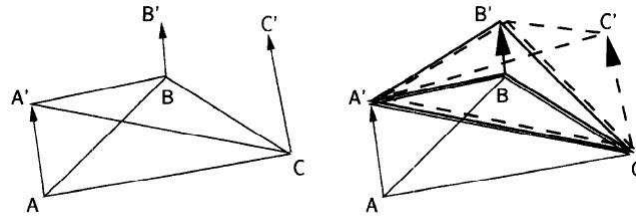


FIG. 3.18: La variation volumique induite par le déplacement des sommets A vers A' , de B vers B' et de C vers C' est la somme des volumes des tétraèdres (A, B, C, A') , (A, B, C, A') et (A, B, C, C') [AB97].

L'idée est d'approcher la valeur zéro de $\Delta V(P)$ par des approximations successives des termes α_i . La conservation volumique se réalise de manière itérative et se base sur l'algorithme proposé par les auteurs dans [AB97] et présenté dans la section 3.4.2.

Afin de minimiser la variation volumique, les auteurs proposent une formulation de l'équation (3.9) en ajoutant un terme d'optimisation :

$$d^t(U) = f^t(U)(X^+D + (I - X^+X)\xi), \quad M^t = X^+D + (I - X^+X)\xi \quad (3.10)$$

où

$$X = \begin{pmatrix} f^t(V_1) \\ \vdots \\ f^t(V_{n_c}) \end{pmatrix}, \quad D = \begin{pmatrix} d^t(V_1) \\ \vdots \\ d^t(V_{n_c}) \end{pmatrix}$$

et $(V_i)_{i \in [1, n_c]}$ représentent les n_c contraintes de déplacement. Le rôle de la matrice ξ est d'aider à la préservation volumique. Elle détermine des attractions ou des répulsions de parties de l'objet modélisé qui ne sont pas fixées par les contraintes de déplacement (voir Fig. 3.20).

Cette méthode de préservation volumique semble prometteuse et convient à notre approche de modélisation : elle s'applique à des maillages triangulaires, l'algorithme est simple et les contraintes de déplacement ainsi que la contrainte volumique sont respectées simultanément de manière automatique. L'algorithme proposé permet la conservation du volume globalement, mais il est possible de le réaliser localement sur plusieurs parties de l'objet. D'autres contraintes géométriques pourraient être imposées avec une adaptation de l'algorithme. La méthode de Borrel et Bechmann a été testée par une maquette de déformation 3D DIJA pour la conception des flacons de parfum et présentée dans la section 3.4.2.

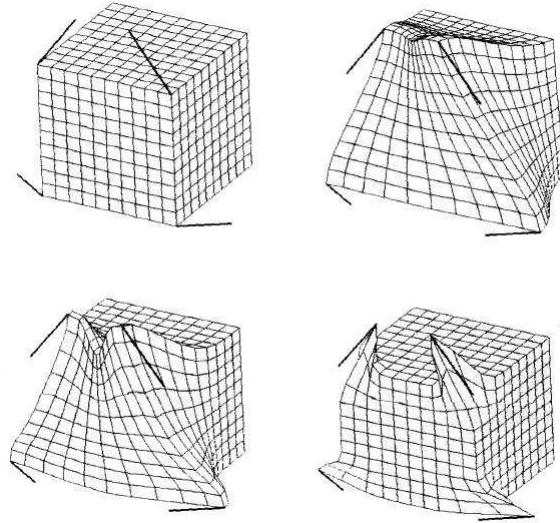


FIG. 3.19: Déformation DOGME avec une fonction B-spline [BR94] : différentes solutions possibles pour le même objet initial et les mêmes contraintes de déplacement, en fonction du nombre de points de contrôle.

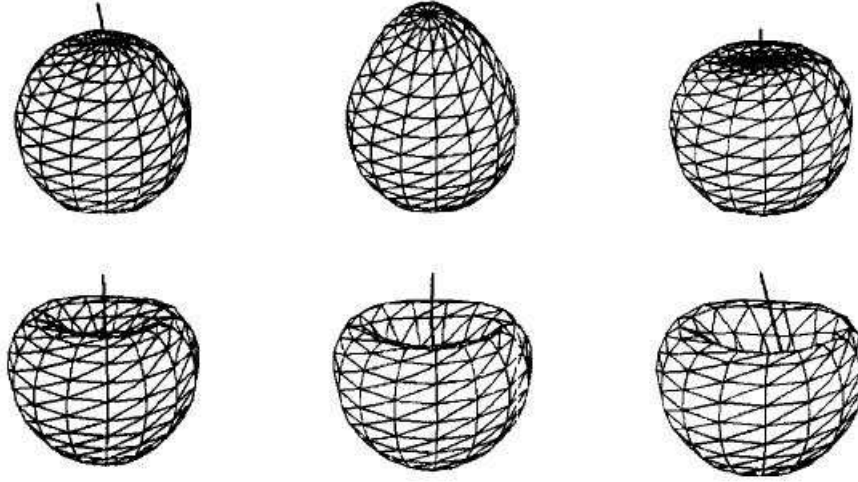


FIG. 3.20: Attraction et répulsion des parties déformables de la sphere pour la même contrainte de déplacement lors d’une déformation avec conservation du volume [BR94].

Une approche différente pour la conservation volumique en s’appuyant sur les propriétés géométriques du modèle a été proposée par Lipman et al. dans [LSCO⁺04]. Il s’agit d’une technique de représentation *moving frames* qui s’appuie sur les coordonnées différentielles d’un maillage triangulaire.

Le volume est estimé par rapport à cette représentation, localement pour chaque point de la surface. Une *moving frame* (e_1, e_2, e_3) est associée à chaque point du maillage. Elle est transformée dans un volume D en la translatant dans la direction $(-e_3)$ de la normale comme montré dans la figure 3.21.

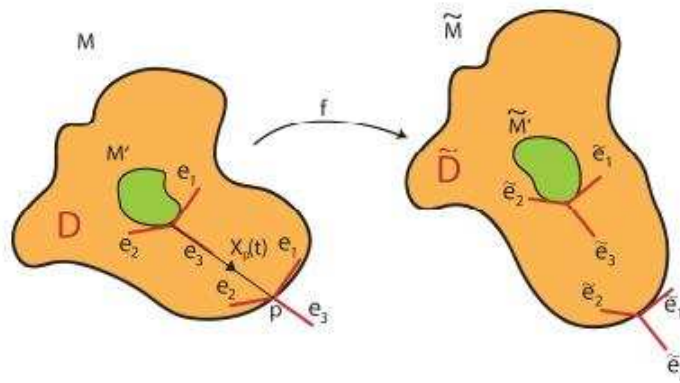


FIG. 3.21: Estimation du volume [LCOGL07].

La préservation volumique se fait seulement par des opérations sur la surface de l'objet. Un facteur d'échantillonnage et des contraintes de rotation qui se basent sur une fonction Dirichlet sont employés afin de préserver le volume et les détails de la surface localement. Le facteur d'échantillonnage dérive de la relation définie entre le volume local et la courbure de la surface qui lui correspond. Un exemple de déformation d'une pièce de lego est montré sur la figure 3.22.

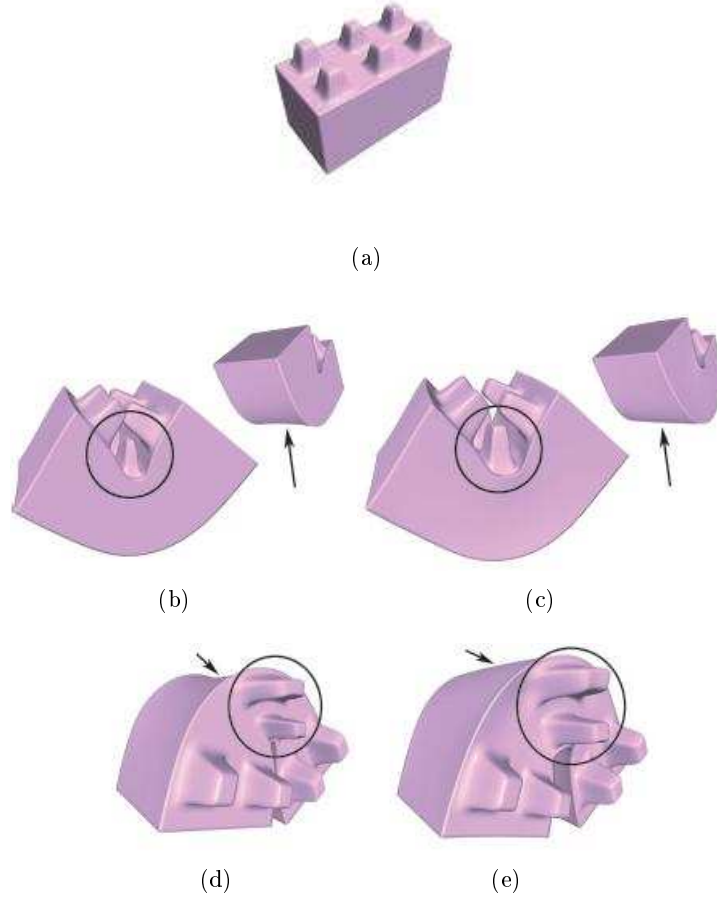


FIG. 3.22: Déformation d'une pièce de lego avec contrainte volumique [LCOGL07] : (a) pièce initiale ; (b-d) pièce déformée sans contrainte volumique ; (c-e) déformation avec correction du volume.

Même si cet algorithme donne de très bons résultats, l'existence du facteur d'échantillonnage empêche son utilisation sous DIJA, puisque celui-ci ne permet pas le respect des autres contraintes sur les points de la surface (tel que les contraintes de déplacement par exemple).

Les cinq méthodes que nous avons présentées constituent un état de l'art non exhaustif des méthodes de déformation des surfaces de formes libres avec contrainte volumique. Elles sont résumées dans le tableau 3.1.

Méthode	Modèle	Calcul du volume	Préservation du volume	Avantages/ Inconvénients
Rappoport et al. 1996	Primitives : volumes de Bézier	Explicitement : intégrale triple	Calcul d'énergie Algorithme Uzawa Multiplication Lagrangienne	i : Calculs lents i : Pas de manipulation directe i : Modèles continus
Aubert et Bechmann 1997	B-Rep Maillages polygonaux	Représentation explicite : Formule directe pour ΔV	Optimisation non-linéaire Approximation itérative Contraintes de déplacement Préservation volumique globale	a : Maillages Δ a : Algorithme simple a : Contraintes de déplacement
Hirota et al. 2000	N'importe quel modèle Maillages polygonaux Optimisation multi-niveaux	Représentation explicite : Formule directe pour ΔV	Energie élastique Approximation Lagrangienne Préservation volumique globale	a : Intuitivité a : Rapidité (calcul, rendu) a : Versatilité i : Masses-ressorts i : Pas de manipulation directe
Zordan et al. 2006	Masses-ressorts Forces de déformation Energie	Représentation explicite $\sum_i V_i$ i -nombre de Faces triangulaires	Forces de pression Préservation volumique globale	i : Pas de description explicite i : Déformation de surfaces implicites i : Pas d'interaction avec l'utilisateur
Lipman et al. 2007	Maillages Δ Coordonnées différentiels (<i>moving frames</i>)	Représentation implicite : Opérations sur la surface seulement Relation volume-courbure de la surface	Localement : facteur d'échantillonnage Contraintes de rotation Préservation volumique locale	a : Représentation discrète i : Facteur d'échantillonnage

TAB. 3.1: Méthodes de FFD avec contrainte volumique.

La plupart des méthodes décrites utilisent des propriétés de la physique pour la préservation du volume, ce qui ne correspond pas à notre approche. Nous rappelons au lecteur que notre approche est purement géométrique et orientée métier, même si le résultat approxime l'objet physique désiré. D'autres ne sont pas nécessairement intuitives, entraînant des déformations nonprévisibles et des calculs lents. La méthode proposée par [AB97] semble la plus proche de nos buts puisqu'elle s'applique aux maillages triangulaires, avec de bons temps de réponse de la part du système. Son adaptation à DIJA est étudiée dans la section suivante.

3.4.2 Intégration d'une contrainte volumique avec les outils 3D de déformation DIJA selon la méthode d'Aubert et Bechmann

Nous montrons, dans cette section, l'intégration automatique d'une contrainte volumique avec les outils de déformation 3D présentés dans la section 3.2 de ce chapitre. La méthode de préservation du volume est celle proposée par [AB97] et résumée dans la section précédente. Cette méthode est testée dans une maquette DIJA de déformation 3D pour la modélisation des flacons de parfum.

Pour l'expression de la fonction d'extrusion f , on choisit le produit matriciel de trois B-Splines

cubiques :

$$f(u) = \bigotimes_{k=1}^3 f^k(u_k), \quad f^k(u_k) = (B_0^{d_k}, B_1^{d_k}, B_2^{d_k})^t, k \in [1, 3] \quad (3.11)$$

où

$$B_i^1(\mu) = \begin{cases} 1, & \text{si } \mu \in [\mu_i, \mu_{i+1}] \\ 0, & \text{si } \mu \notin [\mu_i, \mu_{i+1}] \end{cases}$$

et

$$B_i^d(\mu) = \frac{\mu - \mu_i}{\mu_{i+d-1} - \mu_i} B_i^{d-1}(\mu) + \frac{\mu_{i+d} - \mu}{\mu_{i+d} - \mu_{i+1}} B_{i+1}^{d-1}(\mu)$$

B_i^d sont des fonctions polynômiales définies complètement par $d + 1$ nœuds : $\mu_i, \dots, \mu_{i+d+1}$. Pour notre application, trois splines sont définies au long des trois axes uniformément sur six nœuds, ce qui induit une dimension $m = 9$ à l'espace de la fonction d'extrusion f . Les splines cubiques induisent une continuité C^2 sur la surface déformée. Le nombre de nœuds et leur distribution le long des trois axes déterminent la localisation de la déformation autour des contraintes de déplacement. Plus le nombre des nœuds est grand, plus la déformation est locale. Le nombre et la distribution des nœuds peuvent être ajustés en fonction des besoins de la modélisation. Il est important de retenir qu'un nombre trop grand de nœuds entraîne des calculs additionnels. La figure 3.19 montre l'influence du nombre de nœuds sur la déformation d'un cube avec quatre contraintes de déplacement.

Une fois la fonction f de déformation choisie et les contraintes de déplacement spécifiées, l'utilisateur peut lancer l'opération de déformation. L'algorithme de déformation avec contrainte volumique proposé par [AB97] (que notre maquette implémente) est détaillé dans Algorithme 3.

Le résultat d'une déformation avec contrainte volumique couplée à une contrainte de déplacement appliquée à une forme cylindrique est montré dans la figure 3.23. La correction du volume se répercute, dans ce cas, de manière inattendue, sur une partie déformable de la surface près du cylindre qui représente le bouchon du flacon. La conservation du volume se réalise par plusieurs itérations de l'algorithme 3.

Algorithme 3 L'algorithme de préservation volumique de Aubert et Bechmann

- 1: **Entrée :**
 - Fonction f d'extrusion de dimension m
 - Polyèdre initial P et ses sommets $(P_i)_{i \in [1, n]}$
 - Contraintes de déplacement
 - 2: **Sortie :** l'objet P' qui préserve le volume et qui respecte les contraintes de déplacement
 - 3: **Départ**
 - 4: Calculer X et D à partir des contraintes de déplacement et de la fonction f
 - 5: Calculer la pseudo-inverse X^+ de X
 - 6: $iter = 0$
 - 7: $\xi^{nouveau} = 0$
 - 8: **Répéter**
 - $\xi^{ancien} = \xi^{nouveau}$
 - Calculer $P' = Deformation(P, \xi^{ancien})$
 - Initialiser $\beta_i = (\beta_{i1}, \beta_{i2}, \beta_{i3})$ à zéro $\forall i \in [1, n]$
 - Pour chaque** $\triangle(P_i, P_j, P_k) \in P$
 - $CumulerBeta(\beta_i, \beta_j, \beta_k)$
 - FPour**
 - $B = -\sum_{i=1, j=1..3}^n \beta_{ij}^{(1)} f^t(U_i) X^+ D_j, \forall k \in [1, m], j \in [1, 3]$
 - $A_{kj} = \sum_{i=1}^{(n)} \beta_{ij}^{(1)} f^t(U_i) (I - X^+ X)_k$
 - $\xi_{kj}^{nouveau} = \frac{BA_{kj}}{\sum_{i=1, l=1..3}^m A_{il}^2}, \forall k \in [1, m], j \in [1, 3]$
 - $iter = iter + 1$
 - 9: **TantQue** ($\|\xi^{nouveau} - \xi^{ancien}\| < d_{limit}$ ou $iter \geq iter_{max}$)
 - 10: **Arrêt**
-

Les deux procédures - *Deformation* et *CumulerBeta* - appelées dans l'algorithme précédent sont détaillées par la suite, dans l'Algorithme 4 et l'Algorithme 5.

Algorithme 4 La procédure qui calcule la déformation de P pour un ξ donné

- 1: **procédure** *Deformation*(P, ξ)
 - 2: **Entrée :** P, ξ
 - 3: **Sortie :** P' qui correspond à ξ
 - 4: **Départ**
 - 5: **Pour chaque** sommet $P_i \in P$ faire
 - $P'_i = P_i + f^t(P_i)(X^+ D + (I - X^+ X)\xi)$
 - 6: **FPour**
 - 7: **Fin procédure**
-

Algorithme 5 La procédure qui calcule les nouvelles valeurs $\beta_i, \beta_j, \beta_k$ pour le ξ courant

- 1: **procédure** *CumulerBeta*($\beta_i, \beta_j, \beta_k$)
 - 2: **Entrée** : $\beta_i, \beta_j, \beta_k$
 - 3: **Sortie** : les nouvelles valeurs $\beta_i, \beta_j, \beta_k$
 - 4: **Départ**
 - 5: $\beta_i = \beta_i + 2(P_i P_j \wedge P_j P_k) + P_i P'_j \wedge P'_j P_k + P_i P_j \wedge P_j P'_k + 2(P_i P'_j \wedge P'_j P'_k)$
 - 6: $\beta_j = \beta_j + 2(P_i P_j \wedge P_j P_k) + P'_i P_j \wedge P_j P_k + P_i P_j \wedge P_j P'_k + 2(P'_i P'_j \wedge P'_j P'_k)$
 - 7: $\beta_k = \beta_k + 2(P_i P_j \wedge P_j P_k) + P'_i P_j \wedge P_j P_k + P_i P'_j \wedge P'_j P_k + 2(P'_i P'_j \wedge P'_j P'_k)$
 - 8: **Fin procédure**
-

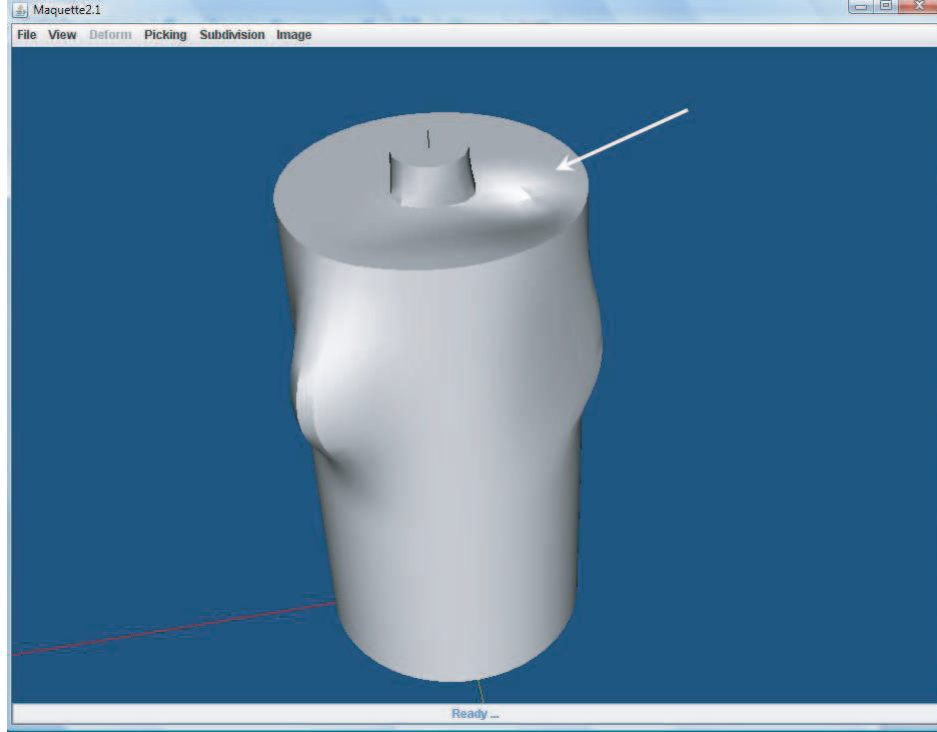


FIG. 3.23: Déformation inattendue sur la surface d'un cylindre représentant une bouteille en forme primaire ; la déformation est générée par l'algorithme de conservation volumique (Algorithme 3) : la partie haute, près du bouchon de la bouteille.

