NUMERO D'ORDRE : **ANNEE : 1997**

Laboratoire d'Informatique Fondamentale de Lille

THESE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

pour obtenir le titre de

DOCTEUR EN INFORMATIQUE

par

Philippe MESEURE

MODELISATION DE CORPS DEFORMABLES POUR LA SIMULATION D'ACTES CHIRURGICAUX

Thèse soutenue le 6 janvier 1997, devant la commission d'examen :

Président :	J.M. Geib	LIFL, Lille
Directeur de thèse :	C. Chaillou	LIFL - EUDIL, Lille
Rapporteurs :	D. Thalmann	EPFL-LIG, Lausanne
	G. Hégron	EMN, Nantes
Examinateurs :	A. Luciani	ACROE, Grenoble
	M. Mériaux	SIC-IRCOM, Poitiers
	H. Delingette	INRIA, Sophia Antipolis
	P. Dubois	ITM-CLARC, CHR&U Lille

UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE
U.F.R. d'I.E.E.A. Bât M3. 59655 Villeneuve d'Ascq CEDEX
Tél. 03.20.43.47.24 Fax. 03.20.43.65.66

Remerciements

- Merci à Christophe Chaillou de m'avoir encadré tout au long de cette thèse. Il a toujours su me proposer de nouvelles pistes lorsque j'étais dans une impasse. Merci à lui et à Sylvain Karpf de m'avoir fait découvrir les simulateurs médicaux.
- Merci à Daniel Thalmann et Gérard Hégron d'avoir accepté de rapporter ma thèse. Leurs remarques pertinentes m'ont permis d'améliorer ce document.
- Merci à Jean-Marc Geib qui m'a fait l'honneur de présider mon jury.
- Merci à Michel Mériaux de m'avoir accueilli au Laboratoire d'Informatique Fondamentale de Lille et d'avoir accepté de juger mon travail.
- Merci à Annie Luciani et Hervé Delingette qui se sont intéressés depuis longtemps à mes travaux et ont accepté d'être examinateurs de ma thèse. Les discussions que j'ai eues avec eux ont toujours été très enrichissantes.
- Merci à Patrick Dubois de participer à mon jury. Il a été durant ces travaux, le coordinateur entre les informaticiens et les médecins. Sans lui et les autres membres de l'ITM, les simulateurs chirurgicaux ne seraient que des concepts. Merci à Jean-Francois Rouland pour avoir eu un jour l'idée d'un simulateur en ophtalmologie et d'avoir ensuite contribué à sa réalisation. Son idée est en quelque sorte à l'origine de l'activité simulateurs.
- Merci aux thésards travaillant sur les simulateurs médicaux : Coyotte, 2ème classe Wibaux et Jérôme ; aux autres membres (ou ex-membres !) de l'équipe : Hervénounet, Samclone, Grégounet, et tous les autres trop nombreux pour être cités. Un grand merci à Scarabée pour avoir saboté tous les PC !
- Merci à Wookie pour nos tentatives réciproques de retarder la thèse de l'autre. Merci à Zeb, par principe.
- Merci à mes parents, mes beaux-parents et mes amis pour avoir supporté mes états d'âme durant cette recherche.
- Et merci surtout à Sandrine, à qui je dédie cette thèse. J'ai toujours pu compter sur elle dans les moments les plus difficiles.

Table des matières

Partie 1. Les simulateurs médicaux

Chapitre 1. Les Environnements Virtuels en Médecine.....17

Introduction.....17

1.1. Applications des environnements virtuels en médecine.....17

1.1.1. Les applications.....18

1.1.2. Les liens entre ces applications.....21

1.2. Apports des simulateurs médicaux.....22

1.2.1. Motivations.....23

1.2.2. Avantages des simulateurs.....23

1.3. Classification des simulateurs médicaux - Etat de l'art.....24

1.3.1. Les simulateurs de navigation.....25

1.3.2. Les simulateurs avec déformations.....26

1.3.3. Les simulateurs avec déformations et retour d'effort.....26

1.3.4. Bilan.....27

Conclusion.....27

Chapitre 2. Les simulateurs médicaux au LIFL.....29

Introduction.....29

2.1. Le GIS simulateurs d'instruments médicaux.....29

2.2. Les projets.....29

2.2.1. Sophocle : simulateur de photo-coagulation laser.....30

2.2.2. Speed : simulateur d'écho-endoscopie digestive.....39

2.2.3. Spic : simulateur de cœlioscopie gynécologique.....41

2.2.4. Les autres simulateurs en projet.....43

2.2.5. Bilan des projets.....44

2.3. La problématique des simulateurs 44

2.3.1. Construction des modèles 44

2.3.2. Réalisme des images 45

2.3.3. Simulation dynamique 45

2.3.4. Protocole d'enseignement 46

Conclusion 46

Partie 2. Modèles mécaniques d'organes

Chapitre 3. Modèles de corps déformables 51

Introduction 51

3.1. Concepts fondamentaux en mécanique 52

3.1.1. Le formalisme de la mécanique 52

3.1.2. La mécanique du solide 53

3.1.3. Quelques définitions 53

3.2. Modélisation de corps déformables 54

3.2.1. Les modèles continus à discrétisation de volume 54

3.2.2. Les modèles à nombre de déformations fini 59

3.2.3. Les modèles discrets 62

3.2.4. Modèles ad-hoc 66

3.3. Les problèmes communs 68

3.3.1. Modèles dynamiques/statiques 68

3.3.2. Méthodes d'intégration 69

3.3.3. La dualité modèle géométrique/modèle mécanique 70

3.3.4. Contrôle de la déformation 72

3.4. Modèles pour la simulation d'organes humains 73

3.4.1. Modèles continus 73

3.4.2. Modèles discrets 75

3.4.3. Bilan 76

Conclusion 76

Chapitre 4. Prise en compte des interactions 77

Introduction 77

4.1. Définition et détection de la collision 77

4.1.1. Définition des collisions 77

4.1.2. Détection des collisions 79

4.1.3. Conclusion générale sur la détection de collisions 87

4.2. Accélération de la détection de collisions 88

4.2.1. Elimination par subdivision de l'espace 89

4.2.2. Elimination par zones d'influence 90

4.2.3. Elimination par critères topologiques 95

4.2.4. Elimination par critères cinématiques 97

4.2.5. Conclusion 99

4.3. Réponse à l'interaction 99

4.3.1. Le point de vue du physicien 99

4.3.2. Corps rigides 100

4.3.3. Corps déformables 101

4.3.4. Modèles d'interactions dépendant du modèle de corps déformables 103

4.3.5. Les problèmes spécifiques des contacts 104

4.3.6. conclusion 107

4.4. Les découpes et déchirements 107

4.4.1. Modélisation de fractures 107

4.4.2. Modélisation de fusion 108

4.4.3. Bilan 108

Conclusion 108

Partie 3. Notre proposition

Chapitre 5. Proposition de modèle déformable avec interactions 115

Introduction 115

5.1. Hypothèses préliminaires 115

5.1.1. Schéma de la simulation 115

5.1.2. Hypothèses 116

5.2. Modèle à ressorts avec composante rigide 116

5.2.1. Pourquoi un modèle masses/ressorts ? 116

5.2.2. Construction du modèle 117

5.2.3. Comportement du modèle 117

5.2.4. Intégration des équations 118

5.2.5. Analyse du modèle 119

5.2.6. Paramétrisation du modèle 120

5.3. Traitement des interactions 121

5.3.1. Accélération de la détection 121

5.3.2. Détection des collisions 123

5.3.3. La réponse à la collision 124

5.4. Traitement des découpes et déchirements 128

Conclusion 130

Chapitre 6. Mise en œuvre 131

Introduction 131

6.1. Détails d'implantation 131

6.1.1. Environnement choisi 131

6.1.2. Optimisation 131

6.1.3. Précision des calculs numériques 132

6.2. Exemple de situations simulées 132

6.2.1. Langage de description 132

6.2.2. Etude d'une situation simple : un objet en chute libre 133

6.2.3. Etude des frottements : cas du corps sur un plan incliné 134

6.2.4. Etude fine des collisions 134

6.2.5. Un cas simple de manipulation 134

6.3. Mesures des performances 136

6.3.1. Influence du nombre de points et de liaisons 136

6.3.2. Etude du temps de calcul dédié aux collisions 137

6.3.3. Bilan 137

6.4. Applications du modèle : comparaison avec les éléments finis.....138

6.5. Perspectives 138

Conclusion 139

Chapitre 7. Conclusion - Discussion.....141

De l'avenir des simulateurs médicaux... 142

Annexe I. Notations mathématiques 143

Annexe II. Les quaternions 144

II.I. Définition 144

II.I.I. Structure d'algèbre 144

II.I.II. Norme du quaternion 144

II.II. Représentation des rotations par des quaternions 145

II.II.I. Quaternion de rotations 145

II.II.II. Equivalence entre matrices et quaternions 145

Annexe III. Les formats de fichiers employés 147

III.I. Fichier SCN 147

III.II. Fichier OJT 148

Bibliographie 149

Introduction

La synthèse d'image est un domaine qui a pour objectif la production d'images à partir d'une description mathématique d'un environnement que l'on appelle *scène*. Lorsque l'on veut calculer une image (on dit effectuer un *rendu* de la scène), il faut choisir un compromis entre d'une part le réalisme et d'autre part la rapidité du calcul de cette image. Ainsi, on considère généralement deux types de *rendu* : le rendu projectif et le lancer de rayons.

Le rendu projectif est une technique produisant des images de qualité faible ou moyenne, mais en un temps très court. C'est un processus qui est câblé dans les machines dédiées aux images 3D. Au contraire, le lancer de rayons est une technique très coûteuse en temps de calcul, mais qui procure des images de très haute qualité. Ces deux techniques peuvent être améliorées en calculant la radiosité qui consiste à évaluer un éclairage global de la scène. Ce procédé est également très coûteux en temps de calcul mais permet d'obtenir des images d'un réalisme remarquable.

Lorsque l'on sait calculer une image fixe, il est intéressant de réitérer le calcul, afin de produire une suite d'images qui, affichées successivement dans le temps, produisent une animation. Dans ce cas, la méthode de rendu employée dépend de l'application. Si l'animation peut être calculée avant sa diffusion devant un public, on utilisera plutôt les méthodes de rendu lentes, mais l'animation obtenue sera de grande qualité. Ainsi, les animations de synthèse utilisées au cinéma ou à la télévision ont généralement nécessité de nombreuses heures voire jours de calcul. Au contraire, on choisira de préférence le rendu géométrique pour afficher des images de façon interactive. La *Réalité Virtuelle*[1] est une application de l'animation temps-réel.

Nos travaux se placent dans le cadre de l'animation temps-réel appliquée au domaine particulier de la médecine. Plus précisément, le contexte est celui de la formation médicale. En effet, la médecine est de nos jours confrontée à l'arrivée rapide de nouvelles techniques. D'une part, l'imagerie médicale a fait d'énormes progrès, les processus d'acquisition regroupent maintenant des techniques aussi diverses que le scanner, la résonance magnétique nucléaire, l'échographie... L'interprétation de ces images n'est pas immédiate et demande un apprentissage spécifique. D'autre part, les techniques opératoires évoluent très rapidement. L'émergence de nouvelles techniques s'accompagne d'un besoin important en formation initiale et continue. Par exemple, les techniques endoscopiques permettent d'effectuer des interventions avec une invasion minimale.

1. Dans ce document, le terme *Réalité Virtuelle* sera peu usité au profit de l'expression plus précise et moins ambiguë d'*Environnements Virtuels*.

Pourquoi l'enseignement de la médecine ne pourrait-il pas bénéficier des progrès informatiques afin d'améliorer la formation des médecins ? A l'instar de la formation des aviateurs, pourquoi ne pas employer des simulateurs qui permettraient aux chirurgiens confirmés ou aux internes de se former aux nouvelles techniques ? Actuellement, l'apprentissage se fait sur patient réel. Pourquoi ne pas essayer d'opérer sur des patients virtuels ?

Nos travaux sont une contribution à la conception et à la réalisation des simulateurs chirurgicaux à but pédagogique. Cependant, par rapport aux classiques simulateurs de vol, le domaine médical nécessite en plus la simulation des interactions mécaniques avec des corps mous pour modéliser les actions des instruments sur les organes. Nous avons donc étudié dans cette thèse comment modéliser le comportement mécanique des organes, comment simuler ce comportement en temps réel. En outre, nous nous sommes intéressés aux interactions entre les instruments et les organes. Enfin, nous cherchons également à simuler les découpes et déchirements des organes. Notre objectif est de reproduire une intervention de façon réaliste et plausible.

Ce document se présente sous la forme de trois parties. La première présente le concept des simulateurs chirurgicaux, en les différenciant d'autres applications, et en dégageant leurs points forts et ce qu'ils apportent. La présentation des recherches faites actuellement dans le monde, sera suivie par l'exposé des projets en cours au laboratoire. Ces projets se sont concrétisés par la création à Lille d'un Groupement d'Intérêt Scientifique. La deuxième partie se focalise sur les problèmes de simulation dynamique, suivant deux directions : d'une part la simulation dynamique proprement dite, à savoir la simulation du comportement d'un corps déformable, en fonction des lois physiques ; d'autre part, la prise en compte des interactions, contacts prolongés ou chocs entre corps. Pour chacun de ces sujets, nous étudions l'adaptabilité à notre projet des solutions qui ont déjà été proposées. Dans la troisième partie, nous exposons notre proposition de modèle pour la simulation à la fois du comportement et des interactions, puis sa mise en œuvre. Cette application nous permet de tirer les avantages à conserver et les améliorations à apporter au modèle. Nous concluons enfin sur les perspectives en recherche des simulateurs médicaux et leur avenir.

Dans cet ouvrage, un certain nombre de notations mathématiques ou physiques sont employées. Leur signication a été regroupée dans l'*Annexe I.*

Partie 1

Les simulateurs médicaux

Théorie de la “périodisation” : après l’époque des cultures où l’Homme explique le monde grâce aux Mythes, puis la civilisation de l’Ecrit dans laquelle c’est par la Théorie qu’il essaie de connaître le monde, l’Homme, arrivé à l’époque de l’informatique, vivrait l’émergence d’un nouveau mode de connaissance : la Simulation.

Pierre Lévy, Les technologies de l’intelligence,
La découverte, 1990.

Introduction

Dans cette première partie, nous décrivons les applications médicales liées aux environnements virtuels. On classe ces applications en trois domaines : la simulation médicale, l'aide à la chirurgie et la télémédecine. Nos travaux sont effectués dans le cadre de la simulation médicale à but pédagogique. Le premier chapitre situe les travaux du laboratoire parmi toutes les applications existantes ou futures, et dresse les implications et extensions possibles. Une fois notre problématique cernée, nous cherchons à dresser une liste, la plus représentative possible, des études déjà réalisées de par le monde sur le thème simulateurs médicaux à but pédagogique. Cette classification nous permet de définir les différentes étapes à franchir dans l'étude et la réalisation de tels simulateurs. Dans le second chapitre, nous présentons les diverses équipes constituant le Groupement d'Intérêt Scientifique dont l'objectif est l'étude des simulateurs médicaux. Nous exposons en détails les projets en cours ce qui nous permet de mettre en évidence les problématiques entrant dans l'élaboration des simulateurs médicaux. L'étude de l'un de ces projets a fait l'objet d'une partie de cette thèse.

L'ensemble de ces chapitres garde une orientation médicale. Seules les diverses mises en œuvre de simulateurs sont étudiées plus techniquement. Nous essayons de mettre à jour les difficultés théoriques et technologiques qui font de la conception des simulateurs médicaux un des défis actuels de la recherche.

1 Les Environnements Virtuels en Médecine

Introduction

Aujourd'hui encore, il est très difficile d'établir une liste exhaustive des domaines concernés par les environnements virtuels. Les applications foisonnent et d'autres ne sont même pas encore soupçonnées. Certaines cependant semblent être suffisamment bien avancées pour avoir, dès maintenant, des répercussions pratiques. Hormis le marché des jeux vidéos et des loisirs en général (qui est sûrement le plus porteur), il est d'autres domaines dans lesquels la *Réalité Virtuelle* a d'ores et déjà acquis ses lettres de noblesse. En architecture, on est capable d'évoluer dans un édifice avant même sa construction. Les problèmes pratiques d'agencement des pièces ou d'esthétisme peuvent être décelés dès la conception. Cette utilisation atteint son paroxysme lorsque tout un lieu public, une gare par exemple, est modélisée afin d'étudier les flux de voyageurs, la signalisation des diverses directions, voire même, la position des affiches publicitaires. Enfin, une autre application pratique est la reconstruction de sites historiques détruits, comme le temple de Karnak, ou difficilement accessibles, comme la grotte Cosquer. Par ailleurs, un autre domaine en effervescence est celui de la communication : par exemple, des personnes non présentes physiquement dans le même site, vont pouvoir dialoguer comme si elles se trouvaient dans une seule et unique salle de réunion. L'enseignement est peut-être, de façon un peu cachée, la plus ancienne activité exploitant les environnements virtuels. Elle reste cependant en pleine évolution. Les environnements virtuels à but pédagogique ont en effet débuté avec les simulateurs de vols, suivis plusieurs années après par les simulateurs de conduite. De nos jours, de nouveaux champs sont susceptibles d'être bouleversés par des formations de ce type : on parle de simulateurs dans la discipline sportive par exemple ou dans le domaine médical. C'est ce dernier secteur, que l'on appelle les simulateurs d'instruments médicaux à but pédagogique, qui nous intéresse.

La médecine ne bénéficie que depuis peu des apports des environnements virtuels. Plusieurs applications bien spécifiques ont été peu à peu mises à jour : l'entraînement avant intervention, l'apprentissage, la réalité augmentée, la téléopération... La distinction entre ces applications n'est malheureusement pas toujours évidente et il y a souvent confusion, ce qui nous mène à présenter et catégoriser, dans la première partie de ce chapitre, tous les aspects des environnements virtuels pour la médecine. Nous positionnons alors nos travaux sur les simulateurs médicaux à but pédagogique. Dans une seconde partie, nous tenterons de justifier l'intérêt que présentent de tels simulateurs. Enfin, nous dresserons une classification des travaux les plus révélateurs qui ont été ou sont réalisés à travers le monde pour la simulation médicale.

1.1. Applications des environnements virtuels en médecine

Les recherches au LIFL concernent les simulateurs médicaux à but pédagogique. Plusieurs autres domaines de recherche en médecine sont connexes à cette activité, et ces applications ne sont pas souvent clairement distinguées, voire, dans la littérature, sont amalgamées. Pourtant, les problèmes d'ordre technique ou éthique sont loin d'être identiques. Il convient de ce fait de clas-

sifier les applications des environnements virtuels en médecine, de préciser leurs contraintes et les questions qu'elles soulèvent. Nous choisissons une classification plus large que celle déjà proposée par [Sat95], en considérant trois points : la simulation, l'aide en cours d'opération et la télémédecine. La simulation comprend les applications précédant l'opération, à savoir l'apprentissage et la planification d'opérations. Au contraire, la réalité augmentée ainsi que la robotique assistent le chirurgien durant l'intervention. Enfin, la télémédecine regroupe la collaboration entre médecins à distance pour établir un diagnostic, ainsi que la téléopération qui fournit les outils nécessaires à un chirurgien pour opérer un patient se trouvant sur un site distant. Dans nos propos, nous traiterons peu de robotique, qui déborde de notre sujet, et de collaboration entre médecins qui relève plutôt des études en travail coopératif. Une fois les autres applications détaillées, nous pourrons situer la place de notre recherche.

1.1.1. Les applications

a) Apprentissage par Simulation

Dans ce type d'applications, on cherche à modéliser fidèlement un appareil afin de procurer un outil d'apprentissage d'un acte difficile à accomplir. Les simulateurs de vol sont les premiers exemples de tels concepts. L'utilité de tels outils n'est plus à remettre en cause et ils ont été généralisés à d'autres type de simulation de conduite comme les automobiles, les chars ou encore les trains. Initialement, l'avantage central était la réduction des coûts. Le risque de perte de l'appareil par un élève inexpérimenté est supprimé et le véritable appareil n'a pas à être monopolisé pour la formation. De plus, la variété des décors et situations ainsi que des problèmes simulés en cours de conduite ont rendu l'outil indispensable dans la formation des aviateurs et bientôt des conducteurs.

Dans le domaine médical, les atouts des simulateurs sont les mêmes. L'idée clef est de remplacer le patient par un modèle informatique le reproduisant le plus exactement possible. Le simulateur devra être très réaliste afin que le passage du simulateur au cas réel se fasse dans les meilleures conditions. L'idéal étant, qu'à l'issue de sa formation sur simulateur, l'élève soit compétent pour opérer des cas réels sans l'aide d'un expert.

Le simulateur ne joue pleinement son rôle que s'il est le plus fidèle possible. Il faut en effet réduire l'effet d'*adaptation* de l'homme vis à vis de la machine. Dans notre cas, c'est la machine qui s'adapte à la situation. Donnons un exemple concret. Si un système a une certaine latence, le cerveau, au bout d'un certain temps d'utilisation, va s'adapter à cette latence et aura même tendance à anticiper en fonction de ce retard. Si un étudiant apprend sur un tel simulateur, il apprendra les gestes en fonction de la latence. Son apprentissage devra alors s'adapter lorsqu'il passera au cas réel avec tous les dangers que cela comporte dans les opérations *in vivo*. Le simulateur se doit ainsi de respecter strictement l'interface d'utilisation habituelle, aussi complexe et peu intuitive soit-elle.

Il existe de nombreux exemples de simulateurs médicaux, que nous détaillerons plus loin dans ce chapitre, au paragraphe *1.3* où l'on dressera une classification selon leur technologie et leur but.

Les problèmes posés sont ici aussi d'ordre technique (le modèle mécanique, le réalisme des images et la présentation à l'utilisateur, les capteurs et les systèmes mécaniques). On cherche des modèles réalistes et génériques, qui semblent se comporter normalement, sans forcément respecter exactement l'organe d'origine.

D'autres problèmes concernent la mise en œuvre des protocoles pédagogiques. Il s'agit de concevoir les exercices de difficulté croissante, de fournir des vues annexes ou des indicateurs que l'on n'a pas forcément durant l'opération réelle, mais qui permettent une bonne compréhension de l'opération et du geste thérapeutique.

Enfin, les questions d'ordre médical sont aussi à prendre en compte. Il s'agit par exemple de modifier le protocole de formation afin de prendre en compte la formation par simulateur. Voire même rendre l'apprentissage par simulateur une étape indispensable et obligatoire dans la forma-

tion. Il faut alors trouver des critères permettant de juger l'acte thérapeutique. Malheureusement, les méthodes thérapeutiques diffèrent d'une école à l'autre. Le simulateur ne peut donc pas se permettre de donner une appréciation systématique.

b) Planification d'opération

La planification d'opération est une voie de recherche très prometteuse. Historiquement, le principe de base s'est particulièrement illustré dans un tout autre contexte, lors de la Guerre du Golfe. En effet, lors d'opérations militaires d'envergure, les pilotes étaient entraînés par des simulateurs. Les pilotes avaient exécuté des simulations de ces missions plusieurs fois avant de les exécuter en "grandeur réelle". Les missions ont été effectuées quasiment parfaitement, car sans surprise. Les succès militaires montrent qu'une préparation dans des conditions les plus proches de la réalité conduisent à un meilleur déroulement de la mission proprement dite. Il est donc intéressant d'étendre ces résultats au domaine médical. Il existe en effet des opérations très délicates où un organe voire même la vie du patient sont en jeu et qui dépendent fortement des caractéristiques du patient. Le chirurgien a donc également tout intérêt à s'entraîner à opérer sur un modèle de son patient avant la véritable intervention. Par analogie avec le domaine militaire, on parlera de *préparation de mission*.

Des équipes se sont par exemple, intéressées à la simulation d'accouchement [BG93a][BG93b] : un modèle du fœtus et du bassin sont réalisés afin de déterminer si l'accouchement est possible sans césarienne. La même équipe propose, sur le même principe de construire l'arborescence des bronches afin de pouvoir déterminer le chemin qu'un endoscope devra emprunter pour atteindre une tumeur, lors d'une *bronchoscopie* [GK95]. Dans un autre exemple, [DSCP94] vont essayer de prédire quelles modifications du squelette il faut effectuer lors d'une intervention de chirurgie crânofaciale. Des prédictions sur le visage du patient après l'opération sont effectuées. Cependant, ces trois exemples n'incorporent pas l'environnement de l'opération, et de ce fait, ne sont pas des simulations de l'opération proprement dite.

La mise en œuvre soulève de nombreux problèmes. Afin de remplir complètement son rôle de préparation, le modèle utilisé doit être très proche de la réalité et se comporter de façon strictement identique. Cela signifie qu'il faut établir un modèle de l'opération très puissant, à la fois pour la restitution visuelle et tactile, voire sonore. Ce modèle est obtenu par reconstruction à partir de données mesurables de l'objet à modéliser. Se posent alors les problèmes de mesures précises (certaines données mécaniques par exemple, sont difficiles ou impossibles à mesurer, et ne peuvent qu'être approchées par un modèle intermédiaire). La forme des organes peut être obtenue par Scanner et/ou RMN suivant leur composition. Ensuite, lors de la simulation, des contraintes techniques de temps de calcul, de fréquence et latence de capteurs, de qualité du rendu, des calculs de la mécanique... rendent la mise en œuvre délicate.

c) Réalité augmentée pour la chirurgie

Lors de certaines opérations, le chirurgien n'a accès qu'à une partie des informations de son champ opératoire. Par exemple, certaines parties du corps où a lieu l'opération sont cachées, ou ne sont pas accessibles directement au chirurgien. Dans ce cas, l'expérience et l'adresse du médecin sont primordiales car elles auront pour tâche d'extrapoler les informations cachées à l'aide des informations connues. Pourtant, il existe des procédés d'acquisition de données volumiques (Caméra miniature, RMN, Scanner...) qui savent s'affranchir des obstacles. Certaines données sont exploitables directement lors de l'opération. C'est le cas de l'endoscopie où une caméra est introduite dans le champ opératoire et va permettre au chirurgien d'observer dans de bonnes conditions l'opération. C'est également le cas de l'échographie. Ces techniques nécessitent un apprentissage afin d'être maîtrisées et devenir pour le chirurgien des outils extrêmement puissants.

Si les données ne sont pas exploitables durant l'opération, il va falloir les acquérir et les communiquer lors de l'opération d'une façon la plus intuitive possible. La robotique peut fournir une solution à ce problème. [CDTL92] proposent une démarche constructive. Leur système *IGOR* est basé sur une méthodologie très rigoureuse. Tout d'abord la phase de *Perception*, où des données

acquises avant l'opération (modèles anatomiques, imagerie 3D comme le scanner, la RMN et la tomographie) et pendant l'opération (échographie, radiographie, vision, aspect tactile...) sont mises en corrélation. Ces données alimentent l'étape suivante, la *Décision*, où l'on va chercher une stratégie chirurgicale. La dernière étape est l'*Action*, où l'on va exécuter la stratégie définie préalablement. L'intervention peut alors être soit passive (le chirurgien met en corrélation lui même la stratégie et les gestes), soit semi-active (la stratégie est indiquée par des repères aux chirurgien), soit active (un robot accomplit l'opération en suivant la stratégie). Ce système a été mis en pratique pour des opérations de *neurochirurgie stéréotaxique* (prélèvement de tumeur au cerveau), de chirurgie de la moelle épinière et enfin pour la *radiothérapie* (par exemple, destruction des tumeurs par irradiation). Un exemple de mise en œuvre du système IGOR dans un contexte de neurochirurgie stéréotaxique consiste à employer un laser indiquant la direction dans laquelle il faut introduire une aiguille. A l'aide de repères, le chirurgien peut déterminer si l'aiguille pointe dans la même direction que le rayon laser.

Lorsque la robotique seule ne suffit pas, les techniques de réalité augmentée apportent d'autres solutions intéressantes. Dans ce cas, on superpose à l'image que voit le médecin, des informations sous forme de schémas, afin qu'il puisse appréhender la position de ce qu'il ne voit pas. Un exemple très intéressant est celui de [PTE95] : il traite le cas des opérations sur la colonne vertébrale (*scolioses*). Lors de cette opération, le chirurgien doit passer un outil dans un orifice qu'il ne voit pas. Cette manœuvre est délicate, mais devient d'une relative simplicité sitôt que l'on superpose un tracé en fil de fer à l'image réelle. La partie cachée de la vertèbre à opérer devient alors visible grâce à cet affichage.

Enfin, une mise en œuvre dans des conditions réelles a été proposée par [BFO92]. Le dispositif a pour but de montrer la position d'un fœtus dans le ventre de sa mère. Il consiste d'abord en une sonde qui acquiert les données ultra-son. Ces données sont reconstituées afin d'obtenir un modèle 3D. A partir de ce modèle, une image est synthétisée et mixée avec l'image réelle. Les positions de l'observateur et de la sonde sont mesurées par des Trackers 6D Polhemus.

Les problèmes soulevés par ces techniques de réalité augmentée pour l'assistance à l'opération ne sont pas l'apanage de la médecine. En effet, les techniques de réalité augmentée exigent dans tous les cas de savoir modéliser de façon très précise l'objet que l'on veut synthétiser, et l'environnement dans lequel on le synthétise. Ensuite, il faut savoir situer très exactement, dans l'espace, l'image réelle afin de positionner correctement l'image synthétisée. C'est l'étape de *calibrage* des capteurs et de recalage de l'image réelle et des objets 3D virtuels. Il faut que l'objet synthétique ajouté s'insère correctement dans l'environnement réel. Cela signifie qu'il faut que l'objet virtuel s'affiche derrière les objets réels situés entre l'observateur et lui, et devant les objets situés derrière lui. Il faut donc connaître les profondeurs à la fois des objets virtuels et des objets réels. Une autre illustration de ce problème vient de l'interprétation que fait le cerveau des images reçues. Par exemple, si on observe un objet virtuel sensé se trouver à l'intérieur d'un objet réel, il faudrait, pour être parfaitement naturel, que l'objet réel devienne transparent. Sans cet artifice, l'objet virtuel semble se trouver devant, ou pire, plaqué sur l'objet réel.

d) Téléopération

L'idée de base de cette technique est de séparer le lieu géographique du chirurgien et du patient. Certaines opérations très délicates ne peuvent être effectuées que par quelques experts dans le monde. Actuellement, l'expert doit se déplacer sur le lieu de l'opération, ce qui n'est envisageable que si l'opération a été prévue suffisamment à l'avance. Dans les cas d'urgence, un chirurgien non habitué à ce genre d'interventions doit intervenir, certes, avec l'appui relatif de l'expert (de façon optimale, avec au moins un échange visuel entre les deux). En cas d'incident, l'expert ne peut que constater les erreurs et les interpréter, il lui est impossible d'intervenir pour "rattraper" la situation. L'application la plus importante de la téléopération concerne les interventions sur les sites à risques tels que les champs de bataille ou les lieux irradiés. Ces recherches sont menées principalement par l'armée.

Dans le cas idéal, il faudrait que le chirurgien opère sur un modèle du patient et que ses agissements soient transmis sur le site distant où se trouve le patient, à un robot qui reproduira préci-

sément tous ses gestes. Le modèle utilisé se doit de retranscrire fidèlement l'aspect visuel et tactile de l'objet. L'ensemble du système est décrit sur la *figure 1.1*. Le chirurgien accomplit l'acte chirurgical sur le modèle. Les informations concernant cet acte sont transmises, via un support de transmission quelconque, vers le robot qui reproduira, avec si besoin, un facteur de réduction, les gestes du chirurgien. Pour le retour d'information, deux choix sont possibles. Soit la résistance mécanique à la manipulation est mesurée et retransmise au modèle, soit ce dernier est suffisamment complet pour pouvoir déterminer par calcul cette résistance.

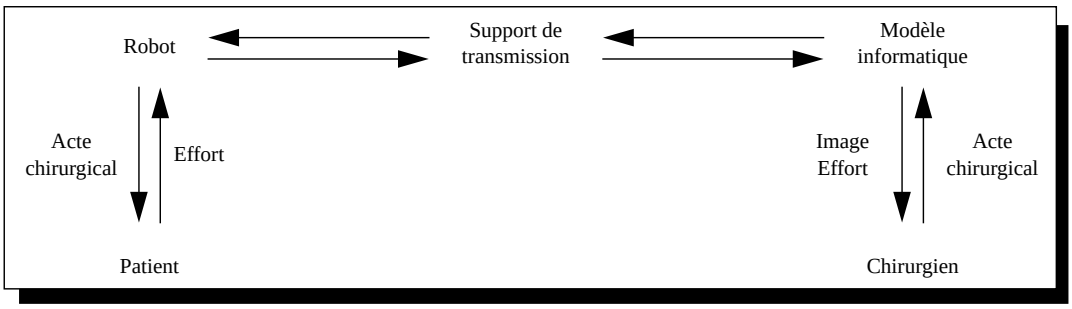

Figure 1.1. *Principe de la téléopération*

Les problèmes concernent la réalisation technique, la fabrication du modèle, les modes de transmission entre les deux sites et la réalisation du robot, mais aussi des questions éthiques : que faire dans le cas d'une rupture de la liaison ? De plus, on sait qu'il est traumatisant pour un patient de se faire opérer par un robot commandé par un chirurgien. Que penser alors d'un robot commandé par un ordinateur ?

e) Autres applications

Les applications citées ci-dessus sont les exemples les plus marquant des environnements virtuels appliqués à la médecine. Cependant, cette liste n'est pas exhaustive et d'autres applications sont possibles. On peut même dire qu'on n'a pas encore déterminé tout ce que les mondes virtuels peuvent apporter à la thérapie par exemple. Une proposition originale est d'utiliser les mondes virtuels comme anti-stress lors d'une intervention. Un exemple pratique est la perturbation amenée par le stress d'un patient lorsque son cerveau est soumis à un examen d'activité. Actuellement, on émet une musique sensée apaiser le patient. Grâce aux mondes virtuels, on pourrait créer un environnement totalement différent de la salle d'opération et permettant au patient de se sentir plus en sécurité. En poussant encore plus loin l'application, certains médecins pensent même à employer les mondes virtuels pour soigner certains problèmes mentaux, notamment chez les personnes introverties. La réhabilitation ou la rééducation sont également envisageables avec ces techniques.

Par ailleurs, [KHDS95] considèrent une salle d'opération complètement modélisée en environnements virtuels. Le but de cette modélisation est de dessiner la salle d'opération du futur capable d'intégrer tous les modes d'opérations, acquisition de données, entraînement à la chirurgie, téléopération, circulation de l'information...

Enfin, une idée intéressante, et plus applicable en l'état actuel de la technologie, repose sur l'utilisation des environnements virtuels à des fins d'amélioration ou d'optimisation des techniques chirurgicales. [SBM94] proposent d'étendre leur simulateur à cette fin.

1.1.2. Les liens entre ces applications

On a présenté le domaine par applications, mais on peut mettre en évidence les problèmes communs :

- Il faut fabriquer un modèle des organes, des outils, voire, de l'environnement de l'opération. Ce modèle est obtenu soit par un modeleur quand seul un aspect réaliste suffit ou par reconstruction à partir de données mesurées. Cette dernière étape est délicate à mettre en œuvre,

requiert du traitement du signal pour isoler les données et éliminer le bruit. Lorsque l'on cherche ensuite à superposer des données venant de sources différentes, il faut procéder à l'étape de *registration*, où on va chercher à mettre en corrélation les diverses sources d'information. La visualisation concerne aussi la façon dont l'objet est vu, d'où le problème de l'acquisition ou de simulation des textures. Construire un modèle peut aussi concerner l'aspect mécanique du modèle d'où la mise en œuvre de protocoles de mesure des données mécaniques ;

- Les modes d'affichage de l'environnement virtuel sont à étudier. Il semble que le port d'un visiocasque ne soit pas bien accepté par les chirurgiens car il modifie l'espace habituel du travail et est assez contraignant à porter. On sait de plus que l'aspect 3D procuré par les techniques d'image stéréoscopique n'est pas habituel et fatigue le cerveau en quelques minutes lors de l'interprétation de l'image qu'il reçoit. Cela devient particulièrement dramatique lorsque le chirurgien doit opérer plusieurs heures. Il semble donc qu'il faille à tout prix limiter l'utilisation du casque et de s'en passer sitôt que cela est possible ;
- Dès qu'il y a simulation de l'opération, que ce soit dans la téléopération, la planification d'opération ou la simulation pédagogique, et que des interactions mécaniques peuvent se produire, il faut des dispositifs permettant au manipulateur de ressentir la résistance mécanique lors de l'interaction. Cette simulation est essentielle car le chirurgien qui opère utilise, sans même s'en rendre compte, ce *retour d'effort* pour interpréter ses propres gestes : appuie-t-il trop sur l'organe ? risque-t-il de découper trop profondément ? etc.

En pratique les applications que nous avons présentées peuvent être regroupées dans un même système. Par exemple, il existe des liens très étroits entre une machine de simulation à but pédagogique et une autre permettant la préparation de mission. Le noyau est strictement le même, la machine a dans les deux cas pour but de simuler une opération. Les seules véritables différences se situent tout d'abord au niveau des données en entrée. La préparation de mission exige une représentation tout à fait fidèle du patient que l'on souhaite opérer, tandis que le simulateur a besoin, lui, de cas peut-être un peu moins fidèles, mais offrant un vaste choix allant des pathologies faciles à traiter et/ou opérer jusqu'à des cas plus subtils. Ensuite la sortie est différente puisque d'une part, c'est le chirurgien qui appréciera son geste, tandis que dans l'autre cas, c'est l'ordinateur qui donnera des critères d'appréciation.

Par ailleurs, il est naturel que tout système de réalité augmentée employé lors d'une opération soit reproduit dans un système de simulation de cette opération. Inversement, si lors d'une simulation, des informations supplémentaires sont fournies à des fins pédagogiques, il est intéressant de chercher à obtenir le même type d'informations en temps réel lors d'une véritable opération. Le problème vient alors de l'acquisition en temps réel et du recalage de ces informations sur les images réelles.

En conclusion, on se rend compte que la simulation est un pré-requis de toutes ces applications. En effet, la préparation de mission, la réalité augmentée et la téléopération nécessitent en étapes préalables de savoir simuler l'opération, d'obtenir un modèle complet des organes, des instruments et de l'opération. Par contre, on ne peut pas douter qu'une fois toutes les applications complètement maîtrisées, on puisse concevoir un système unique pour l'apprentissage, la préparation, la réalité augmentée et la téléopération.

1.2. Apports des simulateurs médicaux

Le but de nos travaux étant maintenant délimité, nous étudions en détail les motivations et apports des simulateurs médicaux.

1.2.1. Motivations

Les simulateurs médicaux n'offrent pas uniquement les apports des simulateurs dans les autres domaines d'apprentissage, mais bien d'autres. Dans la méthode actuelle, l'élève chirurgien est formé en observant son professeur puis en essayant de ré-exécuter les gestes. C'est la technique du *compagnonnage*, qui auparavant était la seule possibilité d'apprentissage, mais qui se révèle très limitée. Par exemple, l'élève et le professeur ne voient pas forcément la même chose d'où des difficultés dans l'échange des informations lors de la démonstration. De plus, on sait parfaitement que, lors d'une démonstration, seule une partie du savoir est effectivement transmise du professeur vers l'élève. Ce n'est que lors de la première exécution de l'acte que l'élève devra se poser toutes les questions au sujet de la manipulation ou de la thérapie à appliquer. C'est malheureusement, cette étape cruciale qui se fera sur le patient, ce qui augmente encore la difficulté de l'intervention. On cherche donc à moderniser le processus de formation en faisant bénéficier l'enseignement des apports de la technologie.