Les déformations inattendues induites par la correction du volume pendant les itérations de l'algorithme 3 diminuent l'intuitivité des déformations, et par conséquent l'intuitivité DIJA. L'algorithme ne donne plus de contrôle à l'utilisateur sur la forme résultante de l'objet, ce qui résulte dans le non respect des règles métier. Le coût de calcul augmente avec la précision de la déformation (qui dépend du nombre des splines et nœuds, dans ce cas), ce qui induit un ralentissement considérable du rendu de l'objet final.

La méthode qu'on vient de présenter s'inscrit dans un contexte FFD (DOGME) [BD93,

BR94]. Nous rappelons que DIJA n'implémente pas la déformation d'un objet en l'englobant dans un volume, mais en appliquant librement et localement des outils de déformation sur sa surface. Pour répondre aux spécifications DIJA, nous avons adapté l'intégration de la contrainte volumique à la philosophie et aux outils de notre système. La section suivante propose une méthode simple qui se base sur un algorithme *marching cubes* et sur la spécificité des outils de déformation que ces travaux de recherche proposent.

3.4.3 Proposition d'une méthode pour l'intégration de la contrainte volumique avec les outils de déformation 3D DIJA. Application pour la modélisation de flacons de parfum et de récipients cosmétiques

Le modèle soumis à la déformation avec contrainte volumique proposée dans cette section s'inscrit dans les modèles de représentation DIJA. N'importe quel modèle est accepté, à condition que le maillage triangulaire correspondant à sa surface puisse en être déduit. Le maillage triangulaire de l'objet est passé à un algorithme de la librairie VTK [VTK] (introduite dans le quatrième chapitre) qui s'appuie sur l'algorithme *marching cubes* [LC87] pour estimer son volume.

L'algorithme de *Marching cubes* est un algorithme d'infographie pour la construction de surface pour la visualisation des données 3D. Il est principalement appliqué dans le domaine de la visualisation médicale pour la reconstruction de surfaces - à partir des images issues de scanners et d'IRM (voir Fig. 3.25) - et pour la modélisation d'objets organiques 3D.

Le principe de l'algorithme *marching cubes* est de produire un maillage triangulaire en calculant des iso-surfaces à partir des données discrètes (voir les annexes A). Il parcourt un champ scalaire, considérant huit points à la fois - qui définissent un cube imaginaire - et détermine les polygones à créer pour représenter une partie de la surface contenue dans le cube. La représentation de la surface dans sa globalité se réalise en connectant les carreaux contenus dans tous les cubes.

Il existe 256 configurations possibles ($2^8 = 256$) d'un carreau dans un cube, en traitant chacune des 8 valeurs scalaires en tant que bit dans un nombre entier de 8 bits. Les 256 configurations se réduisent, par réflexion et symétrie, à 15 cas possibles, illustrés dans la figure 3.24.

La variation $\Delta V = V_{final} - V_{initial}$ du volume se calcule après avoir lancé l'opération de déformation, pour connaître le volume de l'objet final. La correction du volume se fait en propageant la quantité de la variation sur les parties non déformables des flacons, en longueur. Ainsi, le récipient cosmétique sera soit allongé, soit rétréci, selon le sens de déplacement du point initial (vers l'intérieur ou vers l'extérieur du récipient). La valeur avec laquelle l'objet est allongé/rétréci

se calcule selon la formule $z = \frac{\Delta V}{\text{aire}_{\text{base flacon}}}$. L'objet garde sa topologie après la déformation, et le résultat peut être facilement prédit par l'utilisateur.

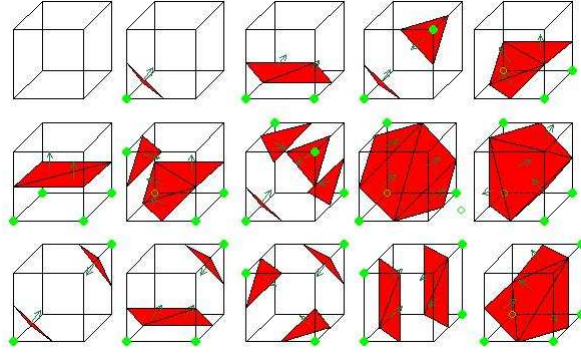


FIG. 3.24: Les configurations d'un carreau dans un cube.

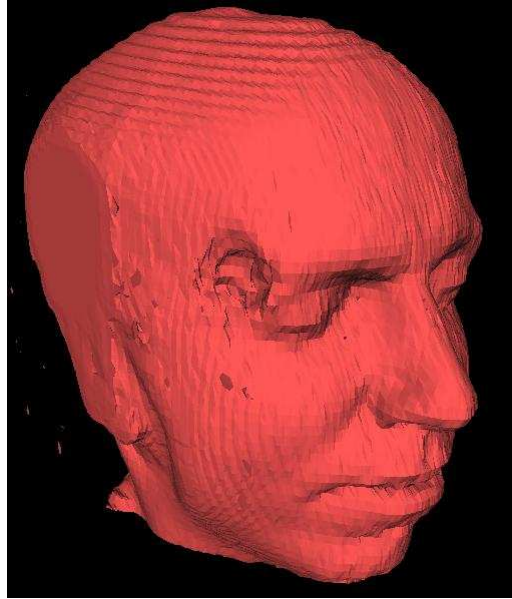


FIG. 3.25: Maillage extrait de 150 coupes IRM utilisant l'algorithme *marching cubes* [Wik].

L'algorithme que nous proposons pour la correction du volume est décrit dans Algorithme 6. Il est adapté aux outils de déformation DIJA 3D. Il donne la possibilité de préserver le volume de manière intuitive, en gardant la forme de l'objet sur les parties non-déformables. L'algorithme n'étant pas itératif, il est très rapide, le rendu de l'objet déformé étant instantané.

La figure 3.26 montre un exemple de flacons de parfum modélisés avec la méthode que nous venons de décrire.

Algorithme 6 La procédure qui calcule la correction du volume

-
- 1: **procédure** *redresserVolume*($P, \Delta V$)
 - 2: **Entrée** : l'objet P après déformation, sans correction volumique et ΔV
 - 3: **Sortie** : P dont le volume est égal au volume initial
 - 4: **Départ**
 - 5: Calculer $A = aire_{base_P}$ //sachant que $base_P$ est une zone non déformable
 - 6: Calculer $dz = \frac{\Delta V}{A}$ //la quantité de translation en z des points dans la partie non-déformable
//sur laquelle la correction volumique s'applique
 - 7: **Si** $\Delta V > 0$ **Alors** //l'objet est allongé
 Pour tous les points $V(x_V, y_V, z_V) \in Zone_de_correction_de_P$ faire
 $z_V = z_V + dz$
 FPour
 - 8: **SinonSi** $\Delta V < 0$ **Alors** //l'objet est rétréci
 Pour tous les points $V(x_V, y_V, z_V) \in Zone_de_correction_de_P$ faire
 $z_V = z_V - dz$
 FPour
 - 9: **FSi**
 - 10: **Fin procédure**
-

Voici quelques exemples d'objets dont le volume est préservé par cet algorithme :

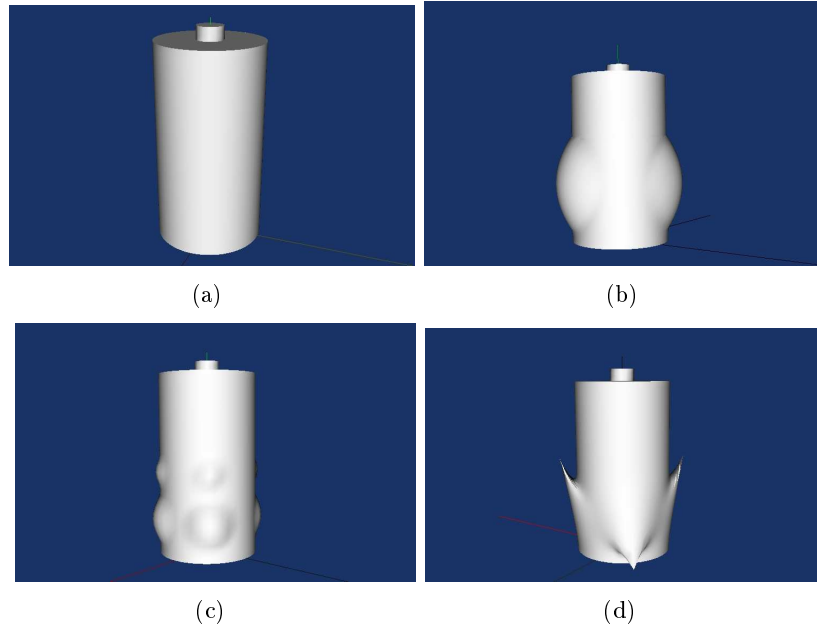


FIG. 3.26: Déformation d'un flacon de parfum avec contrainte volumique selon différents styles ; le flacon est rétréci pendant la correction du volume : (a) flacon initial ; (b) déformation de type « bombé » ; (c) déformation de type « cloche » ; (d) déformation de type « sommet ».

L'algorithme de correction volumique se prête à des objets allongés, représentés soit par des formes de base (cylindres, parallélépipèdes, etc.), soit par des flacons existants dans les bases de

données DIJA. Il est nécessaire, afin de pouvoir appliquer la correction volumique, de fixer des zones non-déformables en longueur (le long de l'axe Oz) sur la surface du flacon modélisé.

Nous avons choisi la contrainte volumique grâce à son importance et son universalité dans les applications industrielles, afin de démontrer l'intégration des contraintes métier avec les outils de déformation DIJA. De la même manière, d'autres contraintes peuvent être intégrées automatiquement dans le système DIJA. La section suivante reprend l'exemple présenté dans la section 3.3 pour montrer cet aspect.

3.5 Intégration d'autres contraintes avec les déformations DIJA. Application pour l'insertion d'un plumier pour une tablette arrière de voiture

L'insertion des plumiers dans une tablette arrière de voiture se fait en appliquant des emboutis sur sa surface. Ceci se fait, comme l'exemple de la section 3.3 le montre, avec un paramétrage spécifique, qui intègre de contraintes métier liées aux limitations spatiales et à la résistance du matériau. Même si les déformations qu'on applique sont purement géométriques, il est possible de réfléchir en termes de lois de la physique, ce qui peut être tout à fait naturel pour certains métiers. Les paramètres font le passage entre les lois réelles et le modèle géométrique et sont raffinés au fur et à mesure jusqu'au niveau géométrique où ils vont représenter la contrainte physique en termes géométriques.

Pour notre exemple concret, l'utilisateur peut appliquer des emboutis sur la surface de la tablette, mais ils ne doivent pas dépasser un certain nombre et leur dimensions doivent être en accord avec les dimensions imposées par les paramètres définis au début de la modélisation. Dans ce cas, le volume ne représente plus une contrainte, mais un paramètre.

Le processus de déformation reste le même qu'avant, seuls les outils de déformation et leurs activation changent, et par conséquent les paramètres de déformation. Plus concrètement, des paramètres additionnels pour le nombre d'emboutis, la distance qui les sépare, le volume et la distance maximaux sont imposés.

L'action de l'outil de déformation change aussi. Pour les cas où l'utilisateur ne respecte pas les contraintes dont les valeurs des paramètres correspondent, la déformation va se propager soit en haut, soit sur la surface autour de la zone déformable (si possible). Prenons l'exemple où l'utilisateur indique une hauteur pour l'embouti qui excède la hauteur maximale permise. Si le volume qui correspond à la hauteur indiquée est inférieur ou égal au volume maximal,

l'outil impose ce volume et la déformation se propage itérativement sur la zone non-déformable qui entoure le contour de la déformation jusqu'à la convergence de la hauteur vers la valeur maximale permise. Si au contraire, le volume calculé avec la hauteur indiquée par l'utilisateur dépasse le volume maximal, l'outil impose la hauteur maximale et la déformation se propage sur la zone autour du contour de déformation jusque le volume devient inférieur ou égal au volume maximal. Ces cas de déformation sont illustrés dans la figure 3.27.

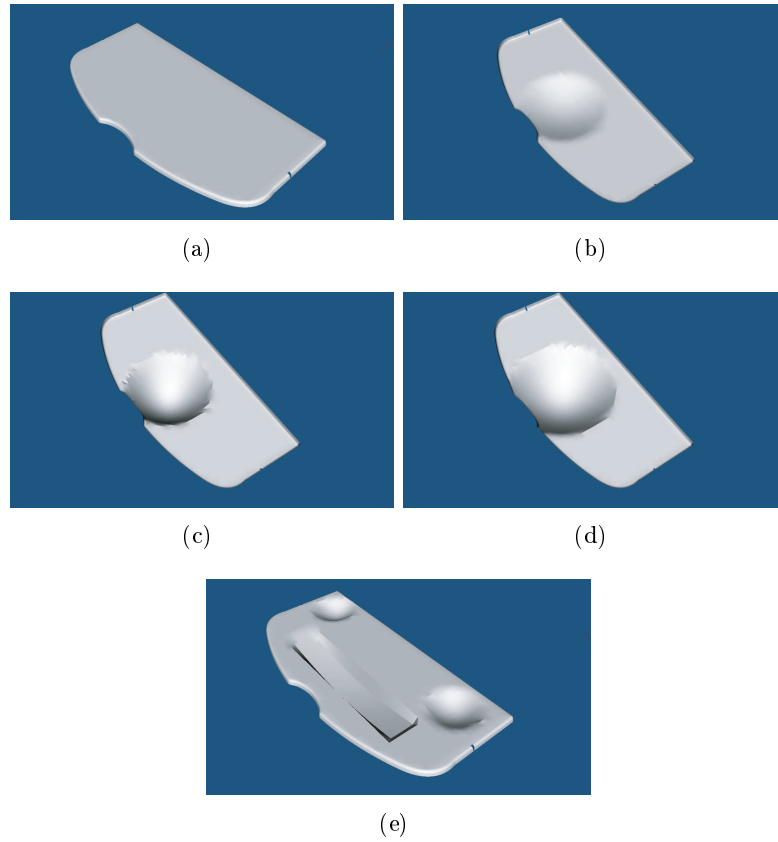


FIG. 3.27: Insertion d'un plumier dans une tablette arrière de voiture par application d'emboutis sur la tablette initiale : (a) tablette initiale (*courtesy of FAURECIA Acoustics and Soft Product Line Mouzon R&D Center*); (b) embouti style « bombé » quand $h \leq h_{max}$; (c) embouti style « bombé » quand $h = h_{max}$; (d) embouti style « bombé » quand $h > h_{max}$; (e) exemple de tablette avec insertion de trois plumiers style « bombé » avec différents contours en respectant les contraintes.

3.6 Conclusion

Nous avons présenté dans ce chapitre les apports que ces travaux de recherche ont amené à la modélisation par déformation proposée par DIJA. Notamment, nous proposons, tout d’abord, l’implémentation des outils de déformation 3D en se basant sur la simplicité des fonctions *decay*. Leur définition et application sont démontrées avec un exemple simple de processus de modélisation 3D.

L’aspect métier des outils proposés est ensuite mis en évidence par un exemple de déroulement des paramètres métier lors de l’insertion d’un plumier dans une tablette arrière de voiture à travers les cinq niveaux de l’architecture DIJA. Le problème qui se pose est comment intégrer de manière automatique des contraintes métier dans le système. Pour répondre à cette question, nous avons étudié l’insertion d’une contrainte métier spécifique, la contrainte volumique, pour la conception des flacons cosmétiques.

Un état de l’art non exhaustif est fait pour la déformation FFD avec préservation volumique. Nous avons comparé l’approche DIJA avec la méthode proposée par Aubert et Bechmann pour la conservation du volume. Les résultats s’avèrent inadéquats pour nos buts de modélisation, car la méthode testée n’est pas suffisamment intuitive et entraîne des calculs lents.

Nous proposons ainsi une méthode simple et rapide pour la correction du volume se basant sur un algorithme de *marching cubes* et sur la spécificité de nos outils de déformation de surface. L’algorithme est très intuitif, car la forme de l’objet modélisé ne change pas avec la correction volumique et donne à l’utilisateur visuellement l’impression d’allongement ou de rétrécissement de son objet en fonction de ses actions, de manière naturelle.

De la même façon que pour la contrainte volumique, on arrive à intégrer d’autres contraintes à travers les paramètres métier, chose montrée dans la dernière partie de ce chapitre, avec l’exemple de l’emboutissage d’une tablette arrière de voiture.

L’implémentation de ces aspects ainsi que les détails techniques sont présentés dans le chapitre suivant.

Chapitre 4

Mise en œuvre informatique d'un modeleur de surfaces discrètes

4.1 Introduction

Nous avons étudié, dans les chapitres précédents, les éléments nécessaires à la définition d'un modeleur de surfaces discrètes dans le domaine de la CAO. La mise en œuvre de ce modeleur étend les travaux présentés dans le deuxième chapitre avec un passage d'un modèle de déformation en deux dimensions à un modèle de déformation en trois dimensions. Cette approche se base sur une proposition, dans le troisième chapitre, d'outils de déformation 3D orientés métier.

Ce quatrième et dernier chapitre se propose de valider de manière pratique le modèle et les outils de déformation proposés, à travers l'implémentation des concepts présentés dans un modeleur de surfaces discrètes pour la CAO.

Les détails d'implémentation sont étudiés dans la section 4.3, après avoir défini le contexte général, les choix techniques et les limites simplificatrices de cette maquette dans la section 4.2. L'interface utilisateur sous-jacente au modèle de déformation, sa structure, le module de visualisation et quelques exemples sont décrits dans la section 4.4. Ce chapitre se termine avec deux applications réelles, présentées dans la section 4.5.

4.2 Contexte

Nous introduisons, avant de présenter la mise en œuvre des déformations 3D, le contexte d'implémentation ainsi que les choix techniques qui en dérivent et les limites de notre maquette.

4.2.1 DIJA

Les travaux de recherche présentés dans ce manuscrit s'inscrivent dans un projet fédérateur, le projet DIJA, qui se construit autour de plusieurs thèmes de recherche, tel que l'IHM [Dan02], l'architecture réseau et l'historique de conception [Den03], la définition d'une base de connaissances orientée métier [GG03], la sécurisation des échanges de données géométriques CAO [MMG04], la modélisation du processus de conception dans un contexte de travail collaboratif [Gar05], l'auto-adaptation des maillages [Hes05] ou l'optimisation des liens entre la conception et la simulation [Rei09]. D'autres travaux de recherche, concernant le passage de la forme à la fonction [BGP06, BGP07] et la modélisation du processus dans le contexte d'une entreprise étendue [BDGY08], sont en cours sous DIJA.

Le projet DIJA se base sur une philosophie de modélisation 'synthétique' qui est à la base d'une approche de conception intuitive orientée métier, qui permet à tout utilisateur de réaliser ses pièces sans avoir une formation technique préalable (voir section 2.2.1). L'implémentation des concepts qui sont à la base de cette approche nécessite, à un niveau matériel, une série des choix techniques, choix qui sont présentés dans la section suivante.

4.2.2 Choix techniques

Les choix techniques concernant le projet DIJA portent sur l'implémentation et la visualisation du modèle de déformation et découlent des spécificités du système DIJA.

Ainsi, le fait que DIJA soit un système de CAO distribué sur Internet, avec une architecture repartie et un niveau élevé de sécurisation du transfert de données, fait que le langage de programmation adopté en unanimité par notre équipe est le langage Java [Jav]. Outre les avantages d'être orienté objet et gratuit, le langage Java est indépendant de la plate-forme et s'adapte bien aux applications réseaux. Cela permet de prendre en compte les contraintes matérielles de la machine client et le fait que l'utilisateur à distance peut être un utilisateur expérimenté, aussi bien qu'un utilisateur novice. De plus, Java donne la possibilité d'intégrer des bibliothèques, tels que Java3D [J3D] pour la visualisation des scènes 3D, JDBC [JDB] pour l'accès aux bases de données, SAX [SAX] et DOM [DOM04] pour l'utilisation du format XML, etc.

Pour la partie visualisation nous avons choisi la librairie VTK [VTK], un outil d'infographie actuellement utilisé dans les domaines du graphisme, imagerie et visualisation. Il permet le rendu d'objets en 3D et la visualisation des simulations en temps réel. VTK utilise des structures de données et d'algorithmes qui s'adaptent à notre modèle de déformation. Cette bibliothèque est

présentée dans la section 4.4.2.

Les données manipulables par les algorithmes de déformation implémentés et par le module de visualisation sont des données de nature discrète (points 3D, arêtes, triangles). Elles sont soit importées à partir des fichiers STL, soit générées par d'autres algorithmes.

Les périphériques d'entrée se restreignent à la souris et au clavier.

En s'appuyant sur ces choix nous proposons par la suite l'extension d'un module du système DIJA, chargé de la déformation de surfaces discrètes 3D. L'objectif est de valider les concepts introduits dans les chapitres précédents et non pas de créer un système de CAO complet. Nous imposons ainsi certaines limites d'implémentation, qui nous permettent de nous concentrer sur les aspects les plus importants de cette étude. Ces limites sont présentées dans la section suivante.

4.2.3 Limites du modelleur

La maquette que nous avons créée met l'accent sur la validation du modèle de déformation 3D présenté en troisième chapitre de cette étude. Ainsi, nous nous concentrons sur l'intégration des règles métier avec des outils de déformation, qui démontrent la validité de l'approche proposée. Les règles et les termes métier sont réducteurs, les outils sont basiques, mais intuitifs et ils permettent d'étudier le déroulement d'un processus de modélisation par déformation.

L'interface graphique a été simplifiée pour des raisons de clarté et d'intuitivité. Les menus sont simples et orientés métier. La visualisation est épurée, avec de possibilités d'ajout d'éléments visuels en fonction des besoins de conception de l'utilisateur. Celui-ci a la possibilité de concevoir ses objets soit à un niveau purement géométrique, s'il le désire, soit à un niveau sémantique plus élevé. Deux interfaces adaptées à des métiers spécifiques sont détaillées dans la section 4.5.

L'aspect intuitif et fonctionnel des outils proposés étant ainsi démontrés, le modelleur peut être étendu par la suite.

Les détails liés à l'implémentation du modèle et des outils de déformation sont décrits dans la section suivante.

4.3 Aspects liés à l'implémentation

Cette section correspond aux détails de représentation interne du modèle de déformation étudié dans les chapitres précédents. Nous présentons dans un premier temps les structures de données correspondant au modèle de représentation. Nous continuons ensuite avec l'étude des structures sous-jacentes aux déformations, en finissant par les algorithmes plus importants qui

les implémentent.

4.3.1 Modèle interne de représentation

Au niveau algorithmique, la représentation géométrique du modèle se réduit, à partir d'un modèle générique quelconque, à une structure de données discrètes qui correspond à un maillage polygonal triangulaire. Les structures de données manipulées par notre modeleur s'appuient sur le modèle B-Rep étendu proposé dans [Dan02] et présenté dans le deuxième chapitre de ce manuscrit et dont le diagramme de classes est illustré dans la figure 4.1.

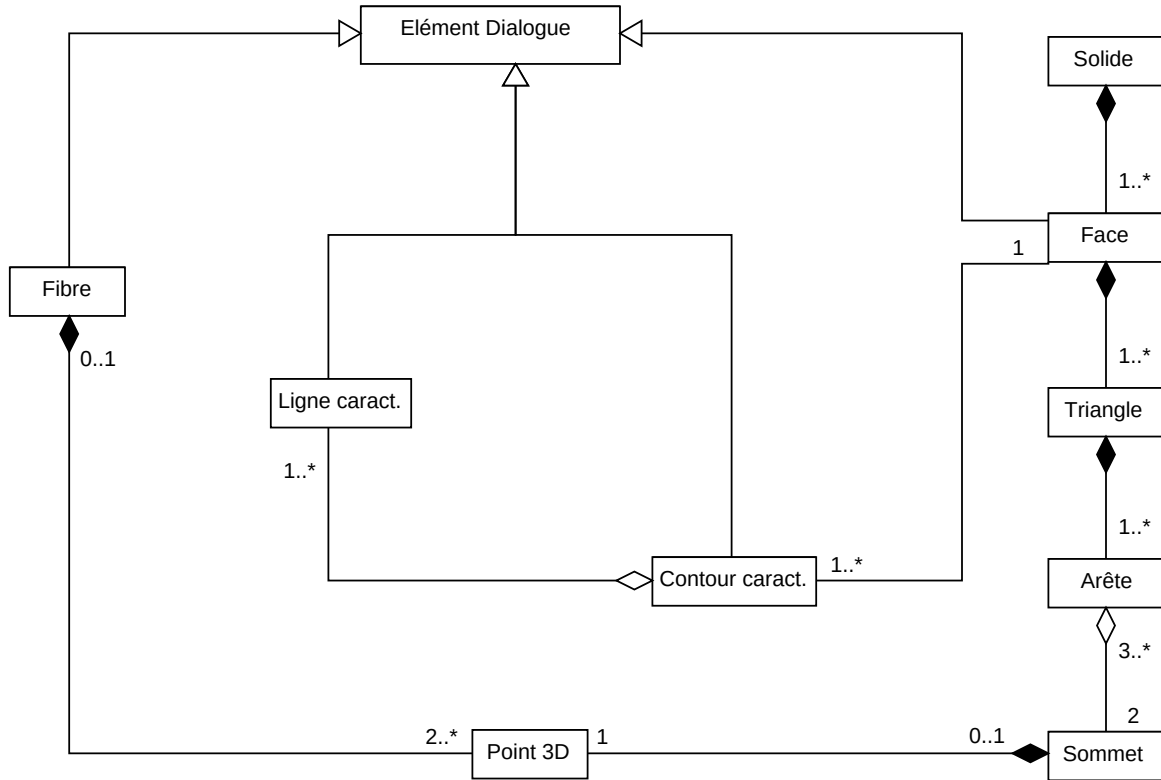


FIG. 4.1: Diagramme de classes UML simplifié du modèle B-Rep étendu [Dan02].

Trois éléments ont été ajoutés au modèle B-Rep classique, qui correspondent aux éléments de dialogue du modèle de représentation DIJA : la fibre, la ligne et le contour caractéristiques, et la face. Ce modèle est fonctionnel depuis le début dans notre système, le seul concept restant à implémenter étant le concept de face et les algorithmes qui lui correspondent.

La représentation d'une surface/solide par faces est implémentée par des structures qui ressemblent au modèle proposé par Danesi, mais qui sont adaptées aux algorithmes de déformation

qui s'appliquent de manière discrète sur les points d'une surface. Ainsi, une surface est représentée par une structure de type « Maillage » (voir le diagramme de classes dans Fig. 4.2) qui contient tous les points 3D, les arêtes et les triangles qui la composent. Le maillage est représenté géométriquement par la totalité des points 3D de la surface qui sont réunis dans la structure « PolygoneContrôle ». La connectivité du maillage est définie par l'ensemble des faces triangulaires, mais également par l'ensemble des arêtes qui le composent. Cette double représentation est nécessaire pour faciliter l'accès et les différents traitements des données pendant l'exécution des algorithmes de déformation et de rendu. Ces structures sont construites à partir d'un fichier STL.

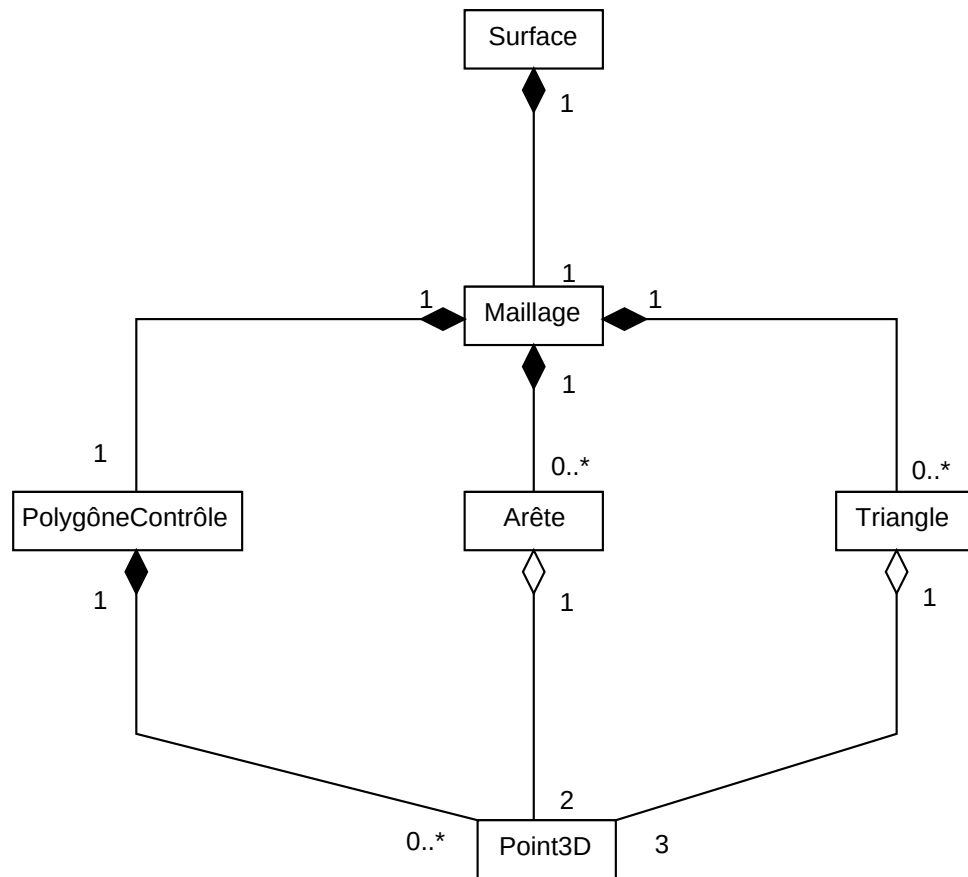


FIG. 4.2: Diagramme de classes UML relative à la structure de données « Surface ».

A cette structure s'ajoute la structure « Zone » qui représente localement la surface à déformer, avec les sommets sous-jacents, et dont le diagramme de classes UML est illustré dans la figure 4.3. Un sommet est une structure qui, en plus des coordonnées 3D du point qu'elle représente, englobe aussi des informations relatives à la déformation (comme la liste des voisins,

la distance jusqu'au point initial de déformation (qui donne la quantification de la déformation dans ce point), si le point 3D est borné ou pas, ancré ou pas, etc.).

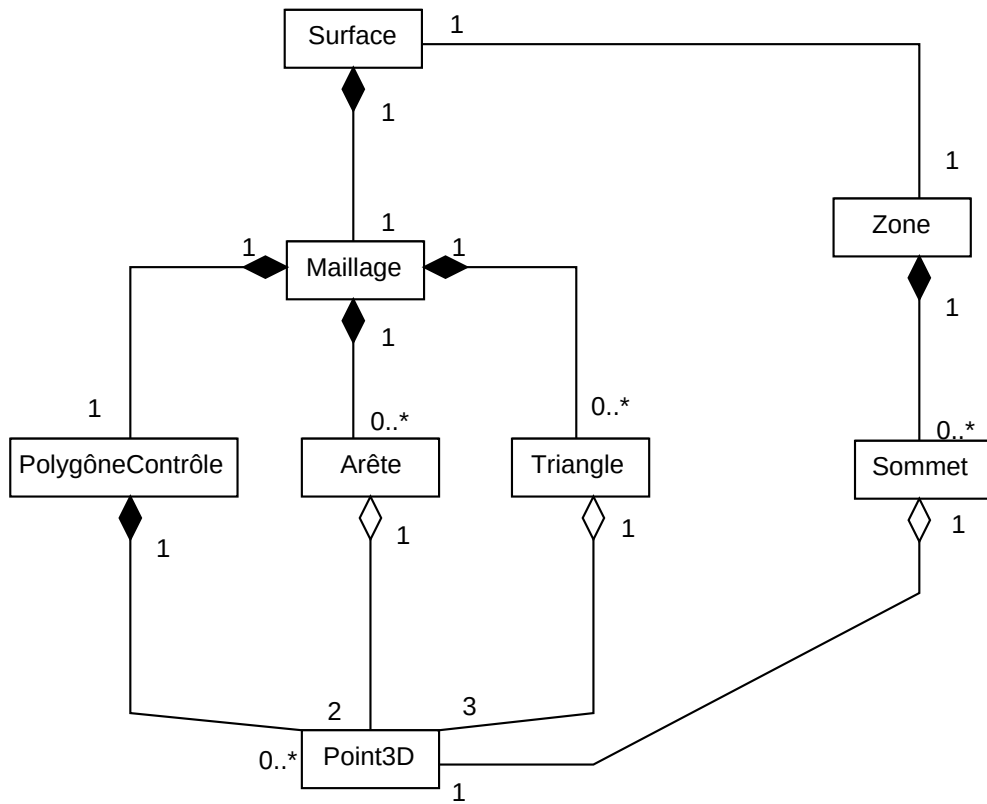


FIG. 4.3: Diagramme de classes UML relative aux structures de données « Surface » et « Zone ».

Ces entités, avec les structures qui correspondent aux algorithmes de déformation, sont illustrées dans la section suivante.

4.3.2 Représentation interne des outils de déformation

La figure 4.4 illustre le diagramme de classes UML représentant les classes utilisées pour l'implémentation des déformations. Une déformation peut s'appliquer à une zone de la surface de l'objet modélisé, à travers un algorithme spécifique qui prend en entrée des paramètres caractéristiques à un certain type de déformation (style, portée, etc.) et des contraintes éventuelles (contrainte volumique, distance, hauteur, etc.). Les caractéristiques métier, ainsi que les paramètres dépendent du style choisi et du type de déformation appliquée. Chaque algorithme de déformation donne lieu à la création d'une interface graphique spécifique, qui propose à l'utili-

sateur des outils adaptés à un certain type de déformation.

La déformation prend en charge le calcul des points affectés par la déformation en fonction du point initial et de la portée/contour de la déformation. Ces points se regroupent dans un objet de type « Zone ». L'algorithme de déformation s'applique, en fonction des choix de l'utilisateur, à cette zone de déformation, le résultat se répercutant à la fin sur la surface de l'objet.

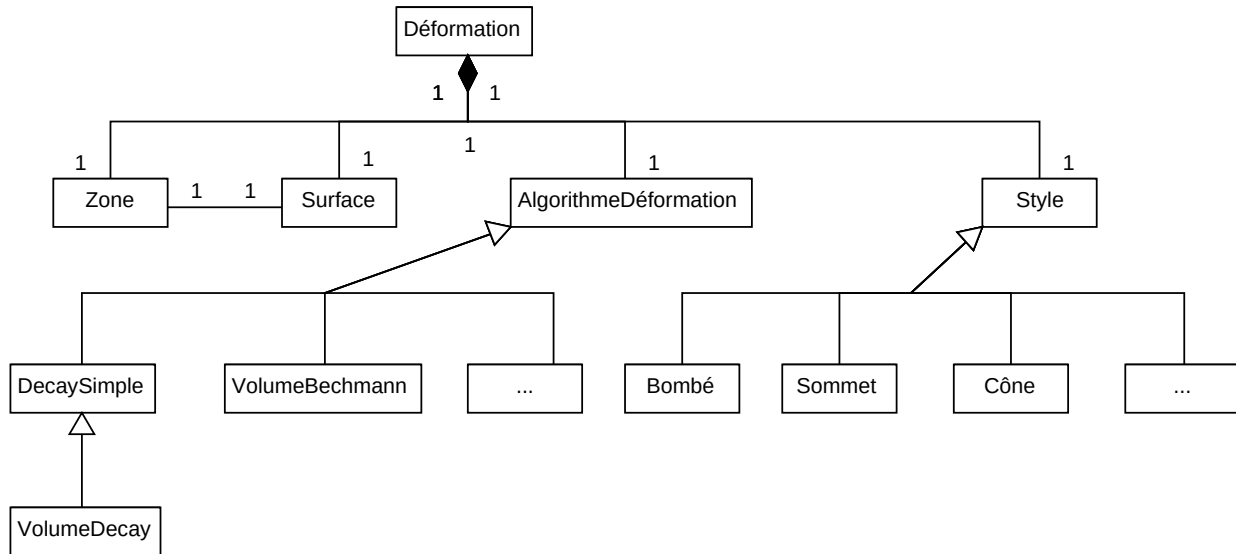


FIG. 4.4: Diagramme de classes UML relative aux déformations.

L'algorithme de déformation *decay* simple, par exemple, a besoin seulement de connaître les points de la zone à déformer dont il calcule le déplacement. Ce n'est pas le cas avec les algorithmes de déformation avec contrainte volumique, qui ont accès à tous les points de la surface, afin de pouvoir effectuer la correction volumique sur les zones non-déformables de l'objet modélisé.

Les différents algorithmes de déformation sont présentés dans la section suivante.

4.3.3 Implémentation de déformations

Nous détaillons par la suite, d'un point de vue algorithmique, les éléments relatifs à la déformation : la forme et l'action. La forme de la déformation, qui résulte du style de déformation choisi par l'utilisateur à un niveau sémantique plus élevé, est implémentée à l'aide de fonctions mathématiques, comme montré dans le troisième chapitre de cette étude (voir section 3.2). Algorithmiquement, chaque style de déformation est englobé dans une classe particulière, qui hérite d'une classe mère, appelée « Style ». La figure 4.5 illustre la diagramme de classes UML qui correspond au style de la déformation.

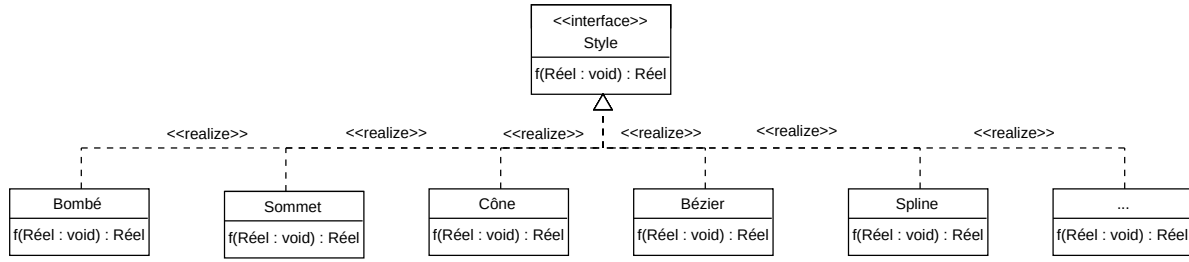


FIG. 4.5: Diagramme de classes UML relative au style de déformation.

Le style de déformation est paramétré par l'utilisateur et instancié avant de lancer l'opération de déformation. L'objet de type « Style » est utilisé ensuite à l'intérieur de la classe « AlgorithmeDéformation » pour calculer le déplacement des points à l'intérieur de la zone de déformation. La fonction *decay* englobée par le style de déformation est appliquée sur les coordonnées x, y, z d'un point 3D $P(x, y, z)$ en fonction de la distance qui sépare le point 3D P du point initial de déplacement $PI(x_i, y_i, z_i)$:

$$valFDecay = style.f(dist);$$

$$P.x = P.x + \Delta x * valFDecay;$$

$$P.y = P.y + \Delta y * valFDecay;$$

$$P.z = P.z + \Delta z * valFDecay;$$

où *style* est l'objet de type « Style », *dist* est la distance - euclidienne ou en termes d'arêtes-entre P et PI , et $\Delta x, \Delta y, \Delta z$ sont les déplacements en x, y et z du point PI .