Une autre motivation pour la construction de tels simulateurs vient justement des progrès techniques. Aujourd'hui, les pratiques médicales évoluent à une vitesse telle qu'il devient difficile pour les chirurgiens de se maintenir à niveau. On passe de plus en plus vite du stade de l'essai clinique à l'usage courant. Ceci n'est pas sans entraîner des risques pour les patients. Par exemple, une toute nouvelle technique, la *cœlioscopie* (laparoscopie), a pour but de réduire l'invasion dans le corps opéré en ne pratiquant qu'une petite incision là où l'on va introduire les outils. Cette technique est sensée réduire la morbidité de l'opération. Cependant, si l'endoscope est mal manipulé, il peut entraîner des dégâts très important. C'est ce qui est arrivé aux Etats Unis par exemple, chez les patients de médecins peu entraînés. L'évolution des méthodes entraîne un accroissement des besoins et donc, d'un point de vue économique, une forte demande en formation.

1.2.2. Avantages des simulateurs

On choisit de classer ces avantages en diverses catégories suivant les personnes à qui elles s'appliquent.

- Apports pour le patient

Avec un tel simulateur, l'acte thérapeutique est séparé de l'acte d'apprentissage. La formation ne se fait plus avec le patient mais sur un modèle. Il n'y a donc plus de risques pour le patient ; la notion de *cobaye* prend un sens plus restreint. De plus, généralement, lors de ses premières manipulation, un étudiant va mettre plus de temps que nécessaire à accomplir l'opération : le patient doit donc supporter également la lenteur de l'exécution. Avec un simulateur, ce n'est que lorsque l'élève a eu son entraînement sur simulateur validé qu'ils pourra tenter des opérations *in vivo*. De plus, une sorte de garantie peut s'instaurer ; on pourrait alors parler d'une *qualification* avec simulateur.

- Apports pour l'élève

Les avantages des simulateurs sont ici multiples. D'une part, l'élève dispose d'un outil qu'il peut utiliser seul, lui permettant de se perfectionner quand il le souhaite. Ainsi, une étape difficile pourra être retravaillée maintes et maintes fois jusqu'à une parfaite maîtrise. Pour lui, un enseignement par difficulté croissante est possible, alors qu'auparavant, ses essais étaient conditionnés par la disponibilité des cas. Ici, sous réserves d'avoir une base de cas complète, n'importe quelles pathologies, y compris des cas rares sont possibles. L'élève a la garantie de rencontrer tous les cas et donc, une fois dans la vie active, de savoir les traiter.

- Apport pour le professeur

Un tel outil fournit au professeur des critères permettant de cerner exactement le niveau de son ou ses élèves. Les élèves sont évalués par des mesures objectives. Si, de plus, un logiciel d'EAO est également fourni, le professeur pourra administrer une base de données de ses étudiants et connaître leur niveau à n'importe quel moment, savoir s'il a progressé, et à quel rythme...

Un autre avantage, et non des moindres, est que la simulation et l'évaluation peuvent avoir lieu à des moments différents. Il suffit de garder une trace de l'acte, par exemple sous forme d'un film. Ceci implique que le professeur n'a pas besoin d'être présent lors de l'exercice. Cela se révèle particulièrement intéressant lorsque l'étudiant maîtrise encore mal la technique et qu'il met trois ou quatre fois plus de temps à accomplir l'acte thérapeutique qu'une personne expérimentée. Enfin, on ne remplace pas le formateur puisque c'est toujours à lui qu'incombe la validation de l'opération. Tout est mis en place afin qu'il puisse s'approprier l'outil.

- Apport pour l'administration hospitalière

L'usage d'un tel simulateur est intéressant pour l'hôpital s'il est d'un coût moindre que celui de l'outil simulé. En effet, lors de la formation, l'élève utilise le simulateur et ne monopolise plus l'outil, pendant ce temps, ce dernier peut être employé pour les soins des malades. Par ailleurs, par une évolution culturelle, il est devenu délicat de nos jours, et sûrement bientôt impossible, d'utiliser l'animal dans les pratiques médicales. Comme il est difficile de s'en passer pour les tests de médicaments pharmaceutiques, il s'agit donc de limiter son recours dans le cas de la formation. On peut tout à fait envisager que grâce aux simulateurs, le stade d'apprentissage sur animal soit supprimé. Ceci est essentiel dans des pays comme la Grande Bretagne où l'expérimentation animale est interdite. Enfin, d'un point de vue juridique, les simulateurs pourraient fournir une sorte de couverture en cas d'échec d'une intervention chirurgicale, pour deux raisons. Tout d'abord, on souhaite que, grâce à de tels outils, les médecins soient mieux formés et surtout parés à toute difficulté. La seconde garantie serait légale : si un médecin n'a pas l'obligation légale de résultats, il a l'obligation de moyens. Un médecin ayant eu recours à la formation par simulateur aurait un avantage juridique car il bénéficie d'une formation complémentaire lui permettant de mieux pratiquer l'opération.

- Apports pour le chirurgien expérimenté

Le médecin "en ville" peut bénéficier lui aussi des apports des simulateurs. Par exemple, pour sa formation continue, afin d'apprendre les nouvelles techniques ou tout simplement, pour se remettre à niveau. Les organismes se chargeant de la formation des médecins en ville ne peuvent actuellement que donner la théorie des nouvelles pratiques sans possibilité d'entraînement à l'acte, car ce ne sont pas tous des organismes hospitaliers. Les simulateurs peuvent combler cette lacune. Enfin, si deux confrères possèdent un tel simulateur, ils peuvent s'échanger leur base de données de cas respectifs pour avis. Le simulateur offre un très bon moyen d'échange de compétence et d'amélioration des méthodes. Le simulateur se présente alors comme un banc d'essai pour tester des nouvelles procédures thérapeutiques.

Un nouvel avantage apparaît en plus, par effet de bord. En effet, en cherchant à simuler le comportement d'un système, il faut dans une première phase le comprendre complètement. Pour le chirurgien qui participe à l'élaboration de tels simulateurs, cette activité devient une source de compréhension des phénomènes physiques à la base de la procédure qu'il utilisait auparavant. Cette compréhension peut permettre d'améliorer la technique opératoire.

1.3. Classification des simulateurs médicaux - Etat de l'art

Les simulateurs médicaux ont une histoire toute récente, comparés aux simulateurs de vols. Ils bénéficient de ce fait des techniques de génération d'images employées dans ces derniers et bien maîtrisées à l'heure actuelle. Cependant, la génération d'images ainsi que les procédés de visualisation de cette image ne constituent qu'une partie des techniques nécessaires aux simulateurs médicaux. En effet, les opérations simulées incorporent souvent des contacts ou chocs entre objets, ce que l'on regroupe sous le terme d'*interaction mécanique*. Dans de nombreux cas, le chirurgien va devoir pousser ou tirer des organes, couper des tissus. Pour prendre en compte ces actes, le simulateur devra se baser sur un modèle non seulement visuel mais aussi mécanique de l'environnement. Les organes, sur lesquels s'exerce une contrainte, doivent se déformer en fonction de leur déformabilité, et un *système à retour d'effort* doit permettre de retranscrire vers le manipula-

teur la résistance à la déformation de l'organe (comme décrit au paragraphe *1.1.2*). Cette considération nous conduit à une classification naturelle des simulateurs. La première catégorie regroupe les simulateurs ne faisant intervenir que la visualisation, c'est à dire ceux qui se rapprochent le plus, techniquement parlant, des simulateurs de conduite. Ensuite, on trouvera dans la seconde catégorie, les simulateurs incorporant un modèle mécanique qui se déforme suivant les contraintes qu'on exerce sur lui, mais sans que le modèle soit véritablement basé sur la physique et où le retour d'effort, s'il existe, n'est pas une caractéristique essentielle. Enfin, la troisième catégorie regroupe les simulateurs complets ou presque, avec modèles déformables d'organes et systèmes à retour d'effort.

1.3.1. Les simulateurs de navigation

La première étape d'une thérapie passe par le diagnostic. Cette activité se base entre autre sur l'observation visuelle de la zone atteinte par la pathologie. Cette phase d'observation est naturellement critique, car elle conditionne le type de thérapie qui sera appliquée. Elle n'est donc pas négligeable, et il ne faut pas la sous-estimer car certaines pathologies demandent une grande pratique pour être décelée. Un simulateur est à même d'accélérer l'apprentissage de la phase de diagnostics. Par ailleurs, un tel simulateur offre une opportunité fondamentale pour l'étude de l'anatomie.

Le simulateur de [BL93] est une illustration exemplaire. Il a été réalisé en collaboration avec la société SOGITEC en s'appuyant pour la modélisation sur le logiciel SINDBAB. Le médecin peut "voyager" dans l'espace pelvien tel un pilote dans un paysage. Les concepteurs font l'esquisse des premières applications d'un tel simulateur pour la reconnaissance de l'anatomie, la répétition de gestes chirurgicaux et leur validation, et enfin pour la recherche de nouvelles techniques opératoires. Pour tous ces objectifs, le simulateur permet non seulement de naviguer dans l'espace opéré mais aussi de matérialiser les contacts entre outils et organes par l'émission d'un signal sonore ou visuel. Ce simulateur ne semble pas avoir eu une application pratique et semble même avoir été abandonné au profit d'un autre simulateur, réalisé en collaboration avec *Thomson Training&Simulation* [BCCL95]. Dans sa version première, il reprend les caractéristiques du simulateur précédent, avec signal sonore lors de contact, avec une modélisation peut-être plus fidèle des organes. En outre, ce simulateur est étudié pour pouvoir incorporer la partie interaction mécanique, non encore achevée à notre connaissance. Le dispositif est composé également d'un poste instructeur permettant de modifier, durant la simulation, divers paramètres, ajouter des complications voire faire stopper momentanément le simulateur pour commentaires.

A Lille, [DGCL93] utilisent des images pré-enregistrées sur vidéo-disque dans son simulateur d'endoscopie digestive. Le chemin que doit emprunter l'endoscope est fixé et tout éloignement de cette trajectoire se traduit par une erreur dans le simulateur. Le manipulateur doit alors recommencer le passage problématique. Ce simulateur est très réaliste car basé sur des photographies d'appareil digestif réelles, et en plus très simple à réaliser puisque les mouvements commandent le lecteur de vidéo-disques. Cependant il est dommage de limiter la trajectoire de l'endoscope à une seule solution : il n'y pas de chemin unique. Au contraire, un modèle permettrait de savoir précisément si la position de l'endoscope est autorisée ou non et une vue adaptée à la position exacte de l'endoscope pourrait être générée. Le matériel graphique nécessaire est peu coûteux puisque le simulateur est piloté par un PC. Cet argument est également valable pour la machine proposée par [DCH95]. La simulation se fait en affichant des images au format GIF sur un PC. Ces images sont modifiées à l'aide de la souris afin de découper puis d'observer l'environnement. Ce simulateur est dédié à l'apprentissage de l'anatomie.

[GW87][HGW92] proposent quant à eux un environnement pour un simulateur de *colonoscopie*, opération dans laquelle le médecin manipule un endoscope afin d'observer et soigner le colon. Ici, l'image visualisée a été simplifiée, ce qui permet au simulateur d'être implanté sur un PC, tout en gardant des temps de génération de l'image tout à fait acceptables (vingt images par seconde sur un 486DX2). Afin de remplir son rôle, le simulateur détecte quand l'endoscope touche une paroi. Un système à retour d'effort simple a tout de même été ajouté dans une version ultérieure du simulateur, afin d'augmenter le réalisme.

Enfin, [ZMFG95] propose un simulateur d'arthroscopie du genou. Le modèle utilisé est une reconstruction manuelle à partir d'IRM d'où l'on extrait, dans une étape ultérieure, la surface sous forme de triangles. Lors de la simulation, des capteurs électromagnétiques mesurent la position des outils et à partir de ces données, les images obtenues sont calculées par une station *Silicon Graphics Extreme*, pour un taux de rafraîchissement de six à dix images par seconde. Lorsqu'une collision est détectée entre les outils et une partie du genou, un signal sonore est émis. Les auteurs reconnaissent qu'un système à retour d'effort manque à leur simulateur pour remplir parfaitement sa tâche.

1.3.2. Les simulateurs avec déformations

Ces simulateurs sont très innovants puisqu'ils arrivent à retranscrire des aspects dynamiques, sans forcément faire intervenir les lois physiques de la mécanique. On cherche ici à obtenir un résultat suffisamment proche de la réalité sans pour autant devoir requérir des calculs trop coûteux. Des heuristiques sont alors employés.

Au TIMC à Grenoble, [HTB95] proposent un simulateur d'examen du genou par ultrasons pour les phases de diagnostic des *thromboses*. Ce simulateur s'appuie sur des images réelles d'échographie du genou. Lors de la simulation, on détermine la position du plan de coupe grâce à des capteurs de position préalablement calibrés. Le plan de coupe permet d'avoir l'image de référence que l'on déforme ensuite pour tenir compte des déformations dues à la position de la sonde échographique. L'écartement entre la position normale et la position déformée sert ensuite pour le calcul de la force de déformation (approximation par élasticité linéaire).

[KKKH95] proposent d'utiliser leur environnement de simulation KISMET afin de simuler des opérations chirurgicales endoscopiques. Le graphisme proposé est très puissant (*Reality Engine 2*). Un système complet de mesure de position des outils est fourni. Le réalisme est accentué par une simulation mécanique des organes. Leur forme est obtenue par reconstruction. La restitution visuelle se fait grâce à des *nurbs* (voir [Far93], chapitre 15) dont les points de contrôle sont reliés entre eux par des ressorts. La résolution des équations tirées des lois physiques permet d'obtenir un comportement naturel dans le déplacement des organes. Cependant, dans le simulateur actuel, aucun retour d'effort n'est fourni, il n'est envisagé que pour une version ultérieure [KKKN96].

Enfin, [LBLG95] étudient un simulateur de cholécystectomie ouverte. Ce travail est original, car c'est tout l'environnement du chirurgien qui est modélisé. L'environnement virtuel est modélisé sous la forme de polygones. Plusieurs outils, dont le changement se fait par commande vocale, sont à la disposition du chirurgien. Il peut ainsi interagir avec les tissus. Comme aucun retour d'effort n'est calculé, un signal sonore indique lorsqu'une action est impossible.

1.3.3. Les simulateurs avec déformations et retour d'effort

La dernière catégorie est constituée des propositions les plus avancées. Le système est opérationnel ou presque et simule tous les aspects des interventions chirurgicales. Par coïncidence, les deux simulateurs les plus avancés à notre connaissance ont trait à l'ophtalmologie.

[SPHL95], en Géorgie, proposent un simulateur impressionnant puisqu'il intègre tous les composants des simulateurs. Des capteurs électromagnétiques déterminent la position des outils. Des images de synthèse sont ensuite générées. Elles sont d'un grand réalisme puisqu'on utilise à la fois le placage de texture et des effets de transparence afin d'afficher un œil conforme à la réalité. Des mesures de forces sont effectuées afin de connaître précisément la réponse de l'œil aux interactions. Le simulateur possède un modèle *ad-hoc*, qui permet la restitution fidèle des contacts et découpes. Le retour d'effort est assuré par trois moteurs permettant la restitution des trois composantes x, y et z de la force calculée.

En Nouvelle Zélande, [SBM94] proposent un système complet permettant la téléopération. Dans ce système, l'œil opéré a un comportement dicté soit par l'œil véritablement opéré, soit par un modèle informatique, ce qui rend ce système utilisable également en local, à des fins d'appren-

tissage. Les graphismes ont été particulièrement soigné, le moindre détail de l'œil est restitué à diverses échelles. Le modèle mécanique sous-jacent est un modèle à éléments finis (voir paragraphe *3.4.1*).

Citons enfin un simulateur de laparoscopie du foie, en cours de construction [CDCT96a] [CDCT96b] avec un modèle déformable basé sur les éléments finis et un système à retour d'effort, le *Laparoscopic Impulse Engine*. Pour le moment seul le foie est modélisé, sans son environnement.

1.3.4. Bilan

La classification proposée correspond à la difficulté de mise en œuvre de ces simulations, et respecte à peu près la chronologie. Les premiers concepts datent du début des années 1980, sous la forme d'études de faisabilité, les premiers simulateurs opérationnels sortent vers 1987-88. De nos jours, le nombre d'équipes dans le monde qui étudie les simulateurs se chiffre par dizaines.

Une remarque s'impose cependant : les principaux sujets d'études sont les créneaux d'enseignements les plus porteurs à savoir l'endoscopie, l'ophtalmologie ou l'arthroscopie. Par exemple, la demande en formation en endoscopie est telle que de nombreuses équipes, ainsi que des industriels se sont engagées dans cette voie :[HGW92], [KKKH95], le *NYU* à New York, l'université de Houston, des entreprises comme *High Techsplanations* [MMRM95], Ixion ou cine-Medecine... En outre, une entreprise américaine, *Immersive Corp.*, commercialise un endoscope incorporant des moteurs pour le retour d'effort.

Tous ces projets sont réalisés non pas par une seule équipe, mais sont issus de la collaboration de chercheurs de cultures différentes. Ce sont des automaticiens, roboticiens, chercheurs en informatique, artistes, médecins et spécialistes en science de l'éducation. Les procédés employés pour atteindre l'objectif fixé semblent dépendre de la culture du maître d'œuvre. On trouve tout aussi bien des simulateurs basés sur des images réelles transformées, ce qui requiert des compétences en analyse d'images, ou sur des images synthétiques, ce qui nécessite des compétences en modélisation.

Vu ce qui a été présenté, rares sont les simulateurs actuellement au stade d'une utilisation clinique. En effet, outre les difficultés liées à la réalisation technique, il existe également une certaine réticence du corps médical vis à vis de tels outils. La plupart des simulateurs présentés dans ce chapitre sont encore incomplets, sans retour d'effort par exemple. Leur évaluation par des spécialistes est de ce fait difficile. Ensuite seulement, l'investissement sur les aspects plus délicats et l'intégration complète sont effectués. Les résultats des études sur l'apport des simulateurs en médecine n'offrent que depuis peu leurs premières conclusions.

Conclusion

Nous avons présenté dans ce chapitre les diverses techniques utilisant les environnements virtuels en médecine, en nous focalisant plus en détails sur les simulateurs médicaux à but pédagogique, objet de nos études au LIFL. La restriction aux simulateurs pédagogiques se justifie sur plusieurs points :

- ces systèmes sont faisables et fiables grâce à la technologie actuelle ;
- ces simulateurs font partie intégrante de systèmes plus complexes et sont de ce fait une étape préliminaire incontournable à la téléopération ou à l'assistance ;
- les débouchés commerciaux existent ou sont latents ;
- contrairement à la majorité des progrès médicaux, les simulateurs ne posent pas ou peu de problèmes d'éthique ;
- l'entrée des simulateurs dans les hôpitaux universitaires est un moyen efficace de préparer le personnel hospitalier à l'utilisation des nouvelles technologies.

L'état de l'art dressé met en exergue l'essor des études entreprises depuis environ cinq ans dans le monde. Nous présentons dans le chapitre suivant les travaux effectués au LIFL sur ce sujet. La démarche originale qui a été adoptée, fondée sur divers projets de difficulté croissante, tire naturellement sa justification de la classification que nous venons de faire.

2 Les simulateurs médicaux au LIFL

Introduction

L'axe simulateurs médicaux est un domaine récent de recherche au LIFL. Ces recherches sont le fruit d'une collaboration entre l'équipe *graphix* du LIFL, les biophysiciens et les médecins du CHR. Cette collaboration est officialisée par la création d'un *GIS* (Groupe d'Intérêt Scientifique) que l'on décrit dans l'historique formant la première partie de ce chapitre. Les projets de simulation menés dans le cadre de ce GIS sont présentés dans la suite. On montre que ces projets sont d'une difficulté de mise en œuvre croissante, ce qui explique leurs différents degrés d'avancement. Ces projets concernent l'ophtalmologie, l'écho-endoscopie, la cœlioscopie et la chirurgie in utero échoguidée. Dans la dernière partie, on dégagera, à la lumière des projets exposés, les divers points à résoudre dans la réalisation de simulateurs.

2.1. Le GIS simulateurs d'instruments médicaux

Nos travaux sur les simulateurs ont débuté en 1993 par l'étude d'un simulateur de photo-coagulation laser, sous la demande du professeur Rouland. L'équipe *graphix* du LIFL a collaboré avec l'*Institut des Technologies Médicales* (ITM) regroupant les biophysiciens du CHR&U, à la conception du simulateur Sophocle décrit ci-après. Le succès de ce simulateur a démontré la complémentarité des deux équipes. En outre, le recours à la simulation pour la formation intéressait de nombreux médecins, dans d'autres disciplines. Les nouveaux projets nécessitaient les compétences à la fois des informaticiens pour la modélisation visuelle et mécanique des organes et des biophysiciens pour le retour d'effort.

A partir de l'été 1994, une réflexion a été menée, qui a aboutit à la formalisation d'une activité commune entre le LIFL, l'ITM et les médecins du CHR afin de faire de Lille un pôle important sur le thème des simulateurs médicaux. Enfin, ce groupe possède une reconnaissance officielle depuis l'automne 1995 grâce à la constitution d'un Groupement d'Intérêt Scientifique (GIS). Les premiers travaux de ce Groupe Simulateurs d'instruments médicaux ont été présentés lors d'une première séance de travail le 18 décembre 1995.

Les équipes impliquées sont nombreuses et démontrent l'aspect multi-compétence du thème de recherche. Outre l'équipe de biophysiciens de l'ITM (représentée par Patrick Dubois), et les informaticiens du LIFL (Christophe Chaillou), des médecins du CHR représentent les disciplines suivantes : ophtalmologie (Jean-François Rouland), gynécologie (Denis Querleu et Francis Puech), gastroentérologie (Vincent Maunoury).

2.2. Les projets

Nous détaillons dans la suite les divers projets en cours depuis la formation du GIS. Le premier concerne SOPHOCLE, simulateur de photo-coagulation de la rétine par laser, ensuite SPEED, simulateur d'écho-endoscopie digestive, SPIC, simulateur de cœlioscopie gynécologi-

que, et enfin de nouveaux projets en cours d'étude : un simulateur de kératotomie radiaire en ophtalmologie et un simulateur d'amniocentèse. Parmi ces projets, le premier sera très détaillé par rapport aux suivants, car il correspond à une partie des travaux effectués lors de cette thèse.

Pour chaque présentation on s'attachera à définir les aspects suivants : le fonctionnement de la véritable opération et sa transposition à la simulation, la modélisation et la visualisation temps-réel.

2.2.1. Sophocle : simulateur de photo-coagulation laser

Sophocle (Simulateur Ophtalmologique de PHOto-Coagulation par LasEr) est le premier simulateur étudié dans le GIS. Les premières études ont débouché en septembre 1993 sur un premier prototype, rudimentaire, purement en images de synthèse, mais montrant l'intérêt d'un tel simulateur. Ce travail préliminaire a fait l'objet de mon DEA [Mes93]. La version suivante [DRMK95][MRDK95], quasi définitive a été proposée en février 1995.

La motivation initiale est la suivante. La photo-coagulation laser est une technique chirurgicale utilisée couramment depuis une dizaine d'années (trois mille actes ont lieu au CHR de Lille par an). Pourtant, alors que cette thérapie n'est plus à remettre en cause, on ne constate pas d'amélioration globale des pathologies qu'elle est censée traiter. Cela est dû en partie au vieillissement de la population, mais aussi peut-être au manque de préparation des ophtalmologistes à cette thérapie le long de leurs six semestres de formation. Un simulateur est donc une bonne approche pour pallier ces déficiences.

a) Description de l'opération

Principe de l'opération

La *photo-coagulation laser* est un procédé chirurgical permettant de soigner les problèmes de la rétine de l'œil, comme les décollements de rétine dus à un choc, les apparitions de trous chez les personnes myopes causés par des étirements de rétines (myopies compliquées), les apparitions de vaisseaux sanguins annexes chez les personnes diabétiques, ce qui provoque des écoulements de sang dans l'œil (*rétinopathie diabétique*), les déchirements de rétine chez les personnes âgées. La photo-coagulation consiste en l'envoi d'une impulsion laser permettant de fixer la rétine sur le fond de l'œil. Le matériel mis en œuvre comprend un *biomicroscope*, qui supporte l'appareillage optique (binoculaire) permettant au chirurgien de visualiser le fond de l'œil, l'éclairage de l'œil (fente lumineuse), et l'appareillage nécessaire à la manipulation du laser. Sur l'œil du patient, est maintenu par le médecin un *verre à trois miroirs*, permettant l'observation des différentes zones de l'œil. Ainsi, aussi bien les zones du fond que celles les plus en périphérie peuvent être observées. La *figure 2.1* compare la véritable opération avec la simulation que l'on propose.

Le biomicroscope

Le biomicroscope est constitué d'un microscope, d'une *lampe à fente* et d'un socle porte instrument. Le médecin observe l'œil par l'intermédiaire d'un binoculaire. L'œil est éclairé par la lumière émise depuis une fente située à l'arrière de l'appareil. Une première commande (palonnier) est situé à l'avant. Elle permet de déplacer la lampe à fente sur le plan horizontal. Il permet également, par rotation, le déplacement vertical. Le laser est émis par le haut et se réfléchit sur un miroir. Un micromanipulateur permet d'incliner le miroir selon deux rotations. Dans la suite, le terme de *lampe à fente* s'appliquera à l'ensemble du biomicroscope, et non plus à la seule partie éclairage.

Le verre à trois miroirs

Le verre à trois miroirs est placé sur l'œil du patient. Un schéma est donné sur la *figure 2.2* Il est constitué d'une lentille centrale (1) et de trois miroirs de différentes inclinaisons :

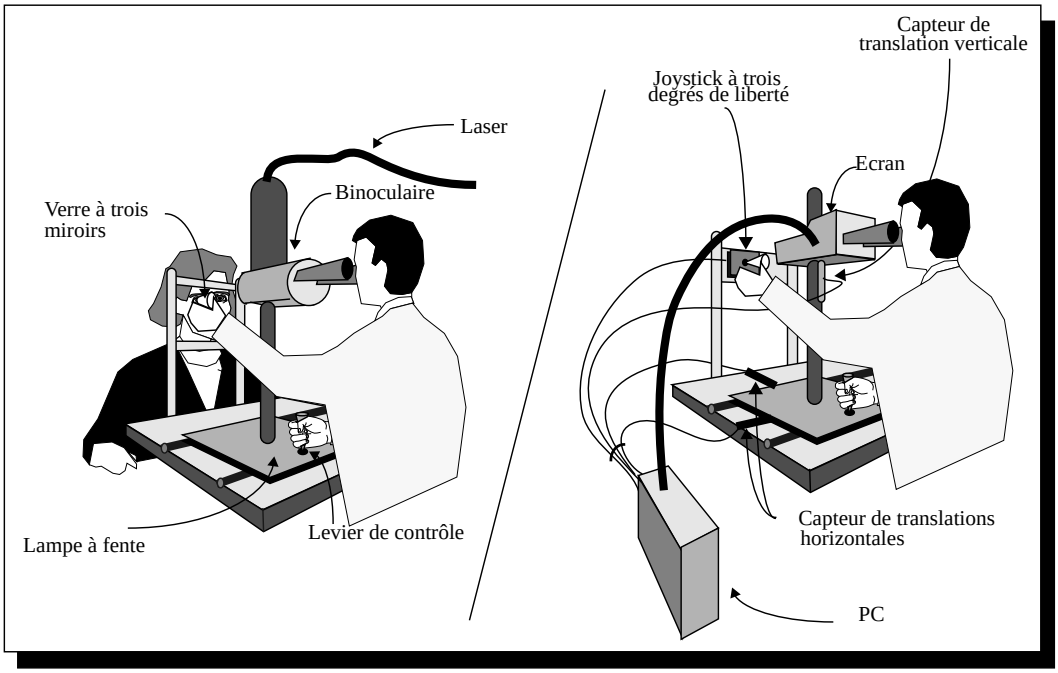

Figure 2.1. *Comparaison opération réelle (gauche) et opération simulée (droite)*

- Le miroir 2 est incliné à 73˚ par rapport au plan frontal, et permet l'observation des régions autour du fond de l'œil ;
- Le miroir 3, incliné à 66˚, permet l'observation des régions périphériques ;
- Le miroir 4, incliné à 59˚, permet l'observation des régions autour de l'iris.

Le verre possède donc trois degrés de liberté, l'un autour de son axe de révolution, les deux autres autour du centre de l'œil (ou plus précisément, du centre de la cornée). On montre que les angles de battement autour de l'œil sont inférieurs à 27˚. Par contre le verre tourne autour de son axe avec un battement de 360˚.

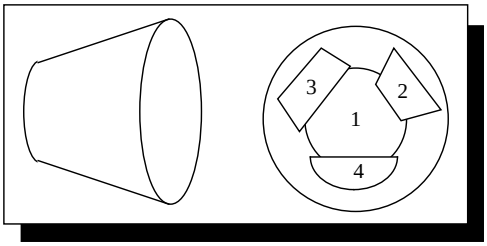

Figure 2.2. *Le verre à trois miroirs*

L'œil opéré

A l'avant de l'œil se situe l'iris (partie colorée) et la pupille (partie par où passe la lumière). Les parties visibles et pertinentes du fond de l'œil sont la rétine, les veines et les artères, la macula (zone située au milieu du fond possédant une grande concentration de cellules et permettant de voir les détails, elle ne doit pas être atteinte par un tir laser), le nerf optique, les cellules, c'est à dire les cônes permettant de voir la couleur et les bâtonnets permettant de voir le contraste (vision nocturne), les vaisseaux choroïdiens, l'*ora serrata*, à l'extrême périphérie de la rétine.

Lors de l'opération, le chirurgien observe l'œil, éclairé par la lampe à fente. Il contrôle directement le verre à trois miroirs, en le faisant glisser sur l'œil du patient. Suivant la zone qu'il faut opérer, le chirurgien observe l'œil dans un des miroirs, ou directement dans le verre central. De même, lors d'un tir, soit le laser atteint directement le fond d'œil, soit se réfléchit sur un miroir du verre avant d'atteindre l'œil. L'emplacement du spot laser est indiqué par l'intermédiaire d'un

faisceau guide. L'envoi d'une impulsion laser a pour effet d'aveugler un bref instant l'œil du médecin. Toutefois, des filtres sont utilisés afin de protéger ce dernier.

b) Le simulateur

Une étude détaillée de l'opération nous conduit à la liste de paramètres suivante : les trois translations de la lampe à fente, les deux rotations autour de l'œil, la rotation du verre autour de son axe de révolution, les deux degrés de liberté pour l'orientation du laser, le diamètre du spot, le grossissement (les trois principaux sont 5, 8 et 12, mais il y a aussi 20 et 30), le déclenchement du tir laser (interrupteur à pédale), largeur et hauteur de fente (le paramètre d'inclinaison est éliminé), la puissance de la lampe, les paramètres contrôlés par le tableau de commande du laser (nature, krypton ou argon, puissance, la durée de l'impulsion). La puissance du laser se doit d'être simulée. En effet, s'il est peu puissant, l'impact est inutile, et si la puissance augmente, la couleur devient plus blanche et si le laser est trop puissant, il y a formation de bulles.

Le déplacement du verre à trois miroirs est mesuré par l'intermédiaire d'un joystick analogique. De même pour la position du micromanipulateur. Les informations de ces deux joysticks sont récupérées par une carte d'acquisition contenant plusieurs entrées analogiques et une entrée digitale. Les informations sont ensuite envoyées vers l'unité centrale : elles sont chacune codées sur douze bits. Le mouvement de rotation autour de l'axe de révolution est géré par le joystick. Les diverses translations sont mesurées à l'aide de capteurs potentiométriques linéaires. Le tir est un simple interrupteur. Les autres paramètres sont, pour la plupart, mesurés à l'aide de potentiomètres ou d'interrupteurs. Le panneau de commande est un ensemble de boutons poussoirs placés en bordure d'afficheurs sept segments.

Le simulateur est implanté sur un ordinateur de type PC haut de gamme, la configuration actuelle est un Pentium 90MHz. La carte graphique à utiliser est une carte accélératrice spécifique, la *Matrox Impression +*, ayant des routines de tracé (2D et 3D) câblées, communiquant avec le CPU grâce à un bus PCI. La résolution de l'affichage est 640x480 en 32768 couleurs ce qui donne un bon compromis temps réel, mémoire vidéo, précision de l'image. De plus, cette carte permet de modifier les fréquences du signal vidéo qu'elle génère. Il est ainsi possible de fournir les signaux nécessaire à l'écran miniature de technologie *nuColor*. Cette technologie exige en effet, un balayage de 180Hz (les niveaux de rouge, vert et bleu sont filtrés successivement d'où le besoin d'une fréquence triple). Cet écran miniature, placé dans le binoculaire permet de présenter l'image de simulation de la façon la moins conforme à la pratique opératoire habituelle. Le réalisme est renforcé par l'ajout de l'ambiance sonore. Un bruit de fond est joué à l'infini et un son est émis lors d'un tir.

La configuration totale, détaillée sur la *figure 2.3*, comprend un autre PC, dit maître, exécutant sous l'environnement Windows, une interface conviviale et un programme d'assistance à l'enseignement (EAO). Ce PC commande le PC de simulation via une liaison série. Le logiciel d'EAO fonctionne comme une suite d'écrans appelés les uns après les autres, il est donc facile d'utilisation. Trois catégories d'utilisateurs ont été répertoriées, à savoir le gestionnaire, les experts et les élèves. Le gestionnaire est responsable du bon fonctionnement du simulateur et de la cohérence de la base de données. Les experts sont l'ensemble des ophtalmologistes ayant des internes à leur charge. Le logiciel leur permet de créer des exercices de photo-coagulation qu'ils proposent à leurs élèves, en choisissant entre autres, le type de pigmentation et la pathologie désirés, le côté de l'œil, sa mobilité... Les élèves peuvent se connecter, afin de s'entraîner, observer une intervention enregistrée ou consulter leur progrès depuis leurs dernières séances.

c) Modélisation

L'œil

L'œil est modélisé par une sphère de 24mm de diamètre. L'iris est modélisé par un anneau plat orthogonal à l'axe optique. La *figure 2.4* représente le modèle synthétique de l'œil, avec les cotations respectives de chaque élément. Tous les points de l'œil sont définis dans un repère lié à

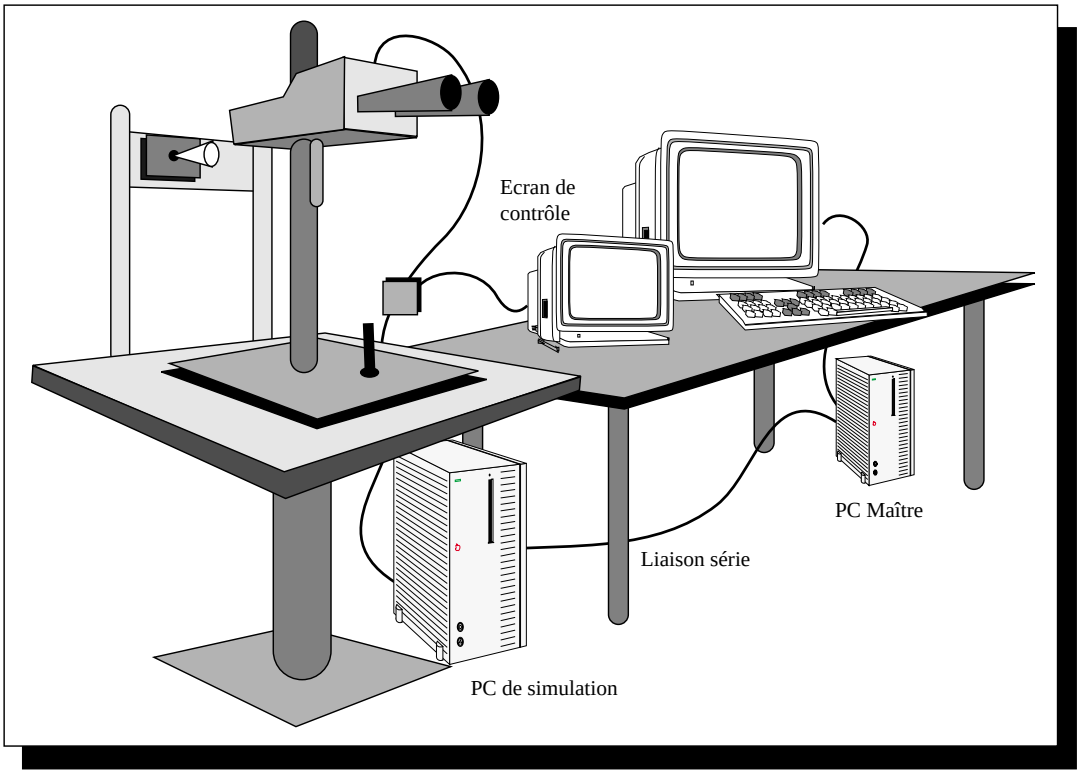

Figure 2.3. *Configuration complète du simulateur*

l'œil. Le centre de ce repère est le centre de la sphère, les axes sont définis ainsi : si on regarde l'œil, avec l'iris devant soi, l'axe $\vec{x}$ est horizontal et pointe vers la droite, l'axe $\vec{y}$ descend et l'axe $\vec{z}$ se dirige de la pupille vers la macula, dans la direction de l'axe optique. Enfin, pour simuler les mouvements de l'œil, les deux angles d'inclinaison le long des axes $\vec{x}$ et $\vec{y}$ sont modifiés suivant une loi gaussienne située autour de la position de repos, le long de l'axe optique. L'écart type est choisi afin d'avoir des mouvements de grande envergure le plus rarement possible.

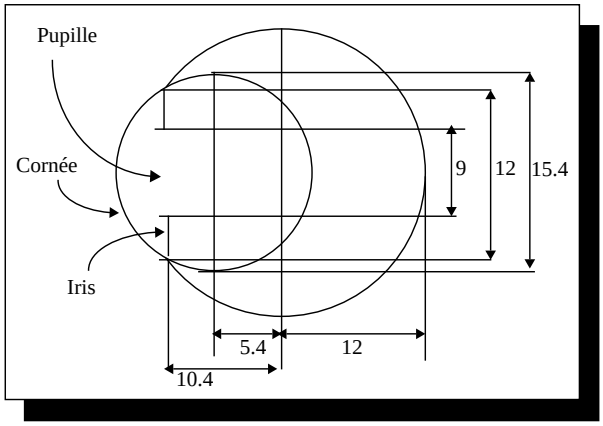

Figure 2.4. *Cotations de l'œil (en mm)*

Modèle approché de l'œil

Un soin tout particulier doit être apporté au réalisme de ces images. Pour cela, on choisit d'utiliser des photographies digitalisées de fond d'œil et de l'iris que l'on emploie afin de reconstruire les textures.