Une première action d'un outil de déformation sur la surface est celle de définir la zone à déformer. Ceci se réalise à partir des choix que l'utilisateur fait au niveau IHM, par une routine englobée dans l'algorithme de déformation, illustrée dans l'Algorithme 7. Plusieurs routines existent dans notre système pour la définition de la zone à déformer (voir Fig. 4.6) :

- définir la zone à partir d'un point initial et une valeur pour le rayon d'influence (qui désigne la distance euclidienne ou la distance en termes d'arêtes entre le point initial et le point cible de déplacement) ;
- définir la zone à partir d'un contour préexistant dans la base de données ;
- définir la zone à partir d'une esquisse à main levée.

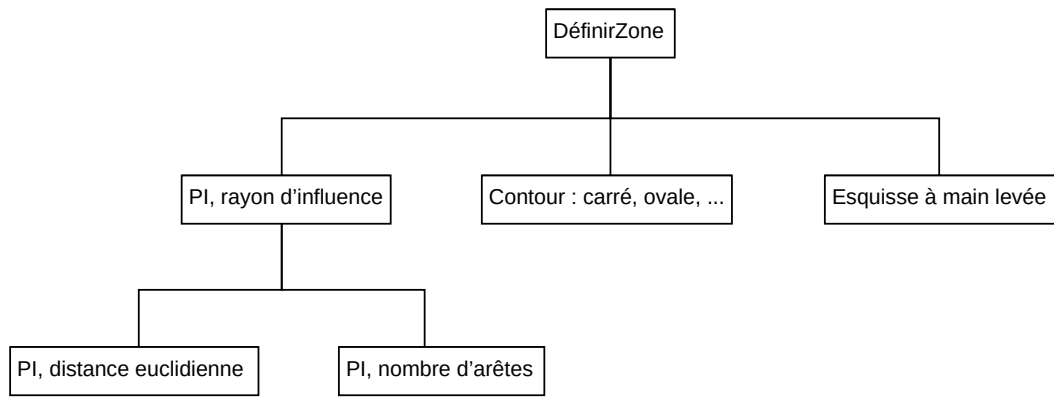


FIG. 4.6: Schéma des routines de définition de la zone de déformation.

Algorithme 7 La procédure qui définit la zone de déformation en fonction du choix de l'utilisateur

```

1: procédure DefinirZone(racine, rayon)
2: Entrée : racine, rayon représentant le point initial de la déformation et le rayon d'influence
3: Sortie : les points de la zone de déformation
4: Départ
5: Si type_zone = « rayon d'influence » Alors
    Si portee = « nombre d'arêtes » Alors
        definirZoneDistArretes(racine, rayon)
    Sinon
        definirZoneDistEuclidienne(racine, rayon)
    FSi
6: SinonSi type_zone = « contour » Alors
    Si contour = « carre » Alors
        definirZoneContourCarre(racine, rayon)
    SinonSi contour = « ovale » Alors
        definirZoneContourOvale(racine, rayon)
    SinonSi ... Alors
        ...
    FSi
7: SinonSi type_zone = « contour esquisse » alors
    definirZoneContourEsquisse(racine, rayon)
8: FSi
9: Fin procédure
  
```

Les points d'une zone de déformation définie par un point initial et un rayon d'influence sont ajoutés à la structure de type « Zone » par un algorithme qui parcourt la liste des sommets niveau par niveau à partir du point initial. L'algorithme, dont le pseudo code est détaillé dans l'Algorithme 8, se base sur une structure de type file. Pour que cet algorithme soit applicable à notre structure de données, un prétraitement est nécessaire afin d'identifier les sommets voisins

de chaque sommet dans le polygone de contrôle.

Algorithme 8 La procédure qui parcourt la liste des sommets par niveau à partir du point initial (racine) et qui calcule la distance entre chaque point et la racine

```

1: procédure ParcoursParNiveau(racine, rayon)
2: Entrée : racine, rayon, représentant le point initial de la déformation et le rayon d'influence
3: Sortie : la liste des sommets dans la zone de déformation
4: Départ
5: distance(racine, racine) = 0
6: quantificateur_de_la_deformation_pour_racine = 1 //la quantification de la déforma-
   tion est maximale (1) pour la racine
7: Créer queue vide q
8: q = q ∪ {racine}
9: sommet = racine
10: TantQue (q ≠ ∅) et (distance(premier_element_de_q, racine) ≤ rayon) and (premier
    élément de q n'est pas ancre)
    sommet = premier_element_de_q
    q = q − {premier_element_de_q}
    Si (sommet ∉ zone_deformable) Alors
        zone_deformable = zone_deformable ∪ {sommet}
    FSi
    Si (distance(sommet, racine) < rayon) Alors
        Pour tous les sommets voisins de sommet Faire
            Si (voisin ∉ zone_deformable) Alors
                zone_deformable = zone_deformable ∪ {voisin}
                distance(voisin, racine) = distance(sommet, racine) + 1
                quantificateur_de_la_deformation_pour_voisin =
                    1 − distance(voisin, racine)/r
                q = q ∪ {voisin}
            FSi
        FSi
11: FTQ
12: Fin procédure

```

Pour les zones de déformation définies par un contour préexistant ou par un contour esquissé à main levée, le même algorithme est appliqué, dès que le point initial de déformation est connu, pour calculer les distances séparant chaque point du point initial.

Le comportement de l'outil de déformation est induit par l'algorithme qui effectue la déformation. L'outil le plus simple de déformation 3D implémenté dans notre maquette induit des déformations localement sur la surface de l'objet modélisé. Ceci se réalise par l'algorithme de déformation à l'aide des fonctions *decay* (algorithme « DecaySimple » dans la figure 4.4) suivant :

Algorithme 9 La procédure qui réalise le déplacement des points de la zone déformable selon une fonction *decay* et selon la distance qui les séparent du point initial de déplacement (*racine*)

```

1: procédure DecaySimple(racine, rayon)
2: Entrée : racine, rayon, représentant le point initial de la déformation et le rayon d'influence
3: Sortie : la surface après déformation
4: Départ
5: Créer queue vide q
6:  $q = q \cup \{racine\}$ 
7: sommet = racine
8: TantQue ( $q \neq \emptyset$ ) et ( $distance(premier\_element\_de\_q, racine) \leq rayon$ ) and (premier
   élément de q n'est pas ancre)
   sommet = premier_element_de_q
   zone_deformable = zone_deformable - {sommet}
    $q = q - \{premier\_element\_de\_q\}$ 
   calculer la nouvelle position du sommet courant
   changer la position du sommet courant
   Pour tous les sommets voisins de sommet Faire
     Si (voisin ∈ zone_deformable) Alors
        $q = q \cup \{voisin\}$ 
       zone_deformable = zone_deformable - {voisin}
     FSi
   FPour
9: FTQ
10: Fin procédure

```

D'autres fonctionnalités peuvent s'ajouter à ce type d'outil, correspondant aux paramètres et aux routines algorithmiques additionnels. Ainsi, les algorithmes « VolumeBechmann » et « VolumeDecay » imposent une contrainte volumique à travers l'implémentation d'une routine de correction volumique présentée dans le troisième chapitre de cette étude. L'algorithme qui correspond à l'outil de déformation de la tablette arrière de voiture intègre des paramètres relatifs à la hauteur, au volume, et au nombre des plumiers insérés, ainsi qu'une routine pour programmer le comportement de l'outil dans le cas où ces paramètres ne sont pas respectés. D'autres paramètres et contraintes métier peuvent être intégrés, de manière automatique, en s'ajoutant au squelette de déformation représenté par l'algorithme de déformation « DecaySimple », décrit dans l'Algorithme 9.

Chacun de ces algorithmes a une interface graphique associée, à travers laquelle l'utilisateur peut paramétrer la déformation de son objet selon les spécificités de son métier. L'interface graphique utilisateur, la structure sous-jacente et le module de visualisation sont présentés dans la section suivante.

4.4 Interface graphique

DIJA propose une interface graphique adaptée à tout type d'utilisateur, grâce aux outils dédiés, en se basant sur une approche métier. Elle donne lieu à un environnement de travail intuitif, qui permet à l'utilisateur de s'immerger dans son espace métier habituel et de modéliser ses objets en exprimant directement son savoir-faire. Ceci se réalise d'une part à l'aide d'une IHM de CAO innovante dont la structure est expliquée dans le paragraphe suivant et d'une autre part à travers un module de visualisation 3D conçu à l'aide de la librairie VTK. Ainsi, l'utilisateur a la possibilité de faire ses choix grâce à une interface intuitive et adaptée et d'analyser en même temps les résultats de manière visuelle.

4.4.1 Structure

L'interface graphique du modéleur renforce l'intuitivité à travers la simplicité. Sa structure, présentée dans la figure 4.7, comprend un menu simplifié et une partie de visualisation. Nous nous concentrons par la suite sur la partie menus/interfaces métier, la partie visualisation étant expliquée dans les paragraphes suivants. La barre de menu de l'interface graphique de l'application se compose de plusieurs sous-menus :

- le sous-menu *Fichier* qui permet l'ouverture d'un fichier STL contenant la forme de départ, l'enregistrement des nouvelles formes en format STL et de quitter l'application ;
- le sous-menu *Visualisation* qui offre plusieurs options de présentation ;
- le sous-menu *Déformation* qui propose plusieurs outils de déformation pour la modélisation de l'objet en cours de conception ;
- le sous-menu *Picking* à travers lequel l'utilisateur a la possibilité de faire un picking à l'aide de la souris du/des point(s) de déplacement ou pour tracer le contour de déformation par esquisse ;
- le sous-menu *Subdivision* qui propose deux outils de subdivision de la surface de l'objet en cours de modélisation ;
- le sous-menu *Image* qui permet l'enregistrement en format eps de l'objet visualisé dans la partie de visualisation.

Le sous-menu *Déformation* donne lieu à des interfaces adaptées au métier choisi qui permettent de paramétrer la déformation selon de contraintes métier spécifiques.

La plus grande partie de l'interface est réservée à la visualisation de l'objet en cours de modélisation. La figure 4.7 représente une forme de départ pour un récipient de liquide.

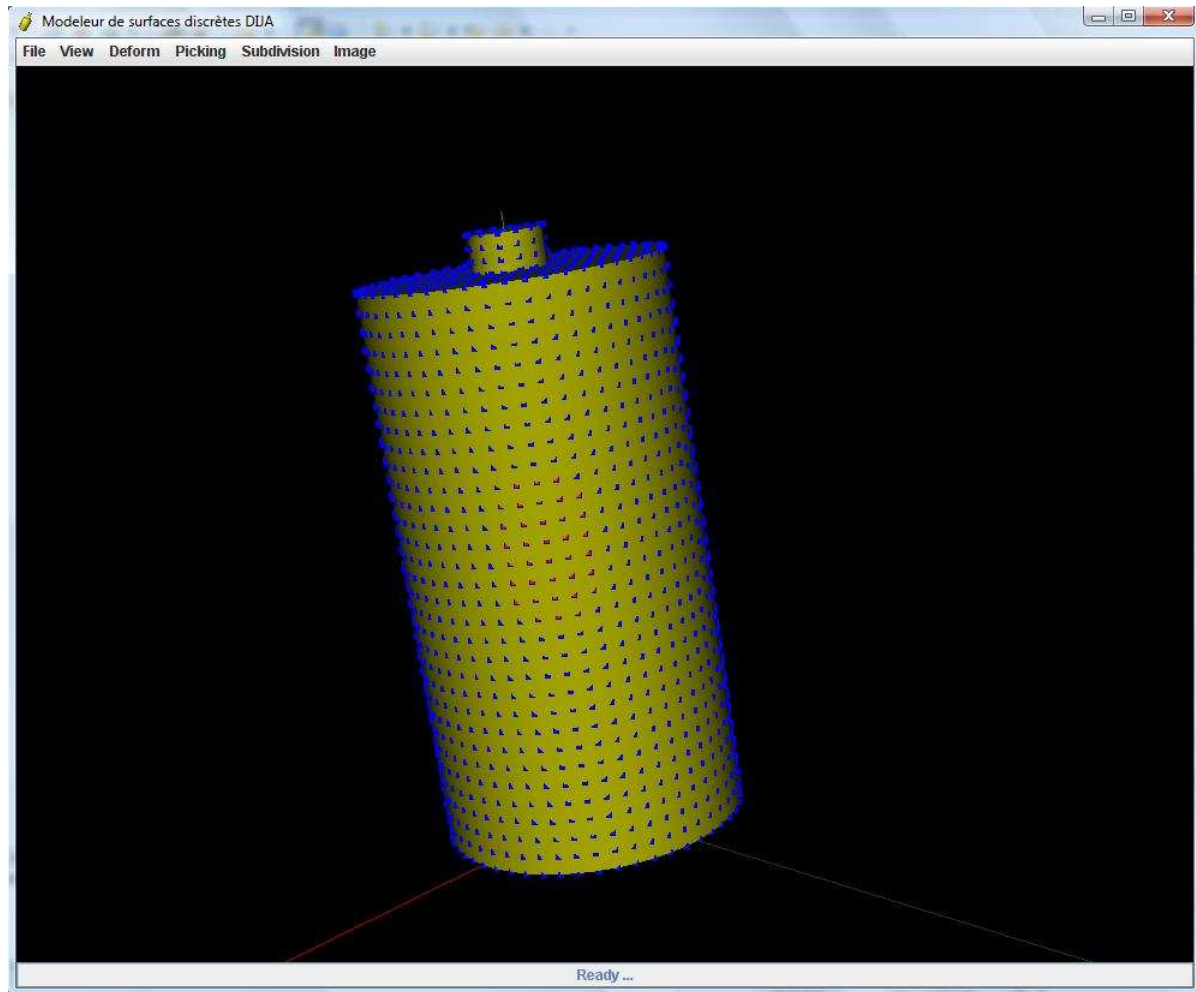


FIG. 4.7: Interface graphique du modeleur.

La surface est affichée en jaune translucide, les points de la surface en bleu et les points de la zone de déformation (l'élément de dialogue de type *face*) en rouge. L'objet affiché est manipulé dans la zone de visualisation à l'aide de la souris. La visualisation 3D avec VTK est présentée dans le paragraphe suivant.

4.4.2 Introduction à VTK

Le système de visualisation VTK [ABB⁺06, SML06] est un logiciel orienté objet dédié à la visualisation scientifique, au graphisme et au traitement de l'image. La librairie VTK est gratuite et se veut indépendante du système d'exploitation et du langage. VTK est adapté aux structures de données et aux algorithmes basés sur les scalaires, les vecteurs, la texture, le volume, et sur des techniques de modélisation avancées, tel que la modélisation implicite, la réduction des polygones, le lissage du maillage, la coupe (*cutting* en anglais), le contour et la triangulation Delaunay. La visualisation et la manipulation des données sont implémentés par des algorithmes de pointe issus de la recherche scientifique [BCE⁺92, NS90, ABB⁺06, SML06, WT91, Wol90].

VTK utilise une approche basée sur un flux de données pour transformer les primitives géométriques en primitives graphiques rendant possible la visualisation. Ceci implique deux types d'objets :

- **les données**, représentant la source du pipeline graphique ; elles sont associées à des attributs tel que scalaires ou vecteurs ;
- **les algorithmes**, représentant les filtres, le traitement du pipeline graphique, qui opèrent sur les données et qui produisent à leur tour des données.

Les algorithmes, avec les données forment des pipelines graphiques dont la structure est montrée dans la figure 4.8, qui réalisent la transformation des primitives géométriques en primitives graphiques et leur rendu.

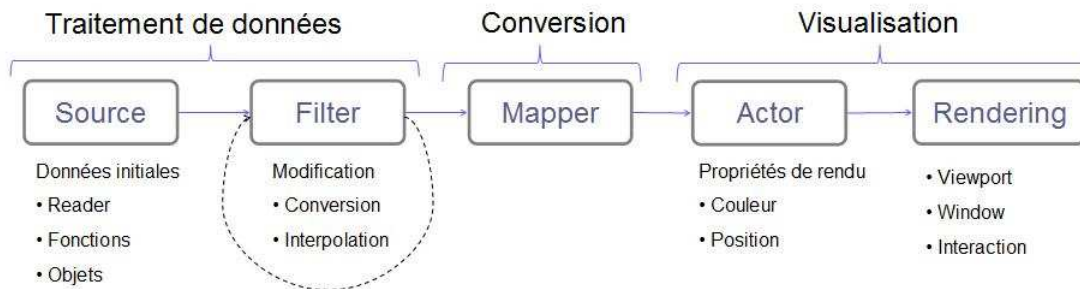


FIG. 4.8: Le pipeline de visualisation VTK. Les arêtes indiquent le flux de données.

Les données, représentées soit par une géométrie polygonale, soit par une image (2D, 3D, *grid*). Leur structure se base sur le principe de *cellule* (voir Fig. 4.9). Les données VTK sont requises via :

- une lecture de fichier (à l’aide des objets de type *Reader* : *vtkSTLReader* pour la lecture d’un fichier en format STL, etc.);
- une construction manuelle (méthode avancée);
- des objets de type *Source* ou *FilterOutput* qui produisent de données.

Des données additionnelles, tels que les couleurs, les vecteurs, les scalaires, peuvent être associées aux cellules VTK.

Les données VTK sont soumises à des traitements à l’aide de filtres (*Filter* en anglais), et qui correspondent à des algorithmes avancés. Le traitement applique des opérations telles que les transformations, le découpage, l’extraction (caractéristiques, VOI, etc.), le contour sur les données VTK. Des exemples des résultats des traitements VTK sont illustrés dans la figure 4.10. Un filtre prend en entrée une ou plusieurs données et produit en sortie un ou plusieurs objets (données).

Dans le pipeline de visualisation, l’entrée d’un filtre représente souvent la sortie d’un autre filtre. L’assemblage du pipeline impose donc que les objets qui le composent soient compatibles. L’exécution du pipeline graphique sous VTK se fait selon un mécanisme qui exécute seulement les portions du pipeline nécessaires à l’actualisation du résultat. Ainsi, une modification en amont du pipeline nécessite la réévaluation du pipeline pour que le résultat soit effectif. L’exécution du pipeline a lieu seulement quand les données sont requises.

Le pipeline graphique du système logiciel VTK représente, pour DIJA, la liaison entre la géométrie et la visualisation du modèle. Il s’adapte aux structures de données employées au niveau géométrique sous DIJA et permet leur rendu, géré par le module de visualisation. Les données sous DIJA sont données discrètes (cellules simplifiées : triangles, arêtes, points), représentées à l’aide des points de $\mathbb{R}^3$ qui forment le maillage de l’objet modélisé, récupéré d’un fichier STL. Les points du maillage sont utilisés pour le traitement algorithmique afin de réaliser les déformations. VTK gère aussi l’interaction Homme-Machine au niveau de la visualisation grâce aux techniques de *picking*. Ces aspects sont détaillés dans le paragraphe suivant, à travers la description du module de visualisation DIJA.

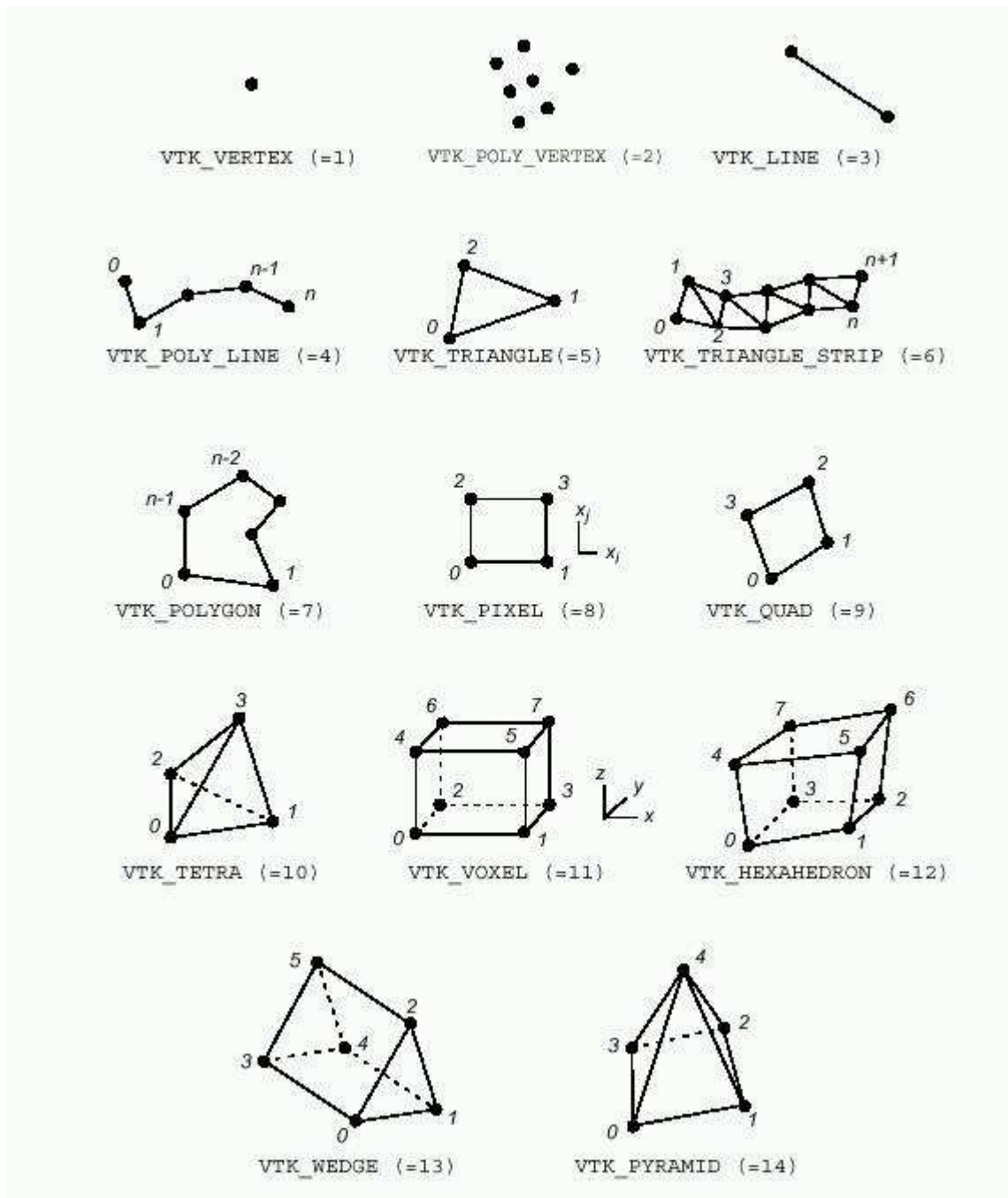
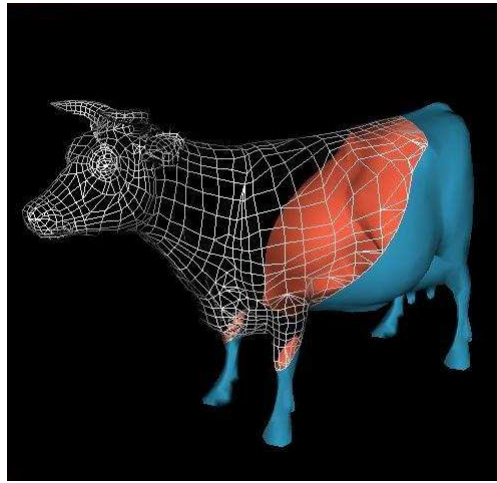
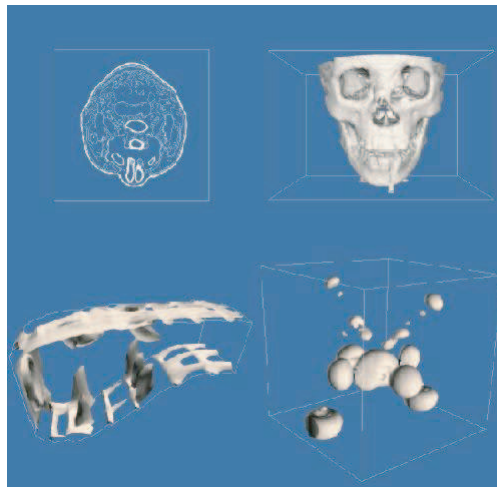


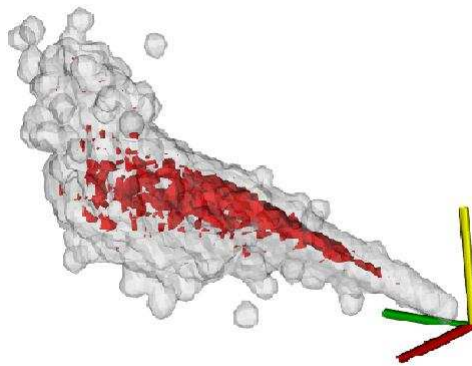
FIG. 4.9: Types de cellules VTK [SML06]. Les nombres désignent l'ordre des points de définition.



(a)



(b)



(c)

FIG. 4.10: Opérations de traitement VTK : (a) Découpage (*clipping* en anglais) [eIPX] ;(b) Contour (*contouring* en anglais) [Joh] ; (c) Extraction (*extract* en anglais) [Hol06].

4.4.3 Module de visualisation

Le module de visualisation DIJA désigne une partie logicielle du système qui joue le rôle d'interface entre le modèle géométrique DIJA et le modèle visuel qui comprend les données nécessaires au rendu dans la scène de visualisation. Le passage d'un modèle à l'autre se fait en utilisant les mêmes primitives géométriques discrètes (points, triangles) à travers les méthodes existantes dans le module de visualisation. Ces méthodes se basent, à leur tour, sur des objets et routines VTK.

Le module de visualisation implémente à la fois le modèle visuel et son comportement. Le modèle visuel se construit à travers le pipeline graphique, à partir des primitives visuelles qui se concrétisent en des objets visuels rendus dans la scène de visualisation. Ces objets sont appelés *acteurs* et possèdent des informations géométriques ainsi que des informations visuelles (tels que la texture, la couleur, la transparence, etc.). La figure 4.11 montre le rendu d'un acteur cube avec différents niveaux d'opacité.

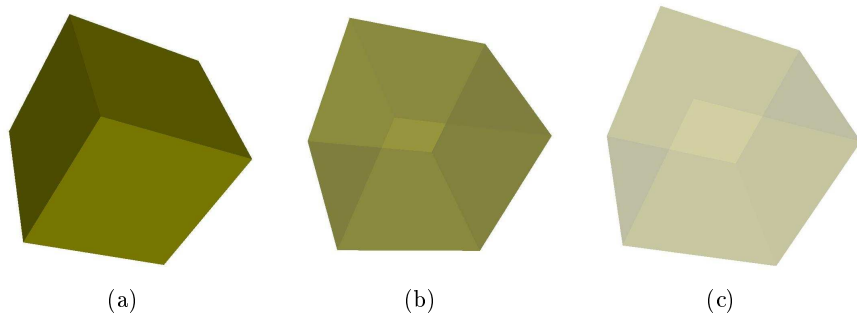


FIG. 4.11: Différents niveaux d'opacité d'un objet : (a) opacité = 1 ; (b) opacité = 0.5 ; (c) opacité = 0.2.

La visualisation des acteurs dans la scène peut s'effectuer de plusieurs façons (voir la figure 4.12) :

- en mode surfacique ;
- en mode filaire ;
- par points.

L'utilisateur a la possibilité d'interagir sur la scène grâce au comportement implémenté dans le modèle visuel. Il peut ainsi naviguer dans la scène (modifier le point de vue), déplacer ou redimensionner les acteurs, choisir le mode d'affichage des acteurs et sélectionner des éléments de la scène par une opération de *picking*.

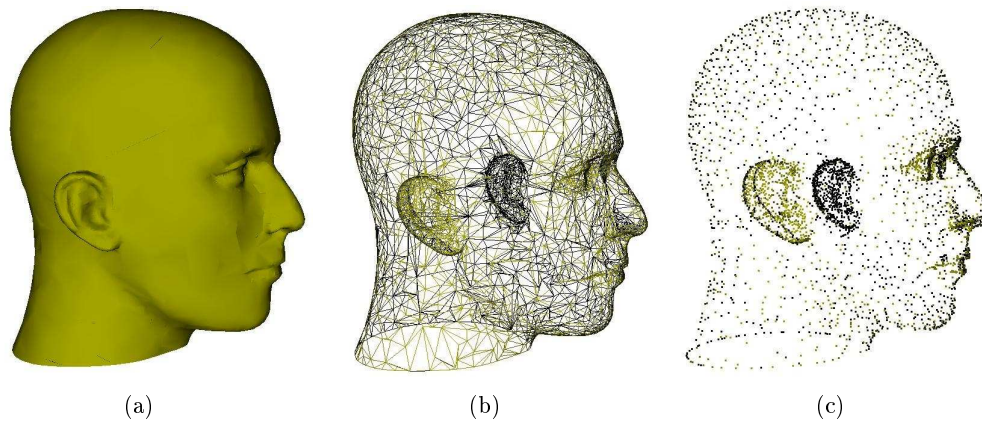


FIG. 4.12: Différents modalités de visualiser un acteur : (a) visualisation en mode surfacique ; (b) visualisation en mode filaire ; (c) visualisation par points.

Si le rendu des acteurs va des primitives géométriques caractérisant l'acteur, à travers des filtres de traitement, jusqu'à des primitives graphiques, le *picking* se réalise par un processus qui va dans le sens inverse. Le système de visualisation récupère, dans ce cas, à partir des primitives présentes dans la fenêtre de rendu (sélectionnées avec à l'aide de la souris), les primitives géométriques de l'acteur (cellules VTK et finalement les primitives géométriques associées).

L'opération de *picking* est utilisé sous DIJA pour sélectionner les points qui forment le contour de déformation, dans le cas d'un contour esquissé, ou pour sélectionner le/les points de déplacement et leurs positions cibles. La figure 4.13 montre la déformation d'une surface plane à l'aide d'une fonction style « bombé ». La zone de déformation est de type circulaire ; elle est déterminée par une opération de *picking* qui donne le point initial de déformation (le centre du cercle) et un point du cercle. La position cible de la déformation est indiquée soit par un *picking* à l'aide de la souris, soit à l'aide de la logique floue (à travers un composant de l'interface graphique ou un terme du vocabulaire DIJA).

L'interaction Homme-Machine mise à la disposition par le module de Visualisation est une alternative à l'approche orientée menus. Elle donne la possibilité d'actionner directement sur la surface de l'objet modélisé pour activer les outils de *picking* ainsi que les outils de visualisation. Les menus sont cependant indispensables dans le cas des déformations qui dépendent des outils plus paramétrés, comme par exemple la création d'un embouti dans une pièce.

Le paragraphe suivant présente, avant de conclure ce chapitre, l'évolution de la GUI pendant la durée de ces travaux de recherche, à partir de la première maquette de déformation de surfaces 3D sous DIJA qui a été ensuite étendue pour créer les GUI correspondantes au design des

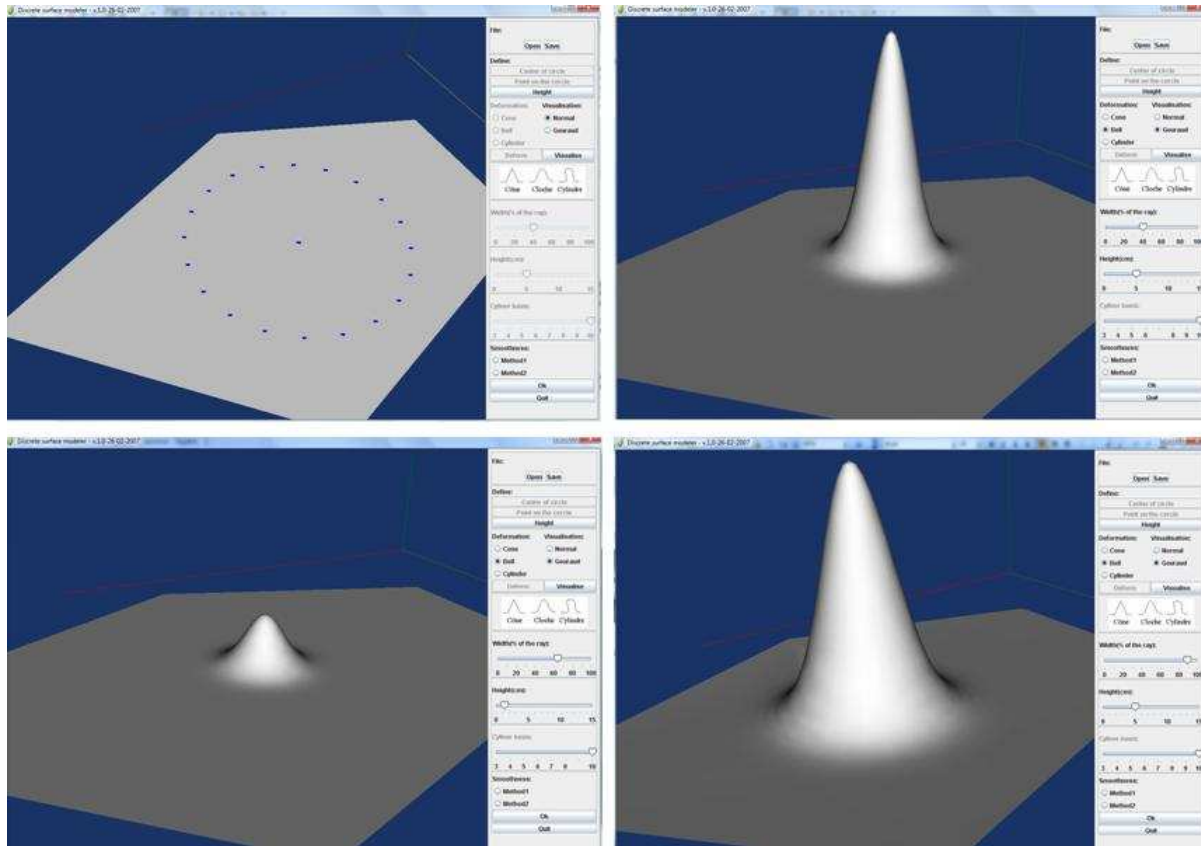


FIG. 4.13: Interaction visuelle pour une opération de sélection dans la fenêtre de rendu pour paramétrer la déformation d'une surface plane.

réipients cosmétiques et aux tablettes arrière de voiture.

4.5 Synthèse de l'implémentation

Algorithmiquement, cette étude a débuté avec la création d'une maquette de déformation des surfaces discrètes planes [CDGP08a], dont l'interface graphique est montrée dans la figure 4.14.

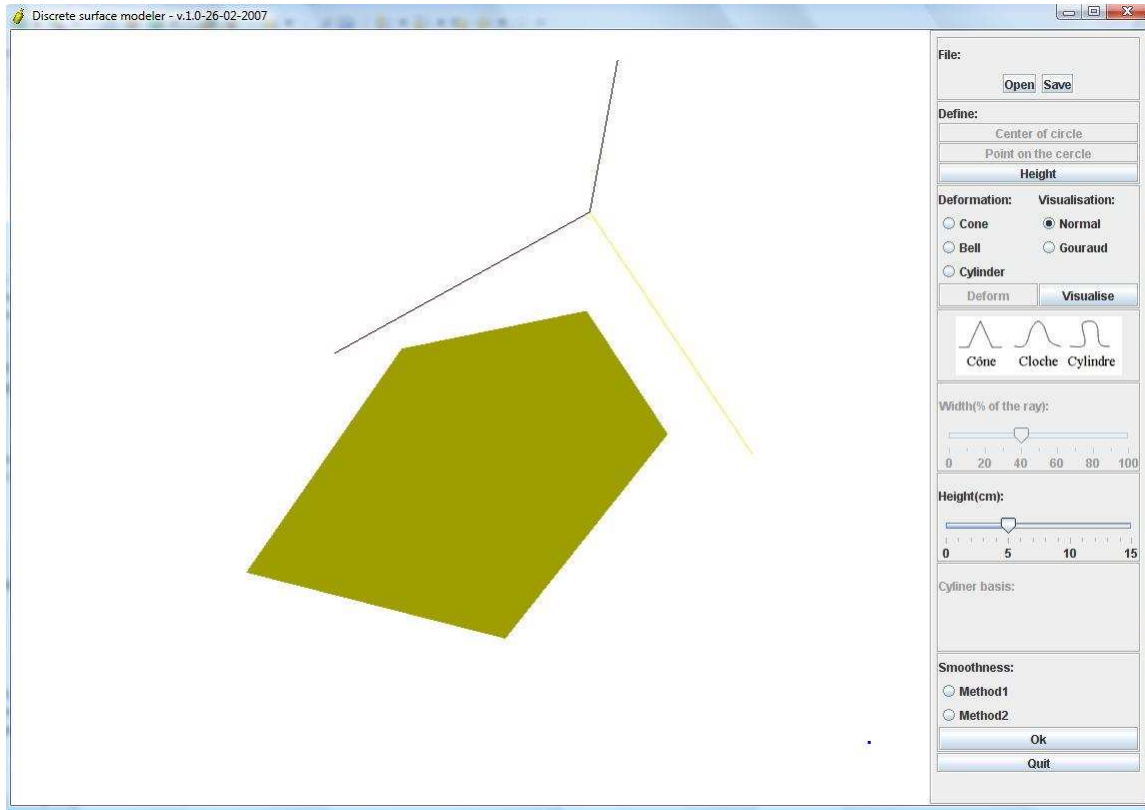


FIG. 4.14: Interface graphique de la première maquette du modeler DIJA de surfaces discrètes.

Le but était de tester de manière simple la déformation d'une partie de la surface en respectant l'approche synthétique DIJA. Trois types de déformation, montrés dans la figure 4.15, ont été proposés dans cette première maquette :

- la déformation de type cône ;
- la déformation de type cloche ;
- et la déformation de type cylindre.

Les trois types de déformation, se basaient sur des fonctions mathématiques tels que l'équation de la droite pour la déformation de type cône, la formule de Gauss pour la déformation de type cloche, et $1/x$ pour la base et l'équation du cercle pour le bout de la déformation de type cylindre.

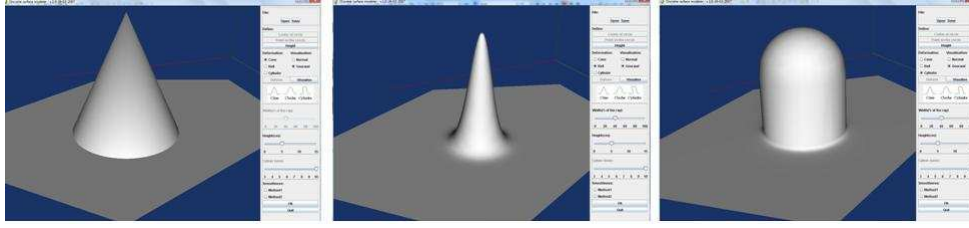


FIG. 4.15: Les premiers types de déformation de surface DIJA.

Cette première maquette nous a permis, non seulement de tester le modèle de déformation DIJA 2D, mais aussi de se familiariser avec les techniques de visualisation et d'interaction VTK.

Les déformations ont été par la suite étendues à des déformations 3D de type *decay* à l'aide des contraintes de déplacement dans une deuxième maquette du modeler DIJA de surfaces. Cette technique permet d'effectuer des déformations sur des objets avec n'importe quelle topologie, à savoir que la forme de la fonction *decay* s'ajoute à la forme de la surface déformée. La figure 4.16 montre différents styles de déformations *decay* appliqués à un objet de base (cylindre).