Les méthodes générales pour le plaquage de textures définissent trois espaces : le premier est l'espace écran, à deux dimensions. Le second est l'espace réel, à trois dimensions et enfin le troisième est l'espace texture, à deux dimensions. Le but est de trouver les fonctions permettant d'aller d'un de ces espaces vers un autre. Dans la transformation perspective directe, on cherche la fonction allant de l'espace texture vers l'espace écran. Elle effectue un accès séquentiel à la texture. Elle est rarement utilisée, car elle peut entraîner des pixels non allumés ou au contraire, plusieurs fois modifiés. Au contraire, dans la transformation inverse, on parcourt la zone recouverte à l'écran, et pour chaque pixel, on trouve quelle est sa couleur dans l'espace de texture.

Dans le cas de la sphère, on peut définir le paramétrage suivant :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} R\cos\varphi\sin\theta \\ R\sin\varphi\sin\theta \\ R\cos\theta \end{pmatrix} \tag{2.1}$$

On en tire les paramètres qui vont décrire l'espace des textures. Ainsi, à chaque point écran, on peut trouver le point de la sphère, de ce point on en tire le couple (φ,θ), grâce auquel on lit la couleur dans le tableau de texture. Cette méthode exige pourtant, pour chaque pixel à texturer, une racine carrée et deux calculs d'arc-cosinus. Elle est donc très lente. On choisit alors la solution de découper la sphère en facette, pour, ensuite, texturer chaque facette. La facettisation peut-être plus ou moins fine. Plus elle est fine, plus il y a de facettes à traiter, mais l'affichage reste le même. La perte ne se situe alors qu'en temps de traitement, pas en temps d'affichage. Une condition essentielle de facettisation est que l'objet-facette doit englober la sphère. En effet, étant donné que l'on regarde l'intérieur de la sphère, s'il existe des volumes à l'extérieur de l'objet-facette, des zones de l'écran ne seront pas texturées pour des angles de visée *rasant*.

Le polyèdre choisi pour représenter la sphère est un dodécaèdre. Cet objet géométrique est composé de vingt triangles équilatéraux. La texture est codée de la manière suivante : on se fixe un numéro pour chaque facette, les textures sont alors codées dans des triangles mis bout à bout dans l'ordre croissant. La structure obtenue peut être insérée dans un tableau bi-dimensionnel. Chaque triangle possède un côté divisé en 256 morceaux et une hauteur échantillonnée de la même manière. La couleur d'un point de la facette est obtenue en projetant vers le centre de la sphère ce point, et en récupérant la couleur du point de la sphère rencontré (le centre de la projection conique est le centre de la sphère).

Obtention de la texture de fond d'œil

On suppose que l'on dispose d'un certain nombre de photographies de l'œil. Les premières expérimentations nous ont conduits à utiliser vingt photographies disposées comme suit : une photographie centrale, six photographies de la première couronne, et treize de la seconde couronne. Chaque photographie est projetée sur une carte de l'œil. La projection est parallèle à la direction orthogonale de la photographie. Cette projection est accompagnée de diverses transformations définissant les degrés de liberté de la photographie : deux translations (origine), une mise à l'échelle et trois rotations (positionnement sur la sphère et orientation). Un logiciel manuel et puis un autre semi-automatique ont été réalisés permettant de créer la carte. Cette carte permet de trouver la texture associée à chaque triangle de l'icosaèdre de l'œil. Le processus de construction est expliqué sur la *figure 2.5*.

La texture de l'œil nécessaire au programme s'obtient par un processus complexe. Une image en seize millions de couleurs, de taille 1024x1024 est fournie en entrée et représente une carte en coordonnées sphériques de l'œil, les pôles étant les points de l'axe optique. Cette image, nette au départ, est transformée en trois autres images avec un niveau de flou croissant. Ce flou servira durant l'opération à simuler la profondeur de champ, par interpolation entre ces niveaux précalculés. Le flou des images est précalculé simplement en moyennant les couleurs dans un carré autour de chaque pixel. Les tailles du carré sont 7x7, 13x13 et 21x21. Ces valeurs n'ont pas de justification théorique, elles se sont révélées bonnes à l'utilisation.

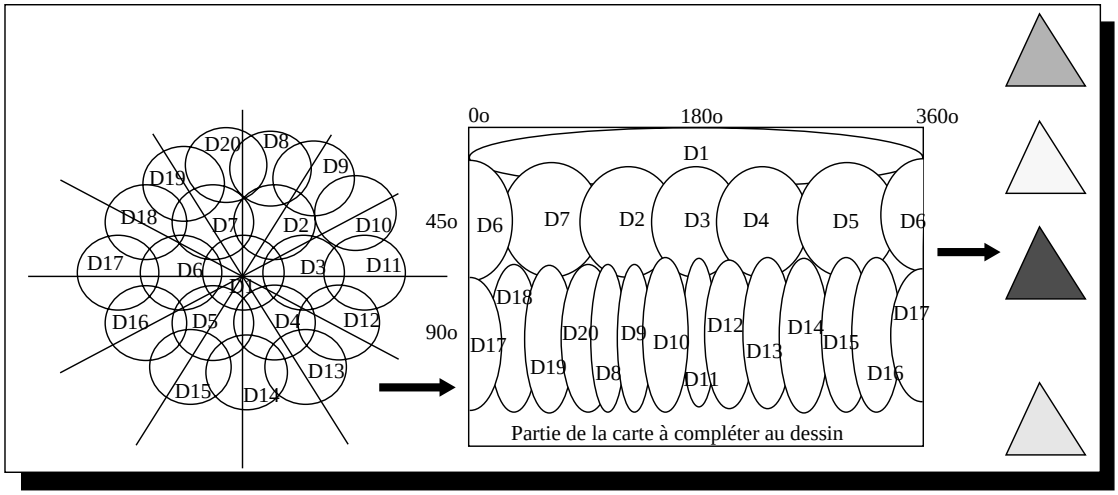

Figure 2.5. *Construction de la carte de fond d'œil*

Modélisation de l'iris

La texture de l'iris est codée de manière tout à fait analogue à celle du fond d'œil. On utilise ici soixante triangles (afin de respecter au mieux l'aspect circulaire de l'iris). A partir d'une photographie de l'iris, on obtient une couronne qu'un programme préliminaire convertit en une bande rectangulaire. Les calculs de flou se font ensuite comme pour le fond d'œil.

Modélisation de l'opération

La modélisation est très dépendante de l'algorithme de rendu utilisé. On choisit donc de modéliser chaque objet de manière simple et d'afficher avec un ordre invariant (ce qui rend tout tri inutile pour l'élimination des parties cachées). Certaines méthodes inspirées du lancer de rayon seront employées pour la gestion du laser.

Modélisation de la lampe à fente

Afin de modéliser la scène, on convient d'un repère absolu, dans lequel toutes les coordonnées seront exprimées. Ce repère est placé, de manière naturelle, au centre de l'œil du médecin (plus précisément, on suppose que le médecin ne travaille qu'avec un œil, alors qu'en réalité, il utilise un binoculaire). L'axe $\vec{x}$ pointe vers la droite du médecin, l'axe $\vec{y}$ vers le bas, l'axe $\vec{z}$ pointe du médecin vers le patient. Dans la réalité, l'œil du patient reste immobile, et c'est la lampe à fente qui est mobile, le médecin suivant les mêmes déplacements. Dans notre modèle, on considère le cas contraire, c'est à dire le médecin et la lampe à fente sont immobiles, alors que l'œil du patient est translaté.

Modélisation du verre à trois miroirs

On modélise le verre à trois miroirs par un cône, on fait une approximation de la lentille sphérique par un cercle dans un plan. Enfin, on modélise les trois miroirs par trois plans : chaque miroir se caractérise par quatre points définissant les bords et une équation de plan. Dans la structure de données, les coordonnées sont exprimées dans un repère centré sur le centre de la sphère de la cornée. En effet, la cornée peut être modélisée par une demi-sphère de rayon 7,7mm, dont le centre est décalé par rapport à celui de la sphère de l'œil. Les axes du repère sont toujours les mêmes. On se fixe une position initiale du verre pour décrire les diverses coordonnées : celle de la *figure 2.2* page 31. Le verre subit les mêmes translations que l'œil, mais se déplace autour de lui. On a dénombré auparavant trois rotations : celle autour de son axe, et deux rotations autour de l'œil. Ces rotation sont exprimées sous la forme d'une matrice dont le calcul dépend de la technologie du joystick analogique utilisé (deux berceaux indépendants ou non) ainsi que de son orientation

lors de son intégration dans le dispositif final. Cette matrice, obtenue par simple multiplication des matrices de rotation élémentaire, permet de connaître les coordonnées dans le repère absolu, des points attachés au verre, ainsi que les nouvelles équations des plans des miroirs.

Modélisation du laser

Un miroir à deux degrés de liberté permet d'orienter le laser. On calcule les coordonnées du vecteur normal au plan du miroir après les rotations. On peut ainsi facilement calculer le point d'incidence du laser sur le miroir et la nouvelle direction par laquelle il le quitte. Par le même procédé, on calcule son éventuelle réflexion sur l'un des miroirs du verre. Pour les trois miroirs, l'intersection M entre le plan et la droite modélisant le laser, doit être déterminée. Ensuite, on vérifie si M est à l'intérieur du miroir (algorithme de produits mixtes). Ensuite, on calcule l'intersection entre le rayon, direct ou réfléchi, avec le bord du verre et l'iris afin de déterminer s'il y a un obstacle. Si ce n'est pas le cas, on calcule alors son intersection avec l'icosaèdre afin de déterminer le point d'impact sur la rétine.

d) Visualisation

Nous dessinons d'abord à l'écran le verre central, puis la vue de l'œil qu'a l'observateur au travers de ce verre. Ensuite par une boucle, on traite les miroirs un par un en dessinant l'image du verre central dans le miroir et la partie de l'œil vue dans cette image.

Découpage de polygones

Certains objets doivent être dessinés uniquement dans d'autres objets. Par exemple, l'iris est vu au travers du verre central, seule la partie de l'iris se trouvant à l'intérieur du contour du verre doit donc être tracée. L'ensemble des objets à tracer sont définis à l'aide de polygones : les cercles de bord de lentille, de bord du verre frontal, les miroirs, l'iris, la fente... Ils apparaissent à l'écran sous forme de contours fermés polygonaux. On tire donc profit de ces polygones, pour délimiter les zones à tracer. Par exemple, le polygone de l'iris est découpé par le polygone de la lentille centrale (voir *figure 2.6*). L'algorithme utilisé est le *clipping* de Sutherland-Hodgman (voir le chapitre 3 de [FDFH90]). Il faut remarquer que le découpage de deux polygones convexes bien orientés donne un polygone convexe bien orienté. Ceci signifie que le résultat du découpage peut ensuite servir de polygone de découpage.

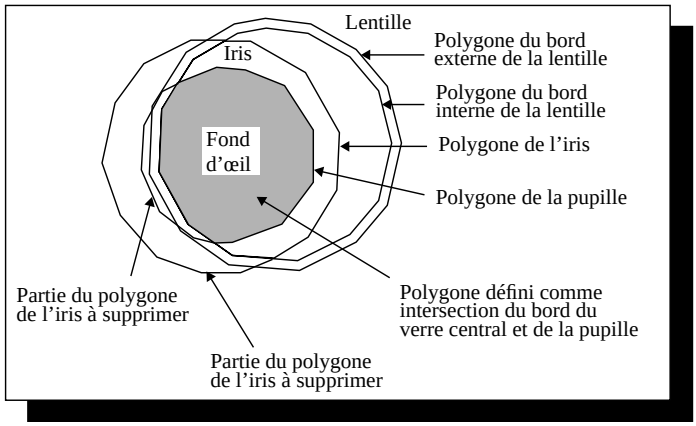

Figure 2.6. *Découpage de polygones*

Algorithme de tracé de l'image

Nous allons maintenant étudier les différentes étapes de la boucle principale du simulateur. Après avoir acquis les paramètres grâce aux valeurs des multiples capteurs, on effectue un certain nombre de calculs, qui peuvent être effectués avant la phase de tracé (matrices, identification du point d'impact du guide,...). Ensuite, le tracé est effectué, on utilise alors un *double-buffer* afin de

créer l'animation, c'est à dire que l'écran visible et l'écran de tracé n'est pas le même, ce qui fournit une image stable.

Le tracé proprement dit est simple. Le principe général de l'algorithme est le suivant : l'objet est toujours à tracer dans une certaine zone, d'où on le restreint à cette zone et on le trace, et il définit la zone dans laquelle il faudra dessiner l'objet suivant. Ainsi, si le verre central est invisible, alors rien n'est visible (on quitte tout de suite). Sinon, le bord du verre doit être tracé dans la zone couverte par le verre frontal, et ainsi de suite... On utilise la routine de découpage de polygones décrite ci-dessus. Il faut cependant noter que l'iris (respectivement le fond d'œil) est tracée à l'intersection de la lentille centrale (respectivement de la pupille) et de la zone éclairée par la lampe à fente. Ainsi, pour l'iris, la détermination du polygone de découpage est plus complexe puisque c'est l'intersection de la zone éclairée et de la lentille centrale découpée (mais pas pour le fond d'œil car il utilise le polygone de découpage de l'iris, lui-même découpé par la zone éclairée). Au sujet du tracé dans les miroirs, le principe reste le même, mis à part qu'avant la projection sur l'écran, on calcule l'image de l'iris, de l'icosaèdre, du verre central..., par la symétrie par rapport au plan du miroir.

Les parties texturées, vues au travers du miroir ou non, sont tracées suivant le même principe.Chaque triangle du fond d'œil ou de l'iris est traité séparément. Si il est mal orienté, on le supprime, sinon, on le découpe avec le polygone de découpage, puis on divise à nouveau le polygone obtenu en triangles. Le placage de texture utilisé est un placage linéaire qui ne tient pas compte de l'effet perspective, c'est à dire que les coordonnées de texture dépendent linéairement des coordonnées écran, mais pas de la coordonnée de profondeur.

Ce choix de placage est justifié car les facettes que l'on observe sont celles qui sont dirigées vers l'observateur, ou presque... Ces facettes étant quasiment parallèles au plan de l'écran, l'effet de la perspective est négligeable. En outre, la facettisation de la sphère de l'œil est susceptible de présenter quelques inconvénients. En effet, la méthode de facettisation garantit une continuité G_o de la texture, mais pas une continuité G_1 sur les bords de deux facettes. Cela n'est pas gênant dans le cas général, même lorsqu'un vaisseau sanguin passe d'une facette à une autre.

Les textures ajoutent un réalisme indéniable à l'image. Pourtant, elles ne suffisent pas. On choisit donc de les renforcer par une simulation de l'éclairage de la fente et du flou généré par les différentes lentilles.

Simulation de l'éclairage

Il est important d'apporter une solution à l'éclairage car sinon la différence de couleur entre la zone texturée et le reste de l'image (principalement noir) est trop marquée et n'est pas réaliste (voir *figure 2.7*). Pour cela, la zone éclairée est divisée en neuf zones. L'algorithme ci-dessus reste vrai, mise à part que chaque triangle est découpé successivement par les neuf zones différentes. On considère un indice d'atténuation faisant référence à une table précalculée, donnant pour chaque composante R, V et B sa valeur atténuée. L'atténuation dépend soit de x soit de y soit des deux, suivant la zone, elle est fonction de la distance par rapport au bord de la partie centrale (sans atténuation) selon une fonction en cosinus au carré (assez proche d'un modèle physique simple).

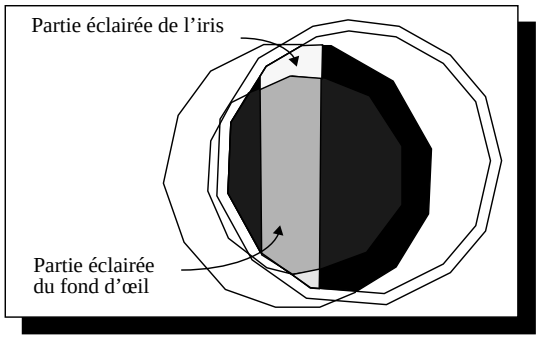

Figure 2.7. *Eclairage de l'œil*

D'autre part, simuler l'éclairage signifie aussi qu'il faut simuler l'éblouissement. Ainsi, en fonction de la largeur de la fente, on fait tendre les couleurs vers le blanc.

Simulation du flou

La visualisation se faisant à l'aide d'un certain nombre de lentilles, il arrive souvent que l'image soit floue. Le réalisme de la simulation se doit d'incorporer ce flou, car il est d'une part difficile à obtenir lors de l'opération et qu'il permet un certain repérage en profondeur. Plusieurs solutions ont été envisagées pour simuler ce flou. En effet, un calcul exact [PC81] ne peut pas être envisagé en temps réel et les solutions proposées habituellement sont fondées sur une méthode multi-passes, s'appuyant sur la partie matérielle appelée *Accumulation Buffer* [HA90]. La première solution est de créer un flou optique, par l'intermédiaire d'une lentille que l'on défocaliserait à l'aide d'un moteur pas à pas. Utiliser une lentille devant un écran présente des inconvénients : on risque, par exemple, de voir les pixels de l'écran. Une autre solution est d'utiliser un écran à tube cathodique. En jouant sur le focus, on pourrait créer une sensation de flou. Ces deux solutions ont cependant le désavantage de créer un flou uniforme sur toute l'image. On ne peut pas choisir des zones de l'écran à rendre plus ou moins floues.

Au contraire, un flou logiciel est un avantage certain, puisque c'est le programme qui décide ce qui doit être flou ou non. Afin d'implanter ce flou, on décide d'un certain nombre de niveaux de flou, qui sont calculés exactement. Pour le calcul d'un flou, pour chaque pixel, on prend la moyenne des couleurs des points d'une zone carrée située autour du pixel considéré. La taille de la zone définit le niveau de flou plus ou moins grand. On précalcule donc trois niveaux de flou et on crée la texture pour chaque niveau. On obtient donc quatre textures du fond d'œil, la texture nette et trois textures floues. Lorsque l'on trace le fond de l'œil, la couleur d'un pixel est obtenue en interpolant linéairement entre deux niveaux consécutifs parmi nos quatre versions de la textures.

Afin d'augmenter le réalisme, le niveau de flou n'est pas le même pour l'iris et le fond d'œil. De plus il n'est pas le même dans la vue directe et dans les différents miroirs. Par contre, le niveau de flou de l'iris ou du fond d'œil reste constant sur tout l'objet. Son niveau ne dépend respectivement que du centre de la sphère ou de l'anneau. De plus, le diamètre de l'iris est modifié afin de l'agrandir dans le cas du flou.

En conclusion, le placage de texture se fait d'une part en trouvant la couleur par interpolation entre les niveaux de texture, puis saturation de la couleur vers le blanc puis atténuation de la couleur vers le noir en fonction de l'éclairage. Dans [MKCD95], on trouvera l'algorithme complet de tracé à l'écran, ainsi qu'une description plus précise de la gestion du flou.

Tracé du spot et des impacts

Soit le laser passe directement dans la lentille centrale, soit il passe au préalable par l'un des miroirs du verre. Dans tous les cas, on obtient l'emplacement du spot sur le fond d'œil. Afin de donner un aspect *translucide* au spot, on choisit de le dessiner avec un motif quadrillé selon la technique du *screen-door transparency.* Ainsi, un point sur deux est affiché avec la couleur de spot, les autres points ne sont pas modifiés. Cette technique est employée afin de simuler un aspect flou du spot : le centre est en couleur uniforme, alors que le contour, de rayon croissant en fonction du flou, est dessiné en quadrillé. Lors du tir, les impacts sont d'une part stockés en mémoire pour l'évaluation et stockés dans la texture pour l'affichage.

En effet, afin de ne pas avoir à colorer plusieurs fois un pixel, à la fois pour le fond d'œil et pour les impact, et afin de pouvoir afficher des impacts flous, on choisit d'incorporer les impacts dans la texture. Cette opération est relativement délicate. La zone recouverte par un impact n'est pas forcément restreinte à une facette. Or, les textures associées à deux facettes adjacentes ne sont pas forcément adjacentes dans le tableau de texture. Un traitement spécial doit donc être effectuer afin de déterminer les textures qui seront modifiées. Ce traitement s'appuie sur une structure topologique de l'icosaèdre. Enfin, l'impact doit être plaqué dans les nivaux flous de la texture. On se base donc sur la texture nette où l'on vient de plaquer l'impact et on utilise une méthode locale

heuristique, de calcul de flou. Cette méthode étant relativement lente, seul le niveau le plus flou est calculé ainsi, les deux autres niveaux intermédiaires sont obtenus par interpolation entre le niveau net et le plus flou.

e) Bilan

Un brevet [CDKM94] a été déposé et le prototype est actuellement en phase d'industrialisation. Il a été présenté et très bien accueilli lors de nombreuses expositions. Il a de plus fait l'objet d'une étude par un interne en ophtalmologie [Peu96] lors de son DEA de génie biomédical. Son impact sur l'apprentissage des internes en ophtalmologie est mesuré au CHR&U de Lille. En outre, il a également été installé à la clinique ophtalmologique du Professeur Coscas de l'hopital intercommunal de Créteil où il a tout de suite été très bien accueilli et immédiatement apprécié. L'étude a consisté à considérer deux groupes d'étudiants. Les étudiants du premier groupe on subi une formation normale tandis que les étudiants de l'autre groupe ont eu un apprentissage de la photo-coagulation laser uniquement par simulateur. A la fin de la période, les deux groupes étaient évalués sur patients réels. Les résultats ont montré que la formation sur simulateur était équivalente à la formation normale. L'impact de Sophocle sur le processus de formation est donc pratiquement démontré.

Ce projet est véritablement la première étape idéale pour notre étude sur les simulateurs médicaux. Il a permis aux équipes du GIS d'aborder à la fois les problèmes de choix et intégration des capteurs, conception de la modélisation d'un organe, des algorithmes de tracés choisis pour la restitution visuelle, ainsi que des moyens de renvoyer cette image vers le médecin sans perturber la pratique opératoire. L'absence d'interaction mécanique et de modèles de déformation lui confèrent une simplicité qui nous a permis de démontrer la faisabilité de simulateurs réalistes sur du matériel grand public peu onéreux. C'est cet aspect qui est primordial si l'on souhaite que les hôpitaux puissent s'équiper avec de tels outils de formation.

2.2.2. Speed : simulateur d'écho-endoscopie digestive

L'écho-endoscopie a permis des progrès importants dans le cadre de l'exploration du tube digestif. Cette technique permet l'exploration de l'œsophage, de l'estomac, du duodénum, du rectum et des organes voisins. Comme 75% des observations écho-endoscopiques se situent au niveau du duodénum, un simulateur d'écho-endoscopie bilio-pancréatique est réalisé dans un premier temps. Ce simulateur est décrit plus précisément dans [VCDM97].

a) Description de l'opération

Un écho-endoscope est un endoscope souple muni d'une sonde échographique à son extrémité. Cette sonde permet de visualiser une image échographique sur un écran vidéo. Cette image correspond à une coupe perpendiculaire à l'axe de l'extrémité de l'endoscope. L'image a pour centre la sonde. Pour avoir une image de qualité, la sonde de l'endoscope est entourée par un ballonnet gonflé d'eau, l'eau permettant la transmission des ondes ultrasonores. Afin d'améliorer encore l'image échographique, l'opérateur appuie le ballon contre la paroi de l'intestin dans la direction qu'il veut observer. Cette pression est obtenue en manœuvrant l'extrémité de l'endoscope par le biais des molettes de commande. Cette manœuvre est appelée *béquillage*.

Deux phénomènes d'ordre mécanique ont été déterminés :

- la déformation des organes lors du béquillage ;
- les frottements et glissements de l'endoscope à l'intérieur des voies digestives.

La pression du ballonnet sur l'intestin lors du béquillage entraîne une déformation du ballonnet, de la surface du duodénum et des organes à la périphérie. L'enroulement de l'endoscope dans l'œsophage, l'estomac et le duodénum contrarie les mouvements de pénétration, de retrait et de rotation sollicités par la main du médecin.

La technique échographique repose sur la propagation des ondes ultrasonores dans les organes. Lors de cette propagation, trois phénomènes (la réflexion, la diffusion et l'atténuation) permettent de construire les images vidéos ultrasonores. Lors du passage d'un milieu à un autre, une partie de l'énergie de l'onde incidente est réfléchie. La proportion réfléchie est fonction des impédances acoustiques des deux milieux. La diffusion correspond à l'émission d'une partie de l'énergie incidente dans toutes les directions de l'espace. Ce phénomène apparaît lorsque l'onde touche une structure dont la taille est petite en comparaison de la longueur d'onde du faisceau ultrasonore émis par la sonde. L'atténuation résulte de la réflexion, de la diffusion, et de l'échauffement des tissus au passage de l'onde ultrasonore. L'intensité décroît exponentiellement avec la profondeur de pénétration.

b) Le simulateur

Le modèle doit intégrer les données physiologiques, les caractéristiques échographiques des organes et les données nécessaires aux interactions entre les organes et l'endoscope. Il doit prendre en compte les interactions mécaniques. En effet, les organes subissant des déformations lors de la manipulation de l'endoscope, il est nécessaire de modéliser leurs élasticités. Ce modèle doit être compatible avec la contrainte temps-réel d'un simulateur.

L'écho-endoscopie est fondée sur la propagation des ondes ultrasonores et sur le retour des échos. Tout comme le lancer de rayons reproduit le parcours des rayons lumineux, on propose de modéliser la propagation des ondes ultrasonores en faisant un lancer d'ondes.

L'endoscope dispose de quatre degrés de liberté : deux degrés pour la déformation de l'extrémité de l'endoscope, un pour le déplacement axial de l'endoscope et un pour la rotation axiale de l'endoscope. Il faut déterminer les limites de ces différents mouvements. Pour chaque degré de liberté, il faut placer une mécanique pour simuler le retour d'effort vers l'opérateur et adapter la sensibilité de cette mécanique. On garde un endoscope réel du coté de l'opérateur. Actuellement, le simulateur incorpore un endoscope avec capteurs optiques de position. Il est en phase de modifications afin d'incorporer des moteurs à courant continu pour le retour d'effort. Cette interface est reliée au PC de simulation via une liaison série.

Il faut construire une structure de données définissant le duodénum et les organes à sa périphérie. Cette structure doit satisfaire les besoins mécaniques. De plus il faut qu'elle soit adaptée à la méthode de calcul envisagée pour générer les images échographiques.

c) Modélisation

Les organes sont modélisés sous forme de surfaces implicites [Gas93]. Les surfaces implicites sont des surfaces mathématiques définies par une équation du type $f(p) = 1$, où f est une fonction et p un point de l'espace. Des iso-surfaces générées par des squelettes sont utilisées. En effet, l'utilisation d'iso-surfaces permet un calcul immédiat des déformations du ballonnet sur le duodénum. De plus les iso-surfaces à squelette ont l'avantage de permettre une définition compacte de surfaces libres et continues comme celles des organes

Les surfaces implicites choisies sont composées par des sphères implicites (le squelette de nos iso-surfaces sont des points). Les propriétés des iso-surfaces (généricité, continuité de la normale, contrôle intuitif des formes) sont adaptées à la modélisation d'organes. De plus ces surfaces permettent de modéliser l'élasticité des objets en jouant sur la dérivée des fonctions potentielles, d'évaluer précisément la surface de contact entre deux objets et d'en déduire les forces d'interactions résultantes.

Un modeleur à base de surfaces implicites sphériques est en cours de développement. Les premiers organes modélisés à la main avec une version rudimentaire du modeleur semblent montrer le bien fondé de ce choix.

d) Visualisation

Le lancer rayon est la méthode la plus naturelle pour simuler la propagation des ondes. Le lancer de rayons classique est lent pour l'affichage de surface implicite, aussi les surfaces sont discrétisées dans un *octree* et l'image obtenue est post-déformée. Le peu de rayons nécessaires à la création d'une images (environ mille deux cents) et l'accélération de la structure octree permettent un affichage temps réel des coupes ultrasonores

Pour construire l'image échographique, la technique du lancer de rayons est adaptée au parcours des ondes ultrasonores. L'affichage de surfaces implicites avec un lancer de rayons étant très coûteux un lancer de rayons discrets est employé. Notre dispositif se décompose en deux parties : la discrétisation des organes et la simulation ultrasonore dans l'espace discret.

Les surfaces implicites sont discrétisées une fois pour toute, pour assurer une vitesse d'affichage compatible avec les contraintes temps réel d'un simulateur. Les données géométriques et ultrasonores sont stockées dans un tableau de dimension *NxNxN*. Chacun des voxels de ce tableau est ensuite décomposé suivant un arbre de sous voxels. Chaque nœud comporte huit fils. Ce sont les sous voxels résultant de la décomposition du voxel en huit parties égales. Les feuilles de l'arbre sont des voxels de deux types : surface ou matière. S'ils sont coupés par une surface alors on leur associe les caractéristiques acoustiques de la surface (coefficient de réflexion, une fonction de la surface implicite,...). Dans le cas contraire, le voxel contient de la matière organique et on lui associe les caractéristiques acoustiques de celle-ci (coefficient de diffusion, texture, atténuation...). La subdivision s'arrête quand aucune surface n'est présente dans le voxel ou quand la taille de la grande diagonale du voxel est inférieure à la taille d'un pixel. Cette limite à la précision géométrique des organes semble compatible avec la faible qualité des images vidéos ultrasonores

Nous décrivons maintenant la visualisation à partir de la structure discrète. Dans un premier temps, on positionne un écran virtuel (un rectangle dans un plan de coupe) dans l'espace des organes. Le centre de l'écran coïncide avec le centre de la sonde échographique. Les quatre coins de l'écran sont positionnés sur le plan perpendiculaire à l'axe de la sonde. Le point haut gauche de l'écran est positionné dans la direction du premier faisceau émis par la sonde. Les rayons sont issus du centre de l'écran et se terminent au bord de l'écran. Pour simuler la rotation de la sonde, on déplace la cible vers un autre point du bord de l'écran. L'angle formé entre deux cibles successives est fonction de la vitesse de rotation de la sonde. La propagation des ondes est simulée par un suivi de rayons. La traversée des données est accélérée par la structure en octree. Les grands volumes homogènes sont composés de voxels de grande taille et sont traversés rapidement par le rayon. Les parties de l'espace proche des surfaces sont discrétisées plus finement afin de localiser précisément les points d'intersection rayons-surfaces.

e) Bilan

Les études actuelles cherchent d'une part à améliorer l'image obtenue par le simulateur afin de rendre compte des phénomènes parasites. D'autre part, un modèle mécanique de l'interaction entre l'endoscope et les voies digestives est en cours d'étude afin de calculer le retour d'effort et de rendre compte de la difficulté de progression de l'endoscope.

2.2.3. Spic : simulateur de cœlioscopie gynécologique

Le travail décrit ici est encore aux stades de conception et d'implantation [Jam96]. Un premier prototype simplifié est en cours.

a) Description de l'opération

Le cœlioscopie chirurgicale à visée gynécologique est une technique opératoire dont les indications s'élargissent sans cesse et qui bénéficie d'une large diffusion dans les services de gynécologie. Elle permet d'opérer *à ventre fermé*.

Une fois la patiente endormie, on réalise une insufflation de gaz dans la cavité péritonéale. Ensuite on introduit une caméra dans la cavité abdominale au niveau de l'ombilic par l'intermédiaire d'un trocart. Cette caméra fournit sur un moniteur vidéo la vue avec laquelle travaillera le chirurgien. Un ou plusieurs instruments sont introduits, toujours par l'intermédiaire de trocarts, au niveau des fosses iliaques. Ils permettent de réaliser le geste opératoire.

La cœlioscopie offre de nombreux avantages : une diminution de la durée d'hospitalisation, un allégement des suites opératoires, une diminution de la morbidité et enfin un moindre préjudice esthétique. Beaucoup d'interventions chirurgicales gynécologiques sont réalisables par cœlioscopie. De plus, certaines constituent presque exclusivement une indication de cœlioscopie, par exemple les kystes bénins, l'endométriose ou la levée d'adhérences péritonéales. C'est par ailleurs un formidable outil diagnostique fréquemment utilisé dans les bilans d'infertilité.

Toutefois, cette technique d'intervention présente des difficultés spécifiques. En effet, le chirurgien ne possède qu'une vue indirecte du site opératoire (via la caméra). De plus cette vue n'est qu'une projection de l'intérieur de la cavité abdominale. Le chirurgien est donc amené à réaliser des gestes, souvent déjà difficiles en eux-même, dans des conditions lui demandant une grande capacité d'abstraction afin de reconstruire mentalement la réalité à partir de la projection vidéo et de la position des extrémités de ses instruments. Un apprentissage rigoureux à cette technique d'intervention est donc nécessaire.

b) Le simulateur

Un premier simulateur est cours de développement. C'est un outil simple ayant pour objectif de former les débutants en cœlioscopie au maniement de la caméra et des instruments dans la cavité abdominale. Tout contact de l'extrémité des instruments (ou de la caméra) avec le patient entraîne l'arrêt de la simulation. Un mécanisme permettant de simuler les trocarts et de mesurer le déplacement des instruments dans celui-ci a été conçu à l'ITM. La première version ne permet pas de gérer les déformations et le retour d'effort.

c) Modélisation

La cavité abdominale, les organes observés et les instruments sont construits avec un modeleur du commerce à partir des données métriques de la littérature ou mesurée dans des conditions identiques à celles de la véritable opération. Les formes obtenues par le modeleur sont surfaciques et décrites sous forme de maillage de triangles.

d) Visualisation

Le logiciel permettant de prendre de visualiser les images en fonction des déplacements des divers instruments est développé avec la librairie OpenGL. Cette librairie étant portée sur des machines de performances graphiques diverses, le logiciel se révèle indépendant du matériel. Comme les machines de synthèse d'images traitant les textures sont encore relativement chères, on choisit dans un premier temps de ne pas recourir à du photo-mapping. Cependant, afin d'obtenir un réalisme suffisant et d'éviter d'afficher des organes de couleur uniquement fonction de l'éclairage, on choisit d'affecter une couleur (diffuse) différente en chaque point du maillage. La couleur diffuse est alors interpolée dans la maille par le matériel, via le calcul d'interpolation de Gouraud qui est câblé sur tous les accélérateurs 3D. Cette solution est très intéressante puisqu'elle procure un certain réalisme à un coût en temps très faible.

e) Bilan

Un protocole d'utilisation de ce simulateur, dans la formation des internes de gynécologie, est en cours de conception dans l'équipe du professeur Querleu. La première version, sans retour d'effort, mais incorporant une détection des contacts entre les outils et les organes, devrait être achevée durant le premier semestre 1997. La version suivante devrait inclure le modèle mécanique complet. Ce modèle fait l'objet du *chapitre 5* de ce document.

2.2.4. Les autres simulateurs en projet

a) Simulateur d'opération ophtalmologique

Ce simulateur n'a pour l'instant fait l'objet que d'études préliminaires. Nous choisissons tout de même de le mentionner, car il illustre certaines difficultés dans la modélisation mécanique d'une opération. On cherche à simuler une opération de kératotomie radiaire, opération consistant à faire des incisions dans la cornée de l'œil.

Description de l'opération

La kératotomie radiaire a pour but de soigner et parfois même complètement résoudre certains problèmes de myopie. En effectuant des incisions dans la cornée, on permet à celle-ci de s'assouplir sous la contrainte. Une métaphore suivante illustre très bien le phénomène : il est plus facile de courber une branche d'arbre si on a effectué quelques incisions transversales à intervalles réguliers le long de la branche. Une fois la cornée plus souple, elle pourra mieux s'adapter aux diverses pressions auxquelles elle est soumise et prendre une forme assurant une meilleure convergence de la lumière. Les patients soignés par cette thérapie sont généralement peu myopes et cette correction leur permet de regagner une vue quasiment normale.

Cette opération s'effectue sous binoculaire. A l'aide d'un scalpel en diamant, le chirurgien effectue des incisions radiales d'une part et concentriques d'autre part, dont la profondeur dépend de tables données, fonction de l'épaisseur de la cornée. Le médecin doit prendre garde à ne pas inciser trop profondément, car il peut dépasser la cornée. Dans ce cas résulte une plaie dans la cornée et un écoulement d'humeur vitrée.

Le simulateur

Les études déjà effectuées consistent d'une part en la mesure des propriétés mécaniques de la cornée : un protocole de mesure évalue la force retournée en fonction de la position d'un outil posé au sommet de la cornée. D'autre part, des travaux basés sur les recherches de [Jou93] essaient d'étudier la simplification de son modèle *éléments finis* afin de l'adapter au temps réel.

b) La chirurgie in utero échoguidée

Cette étude est récente car elle a débuté en septembre 1996. L'amniocentèse est une intervention de prélèvement d'un petite quantité de liquide amniotique guidée par échographie. Son indication principale est aujourd'hui la détection de la trisomie.

Description de l'opération

L'intervention consiste à enfoncer une seringue dans l'utérus en traversant la paroi abdominale afin de faire une ponction de liquide. La mise en place de la seringue est contrôlée par échographie, la difficulté de l'intervention réside dans le contournement du placenta et la préservation de l'embryon. Techniquement, il faut, tout au long de l'intervention, maintenir l'aiguille dans le plan de coupe échographique.

Le simulateur

Pour réaliser un simulateur d'amniocentèse, il faut :

- recueillir des images échographiques de référence ;
- élaborer le cahier des charges du simulateur ;
- concevoir une interface mécanique afin de simuler la partie terminale de l'échographe et de l'aiguille ;
- modéliser les parties du corps sur lesquelles on intervient ;

- simuler le fonctionnement de l'échographe.

Ce simulateur est un premier pas vers la simulation de ponction de sang au cordon ombilical et à long terme de la pose d'un drain vésical ou des ligatures du cordon.

2.2.5. Bilan des projets

Ces projets ont été présentés à la fois selon leur stade d'avancement, mais pas selon leur difficulté de réalisation. En effet, certains nouveaux projets ont une difficulté moindre que d'autres plus avancés. Cependant, on peut dire que Sophocle est un simulateur où les seules interactions sont lumineuses, le problème de la simulation est donc essentiellement la mesure des degrés de liberté et la restitution visuelle. SPEED incorpore des degrés de difficulté supplémentaire. D'une part il nécessite une interaction mécanique limitée et des déformations d'organes simples. D'autre part, l'image restituée est une image acoustique qui est bien différente des images habituellement calculées en synthèse. Parallèlement à SPEED, la difficulté du projet d'amniocentèse est la restitution fidèle du retour d'effort. SPIC est plus complexe encore puisque de multiples interactions mécaniques, avec plusieurs organes, et un retour d'effort dans toutes les directions sont possibles. Il faut pouvoir à la fois tirer, pousser, retenir, découper,... Enfin, le projet d'ophtalmologie, même s'il restreint les interactions mécaniques à un seul organe (la cornée), nécessite une précision nettement plus grande pour les déformations et le retour d'effort puisque toute l'opération repose sur la perception mécanique et visuelle du geste effectué.

Par ailleurs, ces projets ont une caractéristique commune : ils ne requièrent aucun port de visiocasque. En effet, soit l'opération se fait sur écran, auquel cas la restitution visuelle de la simulation est immédiate, ou sous binoculaire, dans ce cas on incorpore un ou deux écrans miniatures.