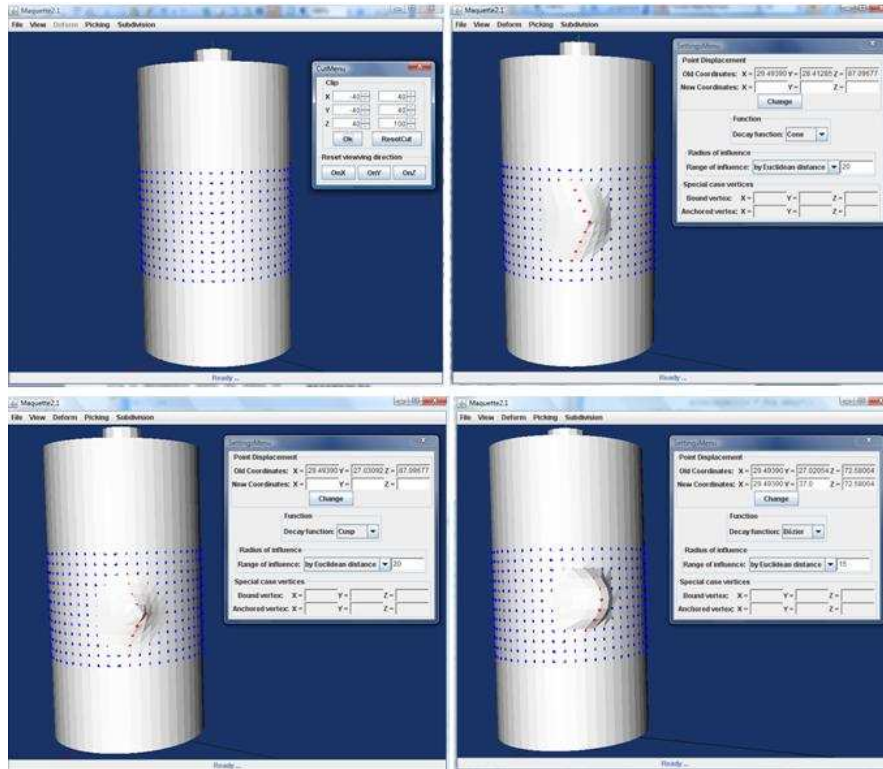


FIG. 4.16: Déformations *decay* appliquées à un cylindre : en haut à gauche : la zone de déformation sur l'objet initial ; en haut à droite : déformation de type « cône » ; en bas à gauche : déformation de type « sommet » ; en bas à droite : déformation se type « bombé » (Bézier) .

En se basant sur cette deuxième maquette nous avons créé une troisième qui implémente l'intégration des règles métier avec les déformations de surfaces. Deux interfaces, correspondant à deux applications ont été créées. Une première qui permet le design des récipients cosmétiques avec l'intégration de la contrainte volumique [CDGP08b], et une deuxième pour le design des tablettes arrière de voiture avec l'intégration des outils plus paramétrés (incluant la contrainte volumique). Les deux dernières applications ont déjà été présentées dans cette étude (dans le troisième et dans le quatrième chapitre).

Comme nous l'avons déjà annoncé au début de ce chapitre, l'approche proposée a été testée sur des éléments simplifiés. Les principes étant validés, le modèleur peut être étendu par la suite par l'intégration d'autres contraintes métier. La base de connaissances pour la déformation de surfaces peut être enrichie, mais sans rien apporter de plus sur la validité de l'approche proposée par cette étude.

4.6 Conclusion

Ce dernier chapitre présente la mise en œuvre informatique des déformations de surface orientées métier DIJA. Les concepts introduits par cette étude sont validés à travers leur implémentation dans un modèleur de surfaces discrètes.

Le contexte technique du modèleur est celui du projet DIJA, un projet fédérateur construit autour de plusieurs thèmes de recherche. Nous avons montré, dans ce contexte, l'implémentation du modèle et des outils de déformation proposés dans le troisième chapitre, ainsi que l'interface graphique et le module de visualisation sous-jacentes. Un accent spécial est mis, dans le développement de la GUI du modèleur, sur la visualisation avec VTK, qui s'avère un outil très prometteur de l'infographie à l'heure actuelle. Le chapitre se termine avec une synthèse de l'implémentation des déformations de surface DIJA qui montre le fil conducteur de cette thèse au niveau implantation/implémentation et qui se construit autour du fil conducteur théorique, concernant les techniques de déformation de surfaces et leur adaptation au contexte DIJA, fil conducteur qui a été présenté dans les chapitres précédents.

Le modèleur de déformation de surfaces discrètes, même si développé à un niveau de maquette, prouve la supériorité de l'approche de modélisation synthétique DIJA par rapport à l'approche constructive classique. Les bases étant validées, le modèleur peut être étendu à des applications plus évoluées, adaptées aux différents métiers.

L'implémentation des déformations de surface orientées métier renforce les aspects fonctionnel

et intuitif du système de CAO DIJA et se traduit, au niveau de la chaîne de production, en gains en termes de temps et de coût. À un niveau recherche, le modéleur prouve la validité des techniques de déformation proposées, ainsi que l'amélioration de l'interaction Homme-Machine à travers des outils dédiés.

Conclusion et Perspectives

En conclusion de ce manuscrit, nous présentons le bilan des travaux sous-jacents. Ceci nous permet de souligner la contribution majeure de ce travail de recherche, qui consiste dans une approche orientée métier pour la déformation des surfaces discrètes dans un système de CAO. Nous terminons en suggérant quelques pistes de recherche que cette étude ouvre.

Le premier chapitre réalise un état de l’art des techniques d’interaction basées sur un modèle de déformation, qui permet d’isoler les éléments nécessaires à la mise en place d’un modelleur de surfaces dans un système de CAO orienté métier. Les éléments les plus adaptés à notre approche sont réunis pour permettre également de réaliser une étude en parallèle entre l’approche CAO classique et la méthode « synthétique » DIJA, en mettant en avant les améliorations proposées par notre système.

Nous étudions, dans un premier temps, les différents modèles de représentation des surfaces en CAO, pour se focaliser à la fin sur ceux qui s’adaptent mieux à notre approche. Ce sont les modèles discrets, qui, par rapport aux modèles classiques, présentent l’avantage d’être simples à représenter (géométriquement et topologiquement). La méthode proposée par ces travaux se base sur une triangulation de la surface comme modèle géométrique ultime, mais en acceptant n’importe quel modèle de départ. En s’appuyant sur les maillages triangulaires, nous réalisons ensuite un état de l’art des techniques de déformation de surfaces, pour compléter notre modèle. Ceci nous permet de découvrir des nouveaux axes d’autres thématiques de recherche introduites dans la CAO graphique. Des techniques intéressantes, comme la déformation avec contrainte de déplacement, les contraintes géométriques, les fonctions *decay* ou les surfaces de subdivision semblent s’adapter aux déformations DIJA, mais il manque toujours l’élément métier. Les contraintes métier sont imposées, dans notre système, à travers des outils de déformation. Nous étudions également les outils virtuels de déformation de surfaces d’une manière générale, pour se diriger, dans le deuxième chapitre, vers des outils plus adaptés de déformation et leur intégration

dans DIJA.

Nous ne pouvons naturellement pas continuer à présenter le fil conducteur de cette thèse sans s'arrêter, dans le deuxième chapitre, sur le contexte de ces travaux, le projet DIJA. Pour mieux comprendre ce travail, nous présentons la philosophie DIJA qui a permis de créer un système de CAO innovant, fonctionnel, distribué, collaboratif et intuitif. L'approche « synthétique » permet de créer ses pièces de manière intuitive, dans un processus de conception renversé. Grâce aux outils métier proposés par le système, l'utilisateur immerge dans son environnement de travail habituel et raffine sa pièce au fur et à mesure par des déformations successives, jusqu'à aboutir à la forme finale désirée. Les déformations s'appliquent sur des éléments visibles sur la peau de l'objet, ce qui rend le processus plus intuitif. Cette approche est possible grâce à une architecture DIJA multicouches et à une approche modulaire qui permettent de représenter la connaissance à différents niveaux sémantiques. Ainsi, elle est disponible à un niveau métier pour qu'elle soit compréhensible et manipulable par l'utilisateur et raffinée ensuite pour qu'elle soit représentable (et manipulable) à un niveau géométrique. Cette architecture permet des interactions Homme-Machine sur plusieurs niveaux, selon les compétences de l'utilisateur. Le cœur de l'interaction DIJA est représenté par les déformations qui ont été jusqu'à présent implémentées sur des éléments de dialogue 2D et qui sont étendues par une méthode proposée dans le troisième chapitre.

Le chapitre 3 fait donc une proposition pour la réalisation et l'intégration d'outils de déformation DIJA 3D. La méthode proposée reprend les principes énoncés dans le deuxième chapitre en ajoutant une troisième dimension aux outils métier, pour qu'ils soient applicables à une surface. Ce chapitre commence avec une étude de la corrélation entre les fonctions *decay* et les déformations métier. Les outils de déformation 3D et leur application sont ensuite démontrés avec un exemple de déformation d'une tablette arrière de voiture pour l'insertion d'un plumier. Cet exemple illustre le principe de déroulement et d'application des paramètres métier, du niveau sémantique plus haut, au niveau sémantique plus bas (géométrique). L'intégration des contraintes métier est ensuite démontrée à travers deux applications. La première concerne l'intégration de la contrainte volumique avec les déformations pour le design des flacons cosmétiques. La deuxième application intègre, de la même façon, d'autres contraintes que celle volumique (distance, hauteur, rapport hauteur-volume, etc.) pour le design des tablettes arrières de voiture.

Pour valider notre approche nous avons présenté, dans le quatrième chapitre, la mise en œuvre du modelleur de déformation de surfaces DIJA. Les détails techniques, le contexte et les

limites de la maquette sont présentées ici. Ce chapitre illustre les structures de données et les diagrammes de classes utilisés pour l'implémentation du modèle de déformation avec les outils sous-jacents proposés dans le chapitre précédent. L'interface graphique est aussi présentée, avec le module de visualisation VTK intégré. Le chapitre se termine avec une synthèse, du point de vue implémentation, des maquettes DIJA pour la déformation de surface.

Les concepts proposés sont ainsi validés, le système de déformations DIJA est à présent complet - par l'ajout d'une troisième dimension - laissant voie libre aux améliorations et aux développements futurs.

L'introduction des notions métiers dans les déformations de surfaces en CAO constitue l'apport majeur de ce travail de recherche. A court terme, le développement autour du squelette du modeleur de surfaces discrètes créé constitue une première direction à poursuivre dans la vue de l'exploitation et de la mise en valeur des résultats obtenus. La pluridisciplinarité des projets traités dans le domaine de la CAO impose une adaptation des interfaces en fonction de la spécialité du concepteur. Ceci repose sur le développement des nouveaux outils et des styles de déformation, sur l'enrichissement du vocabulaire et des bibliothèques de formes DIJA au niveau de la base de connaissances de chaque niveau de raffinement, tout en se basant sur la méthodologie présentée.

Une autre voie de recherche envisageable à court ou moyen terme autour du projet DIJA - qui découle de celle citée dans le paragraphe antérieur - consiste en une approche ontologique pour la représentation des connaissances dans un contexte de travail collaboratif. Les ontologies fournissent un vocabulaire accepté en unanimité par les membres d'un projet pour décrire les différents éléments du modèle (que ce soit modèle géométrique, processus de modélisation, etc.) à des différents niveaux. Les utilisateurs, les organisations, les systèmes de CAO peuvent ainsi communiquer, bien que des besoins et des expériences différents impliquent des différents points de vue. La thèse de Benoit Bertrand [BDGY08], actuellement en cours dans notre équipe, prend en compte ce sujet.

Les interfaces, quant à elles, pourraient être enrichies aussi au niveau IHM, par l'ajout des options (menus) avancées de conception (adaptées au métier), mais aussi au niveau de la visualisation, dévoilant au concepteur un nombre de détails visuels aussi grand qu'il le désire sur l'objet modélisé. Le modèle de visualisation peut ainsi être enrichi par une annotation basée, elle aussi, sur une approche ontologique. L'annotation du modèle de conception 3D peut être classifiée comme textuelle, quand elle se base sur les sémantiques, ou spatiale, quand elle est représentée

par des éléments de l'espace 3D. L'annotation pourrait se faire de manière manuelle, automatique ou semi-automatique, en fonction des préférences et de l'expérience de l'utilisateur, mais aussi en fonction des fonctions du système. L'annotation se réalise souvent de manière interactive, en plaçant le curseur sur un point quelconque de la scène de rendu du modèle 3D conçu. Selon le domaine, plusieurs fichiers de connaissances sont associés au modèle 3D. L'utilisateur a donc plusieurs choix lors de l'annotation, en fonction des fichiers auxquels il a accès par l'intermédiaire de l'application.

Au niveau géométrique, la généricité des modèles acceptés par DIJA implique l'utilisation des algorithmes de conversion continu-discret-continu. Or, la conversion entre les modèles de représentation et son influence sur le résultat obtenu est un sujet qui n'a pas été traité dans ces travaux de recherche. Dans les applications présentées on suppose qu'on débute dans le processus de modélisation avec un maillage de départ correct. Le résultat de la déformation dépend en grande partie de la régularité du maillage initial. Il est souvent nécessaire d'effectuer un pré- ou un post-traitement (décimation, subdivision) sur le maillage à déformer pour améliorer le résultat. Pourtant, pour obtenir des résultats optimaux, un maillage de départ propre s'impose, pour être en accord avec les spécificités des fonctions *decay*, et surtout avec le fait que la forme de la déformation s'applique à la forme du maillage.

Dans le contexte industriel, le même modèle géométrique est souvent utilisé pour des applications très variées : usinage, prototypage rapide, rendu, etc. Ces applications font souvent appel à une représentation discrète de la géométrie de l'objet : le maillage. Le passage du modèle continu au modèle discret se fait, généralement, sans tenir compte des besoins de l'analyse (spécificités métier, etc.). Les méthodes et les algorithmes de conversion des modèles génériques en maillages sont nombreux, mais souvent la conversion génère des effets indésirables sur la forme de l'objet modélisé (manque de précision, fermeture des lignes de contours, carreaux dégénérés, etc.). Une étape d'adaptation du maillage s'impose la plupart du temps. On voit donc la nécessité des algorithmes de conversion adaptés à notre méthodologie, et plus précisément aux spécificités des fonctions *decay*, pour assurer leur propagation correcte sur les points dans la zone de déformation.

Restant au niveau géométrique, un autre sujet envisageable sera la reconstruction des surfaces à partir des éléments de dialogue. Sous DIJA, les éléments de dialogue simples (comme les fibres et les contours caractéristiques) donnent une silhouette à l'objet modélisé et représentent donc un modèle synthétique (intéressant quand on veut le transmettre, par exemple). Il est donc intéressant de savoir reconnaître les éléments de dialogue d'un B-Rep classique, et aussi de

reconstruire la surface une fois les éléments de dialogue transmis.

Le succès de la modélisation sous DIJA est relatif à l'intuitivité du système et aux outils qu'il nous met à disposition. L'intégration des outils de réalité augmentée - par l'utilisation des techniques avancées de visualisation et par l'intégration d'un outil de retour de force - constitue une autre voie de recherche abordable sous DIJA dans un contexte IHM, pour accroître l'intuitivité et par conséquent l'aspect métier. Cela pourra conduire, finalement, à une extension du modèle DIJA par la prise en compte des modèles basés sur des propriétés mécaniques physiques, afin de rendre possible leur simulation. DIJA est actuellement intégralement basé sur des déformations purement géométriques. L'intégration des modèles de déformation basés sur les propriétés physiques des matériaux pourra élargir le spectre d'application de la philosophie DIJA dans des projets médicaux, tels que la modélisation des tissus humains, la chirurgie assistée par ordinateur ou dans des projets de modélisation 3D tels que le design des vêtements, pour donner quelques exemples.

Annexes

Annexe A

Eléments de géométrie discrète

Ontologie

Les travaux de recherche présentés dans ce manuscrit s'intéressent, dès le début, à la représentation discrète des surfaces. Dans cette démarche, nous avons étudié les concepts fondamentaux, les techniques et les applications sous-jacentes à la géométrie discrète [CM91]. Dans la géométrie discrète, les espaces de représentation sont discrets, réguliers et dénombrables dans les entiers. Le but est de traiter et analyser des données et des structures discrètes (pixels, voxels, surfels, segments, droites, plans, surfaces, etc.) dans des applications d'images et volumes numériques. Or, dans le domaine de la CAO, l'espace discret $\mathbb{Z}^n$ ne convient pas.

Sans être exhaustive, cette ontologie se propose de situer la problématique discrète de cette thèse. Nous rappelons au lecteur que, par discret, on comprend les points du maillage appartenant à l'espace continu $\mathbb{R}^3$ - pour le traitement - et les cellules discrètes que ces points forment (arêtes, triangles, etc.). On retient, cependant, quelques notions [Coe02, JF06, Lac05, RM89] qui représentent une mini-ontologie de géométrie discrète :

- **L'espace continu** de dimension n est l'espace $\mathbb{R}^n$;
- **L'espace discret** de dimension n est l'espace $\mathbb{Z}^n$;
- **Représentation analogique** : toute représentation associée à une variable qui varie de manière continue dans l'espace $\mathbb{R}^n$;
- **Représentation discrète** : toute représentation associée à une variable dénombrable que l'on peut mettre en correspondance avec l'espace $\mathbb{Z}^n$;
- **La géométrie discrète** se définit comme l'étude des propriétés géométriques d'ensembles de points représentés sur un maillage et produits par la discrétisation d'objets et de courbes du plan analogique ;
- **Un point discret** est un point qui appartenant à l'espace discret : $P \in \mathbb{Z}^n$ (voir Fig. A.1

- (a));
- Un **pixel** est un carré unité centré en P , où $P \in \mathbb{Z}^2$;
 - Un **voxel** est un cube unité centré en P , où $P \in \mathbb{Z}^3$ (voir Fig. A.1 (b));

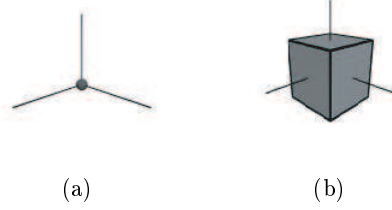


FIG. A.1: Un point discret et le voxel correspondant [JF06] : (a) $P \in \mathbb{Z}^n$; (b) voxel associé à P .

- Un **spel** est un cube unité de dimension n centré en $P \in \mathbb{Z}^n$;
- les éléments de l'espace discret (appelés *cellules*) peuvent être décomposés en des éléments de dimension inférieure au polygone pavant l'espace discret (voir Fig. A.2) :
 - le **pointel** est un élément de dimension 0 de l'espace discret;
 - le **lignel** est un élément de dimension 1 de l'espace discret;
 - le **surfel** est un élément de dimension 2 (une face) de l'espace discret.

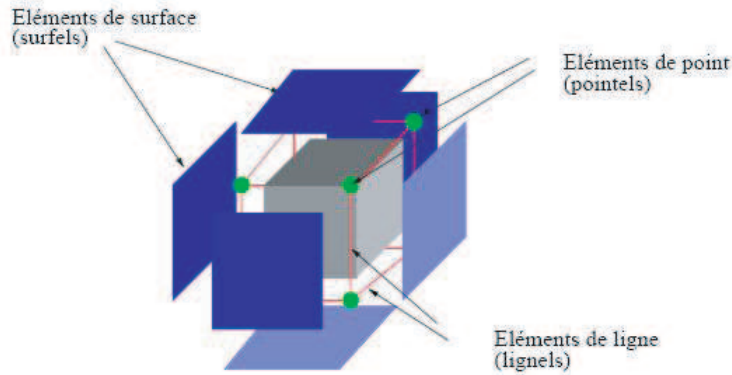


FIG. A.2: Décomposition d'un voxel en cellules de dimensions inférieures [Coe02].

- **ω -adjacence** de deux voxels : deux voxels sont ω -adjacents si leur distance bloc est égale à 1;
- En 3D, il existe trois types d'adjacences : adjacences par face (6-adjacents), par arête (18-adjacents) et par sommet (26-adjacents) (voir Fig. A.3);

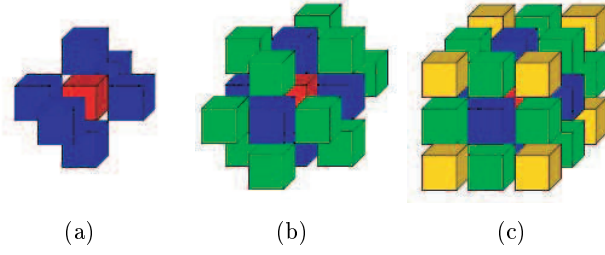


FIG. A.3: Adjacence dans l'espace 3D [Lac05] : (a) 6-adjacence (par face) ; (b) 18-adjacence (par arête) ; (c) 26-adjacence (par sommet).

- **Un objet discret** dans l'espace 3D est un ensemble de *voxels* adjacents (connexes) ;
- **Une surface discrète** est un ensemble de surfels munis d'une relation d'adjacence ;
- **Une isosurface** représente le lieux des points de même valeur par I ; l'isosurface de valeur s dans I est définie par :

$$(x, y, z) \in \mathbb{R}^3 | I(x, y, z) = s.$$

Les isosurfaces peuvent être calculées par l'algorithme de Marching-cube [LC87]. Il résulte une surface triangulée approchant l'isosurface s de I .