2.3. La problématique des simulateurs

De l'étude des simulateurs précédents ont émergé des difficultés communes. Nous choisissons de les classer en quatre catégories. D'une part les problèmes liés à la construction des modèles, au réalisme de l'image, à la mécanique, et enfin au logiciel d'enseignement.

2.3.1. Construction des modèles

a) Forme

Afin d'être le plus conforme possible à la réalité, il faut construire des modèles des organes de la même forme que l'original. Comme on l'a vu, il est possible de reconstruire la forme à partir de données fournies par des procédés d'observation habituels. Nous préférons l'autre possibilité, à savoir la modélisation des organes avec des outils de modelage classiques. Cette solution a l'avantage de fournir le modèle sous la forme mathématique désirée (polygones, surfaces implicites...). On se base sur des cotations et des approximations de formes. Une difficulté dans ce cas peut venir des différences entre les cotations de l'organe lorsqu'il est au repos et dans les conditions d'interventions. Par exemple, en cœlioscopie un gaz est employé afin de maintenir une pression dans le ventre du patient. Cette pression modifie la forme habituelle de l'organe.

b) Aspect visuel de l'objet

Afin d'être le plus réaliste possible, il s'agit également que l'organe simulé ait un aspect identique à l'original. Cela se traduit par le besoin de reproduire la texture de l'objet. L'idéal est évidemment de se baser sur des photographies (on parle de *photo-texture*). L'exemple donné par Sophocle montre qu'il est délicat de reconstruire la texture d'un organe pour plusieurs raisons : certaines parties sont difficiles à photographier ; les photographies obtenues sont délicates à juxtaposer pour rétablir toutes les continuités ; des différences d'éclairement doivent être traitées car elles soulignent les discontinuités entre photographies.

c) Caractéristiques physiques

Si le corps doit être modélisé mécaniquement, il s'agit de pouvoir récupérer ses caractéristiques, certaines simples comme la masse, mais d'autres plus complexes comme le module d'Young ou le coefficient de Poisson. Ces paramètres ne sont pas forcément mesurables. Un autre problème lié aux conditions d'observation apparaît. La forme de l'organe obtenu est celle de l'organe soumis à la pesanteur. Or, lors de la simulation, cet organe se déformera de nouveau sous l'effet de la même pesanteur. L'idéal serait de trouver la forme de l'organe lorsqu'il n'est soumis à aucune force extérieure.

Parfois, les caractéristiques optiques de l'organe doivent être connues. Pour Sophocle par exemple, le trajet du rayon laser est théoriquement modifié par la traversée des différents milieux (air, cornée, humeur vitrée, cristallin). Ces modifications ont été négligées afin de ne pas ralentir les calculs d'intersection liés au laser d'une part, et parce que le chirurgien se base sur la trace du laser du l'œil et non sur la position du miroir à orienter, d'autre part. Dans les simulations en général, il faut, dans la majeure partie des cas, déterminer le comportement de l'organe face à la lumière diffuse et spéculaire.

Enfin, pour certains examens, des données physiques supplémentaires sont demandées et doivent donc pouvoir être quantifiées. Par exemple, les propriétés acoustiques lors d'un examen échographique ou l'absorption des rayons X, pour un examen par scanner.

2.3.2. Réalisme des images

Le réalisme des images passe par trois points importants. La restitution 3D lorsqu'elle est nécessaire, la simulation de l'éclairement puis des couleurs et textures, la précision des contours. Ces problèmes sont de nos jours bien maîtrisés, mais exigent encore maintenant du matériel dédié. Plaquer des textures sur des images de synthèse en temps réel reste coûteux. Cependant le matériel devient de plus en plus grand public [Kar95], et nous espérons pouvoir bientôt bénéficier de cette démocratisation.

Néanmoins, certains éléments intervenant dans les opérations sont plus difficiles à reproduire, qui plus est en temps réel, comme la fumée et les écoulements de fluides. De plus, des parasites ou des problèmes de mise au point peuvent entraver la qualité de l'image originale et doivent donc être reproduits. Des modèles approchés spécifiques sont donc à trouver. Enfin, certaines simulations exigent de générer des images d'un type nouveau. Les opérations par échographie en sont un exemple.

2.3.3. Simulation dynamique

a) Modèle informatique

Il s'agit de trouver comment simuler le comportement cinématique et dynamique des organes. Pour cela, on a le choix de se baser soit sur des modèles *ad-hoc*, soit sur les lois de la physique. Cette dernière solution donne naturellement les résultats les plus proches de la réalité, mais sont aussi les plus difficiles à implanter. Les méthodes numériques de simulation des comportements mécaniques sont répandues et largement utilisées en résistance des matériaux. Pourtant elles exigent des puissances de calcul considérables et leur utilisation dans un contexte temps-réel n'est envisageable qu'au prix d'un gros effort de simplification et d'approximation. De plus, on cherche un système robuste, c'est à dire qu'on ne souhaite pas exhiber des comportements divergents dus au mauvais conditionnement de la résolution des équations mathématiques.

Par ailleurs, on souhaite une simulation dynamique complète, permettant des déplacements rigides et des déformations. On doit pouvoir tirer, pousser, découper... Cela requiert de traiter les interactions mécaniques entre les objets. Ces interactions se présentent sous la forme de collisions ou de contacts. Traiter les interactions signifie d'une part détecter quand et où elles ont lieu, puis de calculer la réaction des objets.

b) Retour d'effort

En outre, il s'agit de trouver les moyens de restituer les efforts mécaniques calculés. Ce retour d'effort peut utiliser diverses techniques (électro-mécaniques, électrostatiques...). Cependant, il n'est pas toujours facile d'intégrer ces dispositifs sans gêner le respect de la prise en main de l'outil que l'on simule par rapport à la réalité. Des contraintes spatiales peuvent être à l'origine de tels problèmes. De plus, le poids ajouté n'est généralement pas complètement négligeable.

2.3.4. Protocole d'enseignement

Une interface conviviale permettant à la fois une richesse des simulations possibles et une simplicité d'utilisation est à définir pour chaque simulateur. Il s'agit aussi de définir les autres points de vue possibles, ou par exemple, les indications susceptibles d'aider l'élève lors d'un exercice. Il faut savoir que ces logiciels seront utilisés par des non habitués aux outils informatiques. Les divers environnements Windows sont les plus habituels et sont donc ceux que nous choisissons de façon préférentielle.

Une évaluation est souvent demandée en complément de ces simulateurs. Il faut cependant pouvoir être indépendant des écoles de thérapie. On évalue donc précisément ce qui est admis unanimement et on donne de simples critères pour les points à polémique.

Conclusion

Cette présentation du GIS a démontré d'une part la logique de nos travaux. Les choix des simulations que nous avons faits ont été motivés par des critères de faisabilité et de difficulté croissante. On a pu aussi définir les divers problèmes et les intervenants concernés. La répartition des tâches se fait de façon immédiate :

- Les informaticiens ont en charge la construction et la définition des modèles visuels et mécaniques et des procédés de génération de l'images ;
- Les biophysiciens (ITM) ont en charge la mesure des caractéristiques physiques des objets simulés, le choix et l'intégration des capteurs et des dispositifs de restitution de l'image générée. Il sont responsables également des outils pour le retour d'effort ;
- Les médecins ont un rôle de contrôle de la conformité au réel du simulateur ainsi que de la mise en place du protocole de formation et de la convivialité de l'interface. Ils ont en outre la tâche d'étudier l'impact des simulateurs.

Tous les aspects abordés dans la problématique exposée sont difficiles et demandent des études approfondies. Nous présentons, dans la partie suivante, notre contribution à l'un des problèmes exposés, à savoir la modélisation mécanique des organes. Plus précisément, le but de nos travaux est de concevoir un modèle temps-réel interactif (ce terme semble lourd, mais résume notre modèle de façon concise), c'est à dire un modèle basé sur la dynamique, qui peut être sollicité mécaniquement par l'utilisateur en temps réel. Cette étude représente la plus grande partie du travail effectué lors de cette thèse.

Partie 2

Modèles mécaniques d’organes

«A quoi servent les mathématiques ?
– Les mathématiques ne servent à rien,
mais c’est encore ce qui marche le mieux»
Un mathématicien contemporain

Introduction

Dans la partie précédente, nous avons montré que les simulateurs médicaux nécessitaient de simuler les interactions entre les instruments chirurgicaux et les organes opérés. Nous cherchons donc à concevoir un modèle *dynamique interactif temps-réel*. Ce modèle est *dynamique* car il se base sur les lois de la mécanique. Il est *interactif* car il doit être capable de réagir à des sollicitations des instruments contrôlés par le chirurgien. Le modèle doit non seulement pouvoir se déformer mais aussi se déchirer ou être découpé. Enfin, les calculs du modèle doivent être exécutés en *temps-réel* durant la simulation.

L'objet de cette deuxième partie est d'analyser les diverses approches possibles pour obtenir ce modèle. Nous nous concentrons sur deux problèmes difficiles : la reproduction du comportement physique et les interactions des organes entre eux et avec les instruments. La reproduction du comportement physique devant être la plus réaliste possible, nous choisissons d'utiliser les lois de la physique. D'autre part, nous étudions les problèmes des interactions entre les corps simulés, contacts prolongés ou chocs. L'étude des interactions exige d'une part de savoir les détecter, et d'autre part de savoir les traiter, c'est à dire déterminer la façon dont réagissent les corps lors de cette interaction. Les problèmes de reconstruction des propriétés physiques et d'adaptation du modèle à l'objet réel simulé ne sont pas abordés dans cette étude, même s'ils sont un argument entrant en considération dans la généricité des modèles. Cette étude est une étape qu'il conviendra d'étudier dans l'avenir.

La plupart des techniques que nous allons détailler vient de la recherche en robotique ou en animation. L'*animation* est le processus permettant de générer des images différentes qui, disposées successivement dans le temps, donnent l'impression du mouvement au spectateur. Les processus couramment employés en animation ne permettent pas l'interactivité. En outre, en animation, seules les images obtenues importent. Les moyens utilisés pour générer ces images comptent peu [AD92]. Vu le contexte de notre travail, nous choisissons de nous restreindre à l'animation temps réel basée sur la physique.

Les critères que nous considérons pour évaluer les diverses techniques sont les suivants : la conformité au réel, le nombre et le type (au sens caractéristiques physiques) de corps que l'on peut simuler par le procédé, la multiplicité d'actions mécaniques que l'on peut effectuer sur le corps, la complexité des calculs et leur adéquation aux contraintes temps-réel, la robustesse du modèle.

Le premier chapitre aborde les problèmes de modélisation de corps rigides, puis déformables. Les techniques destinées à rendre le comportement possible à simuler grâce aux procédés infor-

matiques et les méthodes de résolution des équations physiques sont abordées. En fin de chapitre, nous détaillons les modèles mécaniques déjà proposés dans le cas de la simulation d'organes, car ils ont fait l'objet ces dernières années d'une recherche intense aux Etats Unis ou en Europe.

Le chapitre suivant aborde le problème spécifique des interactions, généralement appelé *détection de collisions* dans la littérature. Ce problème concerne les phénomènes de non interpénétration, prenant la forme de chocs lorsqu'ils sont brefs sinon de contacts prolongés. Nous voyons tout d'abord leur définition, puis la façon de les détecter. Nous présentons les diverses méthodes de calculs ainsi que leurs hypothèses d'application, un calcul exact d'une collision pour une trajectoire quelconque et entre objets de forme quelconque étant impraticable dans un contexte temps réel. Ensuite, nous exposons les diverses techniques permettant d'accélérer la détection de la collision. Nous étudions ensuite comment traiter ces interactions c'est à dire calculer la réponse des objets à la collision, la façon dont ils réagissent après le choc ou durant la contrainte. Le cas des interactions extrêmes (découpes, déchirements) est également étudié, bien que peu de propositions existent aujourd'hui.

3 Modèles de corps déformables

Introduction

Le but de ce chapitre est de présenter et de classifier les diverses solutions qui ont été proposées pour la simulation de corps déformables. Même si cette étude a déjà été effectuée auparavant [Gas90][Jim93], elle n'a jamais été faite dans un contexte de simulation médicale. Pour chaque modèle, nous examinons dans quelle mesure il peut s'adapter à notre cadre de travail. Nous souhaitons en effet que le modèle choisi réponde aux critères suivants :

- il doit avoir un comportement mécanique en déplacements et en déformations conforme à la réalité. L'opérateur doit être capable de pousser, tirer, immobiliser l'organe. Il faut donc se baser autant que faire se peut sur la physique. Comme on désire que cela reste valable dans tous les cas de manipulation de l'opérateur en cours de simulation, le modèle doit être robuste. De plus, la pesanteur est à l'origine de mouvements "spontanés" des organes qui gênent le chirurgien au point de rendre certaines opérations très délicates. De ce fait, le modèle doit pouvoir être soumis à un champ gravitationnel et son comportement influencé en conséquence ;
- les propriétés de flexibilité de l'organe doivent être respectées (c'est à dire son élasticité, linéaire ou non, sa plasticité...). Le modèle doit donc pouvoir simuler plusieurs types de déformabilité ;
- l'environnement de la simulation peut contenir de nombreux objets ;
- la simulation médicale exigeant une grande précision, il est nécessaire que le contour de l'objet soit exactement défini et repérable mathématiquement pour le traitement des zones de contact ;
- le modèle doit pouvoir être modifié dans sa structure sans que son comportement en pâtisse complètement. Plus précisément, il faut pouvoir y appliquer des découpes et des déchirures ;
- il faut un réalisme visuel suffisant, une surface suffisamment lisse et éventuellement munie d'une texture ;
- le coût de calcul pour la mécanique et la restitution visuelle doit être suffisamment faible pour supporter le temps réel sur des machines graphiques de puissance standard. Le coût mémoire doit rester raisonnable.

Il apparaît immédiatement que ces critères sont opposées entre eux : de ce fait, le modèle répondant à tous ces critères n'existe pas. On veut trouver alors un compromis entre le réalisme, la vitesse d'exécution et le coût mémoire.

La première partie de ce chapitre est consacrée au formalisme de la mécanique du point tout d'abord puis des corps rigides, ce qui nous permet de préciser comment traiter les problèmes de déplacement par rotation et translation globale. Ensuite, nous étudions les divers modèles de corps déformables possibles. La classification que nous avons choisie repose sur le choix des degrés de

liberté dont le modèle est doté. Enfin, dans la dernière partie, nous présentons les diverses propositions de modélisations des organes que d'autres équipes de recherche dans le monde ont déjà proposées.

3.1. Concepts fondamentaux en mécanique

3.1.1. Le formalisme de la mécanique

La relation fondamentale de la mécanique (RFD) ou loi de Newton est bien connue et s'exprime sous la forme suivante :

$$\sum \vec{F} = m\vec{a} \tag{3.1}$$

Dans cette définition, m est un coefficient de proportionnalité appelé *masse d'inertie*. L'*accélération* $\vec{a}$ est obtenue en dérivant deux fois la position du point considéré par rapport au temps. Ainsi :

$$\vec{a} = \frac{d\vec{v}}{dt} = \frac{d^2\vec{x}}{dt^2}$$

où $\vec{v}$ est appelée *vitesse instantanée du point* et $\vec{x}$ la position du point exprimée dans un *référentiel galilléen* (en translation rectiligne uniforme par rapport à un repère absolu immobile).

Enfin $\vec{F}$ est la force qui s'applique sur le corps. Elle peut prendre différentes formes :

- la force de pesanteur $\vec{F} = m\vec{g}$ où m est la *masse inerte*, c'est à dire la quantité de matière de l'objet (en mécanique classique, elle est égale à la masse d'inertie) ;
- Les forces de viscosité $\vec{F} = -\lambda\vec{v}$ correspondant aux frottements du point dans le milieu (air ambiant) ;
- Les forces surfaciques (pression) ;
- Les forces de contact, (frottements secs, force de réaction) ;
- Les forces d'interaction, comme par exemple, les ressorts $\vec{F} = k\vec{x}$.

Les notions vues précédemment peuvent s'exprimer sous forme énergétique. On définit l'*énergie cinétique* comme suit :

$$T = \frac{1}{2}mv^2 \tag{3.2}$$

Les forces données ci-dessus peuvent parfois dériver d'une *énergie potentielle V.* On a alors la relation suivante :

$$\vec{F} = -\overrightarrow{grad}V \tag{3.3}$$

Les équations de Lagrange proposent un autre formalisme équivalent à celui de Newton. On considère le système sous forme de degrés de liberté indépendants $\{q_i(t)\}$. Si on connaît la valeur de tous ces degrés de liberté, le système est complètement caractérisé. Ainsi, on va chercher à déterminer l'évolution de la valeur des paramètres du système. Dans la suite, on utilisera la notation $\dot{q}$ pour désigner la dérivée totale par rapport au temps des paramètres (le nombre de points donne le nombre de dérivations successives).

On considère le lagrangien $L(t, q_i, \dot{q}_i) = T(t, \dot{q}_i) - V(t, q_i)$. L'expression précédente est générale, l'énergie cinétique dépendant des dérivées par rapport au temps et l'énergie potentielle ne dépendant que des positions. Le lagrangien obéit à :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q \tag{3.4}$$

Q est la *force généralisée* c'est à dire l'expression de la force en fonction des degrés de liberté.

3.1.2. La mécanique du solide

Tout corps déformable dont moins de trois points sont contraints est susceptible de pouvoir tourner autour de lui-même ou se déplacer en translation d'un point à l'autre de l'espace. Il exhibe de ce fait un comportement solide dont il faut tenir compte. Nous nous intéressons ici au cas des corps solides. Ces corps ont six degrés de liberté, trois en rotation et trois en translation. En mécanique du solide on a pour habitude de séparer l'étude des rotations et des translations en faisant intervenir le centre de masse G du solide dont la position est définie dans le solide par le barycentre des points du solide pondérés par leur masse dans le cas discret ou la densité de masse dans le cas continu. Le mouvement en translation est celui du centre de masse que l'on obtient facilement à partir des équations de Newton *(3.1)*. On considère la somme de toutes les forces appliquées au solide que l'on concentre au niveau du centre de masse. Le mouvement en rotation quant à lui, nécessite le calcul des moments des forces. Si une force $\vec{F}_A$ s'applique sur un point A du solide, on définit le moment par : $\vec{M}_A = \overrightarrow{GA} \wedge \vec{F}_A$. La relation fondamentale devient alors :

$$\sum \vec{M} = \frac{d}{dt}\left(\overset{\vdots}{I}\vec{\Omega}\right) \tag{3.5}$$

Dans cette équation, $\vec{\Omega}$ est le vecteur vitesse de rotation instantanée du solide. La matrice d'inertie $\overset{\vdots}{I}$ est obtenue de la façon suivante dans le cas discret :

$$\overset{\vdots}{I} = \begin{bmatrix} \sum m_i\left(y_i^2 + z_i^2\right) & -\sum m_i x_i y_i & -\sum m_i x_i z_i \\ -\sum m_i x_i y_i & \sum m_i\left(x_i^2 + z_i^2\right) & -\sum m_i y_i z_i \\ -\sum m_i x_i z_i & -\sum m_i y_i z_i & \sum m_i\left(x_i^2 + y_i^2\right) \end{bmatrix}$$

et dans le cas continu :

$$\overset{\vdots}{I} = \begin{bmatrix} \int (y^2 + z^2)\, dm & -\int xy dm & -\int xz dm \\ -\int xy dm & \int (x^2 + z^2)\, dm & -\int yz dm \\ -\int xz dm & -\int yz dm & \int (x^2 + y^2)\, dm \end{bmatrix}$$

Enfin, citons une formulation très intéressante permettant de caractériser l'orientation du solide. Si on appelle *q(t)* le quaternion représentant l'orientation du solide à l'instant *t* (voir *Annexe II.* pour la définition des quaternions), on peut déterminer son évolution en sachant que :

$$\dot{q}(t) = \frac{1}{2}\vec{\Omega}(t)\, q(t) \tag{3.6}$$

3.1.3. Quelques définitions

Nous avons vu comment traiter les corps solides, c'est à dire les corps dont les angles et les distances (en un mot les produits scalaires) sont constants. Les autres corps sont dits déformables ou mous. La déformation peut intervenir de diverses façons, mais ce ne sont que celles qui laissent une structure proche de l'objet initial que nous considérons. Les modélisations de liquides, gaz ou amas ne sont donc pas dans nos objectifs.

On dit qu'un corps est *élastique* si, une fois qu'il n'est plus soumis à une contrainte, il revient à un état d'équilibre stable dit *état de repos*. L'élasticité peut revêtir plusieurs formes. L'*élasticité linéaire* impose que sur une certaine plage, la déformation soit reliée linéairement à la contrainte. L'*hyper-élasticité* suppose que les forces élastiques dérivent d'un potentiel de déformation, c'est à dire qu'on peut trouver V vérifiant l'équation *(3.3)*. La *plasticité* est un cas de non élasticité puisque le corps reste déformé après la contrainte, en d'autres termes, la contrainte modifie l'état de repos.

On ne détaille pas ici la théorie de l'élasticité, on trouvera des éléments dans [Gas90] et [Bai96].

3.2. Modélisation de corps déformables

Trouver un modèle informatique d'un corps déformable revient à réduire le nombre infini de ses degrés de liberté. Les différences entre les modèles tiennent donc essentiellement dans la façon de rendre fini ce nombre, autrement dit, dans le choix de la discrétisation employée. On considère donc les modèles que l'on qualifie de *continus*, où l'on discrétise le volume de l'objet comme les *éléments finis* ou les *différences finies*, ou ceux dans lesquels on rend fini le nombre de déformations applicables à l'objet, comme par exemple, l'*analyse modale*. D'autres modèles sont conçus dès le départ comme discrets, c'est le cas des modèles *masses/ressorts*. Enfin, il reste d'autres méthodes originales qui s'éloignent des lois de la physique, mais dont le formalisme est attractif et qui procurent des solutions simples à certains problèmes.

3.2.1. Les modèles continus à discrétisation de volume

a) Les éléments finis

Principe

La méthode des éléments finis est un procédé complexe qu'il est vain de vouloir décrire complètement en quelques lignes, aussi nous ne présentons ici que les différentes étapes de sa mise en œuvre. Cette méthode a vu le jour progressivement en étant issue des méthodes variationnelles, des différences finies et de l'art de l'ingénieur, jusqu'à sa définition, précisée depuis la fin des années 1960. Elle est surtout employée en mécanique ou en thermodynamique. On trouvera dans la littérature de nombreux ouvrages traitant de cette méthode, citons par exemple [Bath82]. Son application à l'animation est relativement récente [Gou94]. Une étude spécifique a été entreprise au laboratoire [Tur96].

La structure que l'on souhaite étudier est discrétisée en volumes ou mailles élémentaires comportant un nombre connu de noeuds. Chacune de ces mailles est obtenue à partir d'un élément de référence qui peut-être par exemple un tétraèdre ou un parallélépipède. A chacune de ces mailles sont associées des fonctions d'interpolation permettant de déterminer la position de tous les points de la maille à partir de la position des nœuds. Suivant la continuité désirée, la maille parallélépipédique comporte huit nœuds (élément linéaire), ou 20 nœuds (élément quadratique)...

Afin de traiter le problème, on part alors du principe des travaux virtuels qui consiste à dire que pour de petits déplacements, le travail des forces intérieures résistantes qui s'opposent aux déplacements est égal au travail des forces extérieures qui créent les déplacements. Soit, mathématiquement :

$$\int_{V_e} {}^t\delta\vec{\varepsilon}\vec{\tau}\, dV_e = \int_{V_e} {}^t\delta\vec{u}\vec{f}^{\,V}\, dV_e + \int_{S_e} {}^t\delta\vec{u}\vec{f}^{\,S}\, dS_e + \sum_{i=1}^{n} {}^t\delta\vec{u}_i\vec{F}_i \qquad (3.7)$$

Dans cette expression, le vecteur $\delta\vec{\varepsilon}$ représente les petites déformations dues aux déplacements $\delta\vec{u}$, $\vec{\tau}$ représente les contraintes internes, $\vec{f}^{V}$ sont les forces volumiques, $\vec{f}^{S}$ sont les forces surfaciques et $\vec{F}_i$ les forces ponctuelles.

Cette équation est intégrée sur le volume en sommant le calcul sur les mailles. La fonction d'interpolation permet un calcul rapide de l'intégrale sur chaque maille, dans son repère local.

On peut faire deux hypothèses sur le matériau simulé.

- Soit il obéit aux lois de l'élasticité linéaire et le système revient à la résolution de :

$$\bar{\bar{K}}\vec{U} = \vec{R} \tag{3.8}$$

Dans cette équation, $\vec{U}$ représente l'ensemble des déplacements des noeuds. Si la structure comporte N nœuds, le vecteur contient $3N$ lignes (en supposant trois degrés de liberté par nœuds). $\vec{R}$ est l'ensemble des forces s'exerçant sur les nœuds.

- Soit il obéit aux lois de l'élasticité non linéaire. Ce cas apparaît soit à cause des propriétés physiques du matériau, soit parce que l'on applique des grands déplacements à la structure. Dans ce cas, L'équation d'équilibre obtenue devient :

$$\bar{\bar{K}}\left(\vec{U}\right)\vec{U} = \vec{R} \tag{3.9}$$

Cette équation n'étant plus linéaire, sa résolution nécessite des itérations successives jusqu'à convergence. Cela revient à considérer que la contrainte s'effectue progressivement au cours du temps.

Les équations *(3.8)* et *(3.9)* peuvent également s'exprimer sous forme dynamique en sommant le terme d'inertie $\bar{\bar{M}}\ddot{\vec{U}}$ et éventuellement un terme d'amortissement $\bar{\bar{D}}\dot{\vec{U}}$ dans les termes de gauche.

Exemple : cas d'une main agrippant une balle

Une mise en œuvre intéressante a été pratiquée par [GTT89]. Leur objectif est de simuler la pression d'une main sur une balle flexible. Pour cela, la balle est modélisée par éléments finis avec continuité d'ordre 0 en élasticité linéaire. On considère qu'à chaque instant, le système est dans un état d'équilibre. L'ensemble de la simulation est piloté par le déplacement de la main. Ainsi, on partitionne les points de la balle en deux ensembles : on distingue les nœuds en contact et ceux qui ne le sont pas. Ainsi, l'équation *(3.8)* est divisée en deux en tenant compte de cette partition :

$$\begin{bmatrix} \bar{\bar{K}}_{uk} & \bar{\bar{K}}_{kk} \\ \bar{\bar{K}}_{uu} & \bar{\bar{K}}_{ku} \end{bmatrix} \begin{bmatrix} \vec{U}_U \\ \vec{U}_k \end{bmatrix} = \begin{bmatrix} \vec{R}_k \\ \vec{R}_u \end{bmatrix} \tag{3.10}$$

L'indice k (pour *known*) se réfère au données connues (déplacements ou contraintes), l'indice u (pour *unknown*) se réfère aux données inconnues. Aux déplacements inconnus, on associe des forces connues (forces de contraintes internes) et aux déplacements connus (les points de contact) on cherche à déterminer la résistance à la contrainte. Afin d'éviter les interpénétrations, lorsqu'elles sont détectées, on force les points de la balle à de déplacer suivant la normale à la facette de la main avec laquelle elle est entrée en collision.

L'algorithme décrit est le suivant :

Calculer le mouvement du squelette
En déduire les déplacements connus (des points du squelette).

TQ non(convergence) et non(dépassement de seuil) FAIRE
    Evaluer le déplacement des points de la peau.
    En évitant les recouvrements, évaluer la position des points de la balle en contact.
    Résoudre le système permettant de trouver le déplacement des points de la balle et la force de réaction

aux points de contact.
Inverser les forces de réaction afin de déterminer les forces aux points de la peau en contact.
A l'aide des forces précédemment calculées, évaluer le déplacement des points de la balle et du squelette
et les forces de réactions sur le squelette.
FTQ

Par ailleurs un modèle éléments finis est également employé pour les doigts en contact avec la balle afin de simuler proprement les déformations des doigts. La force que l'on fournit à ce modèle est la force des nœuds de la balle en contact que l'on répartit sur les sommets de la facette du doigt en collision. Par contre, si les doigts ne sont pas en contact, on peut employer des modèles de déformation plus rapides, qui ne sont pas obligatoirement basés sur la physique.

Discussion

Les éléments finis sont un modèle très proche des équations physiques, c'est la solution la plus couramment employée lorsqu'il s'agit de résoudre des problèmes de mécanique des milieux continus. Malheureusement, elle s'avère être une méthode très lourde : elle nécessite à la fois une grande puissance de calcul et une importante quantité de mémoire. De plus, elle peut être employée de multiples façons :

- traitements linéaires ou non ;
- étude *dynamique* ou *quasi-statique* (chaque instant est un état d'équilibre) ;
- éléments surfaciques ou presque (cas des coques), ou au contraire volumiques mais avec un choix de géométries variées (parallélépipèdes, tétraèdres...) ;
- fonctions d'interpolation de continuité quelconque (0, 1 ou 2) ;
- noeuds de zéro à six degrés de liberté.

Les hypothèses les plus restrictives donnent les résultats les moins complexes et les plus rapides. Ainsi on peut préférer les éléments de continuité 0, avec seulement trois degrés de liberté en translation par nœud, en élasticité linéaire, dans des états quasi-statiques. Les résultats obtenus sont naturellement très approchés.

b) Méthodes à différences finies

Modèle hyperélastique non linéaire

Ce modèle a été proposé par [TPBF87] et s'appuie sur des données physiques. Soit un corps Ω. A des fins de simplification, on suppose que Ω est un corps à trois dimensions. Tout point de Ω est repéré, de façon paramétrique, par trois coordonnées $a = (a_1, a_2, a_3)$. Les points sont repérés dans l'espace dans un repère global par leur vecteur position $\vec{r} = (r_1(a,t), r_2(a,t), r_3(a,t))$. Le comportement de l'objet est régi par l'équation suivante :

$$\frac{d}{dt}\left(\mu\frac{\partial}{\partial t}\vec{r}\right) + \gamma\frac{\partial}{\partial t}\vec{r} + \overrightarrow{grad}_{\vec{r}}\,\varepsilon(\vec{r}) = \vec{f}(\vec{r}, t) \qquad (3.11)$$

Avec $\mu(a)$ la densité du corps au point a. $\gamma(a)$ la densité de frottements, et $\vec{f}(\vec{r}, t)$ l'ensemble des forces extérieures au corps. Enfin, $\varepsilon(\vec{r})$ représente le potentiel des forces élastiques intérieures.

En effet, cette solution se place dans un modèle d'hyperélasticité, c'est à dire que l'on considère que les forces élastiques dérivent d'un potentiel. Il s'agit alors de définir l'énergie potentielle

d'élasticité. Pour cela, on s'appuie sur la géométrie. Un solide est complètement caractérisé par son tenseur métrique défini par :

$$G_{ij}(\vec{r}(a)) = \frac{\partial}{\partial a_i}\vec{r} \cdot \frac{\partial}{\partial a_j}\vec{r}$$

où le point désigne le produit scalaire. Un potentiel élastique peut alors être obtenu de manière naturelle par le carré de la différence entre la configuration courante du corps et la configuration de repos. Soit :

$$\varepsilon(\vec{r}) = \int_{\Omega} \left\| \bar{\bar{G}} - \bar{\bar{G}}^0 \right\|_{\alpha}^2 da_1 da_2 da_3$$

Dans la norme matricielle utilisée, les coefficients permettent d'ajuster les caractéristiques physiques de l'élément, sa résistance à l'allongement, à la courbure... L'expression du gradient de cette énergie est simplifiée afin d'obtenir l'expression des forces élastiques. Le calcul se fait ensuite par discrétisation. Le corps est discrétisé à intervalle constant selon les paramètres a_i. On forme alors le vecteur $\vec{u}$ regroupant les coordonnées On définit alors des opérateurs de dérivation spatiale. L'équation de la dynamique peut ainsi s'exprimer sous la forme d'une équation matricielle :

$$\bar{\bar{M}}\frac{d^2}{dt^2}\vec{u} + \bar{\bar{C}}\frac{d}{dt}\vec{u} + \bar{\bar{K}}(\vec{u})\vec{u} = \vec{f} \qquad (3.12)$$

Ensuite, une discrétisation temporelle est effectuée, qui permet, connaissant la valeur de $\vec{u}_t$ et $\vec{u}_{t+\Delta t}$ respectivement aux instants t et $t+\Delta t$, de trouver la valeur $\vec{u}_{t+2\Delta t}$, par résolution d'un système linéaire.

Cette méthode se restreint à l'hyperélasticité, mais la variété des objets que l'on peut simuler reste tout de même assez large. Comme l'équation *(3.12)* n'est en général pas linéaire, il faut calculer à chaque pas de temps la matrice $\bar{\bar{K}}(\vec{u})$, sauf si on considère un potentiel d'élasticité linéaire.

Cette méthode, reconnaissent les auteurs, est malheureusement mal conditionnée pour le traitement des objets assez rigides, car alors la résolution ne converge plus. Selon eux, cela vient du fait que le modèle n'est pas linéaire et que de ce fait, la force élastique dépend de la position absolue des points. Enfin le modèle n'est pas non plus satisfaisant car la formulation hyperélastique employée dépend de la paramétrisation de la forme de l'objet. Cela revient à dire que les propriétés physiques du modèle sont conditionnées par des choix mathématiques de représentation de l'objet.

Modèle hyperélastique linéaire avec composante de référence

Dans le modèle précédent, l'utilisation de l'élasticité non linéaire entraîne des problèmes de non convergence des équations lorsque la rigidité du corps augmente. C'est pourquoi [TW88] proposent alors un nouveau formalisme, plus restrictif, mais plus sûr et plus simple, où le comportement rigide est déterminé séparément de l'élasticité.

Un corps déformable est modélisé par une configuration de référence qui obéit aux lois de la physique des solides non déformables et par sa déformation par rapport à ce solide. Les déformations sont donc caractérisées par un déplacement par rapport au solide de référence, ces déplacements obéissant aux lois de l'élasticité linéaire. En élasticité linéaire, l'expression de l'énergie et la force de déformation ne dépendent plus de la position absolue du système, mais seulement de l'écart par rapport à la position de repos, ce qui permet d'obtenir une système plus stable lorsque la rigidité augmente. Il n'est pas nécessaire de rendre l'énergie indépendante de l'orientation et la position de l'objet dans son ensemble. Ainsi, la contrainte sur l'expression de l'énergie de déformation est moins importante puisqu'il suffit qu'elle soit nulle lorsqu'il n'y a pas de déformation et croissante si la déformation augmente.

Le formalisme employé est le suivant : on considère $\vec{c}(t)$ la position du barycentre de l'objet. Soit $d\vec{\theta}(t)$ le vecteur représentant la direction et la valeur de l'angle de la rotation instantanée subie par le corps. On pose :

$$\vec{v}(t) = \frac{d}{dt}\vec{c} \qquad\qquad \vec{\omega}(t) = \frac{d}{dt}\vec{\theta}$$

respectivement la vitesse du centre de masse et la vitesse angulaire instantanée du solide.

Un point du corps déformé est obtenu par addition d'une position $\vec{r}$ sur le solide de référence et d'un déplacement $\vec{e}$ par rapport à cette position. Dans un repère lié au centre de masse, on obtient $\vec{q} = \vec{r} + \vec{e}$. Dans le repère absolu, c'est à dire en tenant compte de la position : $\vec{x} = \vec{c} + \vec{q}$.

En dérivant les équations de Lagrange régissant le système, on obtient les équations suivantes :

$$m\dot{\vec{v}} + \frac{d}{dt}\int \mu \dot{\vec{e}}\, du + \int \gamma \dot{\vec{x}}\, du = \vec{f}^{v} \tag{3.13}$$

$$\frac{d}{dt}\left(I\vec{\omega}\right) + \frac{d}{dt}\int \mu\left(\vec{q} \wedge \dot{\vec{e}}\right) du + \int \gamma\left(\vec{q} \wedge \dot{\vec{x}}\right) du = \vec{f}^{\omega} \tag{3.14}$$

$$\mu\ddot{\vec{e}} + \mu\dot{\vec{v}} + \mu\vec{\omega} \wedge (\vec{\omega} \wedge \vec{q}) + 2\mu\left(\vec{\omega} \wedge \dot{\vec{e}}\right) + \mu\left(\dot{\vec{\omega}} \wedge \vec{q}\right) + \gamma\dot{\vec{x}} + \overrightarrow{grad}_e \varepsilon = \vec{f} \tag{3.15}$$

L'équation *(3.13)* représente le mouvement de translation du corps, dans son ensemble. L'équation *(3.14)* exprime l'évolution de l'orientation du corps. Enfin, l'équation *(3.15)* est l'équation régissant le comportement élastique du corps. Les forces dans les membres de droite sont les expressions des forces généralisées. $\vec{f}^{v}$ est la somme des forces appliquées au système, $\vec{f}^{\omega}$ est la somme des moments de ces forces.

Dans la dernière équation, un potentiel décrivant l'élasticité du système doit être trouvé et on désire qu'il soit linéaire. Le potentiel proposé, en dimension *d*, s'inspire de la théorie des *splines*. Il vaut :

$$\varepsilon = \frac{1}{2}\sum_{m=0}^{p}\sum_{|j|=m}\frac{m!}{j_1!\ldots j_d!} w_j(u)\left|\delta^m_j\vec{e}\right|^2$$

avec $j = \{j_1, \ldots, j_d\}$, où $|j| = j_1 + \ldots + j_d$ et la notation :

$$\delta^m_j\vec{e} = \frac{\delta^m\vec{e}}{\delta u_1^{j_1}\ldots\delta u_d^{j_d}}$$

et où les $w_j(u)$ servent à contrôler la résistance du corps aux étirements, cisaillements...

Une forme simplifiée des forces qui en résultent est proposée :

$$\overrightarrow{grad}_e \varepsilon = \sum_{m=0}^{p}(-1)\,\Delta_{wm}^{m}\vec{e}$$

avec :

$$\Delta_{wm}^{m}\vec{e} = \sum_{|j|=m}\frac{m!}{j_1!\ldots j_d!}\delta^m_j\left(w_j(u)\,\delta^m_j\vec{e}\right)$$

Dans l'implantation choisie par les auteurs, $p = 2$. Les calculs sont effectués de la même manière que dans le cas non linéaire, c'est à dire par une semi-discrétisation spatiale suivie d'une discrétisation temporelle et la résolution d'un système linéaire. Notons que dans le cas élastique, les termes intervenant dans l'équation différentielle sont constants.

Ce modèle a été complété par [TF88] en y ajoutant la possibilité de simuler :

- les phénomènes de plasticité (modification de l'état de repos), en modifiant la forme de la composante de référence ;

- la viscoélasticité par enrichissement de la fonction potentielle ;
- les phénomènes de fracture lorsqu'un seuil de tension est dépassé, auquel cas on annule les coefficients $w_j(u)$ impliqués.

Discussion

La formulation de ces deux problèmes, hyperélasticité ou élasticité linéaire est très proche de celle des éléments finis, et même identique dans la dernière étape de résolution. La différence tient essentiellement dans la façon de calculer les matrices de masses, d'amortissement et de raideur dans l'équation semi-discrète régissant le comportement du corps. La complexité étant équivalente à celle des éléments finis, on retrouve la lourdeur des calculs liés à la construction des matrices (mais dans le cas linéaire, elle est faite une fois pour toute), la quantité de mémoire nécessaire au stockage des matrices et enfin la complexité au minimum quadratique de la résolution finale.

3.2.2. Les modèles à nombre de déformations fini

L'idée de base est de soumettre des modèles géométriques continus à un jeu de déformations connues. Par exemple, des courbures, des torsions... Ces déformations peuvent avoir une expression complexe mais que l'on connaît, et dont seule l'amplitude est paramétrable. Cette amplitude est déterminée par les lois de la physique. Ainsi, on pourra dire par exemple que l'objet peut être soumis à une courbure dont l'amplitude est quantifiée par les lois de la mécanique afin que l'objet s'adapte le mieux à la contrainte qui l'oblige à se courber. L'avantage majeur de cette méthode est de réduire considérablement le nombre de degrés de liberté du système, puisque les seuls paramètres sont les amplitudes du jeu de transformation que l'on se donne. Paradoxalement, cette réduction est aussi l'inconvénient du système puisqu'il faudra avoir un jeu suffisamment important de transformations possibles pour obtenir un système ayant un comportement réaliste.

a) Modèles à déformations globales

[WW90] considèrent des transformations de l'espace particulières, où, à un vecteur $\vec{x} = (x_1,x_2,x_3)$, on fait correspondre un vecteur, $\vec{X} = (X_1,X_2,X_3)$, son image par :

$$X_i = \sum_j R_{ij} p_j(\vec{x})$$

avec $\vec{p}$ un vecteur fonction polynomiale des coordonnées du point, par exemple :

$$\vec{p}(\vec{x}) = \left[1, x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3, x_1^2, x_2^2, x_3^2\right]$$

En dérivant on obtient la vitesse :

$$\dot{X}_i = \sum_j \dot{R}_{ij} p_j$$

L'énergie cinétique totale d'un système contenant n particules subissant la transformation est obtenue de façon immédiate par la formule suivante (ici m, $\vec{x}$ et en conséquence $\vec{p}$ et $\vec{X}$, sont implicitement dépendants de la particule n, on choisit de ne pas surcharger l'écriture) :

$$T = \sum_n T_n = \sum_n \sum_i \frac{1}{2} m X_i^2 = \frac{1}{2} \sum_i \sum_j \sum_k \dot{R}_{ij} \dot{R}_{ik} M_{jk}$$

où la matrice $\overline{\overline{M}}$ est définie comme :

$$M_{jk} = \sum_n m p_j p_k$$

On utilise alors les équations de Lagrange *(3.4)*, en considérant que le lagrangien est égal à l'énergie cinétique seule (sous réserve de convertir tout potentiel en force), soit $L = T$. Les degrés

de liberté du système sont les seuls éléments de l'énergie cinétique susceptibles de varier, à savoir la valeur des coefficients R_{rs}. On démontre que :

$$\frac{\partial L}{\partial \dot{R}_{rs}} = \sum_k \dot{R}_{rk} M_{ks}$$

puis que :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{R}_{rs}}\right) = \sum_k \ddot{R}_{rk} M_{ks}$$

De plus:

$$\frac{\partial L}{\partial R_{rs}} = 0$$

Donc, les équations de Lagrange deviennent, pour chaque r et s :

$$\sum_k \ddot{R}_{rk} M_{ks} - Q_{rs} = 0$$

Soit sous forme matricielle :

$$\ddot{\bar{\bar{R}}}\bar{\bar{M}} = \bar{\bar{Q}} \qquad (3.16)$$

La matrice $\bar{\bar{Q}}$ de force généralisée s'obtient par définition :

$$Q_{rs} = \sum_i f_i \frac{\partial X_i}{\partial R_{rs}} = f_r p_s$$

En inversant la matrice $\bar{\bar{M}}$, qui est constante, on résout le système *(3.16)*, et on trouve la matrice $\ddot{\bar{\bar{R}}}$ que l'on intègre deux fois par rapport au temps afin de caractériser la transformation que le solide subit à l'instant *t*. Les forces généralisées contiennent non seulement les forces extérieures au système, mais aussi les force de déformation internes. L'approche par tenseur métrique exposée dans le modèle de Terzopoulos au paragraphe *3.2.1* est une bonne solution pour exprimer un potentiel interne de déformation.

b) Analyse modale

[PW89] ont expérimenté une solution en se basant sur l'*analyse modale*. Ce calcul consiste à déterminer les modes de vibrations d'une structure soumise à une contrainte. La méthode généralement employée s'appuie sur une décomposition éléments finis de la structure, comme on l'a exposé aux paragraphes précédents. La formule que l'on obtient est du type :

$$\bar{\bar{M}}\ddot{\vec{u}} + \bar{\bar{D}}\dot{\vec{u}} + \bar{\bar{K}}\vec{u} = \vec{f} \qquad (3.17)$$

Où $\vec{u}$ désigne le vecteur déplacement des nœuds de la structure (de dimension $3N$ pour N nœuds). A des fins de simplification, on considère que la matrice d'amortissement $\bar{\bar{D}} = s\bar{\bar{M}}$, avec $s \in \Re$. Comme $\bar{\bar{M}}$ et $\bar{\bar{K}}$ sont définies positives symétriques, il existe une base commune de diagonalisation de ces deux matrices. Cette base est obtenue grâce à l'équation aux valeurs propres :

$$\lambda\vec{\phi} = \bar{\bar{M}}^{-1}\bar{\bar{K}}\vec{\phi} \qquad (3.18)$$

En considérant la nouvelle base formée des $\{\vec{\phi}_i\}$ solutions de l'équation *(3.18)*, on peut réécrire une nouvelle équation de même type que *(3.17)*, mais en $\vec{u}$, $\bar{\bar{M}}$, $\bar{\bar{D}}$ et $\bar{\bar{K}}$. Ces matrices étant diagonales, il est facile de résoudre les composantes de $\vec{u}$. Signalons que les solutions λ_i sont proportionnelles aux diverses *fréquences naturelles* (fréquence d'oscillation sans amortissement) des modes de vibration.

La méthode consiste alors à supprimer les modes dont les fréquences sont trop élevées. En effet, ces modes correspondent à de petites déformations (faible amplitude) ayant des fréquences très élevées : elle n'influencent pas ou peu le comportement global de l'objet. En inspectant les valeurs propres les plus petites, on sélectionne facilement les modes de fréquence plus faibles. Outre le gain en calcul, ce choix limite l'aliassage temporel, car les déplacements et déformations conservés sont de fréquence peu élevée : seuls les mouvements lents sont calculés au détriment des vibrations rapides et de faible amplitude.

Les modes de vibration sont traduits en transformations *polynomiales* de l'espace (applications $\Re^3 \rightarrow \Re^3$). Ces transformations sont appliquées à toutes la structure comme détaillé dans [Bar84]. Cela permet d'étendre le calcul des déformations à toute la structure et non pas aux seuls nœuds de la discrétisation spatiale. Ils proposent plus tard d'employer les fonctions de Garlerkin en remplacement des transformations polynomiales, afin de caractériser l'énergie de déformation de leur modèle [ESP92]. Enfin, les auteurs préconisent l'emploi de *superquadriques* mais seulement à des fins de calcul de collision.

c) Modèle à déformations globales et locales

Tout d'abord, [SP91] proposent de modifier le modèle précédent afin de tenir compte des déformations locales. Les surfaces de base choisies sont les superquadriques (comme elles sont implicites, elles sont plus pratiques pour la détection des collisions, notent les auteurs). Leurs formes restent très variées et de ce fait, le choix des superquadriques n'est pas trop restrictif. Lors d'une animation ou simulation, elles subissent des translations ou rotations, ainsi que d'autres matrices de transformation (par exemple obtenus à l'aide de la méthode des vibrations libres en analyse modale, ou par d'autres méthodes donnant des transformations globales). Les surfaces décrites sont alors de la forme :

$$\vec{\underline{x}} = \overline{\overline{M}}\,\overline{\overline{D}}_u \vec{x} + \vec{b}$$

Où $\vec{x}$ décrit la surface centrée et non déformée, $\overline{\overline{D}}_u$ est la matrice de déformation, $\overline{\overline{M}}$ la matrice de rotation et $\vec{b}$ la translation. C'est dans ce formalisme que les auteurs ajoutent le concept de carte de déplacements, à savoir que l'on rajoute en chaque point de la quadrique un déplacement :

$$\vec{\underline{x}} = \overline{\overline{M}}\,\overline{\overline{D}}_u \vec{x} + b = \overline{\overline{M}}\,\overline{\overline{D}}_u (\vec{x} + d\vec{n}) + \vec{b}$$

où d est une fonction de déplacement dépendant des 2 paramètres décrivant la surface et $\vec{n}$ la normale à la surface non déformée au point $\vec{x}$. Malheureusement, les auteurs ne précisent pas comment faire dépendre cette carte de déplacement des lois de la mécaniques. En effet, leur but premier est de faire de la reconstruction d'objets réels en modelant une super-quadrique.

d) Modèle complet

[MT92] unifient toutes les approches précédentes. Les modèles proposés sont constitués d'une composante déformable, obtenue par déformation globale (du type de celle de [WW90] exposée auparavant) d'un solide de référence (configuration au repos) qui est une superquadrique. Cette composante est ensuite déformée par des déplacements locaux de points obtenus par échantillonnage de la surface et interpolation par fonctions de forme issues de la théorie des éléments finis. Enfin le solide subit des transformations de rotation et de translation afin de le placer au bon endroit suivant la bonne orientation.

Ensuite, la mécanique lagrangienne permet de poser les équations qui régissent le mouvement du corps. Les états (degrés de liberté) du système sont les suivants : les coordonnées du centre de masse, les coordonnées de rotation du corps, les paramètres de la déformation globale et enfin les valeurs des déplacements aux noeuds. On intègre ces équations afin de calculer le comportement complet de l'objet au cours de l'animation ou de la simulation.

Discussion

Toutes ces méthodes ont l'avantage de réduire de façon drastique le nombre de degré de liberté du système. Seuls quelques paramètres décrivent le comportement du système, les calculs étant fonction du nombre de variables à intégrer, s'en trouvent fortement allégés. Pourtant, il semble que ces méthodes basés sur un choix de déformations possibles soient mal adaptées aux déformations locales. En effet, le nombre de paramètres décrivant une déformation globale peut être minimum, par exemple, seulement neuf paramètres décrivent toutes les transformations linéaires possibles. Par contre, les déformations locales nécessitent d'échantillonner la surface et d'étudier la déformation en chaque nœud du maillage obtenu d'où une démultiplication des degrés de liberté.

3.2.3. Les modèles discrets

On regroupe sous cette dénomination les méthodes ne partant pas d'une structure continue préalable, mais qui se basent sur une modélisation par assemblage d'éléments discrets. L'illustration la plus connue de ce principe sont les maillages masses/ressorts. Ces derniers peuvent être employés de façons différentes, aussi nous étudions dans la suite quelques exemples de mise en œuvre. La formalisation est ensuite exposée à partir des travaux de l'ACROE sur le langage CORDIS/ANIMA.

a) Principe de base

Les modèles masses/ressorts supposent un ensemble de particules caractérisé par leur masse m_i et durant la simulation par leur position $\vec{x}_i(t)$. Un graphe est formé sur ces particules, chaque liaison du graphe représentant un ressort. En d'autres termes, deux particules connectées dans le graphe sont en interaction mécanique. La simulation dynamique repose sur la RFD *(3.1)*. Pour chaque particule, on fait le bilan des forces. Ces forces peuvent être la gravité, les forces de frottement visqueux dans le milieu ambiant, ainsi que les forces dues aux liaisons. La force exercée sur la particule *i*, due à un ressort de longueur de repos l_0 et de raideur *k* reliant *i* à la particule *j*, est égale à :

$$\vec{F}^{Elast}_{j \to i} = k\left(\left\| \vec{x}_i - \vec{x}_j \right\| - l_0 \right) \frac{(\vec{x}_i - \vec{x}_j)}{\left\| \vec{x}_i - \vec{x}_j \right\|} \tag{3.19}$$

Si la liaison est visco-élastique, on lui ajoute le terme suivant :

$$\vec{F}^{Visc}_{j \to i} = \lambda \frac{d}{dt} (\left\| \vec{x}_i - \vec{x}_j \right\|) \frac{(\vec{x}_i - \vec{x}_j)}{\left\| \vec{x}_i - \vec{x}_j \right\|} \tag{3.20}$$

Le principe d'action réaction nous donne :

$$\vec{F}^{Elast}_{j \to i} = -\vec{F}^{Elast}_{i \to j} \qquad \vec{F}^{Visc}_{j \to i} = -\vec{F}^{Visc}_{i \to j}$$

Il existe d'autres types de ressorts comme les ressorts angulaires qui sont employés parfois même s'ils requièrent des puissances de calcul importantes. En effet, il est possible de retranscrire le comportement de ressorts angulaires grâce à des ressorts linéaires, mais cela impose d'ajouter des points qui viennent perturber la répartition locale de la masse.

Une fois la RFD énoncée pour chaque particule, on déduit l'accélération que l'on intègre afin de déterminer la vitesse puis la position de la particule concernée. Les méthodes d'intégration sont diverses. La plus employée est la méthode d'Euler dont le principe revient à faire une approximation au premier ordre de la vitesse et au second ordre de la position. On pose, pour chaque particule *i* :

$$\vec{v}_i^{\,t} = \vec{v}_i^{\,t-\Delta t} + \vec{a}_i^{\,t} \Delta t = \vec{v}_i^{\,t-\Delta t} + \frac{\vec{F}_i^{\,t}}{m_i} \Delta t \tag{3.21}$$

Puis :

$$\vec{x}_i^{\,t} = \vec{x}_i^{\,t-\Delta t} + \vec{v}_i^{\,t}\Delta t \qquad (3.22)$$

Cette méthode d'intégration est la plus simple, mais c'est aussi l'une des moins stables numériquement. Nous reviendrons par la suite sur les problème posés par l'intégration et la simulation dynamique (voir paragraphe *3.3.2*).

b) Mise en œuvre particulière : simulation du déplacement d'un serpent

[Mil88] propose une modélisation par ressorts du mouvement des vers et des serpents. Chaque segment du ver ou du serpent est remplacé par un cube de points massiques, relié sur les arêtes et les diagonales par des ressorts. Les ressorts d'arêtes et diagonaux contrôlent le module d'Young de l'étirement, tandis que seuls les ressorts diagonaux affectent les modules de torsion et de cisaillement. L'intégration est faite par la méthode d'Euler.

Le mouvement s'appuyant sur l'interaction entre le corps et le sol, le modèle tient compte des collisions entre les deux. Le sol est modélisé par un plan rigide. On calcule l'intersection P_{tt} entre le plan du sol et le segment $[P^t P^{t+\Delta t}]$ représentant la trajectoire du point issue de la résolution des équations du mouvement. La position du point $P'^{t+\Delta t}$ est calculée ainsi :

$$P'^{t+\Delta t} = P^t + \vec{R} \cdot (P^{tt} - P^{t+\Delta t})$$

où $\vec{R}$ est un vecteur composé du coefficient de réflexion relatif à la direction concernée (normale au plan, ou tangentielle). Le mouvement est alors simple à commander : on utilise une force de mouvement interne et des frictions. Lorsque les ressorts le long du corps se contractent, le point situé le plus en arrière avance, alors que le point situé à l'avant reste fixe. Au contraire, lorsque le ressort s'allonge, le point à l'arrière reste fixe alors que le point à l'avant avance. Cela permet de simuler l'avancée en deux temps du serpent. Le réalisme est alors renforcé en ajoutant un déphasage dans le comportement des ressorts afin de ne pas avoir un mouvement uniforme sur tout le corps du serpent. De plus, une progression ondulatoire horizontale est ajoutée, avec un déphasage de 180° entre la droite et la gauche.

c) Modélisation surfacique : modèle OSEA

[Gas89] propose un nouveau modèle de corps déformables agencé de la manière suivante : autour d'un noyau rigide sont disposés des ressorts, dont l'axe est fixe. La *figure 3.1* montre la coupe d'un tel modèle. Le noyau rigide obéit aux équations de la mécanique des solides (comme exposée au paragraphe *3.1.2*), les forces extérieures au solide incluent la gravité ainsi que la force exercée par ces ressorts. Les équations pilotant le solide sont intégrées par la méthode d'Euler.

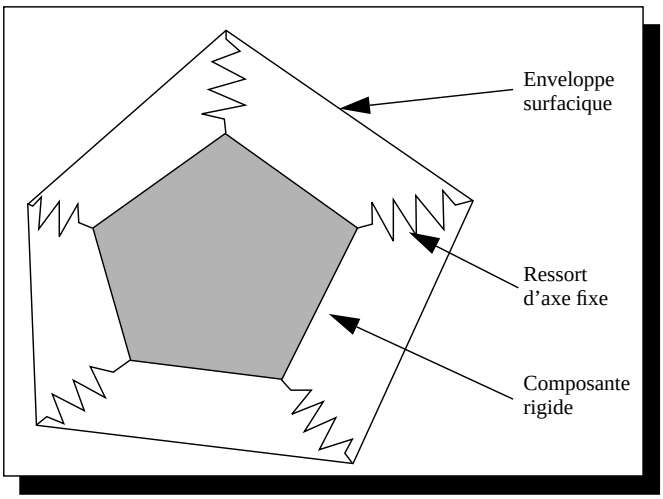

Figure 3.1. *Modèle OSEA*

Les déformations de l'objet sont calculées à partir de l'équation régissant la longueur du ressort et en considérant que l'axe du ressort est fixe. Si l_0 est la longueur du ressort au repos, l_{t1} la longueur du ressort à l'instant t_1, alors, sous réserve que le ressort soit laissé libre, on obtient l'évolution suivante :

$$l^t = l_0 + \left(l_0 - l^{t1} \right) e^{-(t-t_1)\gamma} \cos\left(m_R k_R (t - t_1) \right)$$

Cela ne suffit pas, car on obtient uniquement des déformations locales au niveau du seul ressort déformé. Il reste à propager la déformation. Dans ce but, pour chaque ressort, on définit une zone de propagation, c'est à dire la liste des ressorts voisins affectés par une déformation. La déformation des ressorts voisins est calculée grâce à des contraintes géométriques. Toutes les combinaisons linéaires de matériaux compris entre les deux extrêmes *volume constant* ou *surface extérieure constante* sont possibles. Pour des déformations à surface constante, il n'y a pas toujours de solutions, on cherche à minimiser l'erreur. Dans le cas de solutions multiples, celles qui minimisent le déplacement sont choisies. Un autre modèle de propagation est proposé, plus intuitif, où la force de déformation est compensée par une force de propagation, égale à la première multipliée par un coefficient. On applique une partie de cette force à chaque ressort de la zone de propagation.

L'auteur a choisi de ne pas mettre des ressorts au niveau de la surface, car le contrôle de la déformation devient plus difficile (impossibilité de garantir un volume ou une surface constante), et les équations sont plus délicates à résoudre. La surface extérieure est employée pour le tracé de la structure. Les extrémités des ressorts servent alors de points de contrôle pour un tracé de surfaces *splines*.

d) Modélisation volumique

[Jou95] propose une approche intéressante pour les modèles masses/ressorts volumiques. Il construit son modèle en se basant sur les éléments de base que sont les ressorts linéaires impliquant deux particules, les ressorts angulaires de torsion impliquant trois particules ainsi que les ressorts angulaires de jointure, impliquant quatre particules (voir *figure 3.2*).

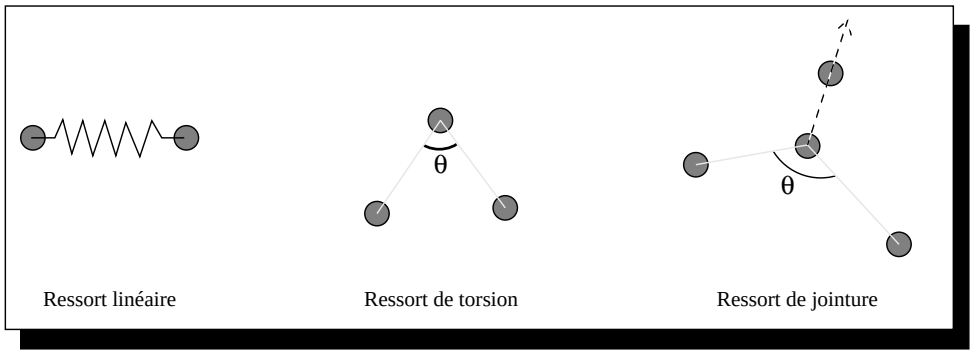

Figure 3.2. *Types de ressorts possibles*

Cette dernière catégorie de ressorts sert à éviter la symétrie des ressorts de torsion (angles positifs et négatifs équivalents).

La première version employée [JBL94] était basée sur le système CORDIS-ANIMA (exposé au paragraphe suivant). La version suivante est une modélisation volumique. Son modèle est intéressant par le procédé de construction qu'il propose. En effet, en se basant sur la géométrie de l'objet, il propose un moyen de découper la structure et de positionner en conséquence les nœuds, afin de respecter au maximum la symétrie de l'objet. Ce choix est justifié par la volonté de positionner les nœuds en fonction des axes principaux du corps à modéliser (cette condition permet de respecter au mieux le comportement durant la simulation). Les étapes pour le découpage sont les suivantes, suivant l'objet à discrétiser :

- Polygones : la facette est discrétisée sur son contour. Si c'est un quadrilatère, on balaye deux côtés opposés le long des deux autres côtés. Si le polygone contient plus de quatre côtés, on

cherche l'isobarycentre de la structure et on construit les secteurs autour de ce barycentre, le long du polygone ;

- Polyèdres : on discrétise ses faces, puis on cherche l'isobarycentre du polyèdre. On trace alors les lignes joignant le barycentre aux points des faces discrétisées ;
- Primitives obtenues par balayage : dans ce cas particulier, on discrétise de façon identique les deux faces opposées et on balaye la discrétisation entre les deux.

Les masses des nœuds sont déterminées ensuite en considérant que la matrice d'inertie doit être égale à celle de l'objet réel. Pour cela, diverses méthodes sont employées afin de subdiviser la structure en tétraèdres. En effet, dans le tétraèdre, le centre de masse est identique au centre géométrique et la solution du problème de respect de la matrice d'inertie possède une solution unique. La méthode d'intégration employée en cours de simulation est basée sur les développements limités, et est de ce fait un peu plus précise que la méthode d'Euler. Si on tient compte du second ordre dans l'expression de la position, on obtient un terme supplémentaire dans l'expression *(3.22)* :

$$\vec{x}_i^{\,t+\Delta t} = \vec{x}_i^{\,t} + \vec{v}_i^{\,t}\Delta t + \frac{\vec{F}}{2m_i}\Delta t^2$$

Afin que le système soit plus stable, la méthode employée pour l'intégration pose que :

$$\vec{v}_i^{\,t+\Delta t} = \vec{v}_i^{\,t} + \Delta t\left((1+\alpha)\,\vec{a}_i^{\,t} - \alpha\vec{a}_i^{\,t-\Delta t}\right)$$

et :

$$\vec{x}_i^{\,t+\Delta t} = \vec{x}_i^{\,t} + \Delta t\vec{v}_i^{\,t} + \frac{\Delta t^2}{2}\left((1+\beta)\,\vec{a}_i^{\,t} - \beta\vec{a}_i^{\,t-\Delta t}\right)$$

Les paramètres α et β servent à contrôler la convergence.

e) Formalisme CORDIS-ANIMA

Ce formalisme prend la forme d'un modeleur-simulateur [LJFC91]. Il a été exploité pour divers cas de figures [JLR90]. Ce formalisme repose sur le concept de *communication instrumentale*. L'homme communique avec la machine sensée simuler l'instrument grâce à des capteurs et la machine produit les résultats sonores, visuels et tactiles via des transducteurs. Le modèle physique a pour but de calculer les résultats que la machine fournit en retour. Il s'appuie sur l'utilisation des éléments *M* et *L*, qui sont les transducteurs des deux variables duales, intensive ou extensive, à savoir la force et la position (de façon équivalente, intensité et tension)[LC86]. Un système physique apparaît alors comme un réseau de bipôles de ce type, comme on le fait de manière courante en électricité et dont la résolution s'appuie sur les *lois de Kirchhoff*.

Les composants de ce réseau sont donc les éléments *MAT* (un atome *M*) et les éléments *LIA* (deux atomes *L*). En effet, d'un point de vue fonctionnel, un atome *M* peut être relié à plusieurs atomes *L*, mais un atome *L* ne peut être relié qu'à un atome *M*. Ces derniers produisent des positions à partir de forces, tandis que les éléments *L* produisent des forces à partir de positions.

Le système est complexifié par différents modules. Les liaisons *L* obéissent par exemple à des transitions d'état, d'où la notion de liaisons conditionnelles. Certaines liaisons peuvent même être dégénérées ce qui permet de simuler les moteurs ou certaines forces comme la gravité. Enfin des modules permettent la modification de la structure, le contrôle des paramètres du système...

Dans tout phénomène physique, l'hypothèse de la causalité est nécessaire. Les positions à l'instant t dépendent des positions et des forces à l'instant $t-\Delta t$. Une constante de temps est donc injectée dans le système et correspond à un pas de calcul.

Les mises en œuvre du formalisme CORDIS-ANIMA sont très nombreuses. Les billes de Jimenez [LJFR89][Jim93] sont une illustration des modèles complexes que l'on peut créer grâce aux liaisons conditionnelles. Dans ces travaux, on cherche à modéliser des agrégats. Pour cela, on considère trois couches de billes : la première couche est constituée des noyaux au centre, assez

rigides, autour de ces noyaux se situe un *derme* qui assure la déformabilité. Le tout est recouvert d'un épiderme assurant la cohésion du tout. La *figure 3.3* illustre la construction du modèle.

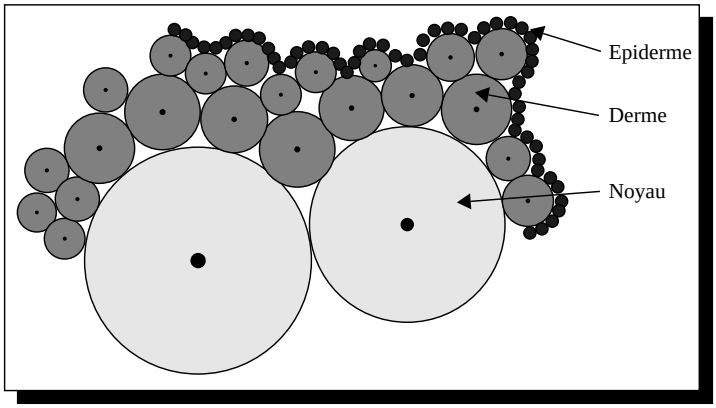

Figure 3.3. *Agglomérats de Jimenez*

L'interaction n'est non nulle qu'à partir d'une certaine distance. La force de répulsion est maximale une fois qu'il y a contact. Un modèle plus complexe d'interaction est également proposé en utilisant la loi de Van der Waals. On considère alors les agglomérats comme étant des *cliques* de masses, les agglomérats sont ensuite mis en relation les uns avec les autres.

Discussion

Les modèles masses/ressorts sont des classes de modèles très employées car elles sont simples à implanter. Cependant sous leur apparente simplicité se cache une complexité de nature assez similaire à celle des modèles continus. Si on considère en effet un graphe de points massiques complètement reliés par des ressorts, on obtient $n(n+1)/2$ liaisons pour n masses, d'où une complexité quadratique. De plus, si le ressort est le symbole même de l'élasticité linéaire, sa loi de comportement ne l'est absolument pas comme on peut le voir dans la formule *(3.19)*. Chaque ressort requiert une racine carrée et une division.

Pour éviter ce calcul, on peut considérer que le ressort est soumis à de petits déplacements, c'est à dire qu'il reste quasiment sur son axe, auquel cas on peut *linéariser* sa loi, mais cela restreint considérablement l'intérêt des masses/ressorts qui sont par nature compatibles avec les grands déplacements.

Enfin, il faut signaler que ces modèles sont très dépendants des conditions d'intégration et peuvent diverger très facilement.

3.2.4. Modèles ad-hoc

a) Modèle à base d'iso-surfaces

Une nouvelle représentation d'objets déformables est proposée par [Gas93]. C'est une représentation par courbe implicite où toute l'information mécanique est contenue dans l'expression de la courbe. Les objets sont définis comme l'union de composants solides, de forme simple, obéissant aux équations de la dynamique des solides, et une partie déformable définie par équation implicite. Ainsi, la forme de l'objet est obtenue en déterminant l'ensemble des points P tels que :

$$f(P) = \sum f_i(P) = 1$$

La sommation correspond aux composantes rigides constituant le modèle. La force est une fonction potentielle dépendant de la distance du point P à la composante rigide. La *figure 3.4* montre le style de corps obtenu.

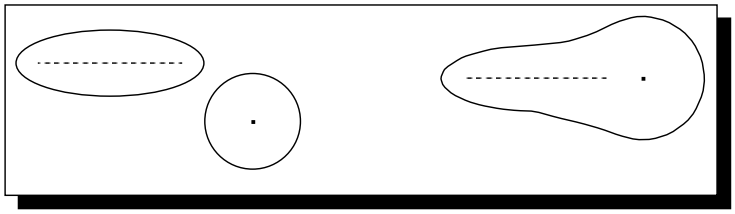

Figure 3.4. *Modélisation par surfaces implicites*

Cette modélisation est dotée de caractéristiques dynamiques. Les seules déformations autorisées sont celles où un point se déplace suivant la normale à la surface. Si un point P passe de X_0 à X, sur cette normale, la réponse à la déformation est :

$$f(X) - f(X_0) = -\left(\vec{N}(P) \cdot \vec{R}(P)\right)$$

où $\vec{N}$ est la normale, $\vec{R}$ est la réponse. De plus la raideur équivalente k est :

$$k(P) = -\frac{df}{dr}(r(P))$$

L'apport véritable de ce modèle tient dans sa formulation élégante des contacts. En effet, seul un terme de déformation est ajouté à la formulation des deux surfaces. Si deux objets sont définis respectivement par les surfaces f_i et f_j, lors du contact les deux surfaces deviennent, pour le premier objet :

$$f_i(P) + 1 - f_j(P) = 1$$

et le second objet :

$$f_j(P) + 1 - f_i(P) = 1$$

dans la zone d'interpénétration. La zone de contact est définie précisément par l'ensemble des point vérifiant :

$$f_i(P) = f_j(P)$$

Un terme de propagation est ensuite ajouté dans une zone de largeur donnée et permet de rendre compte d'une certaine propagation de la déformation. Les fonctions potentielles sont contraintes d'être décroissantes en fonction de la distance. Les fonctions exponentielles ont été choisies. Enfin, la méthode est étendue aux contact avec des objets rigides. La surface en contact est alors définie par :

$$g(P) = 1 - f(P_j) + 1 - f(P_i)$$

Ce modèle est plutôt approprié pour des corps très déformables comme les fluides. Des animations d'écoulement de liquide ont été calculées et donnent un résultat impressionnant.

Discussion

Ce modèle est très attractif car il est d'une formulation originale et très pratique, d'un point de vue de l'implémentation. Malheureusement, il présente un certain nombre de défauts majeurs vu notre objectif. Il repose sur les surfaces implicites et il n'existe actuellement aucun moyen d'afficher des surfaces implicites en temps réel avec précision et sans discontinuité [DTG95]. Un second défaut est le double objectif des fonctions implicites qui sont responsables à la fois de la forme et de l'expression des forces. En effet, une fois la géométrie de l'objet définie, la force de déformation se déduit de cette géométrie. Si par la suite, on souhaite modifier la résistance de l'objet aux déformations, la géométrie de l'objet sera modifiée. Enfin, les formes obtenues sont simples puisque fonctions des distances par rapport à des objets rigides simples. Pour obtenir une forme complexe, il faut augmenter le nombre de corps rigides de base et le nombre de fonctions associées.

3.3. Les problèmes communs

Dans cette partie nous allons cerner les problèmes partagés par les divers modèles que nous avons exposés. En effet ces derniers ne sont pas aussi distincts qu'ils ne le semblent et ont de nombreux points communs. Leur but est de simuler le comportement dynamique d'objet déformable en s'appuyant sur les lois de la physique. Les équations de départ sont donc les mêmes ou sont équivalentes. Les processus de résolution de ces équations sont communs. Par ailleurs, le but étant de produire une image, le problème de la définition de la forme de l'objet affiché se pose. Il faut étudier les liens entre le modèle en charge de la simulation mécanique et le modèle servant à l'affichage ; ces deux modèles (mécanique et géométrique) ont des contraintes opposées. Enfin, les problèmes de conservation de quantités comme le volume ou la surface sont partagés par toutes les méthodes.

3.3.1. Modèles dynamiques/statiques

Les lois, pilotant le mouvement que l'on obtient, sont des équations différentielles, généralement non linéaires, qu'il s'agit de résoudre. Que les modèles soient continus ou discrets, les équations se réduisent à l'expression suivante :

$$\overline{\overline{M}}\ddot{\vec{u}} + \overline{\overline{D}}\dot{\vec{u}} + \overline{\overline{K(\vec{u})}}\vec{u} = \vec{f} \qquad (3.23)$$

L'hypothèse de petits mouvements peut être faite dans cette équation, c'est à dire que l'on suppose que les liaisons entre nœuds ont un axe quasiment fixe. Dans ce cas, la matrice de raideur $\overline{\overline{K}}$ ne dépend plus de $\vec{u}$.

Le premier choix que l'on peut faire est de supprimer la partie concernant les régimes transitoires. On se place alors dans l'hypothèse d'un système *quasi-statique*, c'est à dire que la simulation est une suite d'états d'équilibre. Dans ce cas, on cherche à résoudre le système, dit *statique*, dont la solution est l'état de repos du système :

$$\overline{\overline{K(\vec{u})}}\vec{u} = \vec{f} \qquad (3.24)$$

L'avantage de cette formulation est qu'elle est indépendante du temps et de ce fait, peut être traitée de façon asynchrone. A chaque pas de la simulation, il faut calculer de nouveau la matrice $\overline{\overline{K(\vec{u})}}$ et l'inverser (sauf dans le cas des petits déplacements, où la matrice est constante et inversée une fois pour toutes), et ensuite trouver les valeurs des déplacements. Le système offre la garantie d'être stable et d'avoir une bonne solution. Par contre, comme il manque les régimes transitoires, le système atteint immédiatement sa position de repos, sans oscillation ni temps de réponse. Il est cependant possible d'adjoindre au modèle un mode transitoire précalculé et approximatif.

Ne pas simuler les régimes transitoires au profit des seuls régimes permanents, peut être un inconvénient. En effet, les retours à l'équilibre sont considérés comme instantanés. Cependant, si le système est fortement amorti, le retour à l'équilibre est lent. Au contraire, si le système est faiblement amorti, il y aura des oscillations avant la stabilisation. Dans ces deux cas, les régimes transitoires ont une manifestation ostensible. Il semble donc qu'on ne peut se contenter des régimes statiques que si l'amortissement du système est *critique* (valeur transitoire entre les retours lents et avec oscillations).

Le problème statique peut aussi se formuler sous forme énergétique. La résolution ne consiste plus alors en l'inversion d'un système d'équations, mais en une recherche du minimum d'une énergie potentielle. C'est alors un problème d'optimisation. Soit $\vec{a}$ le vecteur d'état du système, on cherche à minimiser l'énergie $f(\vec{a})$.

Une des méthodes classiques est la *descente de gradient*. On fait évoluer le système vers un minimum en faisant varier l'état du système selon la règle :

$$\dot{a}_i = -\frac{\partial}{\partial a_i} f(\vec{a})$$

L'autre possibilité est bien sûr de résoudre complètement toute l'équation différentielle, c'est l'étude *dynamique*. Outre les problèmes liés à l'intégration, des contraintes s'ajoutent au système. En effet, quelle que soit la méthode employée, il faut choisir un pas d'échantillonnage temporel. Ce choix du pas implique qu'il faut être capable de calculer les transitions d'état du système en, au plus, le temps que met le système pour évoluer. C'est la *synchronicité* [ULC94]. C'est une hypothèse plus restrictive que le simple *temps-réel*. Soit un système non synchrone : supposons un pas de temps de $1ms$ nécessitant $10ms$ de temps de calcul. Soit un objet animé d'une vitesse de $1m/s$. En $1ms$, il parcourt $1mm$. Il est perçu par l'observateur comme ayant eu un mouvement de $1mm$ durant $10ms$, le temps du calcul. Il a donc, pour l'observateur, une vitesse de $0.1m/s$. Réciproquement, tout mouvement de l'utilisateur sera décuplé dans le repère de la machine. Cela signifie qu'un geste lent peut être interprété comme un geste très rapide, voire un choc brutal perturbant pour le système. La difficulté réside alors dans le choix du pas de temps qui doit être compatible à la fois avec les possibilités en calcul de la machine de simulation et les contraintes fréquentielles du système simulé.

3.3.2. Méthodes d'intégration

L'intégration est un des points critiques des modèles dynamiques. En effet, le théorème de Shannon précise qu'il faut intégrer à une fréquence supérieure à deux fois la fréquence d'un signal pour le restituer correctement. Dans notre cas, cela veut dire que le pas de temps choisi conditionne directement le réalisme de la simulation. Les méthodes du type Euler ou celles plus complexes comme Runge-Kutta sont dites *conditionnellement stables*. Il faut donc se trouver dans les conditions de Shannon pour avoir convergence du système. Si les conditions ne sont pas respectées, le système diverge et ne retrouve plus de position stable. Or ces conditions sont parfois difficiles à satisfaire. Par exemple, un système constitué d'un ressort non amorti possède une fréquence naturelle égale à :

$$F_N = 2\pi\sqrt{\frac{k}{m}} \tag{3.25}$$

Cela signifie que le système est d'autant plus mal conditionné que l'on augmente la raideur ou que l'on diminue la masse. Dans le cas de la modélisation d'un objet de masse déterminé, plus on discrétise, plus la masse des nœuds est faible et plus la fréquence augmente. Si le système est amorti, la pseudo période est définie par :

$$F_p = 2\pi\sqrt{\frac{k}{m} - \frac{\lambda^2}{4m^2}}$$

On constate alors qu'en augmentant l'amortissement λ, par exemple les frottements visqueux dans l'air, on diminue la fréquence du système.

Il existe d'autres méthodes plus intéressantes, appelées *méthodes implicites*. Ces méthodes sont dites *inconditionnellement* stables. Elles ont la particularité de toujours converger vers la position de repos du système. Si le pas de temps employé n'est pas suffisant, le système converge plus rapidement, sans respecter fidèlement les équations physiques, vers l'état d'équilibre qui lui est un état physique possible. L'idée de base des méthodes implicites est de se baser sur une expression de la force à l'instant $t + \Delta t$ et non pas t, comme dictée dans les méthodes dites *explicites* ou *incrémentales*. Il existe plusieurs méthodes implicites, par exemple, Houbolt, Wilson, Newmark [Gou94].