- **Dualité isosurfaces-surfaces discrètes** (voir Fig. A.4)
- Soit O l'ensemble des voxels u tels que $I(u) \geq s$;
- Les sommets de l'isosurface = les surfels séparant O de $\bar{O}$;
- On appelle *bel* un surfel frontière entre O et $\bar{O}$;
- On appelle *lien* une $n - 2$ -cellule incidente à un bel ;
- Dans l'espace 3D : le graphe d'adjacence de face des bels est proche de l'isosurface correspondante.

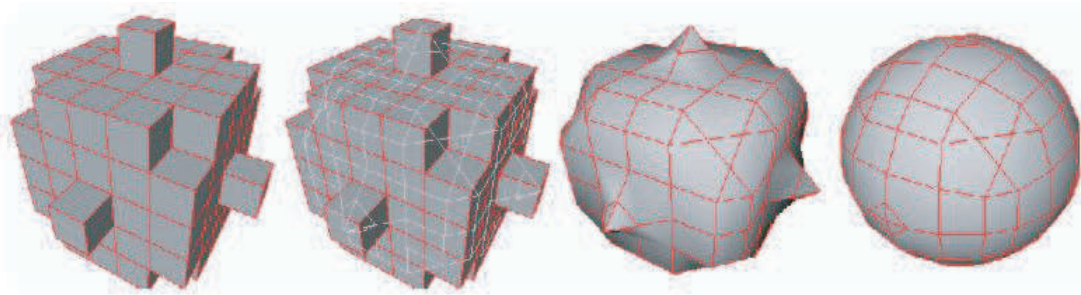


FIG. A.4: Dualité surface discrète-isosurface [Lac05].

- **Une forme discrète** est la discrétisation d'une forme euclidienne (supposée « régulière ») ;

- **Un estimateur géométrique discret** approche une mesure géométrique d'une forme continue en se basant seulement sur sa discrétisation ; les estimateurs géométriques discret se divisent en :

- **estimateurs géométriques globaux** : aire, longueur, moments, etc.
- **estimateurs géométriques locaux** : tangentes, normales, courbure, etc.

Les estimateurs géométriques discrets sont difficiles à valider expérimentalement, car une infinité de formes ont la même discrétisation ; un critère intéressant est la convergence asymptotique de l'estimateur géométrique vers la mesure géométrique ;

- **La continuité** sur $\mathbb{Z}^n$ n'est pas explicitement définie ; elle est estimée sur les invariants géométriques (normales, courbures, etc.) de la surface continue correspondante.
- **La continuité** décrit la relation entre des courbes ou des surfaces (voir Fig. A.5 et A.6). Plusieurs niveaux se distinguent au niveau de la continuité des courbes ou des surfaces [Rhi] :
 - **Continuité de position** (G^0) : elle ne tient compte que de la position des objets ; si les extrémités de chaque courbe/surface se trouvent au même endroit dans l'espace, les deux courbes/surfaces présentent une continuité de position au niveau de leur extrémité commune ;
 - **Continuité de tangence** (G^1) : elle tient compte de la position et de la direction des courbes/surfaces au niveau de leurs extrémités ; si les deux courbes/surfaces sont tangentes au niveau de leur extrémité commune, les courbes/surfaces présentent une continuité de tangence ; les courbes/surfaces G^1 sont aussi G^0 ;
 - **Continuité de courbure** (G^2) : elle tient compte de la position, de la direction et du rayon de courbure aux extrémités ; si le rayon de courbure est le même sur chaque courbe/surface au niveau de leur extrémité commune, les courbes/surfaces présentent une continuité de courbure ; les courbes/surfaces G^2 sont aussi G^1 et donc G^0 .

Chaque niveau de continuité considère que les conditions du niveau précédent sont remplies.

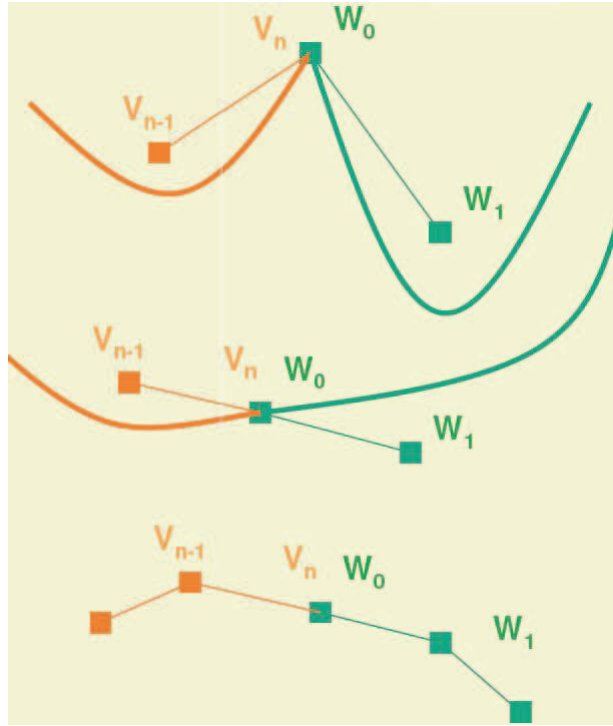


FIG. A.5: Niveaux de continuité des courbes/surfaces [Dub] : continuité de position (haut), de tangence (milieu) et de courbure (bas).

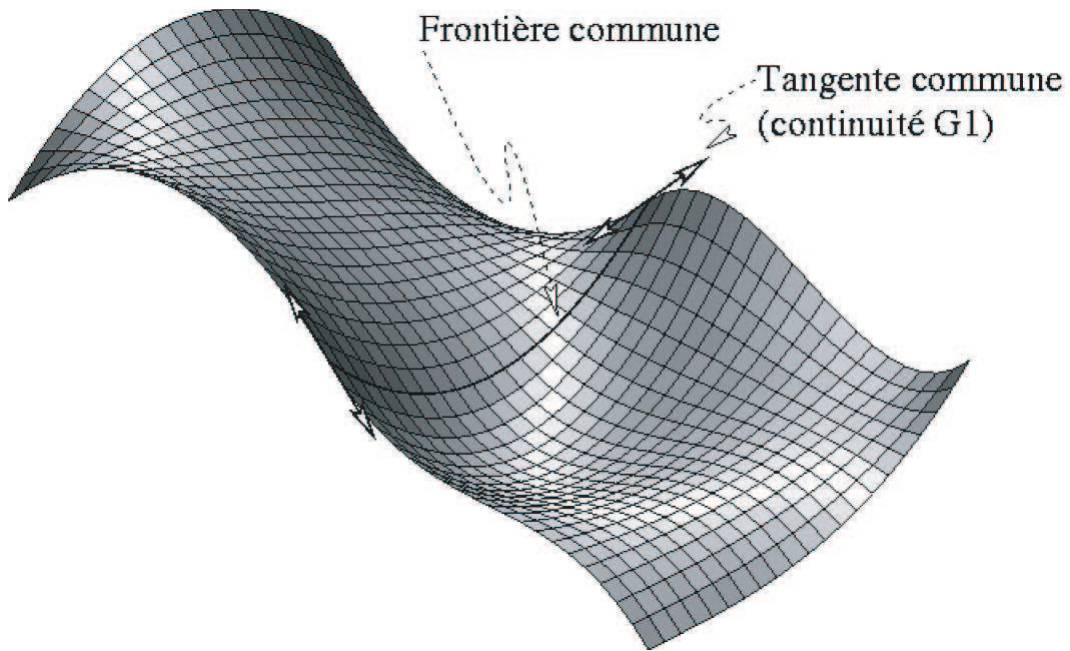


FIG. A.6: Continuité de tangence [Dub].

Annexe B

Fonctionnalités complémentaires DIJA

Afin d'améliorer l'interaction Homme-Machine, le système DIJA propose quelques fonctionnalités complémentaires. L'utilisateur a ainsi la possibilité de définir des nouveaux éléments de dialogue, des nouvelles formes de départ, d'ajouter des mots au vocabulaire et de créer des « macros » (scripts) de conception. Les quatre fonctionnalités complémentaires traduisent et enregistrent le savoir-faire de l'utilisateur en éléments spécifiques à la conception DIJA et accélèrent considérablement le processus de conception, en renforçant la réutilisation. Ces éléments sont détaillés dans les paragraphes suivants.

Création (personnalisation) d'éléments de dialogue

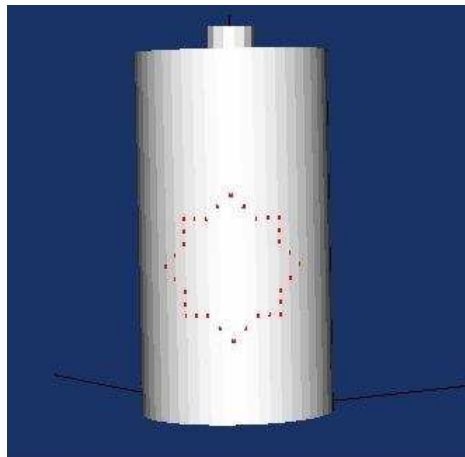


FIG. B.1: Contour caractéristique crée sur la surface d'un cylindre.

Cette opération permet à l'utilisateur de personnaliser les éléments de dialogue qui inter-

viennent dans la déformation, selon son point de vue et son expérience. Il lui permet, lorsqu'il commence sa conception avec une forme complexe choisie dans une librairie de formes, de définir ses propres éléments de dialogue à sa manière. Pour les objets de base (cylindre, boîte, etc.), le système propose des éléments de dialogue créés automatiquement. L'utilisateur peut, tout de même, ajouter ses propres éléments de dialogue à ceux proposés par le système. Si nous prenons l'exemple d'un cylindre, le système propose une fibre qui correspond à son axe de révolution ; supposons que l'axe soit vertical, l'utilisateur peut préférer avoir une fibre horizontale qui correspond mieux à sa vue synthétique de l'objet désiré. La création de nouvelles lignes et de nouveaux contours caractéristiques s'impose aussi, car ils donnent à l'utilisateur la liberté de définir la zone qu'il veut déformer, qui diffère en fonction du problème. Un exemple de définition de contour caractéristique est montré dans la figure B.1.

L'opération de personnalisation d'éléments de dialogue est essentiel à la philosophie de conception DIJA, car il offre à l'utilisateur la possibilité d'exprimer visuellement ses intentions de conception et la liberté de point de vue sur l'objet modélisé, tout comme l'opération de définition d'une forme de départ qui est présentée dans le paragraphe suivant.

Définition d'une nouvelle forme de départ

La forme de départ peut être choisie dans une librairie de formes qui peuvent être représentées par des formes géométriques simples ou par des formes plus complexes, définies par un utilisateur expert. Le choix se fait en fonction de l'expérience et de la vue du concepteur sur l'objet final et même, dans un nombre plus ou moins grand d'itérations, à la forme finale. La librairie de formes prédéfinies est, dans ce sens, très utile, car l'utilisateur a la possibilité de débiter plus près de son objectif qu'avec une forme de base. De plus, avoir plusieurs choix complexes sur la forme de départ peut offrir au concepteur de nouvelles possibilités d'arriver à la forme finale.

Pour définir une nouvelle forme de départ à partir d'un objet créé par l'utilisateur, le système retient le modèle géométrique de la forme, les éléments de dialogue définis sur la forme et un ensemble de notions sémantiques de haut niveau qui leur sont associées. L'ensemble de ces éléments sont enregistrés dans la base de connaissances. L'utilisateur doit donc préciser le comportement de chacun des éléments de dialogue présents sur la forme par rapport aux outils du métier auquel la forme de base fait référence. L'ajout des termes au vocabulaire est présenté dans le paragraphe suivant.

Ajout d'un mot au vocabulaire

Nous avons vu, dans les paragraphes précédents, que lorsque l'utilisateur veut appliquer un outil de déformation, il peut définir son action en employant un mot du vocabulaire spécifique à son métier. C'est une approche très conviviale d'interagir avec le système, qui offre à l'utilisateur une interface adaptée à son contexte de travail habituel. Par exemple, dans un contexte mécanique, on peut appliquer un outil « embouti » pour obtenir un embouti sur la pièce modélisée (voir Fig. B.2).

L'utilisateur a la possibilité d'enrichir le vocabulaire par l'ajout de nouveaux termes qui sont associés à de nouveaux outils de déformation. Pour ce faire, il est nécessaire de préciser explicitement les paramètres caractéristiques du nouvel outil. Chaque outil est considéré comme un ensemble de sous-outils de base paramétrés sur l'objet déformé. La définition de nouveaux outils englobe donc un paramétrage des outils de base pour capturer le savoir-faire de l'utilisateur qui peut ensuite être utilisé par lui-même et par tous les autres utilisateurs dont le métier coïncide avec le sien.

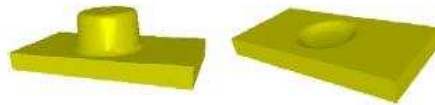


FIG. B.2: Déformation par emboutissage, en appliquant l'outil « embouti ».

Nous présentons dans le paragraphe suivant la modalité de créer et enregistrer un « macro » dans la base de connaissances.

Création d'un « macro » (script de conception)

Dans le contexte DIJA, un « macro » représente une suite d'actions appliquées à un élément de dialogue, qui peut être appliquée à un élément similaire [Dan02]. Un « macro » définit une entité qui permet de dupliquer une déformation. Les « macros » sont enregistrés par le système dans la base de connaissances. Prenons l'exemple de la colonne architecturale présentée dans la section 2.2.1. Pour créer une colonne complexe dont le fût est décoré, l'utilisateur pourrait appliquer un « macro » de déformation dont le paramètre est fourni par une base procédurale qui permet d'appliquer une même déformation sur diverses parties de l'objet (voir Fig. B.3).

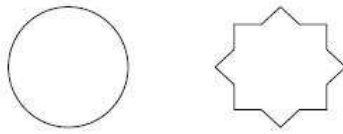


FIG. B.3: Exemple de section du fût d'une colonne architecturale : à gauche, la forme de départ ; à droite, la forme déformée par l'application d'une déformation (angle droit) sur 8 parties du contour.

La puissance de la modélisation « synthétique » DIJA réside dans la possibilité de combiner des outils de déformation de base pour arriver à des outils complexes, réutilisables à l'aide des scripts de conception.

Les quatres opérations présentées dans cette annexe sont complémentaires au sens où elles aident à la représentation des connaissances sous DIJA.

Bibliographie

- [AB97] F. Aubert and D. Bechmann. Volume preserving space deformation. *Comput. & Graphics*, 21 :625–639, 1997.
- [ABB⁺06] L.S. Avila, S. Barré, R. Blue, D. Cole, B. Geveci, W.A. Hoffman, B. King, C.C. Law, K.M. Martin, W.J. Schroeder, and A.H Squillacote. *The VTK User's Guide*. Kitware, Inc., 2006.
- [AHU58] K. Arrow, L. Hurwicz, and H. Uzawa. *Studies in Linear and Non-linear Programming*. Stanford University Press, 1958.
- [AWW89] J. Allan, B. Wyvill, and I.H. Witten. A methodology for direct manipulation of polygon meshes. In *New Advances in Computer Graphics (Proceedings of CG International '89)*, pages 451–469, 1989.
- [BA77] B.A. Brewer and D.C. Anderson. Visual interaction with overhauser curves and surfaces. In *Proceedings of SIGGRAPH*, 1977.
- [Bar81] A.H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1) :11–23, 1981.
- [Bar84] A.H. Barr. Global and local deformations of solid primitives. *Computer Graphics*, 18 :21–30, 1984.
- [BB88] R.H. Bartels and J.C. Beatty. *B-splines*. L'imprimerie Laballery, Clamency, 1988.
- [BB89] R. Bartels and J. Beatty. A technique for direct manipulation of spline curves. In *Proceedings of Graphics Interface*, 1989.
- [BB91] P. Borrel and D. Bechmann. Deformation of n-dimensional objects. *Comput Geom Appl*, 1 :427–453, 1991.
- [BBN00] R.P. Barneva, V.E. Brimkov, and P. Nehlig. Thin discrete triangular meshes. *Theoretical Computer Science*, 246 :73–105, 2000.

- [BCE⁺92] K.W. Brodlie, L.A. Carpenter, R.A. Earnshaw, J.R. Gallop, R. Hubolt, A.M. Mumford, C.D. Osland, and P. Quarendon. *Scientific Visualisation Techniques and Applications*. Springer-Verlag, 1992.
- [BCF94] V. Branco, A. Costa, and F.N. Ferriera. *Sketching 3D models with 2D interaction devices*, 13(3) :489–502, 1994.
- [BD93] D. Bechmann and N. Dubreuil. Animation through space and time based on a space deformation model. *Journal of Visualisation and Computer Animation*, 4(3) :165–184, 1993.
- [BDGY08] B. Bertrand, F. Danesi, N. Gardan, and G. Yvon. Modeling and management of process in numerical engineering. application in foundry. In *Proceedings of the 7th WSEAS International Conference on SYSTEM SCIENCE and SIMULATION in ENGINEERING (ICOSSE'08)*, pages 206–210, 2008.
- [BDHO92] J. Butterworth, A. Davidson, S. Hench, and T.M. Olano. 3dm : A three dimensional modeler using a head-mounted display. *Computer Graphics*, 25(2) :135–138, 1992.
- [BGP06] E. Brosse, Y. Gardan, and E. Perrin. Dija project : Forging preforms design using trade knowledge. In *International Conference on Geometric Modeling and Imaging (GMAI'06)*, pages 244–249, 2006.
- [BGP07] E. Brosse, Y. Gardan, and E. Perrin. Forging-dedicated cad : how getting forging perform by using trade knowledge and feature reorganization. In *4th International Conference on Computer Graphics, Imaging and Visualisation (CGIV'07)*, pages 211–216, 2007.
- [BK] M. Botsch and L. Kobbelt. Multiresolution surface representation based on displacement volumes. In *Proceedings Eurographics. Computer Graphics Forum*, pages 483–491.
- [BL94] J.R. Bill and S.K. Lodha. Computer sculpting of polygonal meshes using virtual tools. *Tech. Rep. UCSC-CRL-94-27, Baskin Center for Computer Engineering and Information Sciences, University of California, Santa Cruz, U.S.A.*, 94-27, 1994.
- [Bli82] J.F. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3) :235–256, 1982.
- [Blo87] J. Bloomenthal. Polygonalisation of implicit surfaces. Technical report, Xerox Technical Report CSL-87-2, 1987.

- [BPR⁺] M. Botsch, M. Pauly, C. Rossl, S. Bischoff, and L. Kobbelt. *Geometric Modeling Based on Triangle Meshes*.
- [BR94] P. Borrel and A. Rappoport. Simple constrained deformations for geometric modeling and interactive design. *ACM Transactions on Graphics (TOG)*, 13 :137 – 155, 1994.
- [BW92] D. Baraff and A. Witkin. Dynamic simulation of non-penetrating flexible bodies. In *Computer Graphics Proceedings, Annual Conference Series, Proceedings of SIGGRAPH*, 1992.
- [Béz78] P. Bézier. General distortion of a set of biparametric patches. *Computer-Aided Design*, 10(2) :116–120, 1978.
- [Béz87] P. Bézier. *Courbes et surfaces*. L'imprimerie Laballery, Clamency, 1987.
- [BZWB89] A. Barr, D. Zeltzer, A. Witkin, and J. Blinn. Physically-based modeling : Past, present, and future. In *SIGGRAPH Panel Proceedings*, 1989.
- [CB07] D. Culler and W. Burd. A framework for extending coputer aided process planning to include business activities and computer aided design and manufacturing (cad/cam) data retrieval. *Robotics and Computer-Integrated Manufacturing*, 23 :339–350, 2007.
- [CC78] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10,6 :350–355, 1978.
- [CDGP08a] I.G. Ciuciu, F. Danesi, Y. Gardan, and E. Perrin. Surface deformation on a discrete model for a cad system. In *EuroCG*, pages 185–188, Nancy, France, March 2008.
- [CDGP08b] I.G. Ciuciu, F. Danesi, Y. Gardan, and E. Perrin. Trade oriented direct deformation of surfaces. In *International Conference On Geometry And Graphics (ICGG08)*, Dresden, Germany, August 2008.
- [CG91] G. Celniker and D. Gossard. Deformable curve and surface finite-elements for free-form shape design. *SIGGRAPH 1991 Proceedings. Computer Graphics*, 25 :257–266, 1991.
- [CHP89] J.E. Chadwick, D.R. Haumann, and R.E. Parent. Layered construction for deformable animated characters. *Computer Graphics*, 23(3) :243–252, 1989.
- [Cla76] J.H. Clark. Designing surfaces in 3-d. *Communications of the ACM*, 19(8) :454–460, 1976.