Comme les équations dépendent des conditions de l'instant $t + \Delta t$, on doit résoudre un système non linéaire à chaque pas de temps. Ces méthodes implicites sont, par conséquent, gourmandes en calcul et sont encore de nos jours difficiles à concilier avec les contraintes temps-réel. C'est

pourquoi, on préfère encore employer les méthodes explicites, mais en leur ajoutant des heuristiques permettant de garantir la convergence des équations. [Pro95] a proposé de borner les distances entre les masses ce qui empêche la dislocation de la structure lors de la divergence. Par contre, il ne démontre pas que le système finit par se stabiliser. S'il doit absorber trop d'énergie pour empêcher la divergence, le système risque donc d'osciller perpétuellement.

Enfin, dans le modèle de [Jou95], le simulateur étudie l'état de l'énergie du système au cours du temps. Dans un système physique, l'énergie totale est constante. Sitôt qu'elle augmente, cela signifie que les équations sont en train de diverger. Si une variation positive de l'énergie est constatée, le pas de temps d'échantillonnage est divisé par deux, jusqu'à convergence. Si par contre, l'énergie n'augmente pas, le pas de temps est multiplié afin d'accélérer le processus. On constate immédiatement qu'une telle méthode n'est pas compatible avec un système de simulation synchrone.

En conclusion, on peut dire que le problème de la convergence des équations est difficile. Il est évident que l'algorithme de simulation doit prendre en compte le contrôle de la convergence afin d'être robuste et ne pas exhiber d'explosions des structures, même en cas de choc violent sur l'objet. Cette convergence peut-être soit garantie par des méthodes inconditionnellement stables, mais ces dernières sont très gourmandes en temps machine, soit plus ou moins contrôlée, mais non assurée, durant la simulation par des heuristiques.

3.3.3. La dualité modèle géométrique/modèle mécanique

a) Visualisation des structures déformables

La mise en œuvre d'une visualisation est difficile du fait de la dualité entre la représentation mécanique de l'objet et sa représentation visuelle. La géométrie se doit d'être précise et fine pour une visualisation réaliste du corps modélisé. Les points nécessaires à l'affichage doivent être nombreux si on ne veut pas afficher des contours trop anguleux. Cependant seuls les points en surface sont utiles. Pour la visualisation, il est nécessaire de traiter un grand nombre de points surfaciques.

Au contraire, la modélisation mécanique a besoin d'un nombre minimum de points. La position des nœuds est généralement contrainte à la fois pour satisfaire une bonne restitution des propriétés physiques (calcul de moment d'inertie) et un bon conditionnement dans la résolution des équations. La première contrainte impose que les points soient disposés symétriquement sitôt que la structure possède une ou plusieurs symétries. Leur disposition doit également être compatible avec les *axes principaux* de la structure physiques (ces axes sont définis comme étant le repère dans lequel la matrice d'inertie est diagonale). La seconde contrainte dépend de la résolution du système physique. Par exemple, dans le cas des éléments finis, il faut que les éléments soient de dimensions équivalentes et que les angles ne soient ni trop aigus ni trop obtus. Le contour des formes obtenu est anguleux si les éléments employés sont des parallélépipèdes. L'usage de tétraèdres est alors entendu possible (le contour est défini par un maillage de triangles), mais les fonctions de bases associées sont plus complexes. De façon identique, dans le cas de ressorts, des pôles dans la structure rendent délicate la définition des coefficients d'élasticité aux alentours.

Cette dualité entre la géométrie et la mécanique s'inscrit dans la problématique plus générale *forme/mouvement*. Plusieurs solutions ont déjà été proposées. Dans le cas des éléments finis, les fonctions d'interpolation permettent de déterminer la position de tous les points de la structure par interpolation des positions des nœuds de la discrétisation.

Les maillages masses/ressorts sont plus problématiques. Si on se contente de former des facettes s'appuyant sur les ressorts, le résultat visuel n'est pas satisfaisant et exige alors une multiplication du nombre de ressorts. Une solution est d'employer les nœuds comme points de contrôle de surfaces *splines*, Bézier ou FFD (voir [SP86]). C'est un choix qui s'inspire de la solution de [CHP89] qui modélisent les déformations de muscles sous forme de FFD, contrôlés par la position de solides articulés représentant les os.

[LC86] proposent de rajouter des points dits géométriques, sans masse, dont la position se détermine à partir des points mécaniques par des contraintes géométriques (volume, surface, distan-

ce constante). Ces points géométriques ajoutent des nouveaux points de contrôle pour les *splines* formant le contour des objets (voir *figure 3.5*). Dans les modèles suivants de CORDIS-ANIMA, la *forme* de l'objet est obtenue par les contours de fonctions d'affluence (sphère, cercle) autour des particules.

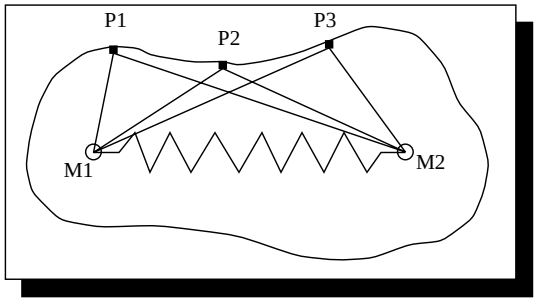

Figure 3.5. *Modélisation par points géométriques et points mécaniques*

b) Définition du modèle géométrique et du modèle mécanique

De l'analyse précédente, on peut déduire que tout modèle de corps déformable doit se composer de deux modèles distincts séparant la géométrie de la mécanique. Le modèle possède deux unités :

- l'unité mécanique, responsable du calcul du mouvement, caractérisée par des valeurs physiques et des lois physiques de comportement ;
- l'unité géométrique, responsable de la forme de l'objet. Ce contour sert non seulement à la restitution visuelle mais aussi à la détection des collisions. C'est en effet la zone où ont lieu les contacts.

Grâce à cette séparation, il est possible d'avoir un nombre minimum de points mécaniques et important de points géométriques. Le problème qui se pose est de savoir le type des informations échangées entre les deux unités. L'unité géométrique, par nature, ne peut interpréter que des données de positions. L'unité mécanique, par contre, ne réagit qu'à des forces (ou de façon équivalente à des énergies potentielles). Les deux questions fondamentales sont alors : quels types de positions doit communiquer l'unité mécanique ? Comment calculer les forces que retourne l'unité géométrique ?

Le diagramme sur la *figure 3.6* montre la décomposition du modèle en deux unités.

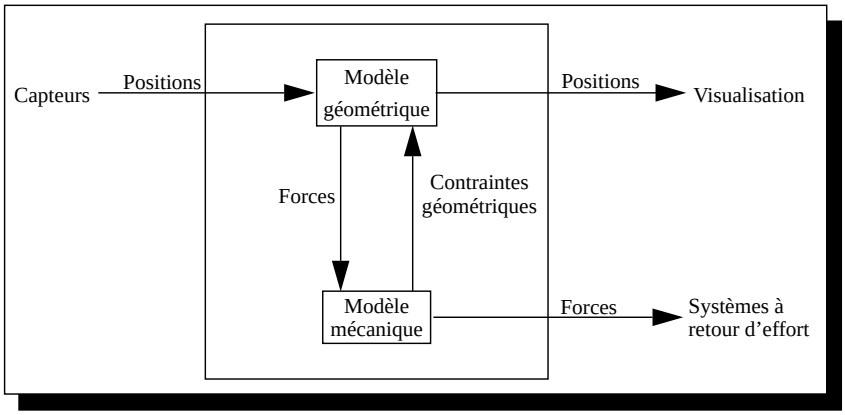

Figure 3.6. *Décomposition du modèle*

Dans les diverses solutions exposées depuis le début de ce chapitre, on constate qu'un modèle de ce type est toujours sous-jacent. En réalité, tous les modèles de corps déformables possèdent de façon plus ou moins directe ces deux unités.

Le modèle géométrique est en réalité constitué des diverses primitives de déformation employées classiquement en modélisation. On peut citer :

- Un maillage de points reliés entre eux. La déformation s'effectue en modifiant la position d'un ou plusieurs points. La forme globale est obtenu soit en formant des triangles à partir des points (fonctions de bases linéaires) soit en utilisant des fonctions de bases plus complexes comme celles définies par les éléments finis ;
- Une surface définie à l'aide de points de contrôle. La déformation s'effectue en modifiant un ou plusieurs points de contrôle. Les primitives sont des surfaces de Bézier ou des surfaces *splines*, par exemple. Un cas particulier de cette technique est formé par les FFD [SP86] ;
- Les déformations globales de [Bar84] ;
- Les formes à base de potentiels comme les surfaces implicites ou les sphères d'influence.

Le modèle mécanique fait en sorte que les déformations exposées ci-dessus soient contrôlées par les équations de la physique. Les formalismes étant équivalents, on utilisera suivant les cas, soit la mécanique du point et/ou des corps rigides, soit les équations de Lagrange soit enfin le principe des travaux virtuels. Seule la façon d'exploiter ces lois permet de faire la distinction d'un modèle à l'autre.

Illustrons ces notions. Les modèles éléments finis s'appuient, pour la partie géométrique, sur des fonctions de bases et pour la mécanique, sur le principe des travaux virtuels. Le module géométrique fournit au modèle mécanique les forces de contact appliquées aux nœuds. Les positions des nœuds sont fournies en sortie au module géométrique qui en déduit la forme globale de l'objet. Dans le modèle de [WW90], le module géométrique est composé de déformations globales. Le module mécanique est fondé sur les équations de Lagrange. Les contacts sont traduits par le module géométrique en potentiels. Ces potentiels sont fournis au module mécanique qui intègre les équations en tenant compte de ces potentiels. En retour, il fournit au module géométrique la valeur des paramètres des transformations globales. Enfin le modèle de [CHP89] est divisé de façon explicite : le modèle mécanique est constitué de masses et de ressorts, chaque masse est un point de contrôle du maillage FFD formant la partie géométrique du modèle.

Le tableau sur la *figure 3.7* récapitule les modèles présentés.

	Mécanique du point ou corps rigides	Lagrange	Travaux virtuels
Echantillonnage, Points discrets	Différence finies [TPBF87], Modèles discrets		Eléments finis
Interpolation (Bézier, splines, FFD)	[CHP89]		
Transformations globales		[PW89], [WW90], [MT92]	
Surfaces implicites	[Gas93], [DG94]		

Figure 3.7. *Classification des corps déformables*

3.3.4. Contrôle de la déformation

Il est évident qu'un ballon de baudruche ne se déforme pas de la même façon qu'un sac de sable. Le premier se déforme à surface constante, le second à volume constant. Il s'agit donc de

contrôler la déformation afin de respecter la conservation du volume ou de la surface. La plupart des méthodes exposées ne peuvent pas garantir de déformations à volume constant, sauf si elles calculent leur déformation à partir de la loi de conservation, comme [Gas89] par exemple. Ce problème est un cas particulier de contraintes dynamiques que nous détaillons dans le paragraphe *4.3.3*, dans le traitement des interactions entre corps déformables. Citons tout de même les méthodes à pénalité. Elles consistent à rajouter des forces permettant de pénaliser toute déviation de la contrainte. Ainsi, si par exemple le volume augmente, une force se crée qui tend à ramener la structure vers un état conservant le volume. Avec cette solution, le volume n'est certes pas constant, mais tend toujours à l'être.

Une autre solution adapté au modèle de corps déformables par iso-surface a été proposée par [DG95] : il considère un territoire autour de chaque particule P_i du noyau rigide de son objet. Ce territoire est défini comme étant la zone de l'objet où le champ associé à la particule est le plus important, comparé aux champs des autres particules. Le volume de ce territoire est calculé grâce aux *Seed* (points que l'on contraint à rester sur la surface implicite et que l'on essaie de répartir uniformément) déjà employées pour l'affichage de leur modèle : on obtient un ensemble de tétraèdres dont on calcule le volume. La variation de volume modifie alors un terme de translation se trouvant dans la formulation du champ qui tend à faire varier le volume vers la solution.

3.4. Modèles pour la simulation d'organes humains

Nous choisissons d'étudier distinctement de l'état de l'art précédent, les modèles d'organes qui ont déjà été proposés pour la simulation d'opérations chirurgicales. En effet, il nous paraît intéressant d'observer en détail les réponses que d'autres équipes dans le monde ont trouvé pour répondre aux problèmes spécifiques que nous avons exposés dans l'introduction de ce chapitre.

3.4.1. Modèles continus

a) Approche 2D

[Bai96] a mis en pratique les divers modèles de corps déformables. Le modèle qu'il propose est 2D mais extensible au 3D. Il consiste en un découpage en éléments finis à neuf nœuds (quadrilatères). Les fonctions d'interpolation sont donc quadratiques. Il caractérise l'énergie de la structure (en calculant la densité d'énergie élastique, fonction du *tenseur de déformation de Green-Lagrange*). La minimisation de cette énergie donne l'état du système, si on se place dans l'hypothèse d'une simulation quasi-statique. Ce modèle comporte un module géométrique séparé puisqu'à la frontière sont disposés des curseurs, sorte de sur-échantillonnage du contour, et qui sont employés pour la détection et le traitement des collisions.

b) Approche par actives cubes

[Bro95] proposent une solution en deux étapes. Tout d'abord, la reconstruction du modèle est basée sur une version étendue des *snakes*, appelés les cubes actifs (*active cubes* en anglais). Le but des snakes est d'extraire la forme d'un objet dont on a une image. Une fonction d'énergie est associée aux snakes, dont le but est de caractériser les propriétés de déformation du snake. L'image de l'objet va permettre de créer les forces animant le snake. La solution, vers laquelle on souhaite que le snake converge, est la forme de l'objet qui doit donc correspondre au potentiel le plus bas. Par rapport aux snakes, les cubes actifs sont volumiques et correspondent en 3D au volume et non pas à la seule surface de l'objet.

Une fois l'objet décrit à l'aide des cubes actifs, on obtient directement une paramétrisation de l'objet. La configuration de référence étant connue, on la caractérise de façon synthétique grâce au tenseur métrique. Ensuite, la structure peut être déformée, une énergie de déformation est alors associée, sur le même principe que [TPBF87] (voir au paragraphe *3.2.1* le modèle *b*). La simula-

tion consiste alors à trouver, en tenant compte des contraintes, l'état de plus basse énergie du système.

La méthode pour déterminer ce minimum d'énergie n'est pas donnée. Les algorithmes permettant de le trouver sont nombreux. Cependant, ils sont coûteux et peu adaptés aux contraintes temps réel. Par contre, un des avantages de cette modélisation est qu'elle est parfaitement compatible avec les découpes, car les actives cubes sont volumiques et qu'une découpe correspond simplement à supprimer une liaison dans le réseau.

c) Approche par précalcul de déformation

[CDBA96] propose une modélisation éléments finis d'un foie. Les éléments employées sont des tétraèdres linéaires isoparamétriques. Ce choix de l'élément de base est conditionné par le logiciel de reconstruction de formes employé pour reconstituer la forme du foie. Le foie est supposé être élastique linéaire et on fait l'hypothèse de petits déplacements. On se trouve donc dans le cas le plus simple des éléments finis. La matrice d'élasticité associée au système est constante. On retrouve l'équation *(3.8)* :

$$\bar{\bar{K}}\vec{U} = \vec{R}$$

Le théorème de superposition est applicable. La déformation engendrée par un ensemble de déplacements est égale à la somme des déformations engendrées respectivement par chaque déplacement. Pour chaque nœud du réseau, et pour les trois axes du repère local, on suppose une contrainte particulière, par exemple un déplacement de 1 selon l'axe, et on calcule le résultat de ce déplacement sur tous les autres nœuds. On stocke le résultat dans un tableau. Lors de la simulation, on répertorie l'ensemble des points en contact ainsi que les contraintes associées puis on calcule une combinaison linéaire des résultats stockées dans les divers tableaux.

Cette solution est remarquable car toutes les étapes longues du calcul des éléments finis sont déportées en début de simulation. De ce fait, le modèle est parfaitement temps-réel. Elles présentent toutefois certains défauts. Tout d'abord la simulation est statique, les régimes transitoires ne sont pas pris en compte. Le comportement, lors des changements d'état, est donc approximatif. Ensuite, l'hypothèse de faibles déplacements a été faite : si de grands déplacements sont effectués, le modèle perd beaucoup de son réalisme. Enfin, les précalculs effectués perdent tout leur sens sitôt que le modèle est modifié, c'est à dire lors des découpes ou des déchirements. Cette solution est incompatible avec les découpes.

d) Approche par condensation

[BC96] proposent une autre solution pour accélérer la méthode des éléments finis en linéaire. Ils partent de la constatation que seule la position des nœuds surfaciques importe. On ne va donc chercher à résoudre que la partie du système décrivant ces nœuds-là. L'équation *(3.8)* est séparé en deux :

$$\begin{bmatrix} \bar{\bar{K}}_{ss} & \bar{\bar{K}}_{si} \\ \bar{\bar{K}}_{is} & \bar{\bar{K}}_{ii} \end{bmatrix} \begin{bmatrix} \vec{U}_s \\ \vec{U}_i \end{bmatrix} = \begin{bmatrix} \vec{R}_s \\ \vec{R}_i \end{bmatrix}$$

L'indice *s* signifie surfacique, l'indice *i* signifie intérieur. Cette équation est très proche de la formulation de [GTT89] que l'on a présentée auparavant. Le système se réduit alors à :

$$\bar{\bar{K}}^* \vec{U}_s = \vec{R}^*$$

avec $\bar{\bar{K}}^* = \bar{\bar{K}}_{ss} - \bar{\bar{K}}_{si}\bar{\bar{K}}_{ii}^{-1}\bar{\bar{K}}_{is}$ et $\vec{R}^* = \vec{R}_s - \bar{\bar{K}}_{si}\bar{\bar{K}}_{ii}^{-1}\vec{R}_i$. Le gain de cette méthode dépend du ratio entre le nombre de nœuds en surface sur le nombre total de nœuds.

Par ailleurs, les auteurs préconisent l'inversion de la matrice $\bar{\bar{K}}^*$ (lors d'une phase de précalcul). En effet, la rapidité du modèle est plus cruciale que sa précision. L'inversion de la matrice n'est pas conseillée si l'on souhaite obtenir des résultats très précis. Dans le cas de la simulation chirurgicale, c'est l'aspect plausible et convaincant qui prime. Une fois la matrice inversée,

une multiplication matrice-vecteur sélective est employée. En effet, les force extérieures sont dues au contact et de ce fait sont locales. On en déduit qu'un faible nombre de nœuds subissent cette force. Le vecteur des forces extérieures est donc peu dense. On restreint en conséquence le produit $\bar{\bar{K}}^{*-1}\vec{R}^{*}$ aux seules composantes de $\vec{R}^{*}$ non nulles.

Cette solution est similaire à la précédente et présente les mêmes limites. Toute découpe brise le temps réel, car toutes les matrices doivent être calculées à nouveau. La matrice $\bar{\bar{K}}$ n'est plus la même, les nœuds surfaciques sont plus nombreux, et la matrice $\bar{\bar{K}}^{*}$ doit être inversée de nouveau.

e) Modèle éléments finis exact

Le modèle proposé par [SBM94] est un modèle éléments finis. La modélisation concerne la cornée de l'oeil. L'élément choisi est donc la coque, en élasticité 3D incompressible à grand déplacement. Le découpage s'effectue en coordonnées sphériques, les angles sont interpolés avec des fonctions de base trilinéaire de Lagrange et la coordonnées de rayon avec une fonction cubique d'Hermite le long du plan de la coque et une fonction linéaire de Lagrange le long de l'épaisseur. Enfin, la cornée est supposée constituée par un matériau homogène, orthotropique et élastique non linéaire. Les fibres de collagènes sont supposées se trouver dans le plan de la cornée.

La résolution du modèle, afin de respecter le temps réel, est implantée sur une machine DEC Alpha 3000/300. Malheureusement, aucun détail n'est donné sur l'algorithme de résolution proprement dit, ni sur les éventuelles hypothèses simplificatrices ou heuristiques employées afin d'accélérer le calcul.

3.4.2. Modèles discrets

a) Modèle masses/ressorts dynamique

[KKKN96] proposent de modéliser les organes par des *nurbs*, dont les points de contrôle ont une masse et sont liés entre eux. Les liaisons peuvent être des ressorts, des amortisseurs ou plastiques. Les *nurbs* définissent le contour de l'organe, d'où une visualisation très réaliste des organes et un comportement dynamique naturel. Par contre, la géométrie des organes obtenue est définie de façon complexe d'où un coût important des calculs des interactions. Les équations de la dynamique sont résolues dans une première version, par une méthode d'intégration explicite ([KKKH95]). La version actuelle repose soit sur un calcul implicite soit semi-implicite (l'algorithme n'est pas complètement décrit dans les articles).

b) Modèle masses/ressorts surfacique statique

[CEO93] proposent de travailler avec un modèle entièrement surfacique composé de masses et de ressorts. L'organe est décomposé en points caractérisés par une position de repos et des liens avec ses voisins. La structure de données pour chaque point est constituée des informations suivantes :

- location du point ;
- booléen indiquant si le point est gelé c'est à dire si sa position est fixée ;
- distance de repos par rapport à tous ses voisins ainsi que l'identification de ses voisins ;
- force totale appliquée au point ;
- position de repos du point.

A partir de la position de chaque nœud, la somme des forces en chaque nœud est calculée. Les auteurs choisissent de ne pas intégrer les équations de la dynamique, mais de prendre un point de vue énergétique. Une fonction de poids est appliquée sur le vecteur somme des forces, afin de

prendre en compte les mouvements des voisins qui bougent : le poids est directement proportionnel au nombre de voisins mobiles.

On obtient alors le vecteur pondéré $\vec{f}_0$. On refait le bilan des forces à une position aléatoire P_1 du point se trouvant dans une certaine zone autour de la position courante P_0. On évalue la force pondérée $\vec{f}_1$ à cette position. La position idéale est obtenue en appliquant la formule suivante pour les trois coordonnées x, y et z :

$$P^k_{min} = f^k_o \frac{P^k_0 - P^k_1}{f^k_0 - f^k_1} - P^k_0 \text{ pour } k \in \{x, y, z\}$$

Ensuite, la nouvelle position est calculée, par heuristique :

$$P = w(P_{min} - P_0)$$

où w est une fonction de pondération. Le modèle tel qu'il est proposé est incomplet et les auteurs pensent incorporer la détection des collisions, les contraintes de contact, les fractures, le placage de textures, le retour d'effort...

3.4.3. Bilan

Toutes les méthodes vues précédemment ont un point commun : elles privilégient le temps réel au détriment du réalisme. Cependant, les méthodes d'accélérations proposées ne sont plus utilisables sitôt que la structure de l'organe est modifiée. Or dans un contexte de simulation médicale, les découpes doivent absolument être simulées.

Conclusion

Dans ce chapitre, nous avons exposé les diverses méthodes existantes à l'heure actuelle pour la modélisation et la simulation de corps rigides ou mous, en se basant sur les lois de la physique. Autant la physique du solide bénéficie d'une théorie robuste, précise et bien connue, au contraire, la physique du corps mou est très complexe, très exigeante en quantité de calculs, variable d'un corps à l'autre, voire même indéterminée sitôt que les corps entrent en interaction. Tous les modèles se séparent en deux composantes : la partie mécanique et la partie géométrique. La partie mécanique s'appuie sur une modélisation continue (éléments finis) ou au contraire discrète (particules). La partie géométrique définit le nombre de degrés de liberté des déformations. Le choix de cette réduction dépend de la préférence que l'on souhaite accorder aux déformations globales ou locales, des contraintes en temps de calcul. Cependant, on a mis en évidence un certain nombre de questions communes, indépendantes ou presque du modèle. C'est par exemple le choix entre un modèle dynamique ou au contraire statique, stable et utilisable en simulation asynchrone, mais qui ne traduit pas toutes les subtilités des déformations au cours du temps. Dans ce dernier cas, se pose le problème de l'intégration par rapport au temps, où l'on constate une dualité entre simplicité et robustesse des calculs.

S'inspirant de ces travaux, diverses solutions pour la modélisation d'organes dans un contexte de simulation ont déjà été proposées. On remarque que le temps réel est le problème crucial de ces approches. En effet, comme on souhaite un réalisme suffisant mais pas forcément le plus exact possible, des solutions approchées sont satisfaisantes.

Tout au long de ce chapitre, nous nous sommes efforcés de séparer la modélisation des déformations de celle des interactions, quoique les deux soient fortement liées. Nous le montrons dans le prochain chapitre traitant tous les aspects des interactions.

4 Prise en compte des interactions

Introduction

Dans le chapitre précédent, nous avons étudié comment modéliser des corps déformables. Cependant, sans interaction mécanique, ces corps ne se déforment quasiment pas. Les interactions sont le garant du réalisme de la simulation puisque c'est par échange d'énergie lors des contacts que le comportement des objets va être altéré. Une simulation sans interaction ressemble à un monde d'objets isolés. Plusieurs concepts se cachent derrière le terme interaction. En effet, la première étape est de détecter les interactions c'est à dire où et quand deux objets quelconques de la scène entrent en contact. Le terme générique employé habituellement dans la littérature est la détection de collision. Ce thème est issu des travaux en robotique, pour la sélection de trajectoires pouvant être empruntées par un robot autonome par exemple. La première étape consiste à définir précisément ce que l'on entend par collision, car le point de vue pour lequel on opte conditionne le choix des algorithmes. Par ailleurs, cette détection a un coût en exécution très important, plus encore lorsque les objets ont une forme variable, d'où des études très fines cherchant à accélérer cette détection.

Une fois les parties des objets entrant en collision identifiées, il s'agit de déterminer la réaction des objets à la collision. En effet, le comportement des objets est différent s'il y a choc ou au contraire contact prolongé. Lors d'un choc, il faut trouver comment les vitesses des objets sont modifiées, comment ils rebondissent l'un sur l'autre. Dans le cas d'un contact prolongé, le comportement des objets dépend des forces de contact entre les deux, des frottements éventuels s'opposant aux glissements.

Contrairement aux corps déformables, il n'existe pas, à notre connaissance, d'état de l'art complet décrivant les trois points clefs du problème des collisions, à savoir la détection, l'accélération et enfin le calcul de la réponse. Les domaines concernés par ce problème sont nombreux : la simulation dynamique et l'animation en général, les mathématiques et la géométrie (calcul d'intersection et topologie), et enfin la robotique (calcul de trajectoire). Nous tentons dans ce chapitre de dresser un état de l'art du traitement des collisions que nous avons voulu le plus représentatif possible du panel des méthodes existantes. Nous ne prétendons pas qu'il soit exhaustif, mais il représente le point de vue de l'informaticien face à ce problème.

4.1. Définition et détection de la collision

4.1.1. Définition des collisions

Une collision entre deux objets est détectée s'il existe un instant où les deux objets ont une intersection non vide. [Boy79] a défini deux façons de considérer les collisions. Soit on échantillonne les trajectoires au cours du temps et on considère les intersections à des instants discrets. Soit au contraire, on conserve l'aspect continu du temps, et on calcule alors des intersection de trajectoires. De cette distinction, on tire alors plusieurs solutions :

a) Intersection 3D

On peut considérer le système à des instants donnés (généralement tous les pas de la simula-

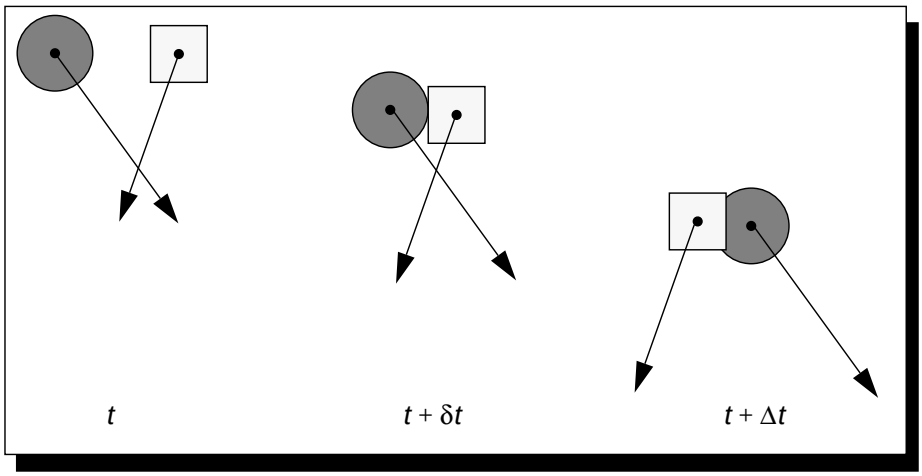

Figure 4.1. *Détection de collisions par interpénétration*

tion) et observer si des objets ont une intersection non vide. Le problème de la détection revient alors à un calcul d'intersection 3D. La complexité du calcul dépend de la modélisation géométrique des objets. Si ce sont des polyèdres, on doit les découper en parties convexes et ensuite faire les tests pour les faces de chaque partie. Si au contraire, le objets sont modélisés par surfaces implicites, il suffit alors d'utiliser la fonction dedans/dehors fournie. Généralement, une fois l'intersection détectée, on revient en arrière dans la simulation. On utilise pour cela une dichotomie, afin de remonter au temps où a eu lieu l'interpénétration. Cette méthode est couramment employée en animation. Sur la *figure 4.1*on suppose deux objets entrant en collision. Au moment t il n'y a pas collision. Au moment $t + \Delta t$ on constate une intersection. On cherche alors par dichotomie l'instant $t + \delta t$ où a eu lieu la collision.

Cependant, dans certains cas, certaines collisions peuvent ne pas être détectées. Sous réserves d'être suffisamment rapides, deux objets peuvent passer l'un au travers de l'autre entre les instants où a lieu le test. Pour éviter ce cas, la simulation doit vérifier la contrainte suivante. Si Δt est le pas de la simulation et que la vitesse maximum des objets est de v_{max}, alors on s'autorise à ne pas détecter d'interpénétrations sur des longueurs d'au maximum $l_{critique} = v_{max} \times \Delta t$. Cette valeur est appelée *résolution* de la détection. Autrement dit, des objets peuvent passer au travers d'un objet possédant une partie dont une dimension ou un diamètre est inférieur à $l_{critique}$. L'exemple de la *figure 4.2* montre le cas de deux objets dont la vitesse est trop élevée. Ni à l'instant t, ni à l'instant $t + \Delta t$ on ne détecte d'intersection et de ce fait, on ne détecte aucune collision, alors que les objets sont effectivement passés l'un au travers de l'autre.

b) Collision par intersection de trajectoire

Afin d'éliminer les cas non détectés de la solution précédente, il faut considérer l'ensemble des positions prises par l'objet au cours du temps. La collision a alors lieu lorsque les volumes décrits par les objets lorsqu'ils suivent leur trajectoire s'intersectent. Plusieurs solutions sont alors envisageables.

Intersection 3D de trajectoires

On considère dans cette solution tout le volume décrit par les objets entre t et $t + \Delta t$. Ensuite, on fait l'intersection 3D de ces deux volumes. Cette méthode n'est pas satisfaisante car ce n'est pas parce que les deux volumes se croisent qu'il y a une collision. En effet, les positions des objets correspondant à l'intersection peuvent être atteintes à des instants complètement différents. La *fi-*

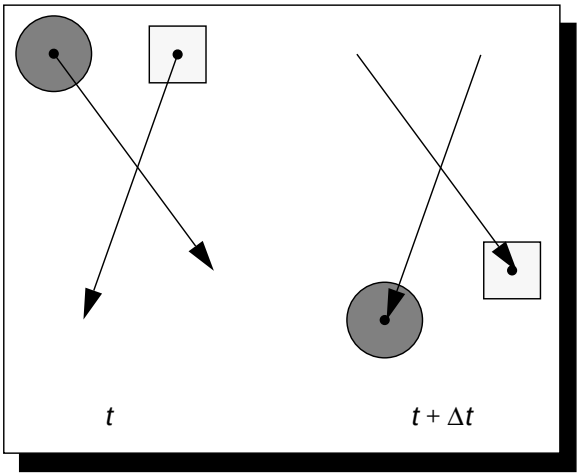

Figure 4.2. *Exemple de collision non détectée*

gure 4.3 illustre ce problème en 2D. Les deux objets ne s'interpénètrent pas durant le pas de temps et pourtant on trouve une intersection de leurs volumes de trajectoire.

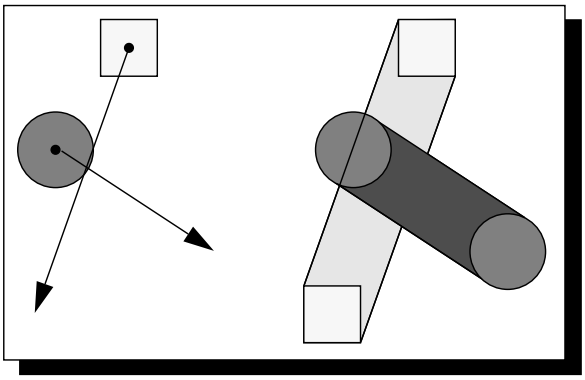

Figure 4.3. *Intersection de volumes de trajectoire*

Intersection 4D de trajectoires

Afin de pallier le problème précédent, on peut se placer en dimension 4, c'est à dire considérer le temps comme une variable d'espace supplémentaire. Dans ce cas, on cherche non plus une intersection de volumes 3D mais 4D. Le problème précédent (intersection de volumes 3D) devient alors l'intersection des projections des volumes 4D sur l'hyperplan 3D de l'espace du monde.

Méthode analytique

Elle est strictement équivalente à la méthode par intersection de volumes 4D mais on regarde le problème sous un angle analytique plutôt que géométrique. Le solide est considéré comme incorporant un paramètre supplémentaire, le temps. Trouver s'il y a interpénétration revient à trouver une solution à un système d'équations. On essaie dans ce cas de réduire le système à une seule équation dépendant uniquement du temps. Cette méthode est réservée à des géométries d'objets relativement simples.

4.1.2. Détection des collisions

Nous dressons ici un état de l'art des méthodes déjà proposées pour la détection des collisions, et pour chacune des solutions dans quelle catégorie elles se situent.

a) Intersection 3D

Deux objets sont en collision lorsque l'intersection mathématique des deux ensembles des points qui définissent l'objet n'est pas vide. Suivant le type de modélisation, l'intersection est plus ou moins complexe.

Modèles polygonaux

[Boy79] distingue quatre cas différents, si les primitives sont polygonales :

- l'intersection est vide ;
- l'un des objets est inclus dans l'autre (deux cas) ;
- les objets s'intersectent mais l'intersection n'est pas égale à l'un ou l'autre des ensembles.

Ce dernier cas est le premier à tester. Le principe de base est alors le suivant : *deux objets sont en intersection si une arête de l'un passe par une facette de l'autre*. Ainsi, on teste pour chaque arête de l'un des corps si les deux extrémités sont du même côté des plans des facettes de l'autre corps. Si c'est le cas, il n'y a pas d'intersection possible. Sinon, il faut tester si le point d'intersection entre le segment et la facette appartient ou non à la facette. Le cas particulier de l'arête complètement incluse dans le plan de la facette est prévu : on teste alors si le segment traverse un des côtés de la facette, si ce n'est pas le cas, on teste si le segment est complètement inclus dans la facette en inspectant si un des points du segment appartient à la facette. Ensuite, afin de discriminer les trois premiers cas d'intersection, il est possible de décider si un solide est dans un autre en testant si un des points de l'un est dans l'autre.

Cet algorithme est relativement analogue à celui proposé par [MW88] mis à part qu'ici ce ne sont que les objets convexes qui intéressent les auteurs. Une version 3D de l'algorithme de Cyrus-Beck (voir [FDFH90], chapitre 3), se basant sur la position des points vis à vis de l'orientation des facettes, est utilisée afin de déterminer si un point est ou non à l'intérieur de l'objet. Tous les points de l'objet A sont testés par rapport à l'objet B, puis tous les points de B sont testés rapport à A. Trois étapes sont utilisées successivement :

- Test si des points du premier objet sont dans le second, grâce à un calcul donnant le signe de la distance relative d'un point par rapport à un plan ;
- Test si les arêtes du premier objet traversent les facettes du second. Pour cela, chaque arête du premier est découpée en petits segments délimités par les intersections avec les plans infinis des facettes du second objet. Le milieu de chaque sous-segment est alors testé afin de savoir s'il se trouve ou non dans le second objet ;
- Test si des facettes identiques et parallèles se sont interpénétrées en inspectant si les barycentres de chaque facette du premier objet se trouve dans le second. C'est un cas très rare de collision.

Enfin, citons l'approche originale de [Hah88], qui, plutôt que de faire une dichotomie pour détecter l'instant de la collision lorsqu'une intersection a été détectée, propose d'interpoler le mouvement supposé linéaire et à vitesse constante afin de remonter au moment où il y a eu intersection. Pour cela un point de collision est désigné qui est translaté selon l'opposé du vecteur vitesse afin de situer le point d'interpénétration c'est à dire le point de contact entre les deux objets lorsque commence l'interpénétration.

Approche par les quaternions

Avec des modèles polygonaux, [Can86] cherche un chemin pour la trajectoire d'un corps solide, dans un environnement rigide. Il préconise l'emploi des quaternions (voir *Annexe II.*) afin de décrire les six degrés de liberté du solide. Les quaternions sont pratiques car ils décrivent, d'une manière élégante, les rotations du solide autour du centre d'inertie.

Trois types de contact sont donnés (voir *figure 4.5*) :

- un point de A est sur une facette de B (Type 1) ;
- un point de B est sur une facette de A (Type 2) ;
- une arête de A touche une arête de B (Type 3).

Ces contraintes sont ensuite exprimées en terme d'équations sur les quaternions. Dans la suite, on donne l'expression de ces contraintes, en supposant que l'objet A est mobile et l'objet B immobile. La position de l'objet A est représentée par un vecteur $\vec{x}_A$ et son orientation par un quaternion Q_A. Soit une facette de l'objet A. On note F_A le quaternion formé pour sa partie réelle par la distance de la facette à l'origine et pour sa partie vectorielle par le vecteur normal au plan. On note $\vec{p}_B$ point de B entrant en collision.

La collision de type 2 obéit à l'équation :

$$\left[Q_A F_A Q_A^* \, (1 + \vec{p}_B - \vec{x}_A)\right] = 0 \qquad (4.1)$$

En employant des notations analogues, le type 1 devient :

$$\left[F_B\left(1 + Q_A \vec{p}_A Q_A^* + \vec{x}_A\right)\right] = 0 \qquad (4.2)$$

Enfin, en nommant $\vec{e}_A$ et $\vec{e}_B$ les directions des deux segments en contact, les collisions de type 3 s'écrivent :

$$\left[Q_A \vec{p}_A \vec{e}_A Q_A^* \vec{e}_B\right] + \left[(\vec{x}_A - \vec{p}_B)\, Q_A \vec{e}_A Q_A^* \vec{e}_B\right] = 0 \qquad (4.3)$$

On peut transformer ces trois expressions afin de les rendre invariantes par multiplication de Q_A par un scalaire. Une intersection entre polygones convexes est alors obtenue en considérant les inégalités associées aux équations précédentes. Ainsi, il y a intersection si par exemple, il existe un point de A situé dans B. Cela se traduit par :

$$\exists \vec{p}_B \in B, \forall F_A \in A, \left[Q_A F_A Q_A^* \, (1 + \vec{p}_B - \vec{x}_A)\right] \leq 0$$

Ce qui se traduit par une disjonction de conjonctions de prédicats. On exprime le même genre de relations pour exprimer le fait qu'un point de B peut être dans A, qu'un segment de B est au travers d'une face de A et inversement. Dans un tel formalisme, on caractérise l'état de l'objet par un espace à sept dimensions dont les paramètres sont les coordonnées du vecteur $\vec{x}_A$ et du quaternion Q_A. Cet espace peut être réduit à six dimensions, car la norme du quaternion n'est pas significative. On peut alors caractériser facilement dans cet espace, les trajectoires sans interpénétration en utilisant l'expression sous forme de prédicats des conditions de non interpénétration.

Modélisation par surfaces implicites

Plusieurs auteurs préconisent l'emploi de fonctions implicites. En effet, ce formalisme procure un moyen simple permettant de décider si un point est à l'intérieur ou non de l'objet. De façon général, la surface est définie par une équation du style $f(x,y,z) = 0$. Les points se situant à l'intérieur vérifient $f(x,y,z) < 0$. Le problème de ces approches survient lorsqu'on cherche à déterminer si deux volumes implicites sont en intersection. Il faut dans ce cas échantillonner la surface de l'un des objets puis inspecter si au moins un des points obtenus se trouve à l'intérieur de l'autre volume. Malheureusement, comme les surfaces sont déformables, on ne peut pas échantillonner une fois pour toute la surface en début de simulation.

[SP91] préconisent l'emploi de superquadriques. Ces surfaces sont définies par la formule suivante :

$$\left[\left(x^{2/\varepsilon_2} + y^{2/\varepsilon_2}\right)^{\varepsilon_2/\varepsilon_1} + z^{2/\varepsilon_1}\right]^{\varepsilon_1/2} = 1$$

Elles procurent un large choix de formes. De plus, il est possible d'associer une paramétrisation des superquadriques. On peut ainsi, en découpant les intervalles de définition des paramètres, obtenir un bon échantillonnage de la surface. Enfin, ces points sont modifiés en sui-

vant les mêmes déformations que la structure dans son ensemble. La détection de collision proprement dite est rendue plus délicate par l'emploi des cartes de déplacement qui modulent la forme des surfaces et compliquent l'équation implicite.

[DG94], dans sa formulation implicite à base solide ne bénéficie pas de cette possibilité de paramétrisation de ses surfaces en temps réel. La solution adoptée consiste en l'emploi de *graines* dont la position est adaptée à la surface à chaque déformation de la structure. Ainsi, au départ, on choisit des directions particulières autour du squelette rigide. A chaque direction est associée une graine. A chaque déformation, la nouvelle position des points est calculée à la partir de la précédente par dichotomie, selon la direction associée à la graine (voir *figure 4.4*). De plus, on emploie un marqueur booléen qui a pour rôle d'indiquer si la graine est à la surface ou non de la structure entière (l'objet est obtenu par assemblage de squelettes entourés de fonctions implicites).

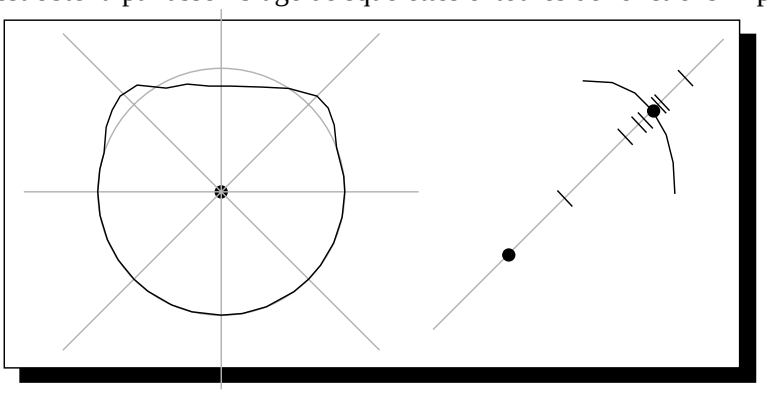

Figure 4.4. *Echantillonnage de surfaces implicites par graines*

Enfin, citons l'approche originale de [SP95] qui proposent un moyen rapide de détecter les collisions à partir d'objets définis par une surface implicite sans employer d'échantillonnage des surfaces. On suppose que les points à l'intérieur de l'objet sont positifs, à la surface nuls et à l'extérieur négatifs. L'intersection de deux corps $f_1(M) = 0$ et $f_2(M) = 0$ est détectée lorsqu'il existe un point M tel que $G(M) > 0$ où :

$$G(M) = min(f_1(M), f_2(M))$$

Le point de collision est considéré, par approximation, comme étant le point où $G(M)$ atteint son maximum, si ce maximum est positif. La méthode pour trouver ce maximum nécessite que la fonction à étudier soit de classe C_1. On change donc la définition de G, afin de la rendre dérivable. La fonction suivante est proposée :

$$G(M) = f_1(M) + f_2(M) - \sqrt{f_1^2(M) + f_2^2(M)}$$

La méthode de détection de collision repose alors sur une recherche de maximum de la fonction. Les méthodes employées sont diverses, mais il faut prendre garde à l'échantillonnage de l'espace lors de la recherche.

Discussion

Détecter des intersections peut sembler plus pratique pour les surfaces implicites que pour les modèles polygonaux. En effet, tester l'inclusion d'un point dans la structure nécessite simplement dans le premier cas l'évaluation d'une expression tandis que dans le second cas, il faut parcourir toutes les facettes de l'objet. Cette différence de complexité n'est qu'apparente. Tout d'abord l'expression d'une fonction implicite peut parfois être très gourmande en temps machine (si elle emploie des expressions trigonométriques ou exponentielles). Ensuite, les fonctions implicites exigent d'être échantillonnées durant la simulation ce qui devient la partie coûteuse de la détection.

b) Intersection 4D

Les algorithmes décrits ci dessous concerne une analyse géométrique de volumes dans un espace à quatre dimensions, qui décrivent les diverses positions des objets au cours du temps.

Décision du vide dans un arbre CSG

[Cam90] étudie le problème de la détection de collisions dans l'espace-temps entre des objets CSG en mouvement (voir [FDFH90], chapitre 12). Construire l'ensemble des positions décrites par l'objet au cours du temps revient à faire une extrusion des objets. L'algorithme de détection revient alors à la détermination du vide. Afin de simplifier l'algorithme, seuls les objets facettisés sont considérés. Tous les corps sont décrits comme une intersection de plans (hyperplans) et le problème s'écrit donc comme une combinaison booléenne sur des demi-espaces. Par ailleurs, on suppose que les corps se déplacent à vitesse constante par intervalle de temps.

Le principe de l'algorithme est alors simple : trouver simplement un point appartenant à l'intersection, ce qui prouve qu'elle n'est pas vide. Trois algorithmes sont utilisés, en chaîne :

- Utilisation de *S-bounds* : on considère des boîtes englobantes autour des objets. L'extrusions dans le temps de ces boîtes donnent l'hyperboîte englobant l'extrusion du corps (une pure boîte englobante dans l'espace 4D n'est pas intéressante car trop générale). On divise le temps en courts intervalles où l'on suppose que les objets se déplacent à vitesse constante ;
- *Diviser pour conquérir* : l'espace de départ est divisé en sous espaces où l'on teste des incompatibilités d'équations, ceci a pour résultat de réduire l'espace de recherche ;
- Découpage de segments : on prend l'intersection de trois hyperplans, on obtient (en général) une ligne qui sera coupée par tous les autres hyperplans, on cumule le découpage puis on teste à la fin s'il reste un segment ou si l'ensemble obtenu est vide.

Intersection par surfaces paramétriques lipschitziennes

[HBZ90] se placent dans le cadre des surfaces paramétriques, de nature quelconque, mais soumis à la condition de Lipschitz, c'est à dire :

$$\left\| \vec{f}(\vec{u}_2) - \vec{f}(\vec{u}_1) \right\| \leq L \left\| \vec{u}_2 - \vec{u}_1 \right\|$$

Dans le calcul on considère des surfaces qui incluent le paramètre temps (on considère donc des super-surfaces $\vec{f}(u, v, t)$). Le principe de base de la détection consiste en une subdivision adaptative des surfaces paramétriques. La condition de Lipschitz et son extension à la matrice jacobienne (matrice constituée des bornes supérieures des composantes de la matrice jacobienne sur le domaine) permettent de créer des sphères ou des boîtes englobantes. On échantillonne l'espace des paramètres de la surface paramétrique en choisissant un intervalle suffisamment petit : si l'on veut des volumes englobants de taille inférieure à D, il suffit d'échantillonner avec un intervalle de taille D/L .

Lorsque l'on détecte une intersection entre deux boîtes englobantes, on divise l'une des deux surfaces en collision (en coupant le volume englobant en deux, suivant sa plus grande dimension u, v ou t), et ainsi de suite jusqu'à trouver une intersection entre deux boîtes de dimensions inférieures à un seuil donné. On considère dans ce cas qu'il y a intersection.

Lorsqu'un nouveau raffinement doit être effectué, c'est à dire lorsque deux volumes englobants se rencontrent, les nouvelles intersections à calculer sont stockées dans une pile, triée par ordre croissant de temps. Cette méthode garantit donc par construction de trouver la première collision dans le temps.

Discussion

On constate que les opérations de calcul d'intersection sur des volumes en 4D sont des solutions très gourmandes en calcul. Les algorithmes proposés reposent la plupart du temps sur une

dichotomie ou des divisions successives... c'est à dire des algorithmes itératifs peu adaptés au temps réel.

c) Calculs analytiques

Dans les approches suivantes, on considère le temps comme un paramètre particulier que l'on cherche à isoler dans les équations afin de trouver le temps exact de la collision.

Calculs exacts de Boyse

[Boy79] affirme que lorsque deux objets polygonaux entrent en collision, c'est toujours lorsqu'une arête de l'un des objets entre en collision avec une face de l'autre objet. Deux situations se présentent alors, soit une extrémité du segment atteint la surface, soit l'arête touche une des arêtes de la surface (voir la *figure 4.5*). Le premier cas se calcule facilement en déterminant l'intersec-

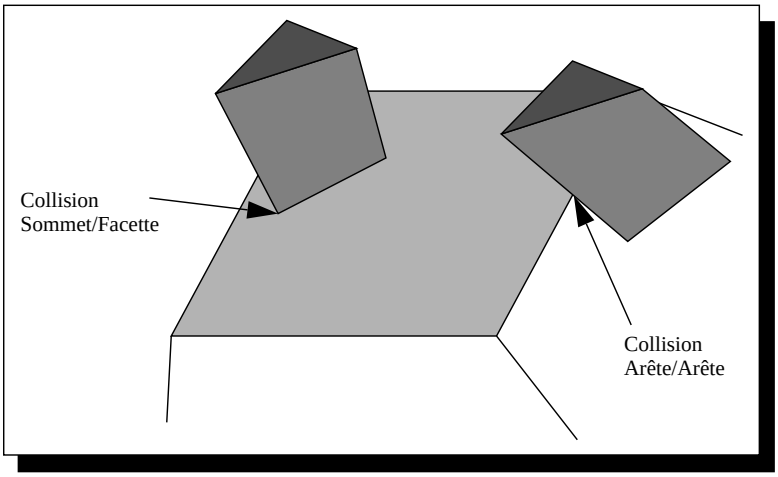

Figure 4.5. *Collision Sommet/Facette et Arête/Arête*

tion entre la surface et la trajectoire des extrémités du segment. Le second cas est plus délicat. On considère la surface balayée par le segment lors de son mouvement. Malheureusement ce cas est plus difficile à détecter pour des mouvements quelconques. Seuls les cas particuliers de la translation et de la rotation sont étudiés et sont traités dans la suite.

Soit une arête en mouvement de rotation. Le premier cas de collision apparaît quand l'une des extrémités du segment entre en collision avec la facette. Pour simplifier le calcul, on suppose, sans perte de généralité, que la rotation a lieu autour de l'axe $\vec{z}$, donc dans un plan du type $z = cste$. Une extrémité de l'arête reste donc sur un cercle d'équation :

$$x^2 + y^2 = R^2$$

Si l'extrémité entre en collision avec la facette, alors il obéit aussi à l'équation de la facette, supposée fixe, soit :

$$ax + by + cz + d = 0$$

En combinant les deux équations, on obtient un système du second degré. Une fois la solution $(x^{t+\delta t}, y^{t+\delta t})$ obtenue, il faut déterminer si elle se trouve sur la partie de cercle correspondant à la trajectoire du sommet. Pour cela on vérifie que la position se trouve bien entre l'angle de départ et l'angle d'arrivée soit :

$$\text{atan}\left(\frac{y^t}{x^t}\right) \leq \text{atan}\left(\frac{y^{t+\delta t}}{x^{t+\delta t}}\right) \leq \text{atan}\left(\frac{y^{t+\Delta t}}{x^{t+\Delta t}}\right)$$

Pour cette inégalité, on suppose que la rotation a lieu à angle croissant, elle doit être inversée dans le cas contraire. Enfin, on suppose que la solution trouvée est effectivement incluse dans les limites de la facette.

Le second cas de collision se résout de la façon suivante. On suppose que l'arête en mouvement subit une rotation d'angle θ autour de l'axe $\vec{z}$. A un instant quelconque, un point se situant sur le segment liant les deux extrémités de l'arête obéit à :

$$x(u,t) = (x_2(t) - x_1(t))\,u + x_1(t)$$

$$y(u,t) = (y_2(t) - y_1(t))\,u + y_1(t)$$

$$z(u,t) = (z_2(t) - z_1(t))\,u + z_1(t)$$

Ce qui nous donne l'équation de la surface balayée par le segment au cours de son mouvement. Durant ce mouvement, chaque point décrit une trajectoire circulaire, ce qui nous permet d'éliminer le paramètre temps :

$$x^2 + y^2 = \left[\left(x_2^{t_0} - x_1^{t_0}\right)u + x_1^{t_0}\right]^2 + \left[\left(y_2^{t_0} - y_1^{t_0}\right)u + y_1^{t_0}\right]^2$$
$$z = \left(z_2^{t_0} - z_1^{t_0}\right)u + z_1^{t_0}$$

D'où, en combinant les deux et en allégeant la notation :

$$x^2 + y^2 = \left[(x_2 - x_1)\frac{(z - z_1)}{(z_2 - z_1)} + x_1\right]^2 + \left[(y_2 - y_1)\frac{(z - z_1)}{(z_2 - z_1)} + y_1\right]^2$$

Ce qui est l'équation d'une quadrique.

Ensuite, on suppose que le point entre en collision avec l'autre arête, supposé fixe :

$$x = (X_2 - X_1)\,v + X_1$$

$$y = (Y_2 - Y_1)\,v + Y_1$$

$$z = (Z_2 - Z_1)\,v + Z_1$$

On injecte ces équations dans l'équation de la quadrique précédemment calculée, et on obtient une équation du second degré en *v*. Si on trouve des solutions comprises entre 0 et 1, alors le segment intersecte la quadrique. On exprime les coordonnées du point d'intersection puis on vérifie qu'il se trouve sur le carreau[1] de quadrique décrit par l'arête au cours du mouvement. Cela se teste facilement en inspectant si la coordonnée z est située entre les deux bornes z_1 et z_2 et si l'angle $\operatorname{atan}\left(\frac{y^{t+\delta t}}{x^{t+\delta t}}\right)$ est bien entre les limites dues à la rotation.

Par ailleurs, [Can86] étend son formalisme à base de quaternions pour incorporer la détection de collision au cours d'un mouvement. On suppose ici que la trajectoire est linéaire uniforme dans l'espace des configurations. Le corps A est supposé au départ à l'origine. Durant le mouvement, il se déplace à vitesse constante vers la position $\vec{x}_A$, sa trajectoire linéaire est donc $t\vec{x}_A$. Son orientation est supposée évoluer de 0 à θ. On caractérise l'orientation finale par le quaternion Q_θ que l'on norme de telle sorte que sa partie scalaire soit 1, on le note $1 + q_\theta$. Durant le trajet, l'orientation du corps est donc de $1 + tq_\theta$. Comme il n'y a pas proportionnalité entre les quaternions et les angles de rotation, il est faux de penser que la rotation se fait à vitesse angulaire constante. L'équation *(4.1)* se transforme en :

$$\left[(1 + tq_\theta)\,F_A\,(1 - tq_\theta)\;(1 + \vec{p}_B - t\vec{x}_A)\right] = 0$$

On obtient donc une équation du troisième degré en *t*. Les équations *(4.2)* et *(4.3)* caractérisant les autres types de collisions deviennent également des équations du troisième degré, car ces équations sont quadratiques vis à vis des composantes du quaternion et linéaires vis à vis des composantes de la translation.

En conclusion, on constate que dans les deux méthodes exposées ci-dessus, même dans des cas très simples de mouvement de rotation et de translation d'un objet en collision avec un autre

1. *patch* en anglais.

objet supposé fixe, les calculs sont lourds. On obtient soit une équation de degré deux demandant le calcul d'une racine carré, suivi de l'évaluation d'un arctangente, soit une équation de degré trois exigeant l'appel à des fonctions trigonométriques. Ces opérations sont très coûteuses et à éviter dans la mesure du possible dans notre contexte temps réel.

Cas particulier de facettes triangulaires

[MW88] font par rapport aux calculs précédents l'hypothèse de mouvements rectilignes uniformes (c'est à dire le long d'une droite et à vitesse constante) entre les instants de la simulation. Pour la détection de collisions, ils affirment que le cas de collision entre un sommet d'un objet et une facette d'un autre objet est suffisant (les cas de collisions entre arêtes sont négligées). Les facettes des objets sont supposées triangulaires, sans perte de généralité.

Si on considère la facette fixe au cours du temps (où si on se place dans un repère lié à la facette), la collision entre un sommet et la facette déterminée par une résolution d'intersection entre le segment représentant la trajectoire du sommet et le plan de la facette. Par contre, si la facette est mobile, et même déformable, la résolution est plus délicate. Le problème se pose de la façon suivante :

Soit P le sommet du premier objet, il est animé d'une vitesse $\vec{V}$. La facette a pour sommets P_0, P_1 et P_2 animés respectivement des vitesses $\vec{V}_0$, $\vec{V}_1$ et $\vec{V}_2$. Le sommet rencontre la facette si on trouve $t \in [0,\Delta t]$, $u \geq 0$, $v \geq 0$, $u + v \leq 1$ tels que :

$$P + \vec{V}t = P_0 + \vec{V}_0 t + \left((P_1 - P_0) + \left(\vec{V}_1 - \vec{V}_0 \right) t \right) u + \left((P_2 - P_0) + \left(\vec{V}_2 - \vec{V}_0 \right) t \right) v \qquad (4.4)$$

Ce système, une fois la variable t isolée, se retrouve sous la forme d'un polynôme de degré 5. Des solutions itératives sont alors utilisées pour le résoudre. L'intervalle $[0,\Delta t]$ est découpé en sous-intervalles et une recherche binaire est effectuée lorsqu'un changement de signe est détecté. Cette solution est rapide et converge toujours.

Discussion

Peu de méthodes exploitant le calcul analytique ont été proposées. En effet, même sous des hypothèses contraignantes (déplacements uniquement rigides ou mouvements rectilignes uniformes) les calculs sont très lourds. Ils sont en général itératifs.

d) Quelques problèmes connexes

La détection doit pour certaines circonstances être complétée afin de régler des problèmes courants comme la collision multi-points ou la collision d'un objet avec lui-même (*auto-collision*).

Cas de collisions multi-points

[SWFC93] traitent du problème général des collisions multiples, où deux cas apparaissent : plusieurs collisions ont lieu en même temps en des points isolés ou une collision a lieu en un nombre infini de points (ex: un tore sur le sol). Pour cela un algorithme est proposé, fondé sur les intervalles. Tout d'abord le problème de collision est transformé en un problème de contrainte en exprimant les faits suivants :

- La collision a lieu en un point situé sur les deux courbes ;
- Les plans tangents des courbes au point de contact sont confondus ;
- Les deux corps se rapprochent.

Les deux premières conditions impliquent des égalités tandis que la dernière implique des inégalités. Cet ensemble d'égalités et d'inégalités forment les contraintes, et c'est dans cet espace

de contraintes que l'on va essayer de minimiser la fonction $f(u) = t$ où u est le vecteur d'état du système et t le temps, c'est à dire trouver la collision la plus proche dans le temps.

L'algorithme fonctionne à l'aide de subdivisions successives et de critères tirés de la théorie des intervalles permettant de déterminer l'existence d'une ou de plusieurs solutions. A la subdivision est ajoutée une méthode de raffinement, qui permet de réduire encore plus rapidement les intervalles (méthode de Newton).

Traitement des auto-collisions

Le problème spécifique de l'autocollision est délicat. En effet, cela ne suffit pas d'appliquer les algorithmes de détection énumérés dans cette partie à un objet sur lui-même. Par exemple, si deux polygones sont côte à côte et partagent une arête commune, ils seront considérés comme entrant en collision arête sur arête. De la même façon, tout sommet d'une facette sera en collision avec cette facette. On pourrait alors tenir compte de la topologie de l'objet pour la détection. Pourtant, de tels critères de juxtaposition ne suffisent pas car, par exemple, deux polygones juxtaposés peuvent effectivement entrer en collision s'ils se replient l'un sur l'autre.

[VT94] se basent sur une propriété particulière des surfaces. Ils donnent une condition suffisante pour exprimer le fait qu'une surface n'entre pas en collision avec elle même :

S'il existe un vecteur $\vec{V}$ tel que pour tout point M de la surface, $\vec{N}_M \cdot \vec{V} > 0$ où $\vec{N}_M$ est la normale à la surface en M et où la projection du contour sur un plan orthogonal à ce vecteur $\vec{V}$ n'a pas d'auto-collision (en 2D), alors la surface n'entre pas en collision avec elle-même.

Afin de profiter de cette propriété, une structure de données hiérarchiques, par régions décrivant les voisinages topologiques est donnée. En effet, lors des test d'autocollision, on ne désire pas s'occuper des effets de voisinages. Ensuite afin de trouver le vecteur $\vec{V}$, on procède avec des échantillons bien répartis de direction de l'espace. Ensuite un champ de bit est propagé dans l'arbre afin de trouver le candidat approprié à l'application de la propriété. Ensuite la vérification de l'intersection sur le plan est effectuée avec des algorithmes classiques.

La robustesse de l'algorithme a été améliorée dans [VT95]. Ainsi, à partir de configurations où il n'y a pas intersection ou au contraire des situations inconsistantes, les auteurs cherchent un algorithme permettant de conserver ou rétablir une situation consistante. Le point délicat à régler concerne le dépistage des orientations incohérentes et la détection du sens des collisions.

4.1.3. Conclusion générale sur la détection de collisions

Les calculs de détection de collisions sont très lourd, quelles que soient la modélisation mathématique et la méthode de détection. La plupart du temps, seule la collision entre un point du premier objet et la surface de l'autre objet est calculée. Certaines collisions ne sont donc pas détectées. Pire encore, dans des modèles polygonaux, si on néglige les collisions entre arêtes, certaines aberrations peuvent survenir. La *figure 4.6* illustre (en vue de coupe) le cas particulier d'un objet en chute libre tombant sur un plan où la collision entre point et surface seule ne suffit pas. A l'instant *(a)*, le cube tombe en chute libre. A l'instant *(b)* on détecte une collision entre l'un des sommets du cube et le plan incliné. Le cube bascule et atteint la position décrite en *(c)*. S'il y a détection de collision entre les arêtes du cube et la bordure du plan, le cube reste en équilibre à sa position ou bascule autour du bord du plan. Sans détection de collisions entre arêtes, le cube continue son mouvement et se retrouve dans la position *(d)*, immobile, car le point initial de la collision est considéré comme étant au dessus du plan et en collision avec lui.

Un autre moyen de lutter contre ces phénomène d'interpénétration par intersection d'arêtes est d'analyser finement les zones de contact et d'ajouter des points sur les arêtes, afin de construire le polygone de contact (polygone résultat de l'intersection des deux facettes en contact). La construction de ce polygone est coûteuse. En outre, le polygone de contact n'est pas forcément fixe au cours du temps (lorsque l'objet glisse sur le bord d'une pente par exemple). On risque alors de démultiplier le nombre de points définissant la zone de contact lors du glissement.

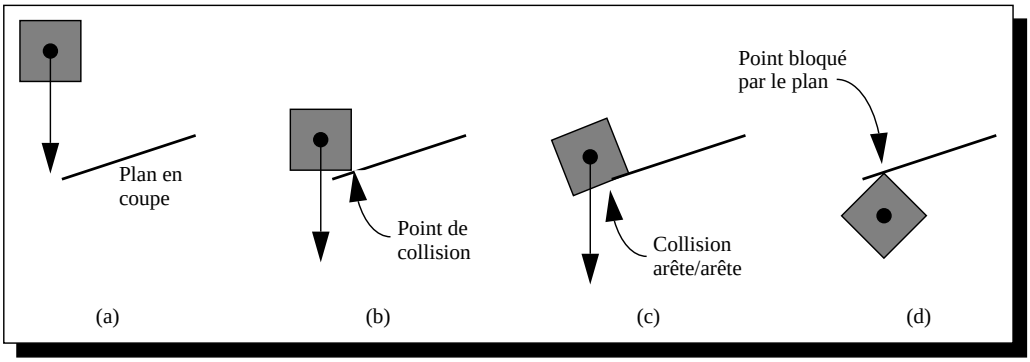

Figure 4.6. *Cas de collision arête/arête critique (vue en coupe)*

Enfin, selon [Hah88] le temps nécessaire à la détection dans un contexte de corps rigides représente entre 70% et 99% du temps de calcul total. On constate donc qu'il faut absolument avoir recours à des algorithmes d'accélération de la détection de collisions. Ces algorithmes, pour être justifiés, doivent être simples et rapides. Nous les présentons dans le paragraphe suivant.

4.2. Accélération de la détection de collisions

Les techniques de détection sont le cas général de complexité $o(n^2)$, où n est le nombre d'objets de la scène. De plus entre chaque objet, la complexité de la détection de collision est en $o(p^2)$ où p est le nombre de primitives moyen par objet. La simulation ralentit de façon alarmante avec l'augmentation du nombre de primitives.

L'accélération de la détection de collisions consiste en la détection de non collision. Le cas idéal serait de fixer que certains objets ne peuvent pas, par hypothèse de départ, construction géométrique ou règle arbitraire, entrer en collision. Par exemple, deux objets mobiles, mais ayant par exemple un degré de liberté contraint les obligeant à toujours rester dans des zones de l'espace disjointes, n'ont jamais la possibilité d'entrer en collision en cours de simulation. Il est donc parfaitement inutile de tester leur collision en cours de simulation. Une solution proposée est par exemple l'usage d'une *matrice de collisions*, constituée de booléens indiquant si deux objets sont, par définition de l'environnement, susceptibles ou non d'entrer en collision [GSF94].

L'idée est de se doter d'une batterie de tests simples et rapides, permettant d'éliminer le maximum de cas d'intersections impossibles. Ces tests évitent ainsi d'exécuter des calculs exacts d'intersection. Par la succession de ces tests, on crée une sorte de pipeline. On effectue les tests les plus rapides tout d'abord, suivi de tests plus complexes et ainsi de suite. A chaque étage on élimine des cas et ainsi on n'effectue les tests les plus coûteux que lorsque cela est véritablement nécessaire. A la fin du pipeline, seuls les facettes, sommets ou arêtes pouvant de façon très probable entrer en collision subsistent.

Naturellement, les tests dans l'accélération dépendent fortement du type de détection de collision que l'on a choisi (intersection 3D ou de trajectoires). Cependant, il nous paraît plus constructif de classer les diverses solutions existantes selon le type de stratégie employé plutôt que sur le type de détection. Les stratégies que nous considérons sont les suivantes :

- subdivision de l'espace : on élimine les couples d'objets ne se trouvant pas dans la même portion d'espace ;
- volumes englobants : on teste d'abord la collision entre des volumes simples englobants les objets ;
- topologie : on considère les distances entre les objets ;
- cinématique : on élimine les couples d'objets dont les trajectoires rendent toute collision impossible.

On remarquera que les méthodes exposées partagent parfois des caractéristiques avec les techniques d'accélérations de lancer de rayons (voir [FDFH90], chapitres 15 et 16). En effet, ces problèmes ont un but commun : savoir détecter si une intersection peut ou non avoir lieu.

4.2.1. Elimination par subdivision de l'espace

L'idée de base de ces algorithmes est très simple. Deux objets se situant en des endroits éloignés de l'espace n'ont aucune chance d'entrer en collision. En subdivisant l'espace, on se fixe des zones bien séparées et on cherche pour chaque objet dans quelle zone il se situe.

a) Subdivision régulière de l'espace

L'idée la plus simple (et une des plus employées dans les algorithmes de jeux vidéos) est de partitionner l'espace grâce à une grille régulière. Tout objet de la simulation peut être placé dans cette grille et on ne fait le test que pour des objets se situant dans la même grille. Cet algorithme est classique en image de synthèse et est couramment employé, y compris dans des domaines relativement différents comme le lancer de rayons ou les machines de synthèse d'images parallèles. Les problèmes liés à cette technique sont bien connus. Le niveau de subdivision de la grille est délicat à définir. Lorsque la grille est trop fine, l'espace mémoire nécessaire au stockage augmente considérablement. Si on contraire, elle est trop grossière, les objets ont plus de chance d'être présents dans plusieurs zones à la fois. D'un point de vue calculatoire, cela revient à considérer que les objets sont dupliqués. Au pire, un objet peut se situer à un croisement de zones et dans ce cas, il est présent dans huit zones simultanément (en 3D).

L'autre inconvénient de la subdivision régulière de l'espace est de ne pas tenir compte du principe de localité. Les objets ne sont généralement pas répartis de façon homogène dans la scène. Les objets sont souvent regroupés dans certaines zones. Ainsi des zones vont être surchargées d'objets tandis que d'autres seront vides. La solution souvent adoptée pour pallier ces problèmes de densité hétérogène est la subdivision adaptative.

b) Subdivision adaptative de l'espace en voxels

Nous présentons ici un exemple de subdivision adaptative, propre au problème des détections de collisions. Dans la simulation de [BT95], l'espace est subdivisé en voxels[1]. Ces voxels ont une structure de type octree (voir [FDFH90], chapitre 12), car ils peuvent être eux-même subdivisés. Les objets sont supposés polygonaux et chaque surface plane est soit triangulaire soit rectangulaire. Chacune des surfaces des objet est balayée, tel un remplissage, grâce à un algorithme de type DDA[2] en trois dimensions (voir [FDFH90], chapitre 3). Lors du balayage on détermine à quel voxel appartient la partie courante de la surface afin d'ajouter la facette à la liste associée au voxel. Ainsi chaque voxel contient la liste des facettes passant par lui. Afin de contrer les erreurs d'arrondi lors du DDA, et surtout le fait que le DDA ne considère que le centre des voxels pour le test d'appartenance de la facette, le remplissage se fait sur un sur-échantillonnage de la division en voxel (d'un point de vue arithmétique, cela revient à travailler en virgule fixe). Cependant, l'environnement est supposé dynamique, les objets étant en mouvement, il faut déterminer à chaque pas de temps dans quels octrees il se situent. Pour cela, on considère le prisme formé par la position à l'instant t et celle à l'instant $t + \Delta t$, et on balaye le prisme de la même façon que pour une facette afin de remplir les octrees.

En cours de simulation, on teste les interpénétrations en inspectant si deux facettes passent par le même voxel. Dans ce cas, on discrétise de nouveau les deux facettes, mais à un niveau plus fin (d'où la structure en octree) afin de déterminer si oui ou non il y a interpénétration. On descend jusqu'à un niveau suffisant dans la structure.

1. *voxel* est la contraction de *volume élément*, soit élément de volume en français
2. *Digital Differential Analyser*

c) Tri selon la profondeur en chaque pixel

[FS91] étudie une nouvelle méthode pour détecter rapidement les interférences entre objets. On projette sur l'écran tous les objets par *scan-conversion* (voir [FDFH90], chapitre 3), et en chaque pixel, on repère le Z_{min} et le Z_{max}, que l'on stocke dans une liste, avec le numéro de la facette concernée. Une fois tous les objets convertis, on trie en Z la liste associée à chaque pixel, et on vérifie si les paires de Z consécutifs sont associées aux mêmes objets. Si ce n'est pas le cas, on a détecté une collision possible. Les numéros de faces stockés dans la liste sont les faces entre lesquelles il faut calculer l'intersection. Afin de limiter les calculs, on peut ajouter un simple test de comparaison entre les valeurs Z_{min} et Z_{max} de chaque objet, avant de dérouler l'algorithme de Z-liste par pixel. Cette méthode est capable d'exploiter le matériel dans le cas d'un tampon de profondeur (Z-buffer) câblé.

4.2.2. Elimination par zones d'influence

Cette stratégie repose sur des volumes (boîtes, sphères) englobants. On suppose que l'interpénétration de volumes simples est rapide à déterminer et peut remplacer avantageusement un calcul d'intersection réel entre objet plus complexes. Par exemple, l'interpénétration de deux sphères consiste simplement à tester si le carré de la distance entre les deux centres est plus petit que le carré de la somme des deux rayons.

a) Comparaison de points et de volumes englobants

[MW88] proposent une optimisation à base d'octrees et de boîte englobante. Dans cette méthode, on ne teste pas les boîtes englobantes ente elles, mais on cherche à comparer les points d'un des objets par rapport à la boîte englobante d'un autre objet. On trie la position des sommets d'un objet dans un arbre. Pour cela, on parcourt les points et on les insère dans l'arbre. Dans cet arbre, chaque point possède huit sous-arbres contenant tous les autres points se situant autour de ce point, dans l'octant de l'espace associé. L'algorithme de construction est simple. On place le premier sommet de l'objet en tant que racine de l'arbre. Ensuite, on place les autres sommets en parcourant l'arbre et en cherchant, à chaque noeud de l'arbre, dans quel octant autour de ce nœud se situe le sommet à placer. Suivant l'octant, on passe à l'un ou l'autre des fils et ainsi de suite jusqu'à arriver à une feuille de l'arbre.

La *figure 4.7* montre sur un exemple 2D (donc avec des quadtrees) le résultat obtenu. On suppose que l'on a placé les sommets dans l'ordre alphabétique.

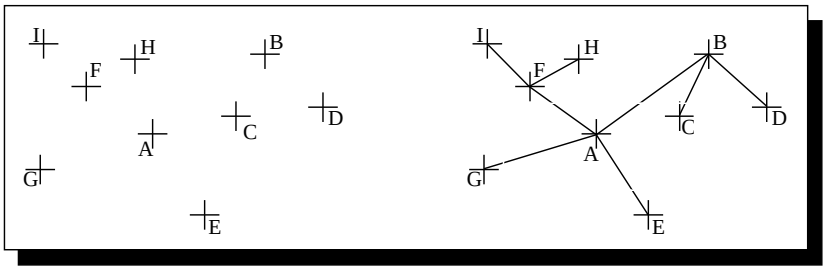

Figure 4.7. *Répartition des points en quadtrees*

La détection peut alors exploiter cette répartition. On entoure les facettes avec une boîte englobante. Si la facette est en mouvement, la boîte doit englober toute la trajectoire de la facette. Ensuite, on trouve rapidement les points situés dans le volume, c'est à dire ceux susceptibles d'entrer en collision avec la facette et pour lesquels il faudra ensuite faire une analyse plus fine.

L'algorithme fonctionne sur le principe suivant : si le point est situé à l'intérieur du rectangle, on inspecte ses sous-arbres. S'il est à l'extérieur, on n'inspecte que les sous-arbres allant dans la direction de la boîte englobante. Ainsi, dans l'exemple de la *figure 4.7*, les sous-arbres du point A sont inspectés car ce dernier est dans la boîte englobante. Par contre, seul le sous arbre du point B

correspondant au quadrant bas, gauche est inspecté, c'est à dire qu'on va faire le teste pour le point *C* et ses sous-arbres, mais pas pour le point *D*.

b) Intersection de volumes englobants

Les deux volumes englobants généralement employés sont la boîte ou la sphère. Ce sont en effet ces deux volumes qui offrent les calculs d'intersection les plus simples. De plus, ils est relativement simple d'englober des objets complexes avec ces deux volumes. Cependant, si les volumes ne s'approchent pas suffisamment de la topologie de départ des objets, on détectera plus souvent des intersections entre les volumes englobants sans qu'il y ait intersection entre les objets. Donc, plus un volume englobant est proche de l'objet de départ, plus il est efficace et remplit pleinement son rôle d'accélération de la détection des collisions. Afin de quantifier cette notion de proximité de forme, on définit la *distance de Hausdorff* comme :

$$\tau = max\{min\{dist(g,b), g \in G\}, b \in V\}$$

On cherche alors à la minimiser. Si le volume englobant est une sphère, la construction peut s'avérer délicate. En effet, si on cherche une sphère englobant un objet extrêmement fin et long, on obtient une sphère de diamètre la longueur de l'objet. La distance de Hausdorff est alors égale au rayon de la sphère, comme illustré sur la *figure 4.8*.

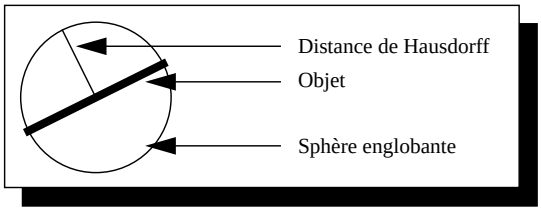

Figure 4.8. *Distance de Hausdorff*

[Hub96] s'intéresse à la création d'une hiérarchie de sphères englobant un objet. L'aspect hiérarchique permet une recherche de la collision par affinements successifs. Une première méthode est de construire un octree autour de l'objet et de définir une sphère autour de chaque nœud de l'octree contenant des parties de l'objet. Cette première solution est simple à mettre en oeuvre mais procure un résultat très approximatif. Les auteurs préconisent alors la recherche des *axes médians* de l'objet afin d'y positionner les sphères.

Dans la suite, les volumes englobants sont supposés être des boîtes, car elles permettent une distance de Hausdorff généralement plus faible.

c) Tris de boîtes englobantes suivant les axes

[Rob95] propose, dans un contexte de programmation de jeux vidéos, une méthode très intéressante afin d'accélérer la détection des collisions entre personnages du jeux. L'ensemble est présenté en 2D mais est tout à fait généralisable au cas 3D. On suppose chaque objet entouré d'une boîte englobante. Ces boîtes englobantes sont triées suivant les trois axes du repère global $\vec{x}$, $\vec{y}$ et $\vec{z}$. On obtient donc trois listes. Par exemple, pour l'axe $\vec{x}$, chaque élément est un couple du type $(x_{min}, \Delta x)$, où x_{min} est la borne inférieure de la boîte englobante et Δx la taille. Les deux autres axes sont obtenus de façon similaire.

L'algorithme part alors de la constatation suivante : soit un couple $(x^p_{min}, \Delta x^p)$ de la liste. On cherche, pour les couples suivants $(x^{p+k}_{min}, \Delta x^{p+k})$ si $x^{p+k}_{min} \leq x^{p+k}_{min} + \Delta x^p$. Si c'est le cas, cela veut dire que la boîte englobante associée est en collision possible avec la boîte du couple considéré. Si ce n'est pas le cas, alors il est inutile de continuer le test pour le reste des couples de la liste, puisque la liste est triée.