-
- [CM91] J.M. Chassery and A. Montanvert. *Géométrie discrète en analyse d'images*. 1991.
- [Coe02] D. Coeurjolly. *Algorithmique et géométrie discrète pour la caractérisation des courbes et des surfaces*. PhD thesis, Université Lumière Lyon 2, 2002.
- [Coq90] S. Coquillart. Extended free-form deformation : a sculpting tool for 3d geometric modelling. *International Conference on Computer Graphics and Interactive Techniques, Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, 24(4) :187–196, 1990.
- [Dan02] F. Danesi. *Contribution à la définition et à l'intégration de notions intuitives dans le dialogue Homme - Machine en C.A.O.* PhD thesis, Université de Reims Champagne-Ardenne, 2002.
- [DDGP02] F. Danesi, L. Denis, Y. Gardan, and E. Perrin. Basic components of the dija project. In *The 7th ACM Symposium on Solid Modelling and Applications 2002 (SM'02)*, 2002.
- [Del94] H. Delingette. Simplex meshes : a general representation for 3d shape reconstruction. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1994.
- [Del99] H. Delingette. General object reconstruction based on simplex meshes. *International Journal of Computer Vision*, 32(2) :111–146, 1999.
- [Den03] L. Denis. *Vers une architecture adaptée à un système de CAO utilisable à travers Internet*. PhD thesis, Université de Reims Champagne-Ardenne, 2003.
- [DGGP03] F. Danesi, N. Gardan, Y. Gardan, and E. Perrin. Synthetic modelling system architecture. In *Preceedings International Conference on Geometric Modeling and Graphics (GMAG)*, 2003.
- [DGMP99] F. Danesi, Y. Gardan, B. Martin, and I. Pecci. 3d design by sketches. In 372-381, editor, *12'th journeys of the AFIG*, 1999.
- [DGMP00a] F. Danesi, Y. Gardan, B. Martin, and I. Pecci. 3d design by sketches. *Revue Internationale de CFAO et d'informatique graphique*, 15 :71–86, 2000.
- [DGMP00b] F. Danesi, Y. Gardan, B. Martin, and I. Pecci. Sketches : definition and usage in 3d design. In *Premières rencontres Jeunes Chercheurs en Interaction Homme-Machine, RJC-IHM*, pages 23–26, 2000.

- [DLG90] N. Dyn, D. Levin, and J.A. Gregory. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9 :160–169, 1990.
- [DOM04] Document object model, version 1.0. world wide web consortium recommendation, 2004.
- [DS78] D. Doo and M. Sabin. Analysis of the behaviour of recursive division surfaces near extraordinary points. *Computer Aided Design*, 10, 6 :356–360, 1978.
- [DTG96] M. Desbrun, N. Tsingos, and M.P. Gascuel. Adaptive sampling of implicit surfaces for interactive modelling and animation. *Computer Graphics Forum*, 15(5) :319–327, 1996.
- [Dub] N. Dubreuil. Modélisation de formes libres et déformations.
- [EHBE97] L. Eggi, C. Hsu, B. Bruderlin, and G. Elber. Inferring 3d models from freehand sketches and constraints. *Computer-Aided Design*, 29(2) :101–112, 1997.
- [eIPX] eXtensible Imaging Platform (XIP). <https://collab01a.scr.siemens.com/xipwiki>.
- [Elb01] G. Elber. Multiresolution curve editing with linear constraints. In *ACM Symposium on Solid and Physical Modeling*, pages 109–119, 2001.
- [Els90a] M. Elson. Displacement animation : development and application. In *Character Animation by Computer : SIGGRAPH 1990 Course Notes, International Conference on Computer Graphics and Interactive Techniques*, pages 14–36, 1990.
- [Els90b] M. Elson. Winged edge polyhedral models : Their use in the construction of characters, speech, emotion, and body language. In *State of the Art in facial Animation : SIGGRAPH 1990 Course Notes*, pages 21–42, 1990.
- [Far] G. Farin. A history of curves and surfaces in cagd. *Arizona State University Tempe, AZ 85257-5406*.
- [FB93] B. Fowler and R. Bartels. Constraint-based curve manipulation. *IEEE Computer Graphics and Applications*, 13(5) :43–49, 1993.
- [Fow92] B. Fowler. Geometric manipulation of tensor product surfaces. In *Symposium of Interactive 3D Graphics*, 1992.
- [FSJ⁺06] J. Feng, J. Shao, X. Jin, Q. Peng, and A.R. Forrest. Multiresolution free-form deformation with subdivision surface of arbitrary topology. *The Visual Computer*, 22(1) :28–42, 2006.

- [FUN⁺08] H. Fujita, Y. Uchiyama, T. Nakagawa, D. Fukoka, Y. Hatanaka, T. Hara, G.N. Lee, Y. Hayashi, Y. Ikedo, X. Gao, and X. Zhou. Computer-aided diagnosis : The emeing of three cad systems induced by japanese health care needs. *Computer Methods and Programs in Biomedicine*, 11, 2008.
- [FvDF⁺00] J.D. Foley, A. van Dam, S.K. Feiner, J.F. Hughes, and R.L. Phillips. *Introduction à l'infographie*. HUROPÉ, Barcelone, 2000.
- [Gai96] J.E. Gain. *Virtual Sculpting : an investigation of directly manipulated free form deformations in a VE*. PhD thesis, Rhodes University, 1996.
- [Gar91] Y. Gardan. *La CFAO, introduction, techniques et mise en oeuvre 3ème édition*. ISBN 2-86601-293-3. Imprimerie Laballery, Clamecy, 1991.
- [Gar05] N. Gardan. *Proposition d'une Méthodologie de Travail Collaboratif : Concepts et Applications*. PhD thesis, Université de Reims Champagne-Ardenne, 2005.
- [Geo08] O. Georget. Courbes et surfaces de bézier. <http://nonifier.ovh.org/wordpress>, 2008.
- [GG03] N. Gardan and Y. Gardan. An application of knowledge based modelling using scripts. *Expert Systems With Applications*, 25(4) :555–568, 2003.
- [GH91] A. Gaylean and J.F. Hughes. Sculpting : An interactive volumetric modeling technique. *Computer Graphics*, 25(4), 1991.
- [Gle92] M. Gleicher. Integrating constraints and direct manipulation. In *Proceedings of Symposium on Interactive 3D Graphics*, 1992.
- [GLS95] M. Green, J. Liang, and C. Shaw. Interactive 3d geometrical modelers for virtual reality and design. In *International Conference on Virtual Systems and Multimedia*, 1995.
- [GM97] S.F.F Gibson and B. Mirtich. A survey of deformable modeling in computer graphics. Technical Report TR-97-19, MITSUBISHI ELECTRIC RESEARCH LABORATORY, 1997.
- [GP89] J. Griessmair and W. Purgathofer. Deformations of solids with tri-variate b-splines. In *EUROGRAPHICS*, pages 137–48, 1989.
- [GP96] Y. Gardan and E. Perrin. *An algorithm reducing 3D boolean operations to a 2D problem : concepts and results*, 28(4) :277–287, 1996.

- [Gre96] J. Greiner, G. Loos. Data dependent thin plate energy and its use in interactive surface modeling. In *Proceedings Eurographics, Computer Graphics Forum*, volume 15, pages 176–185, 1996.
- [HDD⁺94] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *Proceedings of SIGGRAPH*, 1994.
- [Hes05] F. Heschung. *Vers une construction et une adaptation automatique des maillages*. PhD thesis, Université de Reims Champagne-Ardenne, 2005.
- [HHK92] W.M. Hsu, J.F. Hughes, and H. Kaufman. Direct manipulation of free-form deformations. In *International Conference on Computer Graphics and Interactive Techniques, Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 177–184, 1992.
- [Hil08] M. Hilléreau. *Vers une CAO guidée par la connaissance des fabricants*. PhD thesis, Université de Reims Champagne-Ardenne, 2008.
- [HML] G. Hirota, R. Maheshwari, and M.C. Lin. Fast volume-preserving free form deformation using multi-level optimisation. In ACM Press New York, editor, *Proceedings of the Fifts ACM Symposium on Solid Modeling and Applications*, pages 234–245.
- [Hol06] R. Holbrey. Dimension reduction algorithms for data mining and visualization. Technical report, University of Leeds/Edinburgh, 2006.
- [J3D] Java.net. java3d. <https://java3d.dev.java.net>.
- [Jav] Sun microsystems. <http://java.sun.com>.
- [JDB] Sun microsystems. jdbc. <http://java.sun.com/products/jdbc/overview.html>.
- [JF06] D. Jamet and C. Fiorio. Géométrie discrète - droites et plans. Technical report, LIRMM Montpellier, 2006.
- [Joh] A. Johnson. <http://www.evl.uic.edu/aej/526/pics/contouring.jpg>.
- [Kam97] K. Kameyama. Virtual clay modeling system. In *Proceedings of SIGGRAPH*, 1997.
- [KCY92] A. Kaufman, D. Cohen, and R. Yagel. Normal estimation in 3d discrete space. *The Visual Computer*, 8 :278–291, 1992.
- [KP00] V. Kokotovich and T. Purcell. Mental synthesis and creativity in design : an experimental examination. *Design studies*, 21 :437–449, 2000.

- [KRK⁺06] Y.J. Kil, P. Renzulli, O. Kreylos, B. Hamann, G. Monno, and O.G. Staadt. 3d warp brush modeling. *Computers & Graphics*, 30 :610–618, 2006.
- [KS95] P.D. Kaklis and N.S. Sapidis. Convexity-preserving interpolatory parametric splines of non-uniform polynomial degree. *Computer Aided Geometric Design*, 12(1) :1–26, 1995.
- [Lac05] J.O. Lachaud. Surfaces discrètes : définition, extraction et géométrie. Technical report, LIRMM Montpellier, 2005.
- [Las44] J. Lassiter. Principles of traditional animation applied to 3d computer animation. In *Computer Graphics (SIGGRAPH '87 Proceedings)*, 45-44.
- [LC87] W.E. Lorensen and H.E. Cline. Marching cubes : a high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4) :163–169, 1987.
- [LCJ94] F. Lazarus, S. Coquillart, and P. Jancene. Axial deformations : an intuitive deformation technique. *Computer-Aided Design*, 26(8) :607–13, 1994.
- [LCOGL07] Y. Lipman, D. Cohen-Or, R. Gal, and D. Levin. Volume and shape preservation via moving frame manipulation. *ACM Transactions on Graphics*, 26(1), 2007.
- [LG94] J. Liang and M. Green. Jcad : A highly interactive 3d modeling system. *Computer Graphics*, 18(4) :499–506, 1994.
- [LKMTT91] A. LeBlanc, P. Kalra, N. Magnenat-Thalmann, and D. Thalmann. Sculpting with the 'ball and mouse' metaphor. In *Proceedings of Graphics Interface 1991*, pages 152–159, 1991.
- [Loo87] C.T. Loop. *Smooth subdivision surfaces based on triangles*. PhD thesis, Utah University, 1987.
- [LSCO⁺04] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rossel, and H.P. Seidel. Differential coordinates for interactive mesh editing. In *Proceedings of the Shape Modeling International Conference*, pages 181–190, 2004.
- [LW94] H.J. Lamouns and W.N. Waggenspack. Nurbs-based free-form deformation. *IEEE Computer Graphics and Applications*, 14(6) :59–65, 1994.
- [LX01] D.T. Liu and X.W. Xu. A review of web-based product data management systems. *Computers in Industry*, 44 :251–262, 2001.
- [Mau03] B. Maury. Version continue de l'algorithme d'uzawa. *Comptes Rendus de l'Académie des sciences*, I(337) :31–36, 2003.

-
- [MCL08] C.K. Mok, K.S. Chin, and H. Lan. An internet-based intelligent design system for injection moulds. *Robotics and Computer-Integrated Manufacturing*, 24 :1–15, 2008.
- [MCQ05] K.T. McDonnell, Y.S. Chang, and H. Qin. Digital sculpture : A subdivision-based approach to interactive implicit surface modelling. *Graphical Models*, 67 :347–369, 2005.
- [MDA01] J. Montagnat, H. Delingette, and N. Ayache. A review of deformable surfaces : topology, geometry and deformation. *Image and Vision Computing*, 19(14) :1023–1040, 2001.
- [Mil89] G. Miller. Globular dynamics : a connected particle system for animating viscous fluids. *Comput. & Graphics*, 13(3) :305–309, 1989.
- [MJ96] R. MacCracken and K.I. Joy. Free-form deformations with lattices of arbitrary topology. In *Computer Graphics, Proceedings of SIGGRAPH*, pages 181–188, 1996.
- [MMG04] C. Minich, K. Monnier, and Y. Gardan. Vers une sécurisation accrue des échanges cao. In *Conférence internationale MICAD*, 2004.
- [MN94] T. Murakami and N. Nakajima. Direct and intuitive device for 3d shape deformation. In *Proceedings of SIGGRAPH*, 1994.
- [MYS99] A. Miyoshi, G. Yagawa, and Sasaki S. An interface agent that actively supports beginner users in performing analyses. *Advances in Engineering Software*, 30 :575–579, 1999.
- [NS90] G.M. Nielson and B. Shriver. *Visualisation in Scientific Computing*. IEEE Computer Society Press, 1990.
- [OMV01] J. Ouédraogo, F.S. Marzani, and Y. Voisin. Construction of a surfacic mesh from an image acquired with a structured light system. In *International Conference on Image and Signal Processing*, pages 986–993, 2001.
- [Ost87] E. Ostby. Describing free-form 3d surfaces for animation. In *Symposium on Interactive 3D Graphics, Proceedings of the 1986 workshop on Interactive 3D graphics*, pages 251–258, 1987.
- [Par77] R. Parent. A system for sculpting 3d data. In *Computer Graphics (ACM SIGGRAPH Proceedings)*, pages 138–147, 1977.
- [PB81] S. Platt and N. Badler. Animating facial expressions. In *Proceedings of SIGGRAPH*, 1981.

- [PE98] M. Plavnik and G. Elber. Surface design using global constraints on total curvature. In *The VIII IMA Conference on Mathematics of Surfaces*, 1998.
- [Per06] E. Perrin. *HDR : Vers une C.A.O. fonctionnelle : éléments d'architecture et amélioration sémantique des opérateurs booléens*. PhD thesis, Université de Reims Champagne-Ardenne, 2006.
- [PF98] L. Papier and J. Françon. Evaluation de la normale au bord d'un objet discret 3d. *Revue Internationale de CFAO et d'Informatique Graphique*, 13(2) :205–226, 1998.
- [PGL⁺02] J.P. Pernot, S. Guillet, J.C. Leon, F. Giannini, B. Falcidieno, and E. Catalano. A shape deformation tool to model character lines in the early design phases. In *Proceedings Shape Modeling International*, 2002.
- [RB04] D. Rypl and Z. Bittnar. Triangulation of 3d surfaces reconstructed by interpolating subdivision. *Computers and Structures*, 82 :2093–2103, 2004.
- [Ree83] W.T. Reeves. Particle systems : a technique for modelling a class of fuzzy objects. In *Computer Graphics (ACM SIGGRAPH Proceedings)*, pages 359–376, 1983.
- [Rei09] M. Reimeringer. *Une méthodologie et des outils pour concevoir en tenant compte de la simulation*. PhD thesis, Université de Reims Champagne-Ardenne, 2009.
- [Rey97] H. Reynolds. An alternative inter-particle force model for coupled system flexible body dynamics. In *8th Eurographics Workshop on Computer Animation and Simulation*, 1997.
- [Rhi] Rhino. <http://www.fr.rhino3d.com/4/help/Information/Glossary.htm>.
- [RK03] D. Roller and I. Kreuz. Selecting and parameterising components using knowledge based configuration and heuristic that learns and forgets. *Computer-Aided Design*, 35 :1085–1098, 2003.
- [RLR07] K. Russell, W.T. Lee, and Sodhi R.S. On the application of cad technology for the synthesis of spatial revolute-revolute dyads. *Computer-Aided Design*, 39 :1075–1080, 2007.
- [RM89] A. Rosenfeld and R.A. Melter. Digital topology. *The Mathematical Intelligencer*, 11(3) :69–72, 1989.
- [RSB96] A. Rappoport, A. Sheffer, and M. Bercovier. Volume-preserving free-form solids. In *IEEE Transactions on Visualisation and Computer Graphics*, 1996.

- [SAX] Simple api for xml. <http://www.saxproject.org>.
- [SF98] K. Singh and E. Fiume. Wires : A geometric deformation technique. In *Proceedings of the SIGGRAPH conference*, 1998.
- [SHB07] B. Sauvage, S. Hahmann, and G.P. Bonneau. Volume preservation of multiresolution meshes. In *Proceedings Eurographics. Computer Graphics Forum*, volume 26(3), pages 275–283, 2007.
- [SHBE08] B. Sauvage, S. Hahmann, G.P. Bonneau, and G. Elber. Detail preserving deformation of b-spline surfaces with volume constraint. *Computer Aided Geometric Design*, 25 :678–696, 2008.
- [Sim90] K. Sims. Particle animation and rendering using data parallel computation. In *Computer Graphics, Proceedings of SIGGRAPH*, 1990.
- [SML06] W. Schroeder, K. Martin, and K. Lorensen. *The Visualisation Toolkit An Object Oriented Approach to 3D Graphics*. Kitware, Inc., 2006.
- [SP86] T.W. Sederberg and S.R. Parry. Free-form deformation of solid geometric models. *Computer Graphics*, 20(4) :151–160, 1986.
- [ST92] R. Szeliski and D. Tonnensen. Surface modelling with oriented particle systems. In *Computer Graphics (ACM SIGGRAPH Proceedings)*, pages 185–194, 1992.
- [Sta98] J. Stam. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *Proceedings of SIGGRAPH*, 1998.
- [TF08] P. Theodorou and G. Florou. Manufatcuring strategies and financial performance-the effect of advanced information technology : Cad/cam systems. *Omega, The international Journal of Management Science*, 36 :107–121, 2008.
- [TGM03] F. Triquet, L. Grisoni, and P. Meseure. Realtime visualisation of implicit objects with contact control. In *International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, 2003.
- [Toh03] J. Tohka. *Global optimisation-based deformable meshes for surface extraction from medical images*. PhD thesis, Digital Media Institute / Signal Processing, Tampere University of Technology, 2003.
- [TPF89] D. Terzopoulos, J. Platt, and K. Fleischer. From goop to glob : Heating and melting deformable models. In *Proceedings Graphics Interface*, 1989.

- [vFTS06] W. von Funk, H. Theisel, and H.P. Seidel. Vector field based shape deformations. In *SIGGRAPH Computer Graphics 2006*, pages 1118–1125, 2006.
- [VMT97] P. Volino and N. Magnenat-Thalmann. Developing simulation techniques for an interactive clothing system. In *Proceedings of the 1997 International Conference on Virtual Systems and Multimedia*, 1997.
- [VTK] VTK. <http://www.vtk.org/>.
- [WH94] A. Witkin and P. Heckbert. Using particles to sample and control implicit surfaces. In *Proceedings of SIGGRAPH*, 1994.
- [Wik] Wikipedia. <http://fr.wikipedia.org/wiki/Fichier:Marchingcubes-head.png>.
- [Wol90] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1990.
- [WT91] K. Waters and D. Terzopoulos. Modeling and animating faces using scanned data. *Visualisation and Computer Animation*, 2 :123–128, 1991.
- [WW92] W. Welch and A. Witkin. Variational surface modeling. In *Proceedings of SIGGRAPH*, 1992.
- [ZCCD06] V.B. Zordan, B. Celly, B. Chiu, and P.C. DiLorenzo. Breathe easy : Model and control of human respiration for computer animation. *Graphical Models*, 68 :113–132, 2006.
- [ZSD⁺00] D. Zorin, P. Schroder, T. DeRose, L. Kobbelt, A. Levin, and W. Sweldens. *Subdivision for modeling and animation*, course notes of acm siggraph 00 edition, 2000.
- [ZSS96] D. Zorin, P. Schroder, and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. In *Proceedings of SIGGRAPH*, 1996.