L'algorithme fonctionne alors de la façon suivante. On balaye une première liste. Pour chaque élément de la liste, on cherche les couples suivants susceptibles d'entrer en collision. La com-

plexité de cette phase est dans le meilleur des cas $o(n)$ (aucune collision) et dans le pire des cas $o(n^2)$ (tous les objets en collisions). Ce premier balayage a mis en évidence des couples d'objets en collisions possibles. On réitère le processus sur les deux autres axes, ce qui permet d'éliminer certains de ces couples, car l'ordre des éléments est généralement différent pour chaque axe. A la fin de l'algorithme, on obtient la liste des boîtes effectivement en intersection. Les objets correspondant aux boîtes ont donc toutes les chances d'être en collision.

L'auteur propose ensuite une méthode incrémentale permettant de se passer du tri systématique selon les trois axes. On bénéficie de la cohérence temporelle des positions les unes par rapport aux autres. En se basant sur le tri à l'instant précédent, seules quelques positions doivent être modifiées dans la liste. Les listes doivent donc être doublement chaînées.

En conclusion, cet algorithme est particulièrement intéressant pour une première phase de détection, dans des contextes avec un grand nombre d'objets, afin de déterminer rapidement les couples d'objets demandant un traitement plus fin.

d) Subdivision de boîtes englobantes : cas particulier des arbres CSG

[ZPG95] s'intéressent de manière plus formelle à la détection de collisions dans le cadre de la modélisation CSG. A chaque primitive dans l'arbre est associée une boîte englobante. Les opérations dans l'arbre CSG s'appliquent de façon identique sur les boîtes englobantes (appelés des *S-bounds*). Ensuite, les boîtes englobantes sont subdivisées en voxels de telle sorte qu'il y ait le moins de primitives par "voxel". La troisième étape consiste à faire le calcul exact de la collision. L'algorithme distingue les cas suivants, dans le cas de deux primitives par voxel seulement :

- les deux primitives sont polyédriques : les algorithmes testent alors les inclusions des sommets de l'un dans le volume délimité par l'autre, puis les côtés, puis enfin les cas où les faces sont alignées. Cet algorithme est identique à celui décrit au paragraphe *4.1.2* section *a* ;
- Si l'une est une courbe, on teste les points, les côtés et les faces, du polyèdre par rapport à la surface ;
- Si les deux sont des courbes, alors des polyèdres englobant (OBP) et inclus (IBP) sont utilisés afin de savoir rapidement s'il y a intersection. Si les deux OBP sont disjoints, il n'y a pas de collision, si les deux IBP s'intersectent, il y a collision. Il faut cependant souligner que cet algorithme ne fonctionne qu'avec des primitives convexes.

S'il reste plus de deux primitives par voxel, alors un algorithme de détection du vide est utilisé.

Les deux premières étapes de l'algorithme sont compatibles avec des modélisations autres que le CSG. La troisième étape, même si elle traite en théorie de la détection exacte de la collision, nous intéresse également en tant que technique d'accélération. En effet, le traitement par OBP et IBP des objets courbes peut-être considéré comme un moyen rapide de détection de collision ou non collision sûre entre surfaces courbes.

e) Comparaison de boîtes englobantes dans divers repères

[GSF94] proposent un pipeline pour la détection de collisions. Chaque étape est plus coûteuse que la précédente. L'algorithme tire sa justification de la considération suivante : tout algorithme se basant sur des boîtes englobantes est confronté aux problèmes liés au mouvement. Généralement, on définit la boîte englobante d'un objet dans un repère local. Dans ce repère, la boîte englobante est appelé *container* et elle est alignée sur les axes[1]. Une fois l'objet placé dans le repère global ou une fois que l'objet s'est déplacé, l'objet a subi une rotation et sa boîte englobante n'est plus parallèle aux axes. Or, les calculs d'intersection les plus rapides se font sur des boîtes parallèles, et non des boîtes englobantes quelconques. Si on calcule en temps réel une telle boîte autour

1. Par abus de langage, nous dirons boîtes parallèles en désignant les boîtes dont les faces sont parallèles aux axes.

de l'objet, cette dernière n'est généralement pas adaptée ni ne profite de la géométrie de l'objet. Ces problèmes sont illustrés sur la *figure 4.9* On constate que la boîte parallèle définie dans le repère global est assez éloignée du container de l'objet. Par ailleurs, la construction de la plus petite boîte parallèle d'un objet connaissant son orientation est coûteuse car elle demande la plupart du temps de trouver le minimum et le maximum des trois coordonnées de tous les points du corps.

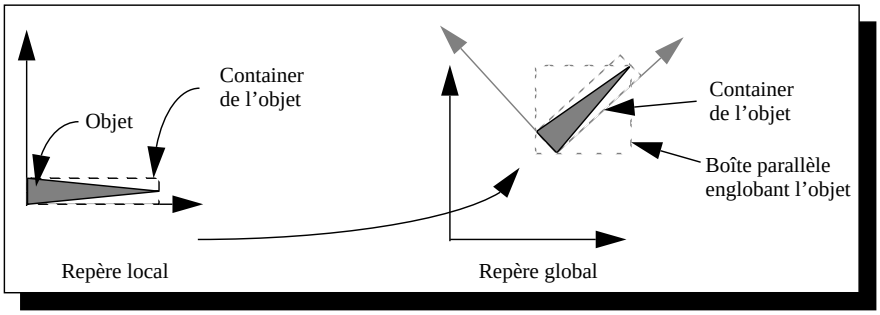

Figure 4.9. *Notion de boîte englobante et de container*

L'algorithme fonctionne en trois étapes :

Interférence de containers

On suppose donc que chaque objet possède une boîte englobante parallèle dans son repère global. On calcule ensuite le container de cette boîte englobante dans le repère global. Ainsi, il suffit de comparer les containers des boîtes englobantes de l'objet A et de l'objet B pour déterminer si les deux objets n'ont aucune chance d'être en collision. Chaque boîte est caractérisée par six paramètres X_{min}, Y_{min}, Z_{min}, X_{max}, Y_{max} et Z_{max}. Deux boîtes ne sont pas en intersection si l'une des conditions suivantes est vérifiée : $X^A_{min} > X^B_{max}$, $Y^A_{min} > Y^B_{max}$, $Z^A_{min} > Z^B_{max}$, $X^B_{min} > X^A_{max}$, $X^B_{min} > X^A_{max}$ ou $X^B_{min} > X^A_{max}$.

Interférence des voxels

Lorsque des boîtes parallèles s'intersectent, chaque container du repère propre de l'objet est divisé en voxels les plus réguliers possible dans le repère lié à l'objet. Ces voxels sont construits en découpant en deux parties égales la plus longue dimension du container, et ainsi de suite, jusqu'à obtenir une résolution voxel suffisante. Chaque voxel est rempli par la liste des facettes passant par ce voxel.

On se place alors dans le repère local de A et on calcule une boîte englobant le container de l'objet B et parallèle aux axes du repère local de A. Inversement, on fait la même opération en intervertissant A et B. On détermine alors la s-matrice d'interférence de A et B, c'est à dire l'ensemble des voxels de A par lesquels passe la boîte englobante de B et l'ensemble des voxels de B par lesquels passe la boîte englobante de A.

L'étape suivante de l'algorithme est de trouver l'ensemble des couples de voxels en intersection. On considère alors, dans le repère local de A, des boîtes englobant les voxels de B, et inversement, dans le repère local de B des boîtes englobant les voxels de A. On en déduit alors des couples de voxels en intersection.

Interférence des facettes

Des couples de voxels en intersection et en ayant éliminé les doublons, on tire alors une liste de couple (facettes de A,facettes de B) ayant une grande probabilité d'entrer en collision.

f) Comparaison de boîtes non alignées sur les axes

[GLM96] partent de la constatation suivante : il est simple de tester des intersections entre boîtes englobantes alignées sur les axes, mais ces volumes sont généralement très éloignés de la

géométrie de l'objet. Les auteurs cherchent donc à accélérer la détection d'intersection entre boîtes quelconques. Utiliser des boîtes quelconques se justifie par plusieurs raisons. Premièrement, ces boîtes sont définies dans le repère local de l'objet : elles sont généralement optimisées à la géométrie de l'objet. Deuxièmement, si l'objet se déplace en translation et/ou en rotation, la boîte englobante subit la même transformation. Enfin, troisièmement, si ces boîtes sont construites hiérarchiquement, pour un même nombre de volumes englobants, les boîtes quelconques permettent de descendre à un niveau plus bas, voire au niveau primitive, d'où une meilleure précision de la détection.

Les auteurs proposent un processus de construction de cette hiérarchie que l'on ne détaille pas ici. Cette phase doit être faite en précalcul, car elle est très coûteuse. Cette méthode n'est donc pas telle quelle, adaptée aux corps déformables (chaque déformation nécessite un recalcul des boîtes englobantes). Avec quelques modifications, elle pourrait être applicable aux corps mous mais ne serait peut-être plus optimale.

Le test d'intersection entre boîtes quelconques proposé part de la constatation suivante : si deux boîtes ne sont pas en intersection, alors il existe un plan les séparant. On montre que ce plan est soit parallèle à l'un des plans des boîtes (deux fois trois possibilités) soit parallèle à un plan formé par une arête de l'une des boîtes et une arête de l'autre (trois fois trois possibilités). En tout, quinze plans sont donc possibles. On considère alors les axes orthogonaux à ces plans. Si les boîtes sont en intersection alors il y a intersection sur chaque axe, des segments obtenus en projetant perpendiculairement les boîtes sur cet axe. Réciproquement, s'il existe un axe tel que les segments des projections des deux boîtes ne se croisent pas, alors il n'y a pas d'intersection.

Le point délicat de l'algorithme revient donc à détection l'intersection des segments sur les axes. Mathématiquement, ce calcul est effectué de la manière suivante : soient deux boîtes A et B dont les centres sont séparés par un vecteur de translation $\vec{T}$. On appelle $\overline{\overline{R}}$ la rotation permettant de passer du repère local de la boîte A vers le repère local de la boîte B. L'axe considéré est muni d'un vecteur directeur $\vec{L}$. La distance entre la projection des centres des deux boîtes sur l'axe considéré est égale à :

$$\left|\vec{T} \cdot \vec{L}\right| \tag{4.5}$$

Le longueur du segment projection de la boîte A est obtenue en projetant le plus grand rayon de la boîte. Si on appelle a_i la moitié de la longueur, la largeur et la profondeur de la boîte, et $\vec{A}_i$ les vecteurs unitaires du repère local de A, le segment a un rayon de :

$$\sum_i \left|a_i \vec{A}_i \cdot \vec{L}\right| \tag{4.6}$$

On obtient une formule analogue pour B. Il faut signaler que dans cette somme de trois termes, il arrive fréquemment que un ou deux termes soient nuls, car l'axe n'est pas quelconque. En affirmant ensuite que les segments ne s'intersectent pas si la distance séparant leur centre est supérieure à la somme de leur rayon, on teste donc :

$$\left|\vec{T} \cdot \vec{L}\right| > \sum_i \left|a_i \vec{A}_i \cdot \vec{L}\right| + \sum_i \left|b_i \vec{B}_i \cdot \vec{L}\right| \tag{4.7}$$

Ce test est très rapide. De plus, les deux sommations de trois termes ne donnent que quatre termes car deux parmi ces six sont nuls, pour chaque axe. Les auteurs montrent comment déterminer rapidement, à l'aide des coefficients de $\overline{\overline{R}}$ la valeur des quatre termes nécessaires.

Ces tests nécessitent deux cents opérations dans le pire des cas (boîtes en intersection), mais que, en moyenne, en parcourant la moitié des axes, on détecte la non-interpénétration. En moyenne, il faut donc cent opérations. En comparaison de cette méthode, si on cherche à déterminer directement l'intersection entre deux boîtes quelconques, il faut calculer les intersection entre chaque segment de l'une des boîtes et les faces de l'autre, et vice-versa. En tout, il faut cent quarante-quatre calculs d'intersection droite/plan.

g) Bilan

Les volumes englobants sont intéressants puisqu'ils permettent des tests simples et rapides éliminant de façon sûre des cas de non-interpénétration. Pourtant, ils ont quelques inconvénients. Si l'objet est solide mais se déplace, la boîte englobante se déplace avec lui et ses plans ne sont plus parallèles aux axes. Il faut dans ce cas calculer une nouvelle boîte englobante ou changer de repère. Sitôt que l'objet est concave ou de longueur nettement plus grande que sa largeur et/ou sa hauteur, la boîte englobante est souvent mal adaptée et laisse beaucoup d'espace vide inutile. Par ailleurs, si le corps est déformable, le volume englobant n'est pas constant et doit être calculé à chaque pas de temps. Ensuite, il faut se soucier de la précision avec laquelle on fait des approximation des objets à l'aide de volumes englobants. Il est facile et rapide de calculer l'intersection de sphères ou de boîtes alignées sur les axes. Cependant, ces objets ne respectent généralement pas suffisamment la géométrie de l'objet. De plus, les boîtes alignées sur les axes doivent être recalculées à chaque rotation de l'objet. Au contraire, des boîtes englobantes quelconques suivent les mouvements rigides des objets et peuvent de ce fait rester toujours très proches de la forme de l'objet. Un nombre réduit de boîtes est alors nécessaire. Cependant, les calculs d'intersection entre boîtes quelconques sont plus délicats. Un compromis est alors à trouver entre le nombre et la précision des volumes englobants et le temps de calcul associé à leur intersection.

Pour l'accélération, il est très intéressant d'employer des volumes englobants hiérarchiques [Hah88], [GLM96], [Del96], [Hub96]. Ce dernier propose une méthode afin d'exploiter la hiérarchie pour le temps réel (cette méthode est qualifiée de *time-critical*). Chaque pas de temps de simulation se caractérise par trois étapes : l'étape d'évolution de la simulation, l'étape de détection des collisions et l'étape d'affichage. En calculant la moyenne des temps nécessaires à la simulation et à l'affichage des pas de temps précédents, on peut trouver une borne supérieure du temps à ne pas dépasser pour la détection des collisions. Par un processus hiérarchique, on parcourt les premiers niveaux de tous les objets entre eux. S'il reste du temps, on passe au niveau suivant de la hiérarchie et ainsi de suite jusqu'à épuisement du crédit de temps. Cette méthode garantit le temps d'exécution de la détection, au détriment de la précision : toutes les *vraies* collisions sont détectées, par contre si des objets sont assez rapprochés, des *fausses* collisions supplémentaires pourront être détectées.

4.2.3. Elimination par critères topologiques

Ce type d'élimination repose sur un principe simple. Pour chaque objet, on détermine les autres objets proches de lui. Ce sont avec ces objets que les collisions sont les plus probables.

a) Position relative des objets entre eux

Le premier principe que l'on peut employer est d'inspecter de quel côté de chaque facette d'un objet B se trouvent les arêtes d'un objet A. Ainsi, on conserve en mémoire le signe de la distance des sommets de A par rapport à chaque facette de l'objet B. On ne calcule l'intersection que s'il y a changement de signe de la distance. Cette méthode est en pratique peu intéressante car deux objets peuvent souvent passer l'un au delà de l'autre tout en étant très éloignés l'un de l'autre. En termes plus mathématiques, dans la méthode exposée, le point d'intersection calculé n'est pas forcément inclus dans la facette. Dans cette méthode, il manque un principe de localité.

Une autre méthode proposée par [Bar90] consiste à exhiber un plan tel que les deux objets à comparer se situent de part et d'autre de celui-ci. Ce plan est généralement supporté par une face de l'un des objets ou une arête de chaque objet. On marque alors, durant la simulation, quelle est la facette ou quelles sont les arêtes concernées pour pouvoir les tester à l'instant d'après. Si on constate que le plan n'est plus valable, il faut chercher un autre plan séparateur, cette partie est très coûteuse. Cette méthode existe aussi pour les corps déformables, mais est moins efficace, car il faut chercher les deux points les plus proches des deux corps.

b) Calcul de distances entre objets

Les méthodes précédentes manquent d'une notion de localité. On cherche alors à déterminer les distances minimales entre les objets de la simulation.

[CC86] proposent un calcul de la distance minimale entre deux objets convexes à savoir la plus petite translation telle que les deux objets soient en contact sans interpénétration. Elle est exprimée comme étant :

$$MTD^*(A, B) = inf\{|t|, contact(A, B + t)\}$$

Comme on désire faire la distinction entre le cas où les corps s'interpénètrent ou non, la Distance Minimale de Translation *MTD* est définie par :

$$MTD(A, B) = \begin{cases} -MTD^*(A, B) ; (A \cap B \neq \emptyset) \\ MTD^*(A, B) ; (A \cap B = \emptyset) \end{cases}$$

Pour la calculer, une propriété est démontrée : la plus petite distance translationnelle entre deux corps A et B est égale à la plus petite distance translationnelle entre l'origine et le polygone différence de Minkowski des deux objets. Ainsi, on montre que :

$$MTD(A, B) = MTD(\{0\}, CON(A, B))$$

La différence de Minkowski de deux solides est définie comme :

$$CON(A, B) = \{b - a; a \in A, b \in B\}$$

Une méthode est donc proposée pour construire la différence de Minkowski, comme intersection de demi-espaces. On cherche donc à trouver les plans qui délimitent ce polygone. On considère un plan passant par une facette de l'un des objets et on le translate pour qu'il atteigne l'autre objet. La normale de ce plan munie de la distance de la translation détermine un des plans définissant la différence de Minkowski. Ainsi, il suffit de parcourir toutes les facettes des deux objets en les translatant afin qu'elles atteignent l'autre objet.

Ensuite, une méthode est proposée pour calculer exactement la distance entre les deux solides. On ne peut pas se contenter de calculer la distance entre l'origine et le plan le plus proche. En effet le point de ce plan le plus proche de l'origine n'appartient pas forcément au polygone. Il faut donc au préalable découper le polygone afin de déterminer la facette s'appuyant sur ce plan. Une fois la facette obtenue, on peut calculer la distance de cette facette à l'origine, qui est la distance minimale de translation. Il faut cependant souligner que cette méthode marche bien en 2D mais est plus délicate en 3D.

c) Calcul incrémental de distances entre objets

Le calcul absolu de la distance séparant les objets étant difficile et coûteuse, [LC91] propose une méthode incrémentale à base de suivi de distance. Une description topologique des objets est nécessaire à l'algorithme. Chaque objet est un polyèdre convexe ou une réunion de polyèdres convexes. Il est décrit comme un ensemble d'éléments (sommet, arête ou facette) pour lesquels on précise :

- ses frontières (pour une arête ce sont les sommets, pour une facette ce sont les arêtes qui la délimitent) ;
- ses co-frontières (pour un sommet, les co-frontières sont les arêtes aboutissant à ce point, pour une arête, ce sont les deux facettes de part et d'autre de cette arête, pour une facette, ce sont l'ensemble des facettes de l'objets) ;
- sa région de Voronoï : elle est définie comme étant l'ensemble des points de l'espace qui sont les plus proches de cet élément par rapport à tous les autres. Autrement dit, cela signifie que la distance des points de la région de Voronoï d'un élément au polyèdre se calcule entre ce point et un point de cet élément.

Un *test d'applicabilité* a pour but de déterminer si un couple d'éléments est celui permettant de calculer la plus petite distance d'un objet à un autre. Si le test échoue, on cherche parmi les

frontières ou les co-frontières après un meilleur couple candidat. A chaque pas, on est certain de diminuer la distance entre les éléments testés. On a donc l'assurance, en fin d'algorithme, d'aboutir au couple d'éléments les plus proches des deux objets.

Le *test d'applicabilité* est simple. Soit E_1 et E_2 les éléments de l'objet 1 et 2 supposés les plus proches. Soit P_1 et P_2 les points les plus proches respectivement de E_1 et E_2. Si P_1 n'est pas dans la région de Voronoï de E_2 ou si P_2 n'est pas dans la région de Voronoï de E_1, alors le couple n'est pas le bon, et on se sert du ou de l'un des plans de la région de Voronoï qui posent problème (c'est à dire que le point testé n'est pas du bon côté de ces plans) pour déterminer quelles frontière ou co-frontière des éléments il faut inspecter.

Pratiquement, les régions de Voronoï ont généralement quatre à cinq côtés. Si la région en a plus, un découpage en polyèdres à moins de côtés est effectué afin de ne pas avoir trop de tests à effectuer. Lors de la simulation, sitôt qu'une distance entre les objets devient inférieure à un certain seuil, on conclut à la collision

Cette méthode peut se généraliser pour tester les intersections entre boîtes englobantes des objets. Ainsi, n'importe quelle primitive peut bénéficier de cette technique d'accélération.

4.2.4. Elimination par critères cinématiques

Comme leur nom l'indique, nous plaçons dans ce type d'élimination, toutes les méthodes impliquant la vitesse relative des objets entre eux.

a) Elimination par orientation de la vitesse relative

Ce type de critères se base sur une constatation évidente. Si deux objets d'éloignent l'un de l'autre, ils n'ont aucune chance de s'interpénétrer. [Van94] s'inspire du *Back-face culling* (voir [FDFH90], chapitre 15) utilisé en synthèse d'images et consistant en l'élimination des facettes qui ne sont pas orientées vers l'utilisateur. Ici, d'une manière tout à fait analogue, pour une facette, on élimine les facettes dont le vecteur vitesse relative n'est pas orienté vers la facette.

Soit un point P de l'espace dont la position est $\vec{p}$, on peut calculer la vitesse relative de P supposé lié rigidement au premier objet, puis au second. La différence de ces deux vitesses donne la vitesse relative des deux objets au point P (à cause de la rotation des objets, la vitesse relative dépend du point de l'espace considéré). On a, si P est rattaché à l'objet i :

$$\dot{\vec{p}} = \dot{\vec{r}}_i + \vec{\omega}_i \wedge \vec{c}_i$$

avec $\vec{r}$ position du barycentre de l'objet et $\vec{c}$ la position de P par rapport au barycentre. On a alors la vitesse relative des deux objets au point P :

$$\dot{\vec{p}}_{ij} = \dot{\vec{p}}_i - \dot{\vec{p}}_j$$

Soit :

$$\dot{\vec{p}}_{ij} = \dot{\vec{r}}_i - \dot{\vec{r}}_j - (\vec{\omega}_i \wedge \vec{r}_i) + (\vec{\omega}_j \wedge \vec{r}_j) + \vec{p} \wedge (\vec{\omega}_j - \vec{\omega}_i)$$

Soit une facette f de l'objet i, dont la normale est $\vec{N}_f$. Si P est un point de la facette, on note $\dot{\vec{p}}_{ij}$ la vitesse relative de P par rapport à l'objet j pour un observateur fixe sur i.

S'il existe un point P sur f tel que :

$$\dot{\vec{p}}_{ij} \cdot \vec{N}_f \geq 0$$

alors P peut entrer en collision avec l'objet j. Si par contre, pour tout P de f :

$$\dot{\vec{p}}_{ij} \cdot \vec{N}_f < 0$$

alors il n'y a aucune collision possible. Afin de limiter le nombre de tests (la propriété doit être valable pour tout P). On se contente de faire le calcul sur les sommets du triangle et on utilise la propriété suivante : la vitesse relative est une propriété linéaire, c'est à dire qu'on peut connaître

par combinaison linéaire la valeur de la vitesse en tous les points d'un segment ou d'un triangle si on connaît sa valeur aux sommets.

b) Elimination par estimation de trajectoire

Si l'on sait calculer les distances séparant les objets et qu'en plus, on arrive à borner la vitesse durant un pas de temps suffisamment grand, il est possible de prévoir quand peuvent avoir lieu des collisions. Ainsi [CK86] utilisent une évaluation de la distance minimale l_{min} entre deux corps et de leur vitesse maximale v_{max}, afin de trouver une borne inférieure du temps où il y aura collision. En effet, il ne pourra y avoir de collision qu'après $t_{min} = \frac{l_{min}}{v_{max}}$. Les évaluations n'ont pas besoin d'être précises : une majoration de la vitesse et une minoration de la distance facilement obtenues valent mieux que des valeurs précises difficiles à calculer.

c) Méthode par piles d'évènements

En exploitant complètement l'estimation des trajectoires, [LC92] utilisent une pile où ils stockent les collisions possibles. En effet, une fois que l'on a obtenu une borne inférieur du temps où risque d'avoir lieu la prochaine collision, on stocke la collision possible dans une pile triée. Ainsi à chaque instant de la simulation, on peut tester si une collision doit avoir lieu en inspectant le temps en haut de pile. On vérifie alors si la collision a lieu. Si non, on évalue à nouveau le temps de la collision et on stocke dans la pile. Si oui, on calcule la réponse dynamique au choc.

Plutôt que de se placer dans une simulation synchrone, [Dwo94][DZ93] quant à eux préfèrent se baser sur une horloge adaptative qui ne s'éveille que lorsqu'un évènement a lieu, c'est à dire, une collision. On maintient pour chaque objet le numéro du premier objet avec lequel il a le plus de chances d'entrer en collision.

Les auteurs se placent dans des environnements épars, c'est à dire où les collisions sont rares. On suppose de plus que les trajectoires sont rectilignes uniformes. Chaque objet est entouré d'un volume englobant, une sphère est le volume le plus intéressant pour les calculs. En effet, la détection de collision entre deux sphères, de rayon constant, en mouvement rectiligne uniforme est une équation du second degré, et correspond au moment où la distance entre les deux centres est égale à la somme des deux rayons. Si une collision à venir est détectée, elle est mise dans la queue du système qui contient, pour chaque élément, le numéro du premier élément avec lequel il peut entrer en collision, le tout trié par ordre croissant. Lorsqu'un évènement a lieu, on calcule les nouvelles trajectoires et on recherche les nouvelles collisions possibles des deux objets.

Dans la méthode, au moins un des deux objets considère la collision comme la plus proche pour lui. Cependant, il peut très bien arriver que l'un des objets (appelé *acteur*) considère qu'il y aura collision avec un autre élément (appelé *cible*), alors que celui-ci doit théoriquement entrer en collision avec un autre objet auparavant. L'objet acteur risque alors de ne pas trouver l'objet cible à l'endroit prévu car celui-ci aura été dévié avant. Cet évènement ne pose pas de problèmes particuliers lors de la simulation, mais crée des évènements obsolètes qui doivent être détectés : ce sont les prédictions qui ne peuvent plus se réaliser, suite au changement de direction d'un objet. Deux cas se présentent :

- Si l'acteur est absent, c'est à dire que l'objet pensait entrer en collision avec un autre objet, mais a été dévié de sa course par un autre : rien à faire dans ce cas, puisque l'objet a déjà calculé une nouvelle cible lors du choc ;
- Si la cible est absente, c'est à dire que l'objet devait entrer en collision avec un autre objet qui lui, a été dévié auparavant : il faut alors calculer une nouvelle cible pour l'acteur.

Afin de détecter rapidement les évènements obsolètes, on utilise un marqueur de temps. Si un des objets impliqués dans la collision a été modifié depuis la date de la prédiction, l'évènement est obsolète.

4.2.5. Conclusion

Les méthodes présentées sont diverses et reposent sur des stratégies variées. Il est difficile de les comparer, certaines cherchent à accélérer les calculs d'intersection 3D, d'autres utilisent des critères cinématiques (donc incluant le temps), certaines s'appliquent au niveau objets d'autres s'appliquent au niveau primitives de l'objet (faces, arêtes et nœuds pour un modèle polyédrique). Ces différences sont un atout car elles permettent de composer ces tests, sans redondance. Il est évident que dans le pipeline, ce sont d'abord les méthodes cherchant à éliminer des collisions entre objet qui doivent être employées suivies par les méthodes liées aux facettes ou surfaces. Les étapes préliminaires concernent donc les boîtes englobantes et les évaluations de distances entre objets. Les étapes suivantes sont les méthodes hybrides, entre le niveau objet et le niveau primitive, comme par exemple la comparaison des primitives de l'un des objets avec la boîte englobante de l'autre objet. Les étapes finales regroupent les techniques comparant les primitives, de voxels ou de comparaison de vitesse relative. Il faut tout de même prendre garde aux calculs nécessités par ces méthodes. Par exemple, sitôt qu'il faut modifier des boîtes englobantes, changer de repère ou détecter des points particuliers, le temps de calcul peut augmenter au point de rendre la méthode inefficace.

Par ailleurs, parmi les méthodes citées, peu ont été étudiées dans un contexte d'objets déformables. Certaines dépendent de la condition de rigidité et sont de ce fait inapplicables à notre simulation. Les objets déformables ont l'inconvénient, outre les équations plus complexes qu'ils engendrent, de ne pas être caractérisés par une boîte englobante fixe. Il faut la calculer à chaque pas de temps.

Dans la suite, nous considérons que nous avons un système quelconque, capable de fournir des couples d'éléments (points, facettes, arêtes) de deux objets entrant en collision à un temps déterminé et connu, que ces couples sont obtenus en temps réel avec des routines d'optimisation efficaces. Nous étudions maintenant comment exploiter ces caractéristiques pour la réponse des objets à l'interaction.

4.3. Réponse à l'interaction

Détecter et empêcher les interpénétrations ne suffisent pas pour atteindre une simulation réaliste. En effet, les objets entretiennent des interactions qui modifient leur comportement. Ainsi, les objets ne réagissent pas de la même manière suivant la violence de l'interaction. L'étude de cette réponse à la collision est l'objet de cette partie. Cette étape de résolution des collisions n'est pas toujours nécessaire. En effet, il existe des cas où la simple détection des boîtes englobantes des objets et la prise en compte de certaines propriétés physiques suffisent à déterminer une bonne approximation de l'effet d'un choc. C'est par exemple le cas du simulateur de champ de bataille de [ZOMP93]. Si on détecte que la boîte englobante du véhicule passe par la boîte englobante d'un obstacle, on en déduit une collision qui se traduit soit par la destruction et l'arrêt du véhicule si l'obstacle est solide, soit au contraire par une destruction totale ou partielle de l'obstacle.

4.3.1. Le point de vue du physicien

La collision est un phénomène complexe. La physique ne procure qu'une théorie partielle car les lois entrant en jeu sont mal connues. Parfois même, certaines collisions ne sont pas déterministes et relèvent de la théorie du chaos. On distingue tout de même deux grandes catégories d'interaction.

- Le choc : il se caractérise en physique par un échange quasi-instantané d'énergie. La vitesse des corps est alors discontinue. Les phénomènes physiques entrent en jeu sur une durée très courte : c'est pourquoi, à la notion de force (produit de la masse par l'accélération) se substitue la notion d'impulsion, homogène à une quantité de mouvement (produit de la masse par

la vitesse). L'impulsion est d'un certain point de vue une force infinie s'exerçant sur une durée nulle (un *dirac* de force).

- Le contact prolongé : il peut être statique ou se manifester sous la forme de glissement. Des forces de frictions (frottement sec) apparaissent dans les zones de contact. Le phénomène qui apparaît est microscopique. Il n'est cependant pas concevable de calculer ces phénomènes durant la simulation. On préfère employer des modèles macroscopiques et de ce fait statistiques, pour rendre compte de ces phénomènes. Malheureusement, ces modèles ne sont pas adaptés à certaines situations et peuvent parfois se comporter de façon inattendue et incohérente. Le modèle le plus souvent employé est celui des frottements de Coulomb, se reposant sur des lois empiriques. Ce modèle distingue trois cas :

  -Soit la vitesse de glissement $\vec{V}_g$ n'est pas nulle, et dans ce cas la force de frottement $\vec{F}_f$ est reliée à la force de non interpénétration $\vec{F}_n$, normale, par la relation :

$$\vec{F}_f = -\mu_c \|\vec{F}_n\| \frac{\vec{V}_g}{\|\vec{V}_g\|} \tag{4.8}$$

  -Soit la vitesse de glissement est nulle, c'est que les forces de frottement viennent annuler la composante tangentielle $\vec{F}_t$ des forces s'appliquant sur la particule, il y a adhérence. On a donc $\vec{F}_f = -\vec{F}_t$. Dans ce cas, la force de frottement doit vérifier l'inégalité suivante :

$$\|\vec{F}_f\| \leq \mu_s \|\vec{F}_n\| \tag{4.9}$$

  -Dans le cas où le coefficient de frottement cinétique μ_c est différent du coefficient statique μ_s, il faut tenir compte d'un état intermédiaire où :

$$\vec{F}_f = -\mu_s \|\vec{F}_n\| \frac{\vec{F}_t}{\|\vec{F}_t\|} \text{ avec } \|\vec{F}_t\| > \mu_s \|\vec{F}_n\|$$

Les interactions entre corps rigides sont les mieux maîtrisées et nous les exposons dans la partie suivante. Nous voyons qu'il est possible, en employant les lois physiques de conservation, de caractériser les phénomènes de collisions entre objets rigides. Ensuite nous nous intéressons au cas général, c'est à dire incorporant des corps déformables ou non. Enfin, nous présentons les problèmes particuliers qui apparaissant lors des contacts prolongés.

4.3.2. Corps rigides

La technique suivante a été adaptée à l'animation simultanément par [Hah88] et [MW88]. On suppose deux corps *A* et *B* entrant en interaction au point *P*. On définit un plan qui est soit le plan de la facette de l'un des objets entrant en collision avec le point d'un autre objet, soit le plan formé par les deux arêtes lors d'une collision arête/arête. On construit alors le repère $\left(\vec{i},\vec{j},\vec{k}\right)$ tel que le vecteur $\vec{k}$ soit la normale de la collision (c'est à dire la normale de la facette ou la normale aux deux arêtes suivant le type de collision.

Chaque solide est caractérisé par sa masse *m*, sa matrice d'inertie $\dot{I}$, sa vitesse linéaire $\vec{v}$, sa vitesse angulaire $\vec{\omega}$, le vecteur $\vec{\rho}$ du barycentre au point *P*. Dans les calculs qui suivent, seront indiquées par une étoile les quantités ayant été modifiées après le choc. On appelle $\vec{R}$ l'impulsion du choc. La conservation de la quantité de mouvement permet d'écrire, pour le corps *A* :

$$m_A \vec{v}_A^{\,*} = m_A \vec{v}_A + \vec{R} \tag{4.10}$$

et pour le corps B :

$$m_B \vec{v}_B{}^* = m_B \vec{v}_B - \vec{R} \tag{4.11}$$

La conservation du moment cinétique permet d'ajouter pour le corps A :

$$\overline{\overline{I}}_A \vec{\omega}_A{}^* = \overline{\overline{I}}_A \vec{\omega}_A + \vec{\rho}_A \wedge \vec{R} \tag{4.12}$$

Ainsi que pour le corps B :

$$\overline{\overline{I}}_B \vec{\omega}_B{}^* = \overline{\overline{I}}_B \vec{\omega}_B - \vec{\rho}_B \wedge \vec{R} \tag{4.13}$$

Les équations *(4.10)*, *(4.11)*, *(4.12)* et *(4.13)* mise sous forme vectorielle, nous donnent douze équations. Les inconnues du systèmes sont les coordonnées des nouvelles vitesses linéaires et angulaires des deux corps ainsi que les coordonnées de l'impulsion ce qui nous donne quinze inconnues. Il nous manque donc trois équations afin de caractériser complètement le système. Les trois équations suivantes dépendent des hypothèses que l'on choisit pour le choc. On peut par exemple, considérer qu'il n'y a pas de friction donc que l'impulsion est sur la normale à la facette entrant en collision. Cela se traduit par :

$$\vec{R} \cdot \vec{i} = 0 \tag{4.14}$$

$$\vec{R} \cdot \vec{j} = 0 \tag{4.15}$$

De plus, on suppose que les deux objets ne rebondissent pas sur l'un sur l'autre et qu'ils restent en contact, c'est à dire que le coefficient d'élasticité de la collision est nul. Ces conditions donnent :

$$\left(\vec{v}_B{}^* + \vec{\omega}_B{}^* \wedge \vec{\rho}_B - \vec{v}_A{}^* - \vec{\omega}_A{}^* \wedge \vec{\rho}_A \right) \cdot \vec{k} = 0 \tag{4.16}$$

On peut prendre d'autres hypothèses pour le choc, les calculs sont un peu plus complexes mais la résolution est possible. Ainsi, [Hah88] propose de ne pas annuler la composante normale de la vitesse relative après le choc, mais de considérer que l'impact est élastique. L'équation *(4.16)* s'écrit alors :

$$\frac{\left(\vec{v}_B{}^* + \vec{\omega}_B{}^* \wedge \vec{\rho}_B - \vec{v}_A{}^* - \vec{\omega}_A{}^* \wedge \vec{\rho}_A \right) \cdot \vec{k}}{(\vec{v}_B + \vec{\omega}_B \wedge \vec{\rho}_B - \vec{v}_A - \vec{\omega}_A \wedge \vec{\rho}_A) \cdot \vec{k}} = -\varepsilon \tag{4.17}$$

Le coefficient d'élasticité ε est choisi entre 0 et 1. De plus, l'auteur préfère considérer que le système permet les frictions et de ce fait, les équations *(4.14)* et *(4.15)* sont modifiées. On préfère supposer qu'il n'y a pas de glissement entre les corps d'où les vitesses tangentielles sont nulles :

$$\left(\vec{v}_B{}^* + \vec{\omega}_B{}^* \wedge \vec{\rho}_B - \vec{v}_A{}^* - \vec{\omega}_A{}^* \wedge \vec{\rho}_A \right) \cdot \vec{i} = 0 \tag{4.18}$$

$$\left(\vec{v}_B{}^* + \vec{\omega}_B{}^* \wedge \vec{\rho}_B - \vec{v}_A{}^* - \vec{\omega}_A{}^* \wedge \vec{\rho}_A \right) \cdot \vec{j} = 0 \tag{4.19}$$

Il faut alors vérifier que le système vérifie la condition de Coulomb de frottement statique, soit :

$$\left\| \vec{k} \wedge \left(\vec{R} \wedge \vec{k} \right) \right\| \leq \mu_s \left\| \vec{R} \cdot \vec{k} \right\|$$

Si ce n'est pas le cas, l'hypothèse de glissement nulle est invalidée, il faut poser les équations suivantes :

$$\vec{R} \cdot \vec{j} = 0$$

$$\vec{R} \cdot \vec{i} = \mu_c \left\| \vec{R} \cdot \vec{k} \right\|$$

Dans ce cas les vecteurs $\vec{i}$ et $\vec{j}$ ne sont plus quelconques dans le plan, mais représentent la direction des frottements et la direction perpendiculaire.

4.3.3. Corps déformables

Le calcul pour les corps rigides n'est pas du tout applicable au corps déformables. En effet, la résolution suppose que l'énergie due au choc (l'impulsion) est répartie entre les modifications

de vitesses des objets et les éventuels frottements de la collision qui se traduisent par l'émission de chaleur. Cette répartition d'énergie est valable pour les corps déformables mais n'est pas complète. En effet, une partie de cette énergie sert aussi à déformer le corps.

a) Méthodes basées sur les contraintes (ou méthodes analytiques)

Ce sont les méthodes les plus intuitives, elles sont détaillées dans [Bar92]. Elles consistent simplement à faire en sorte que la contrainte de non interpénétration soit immédiatement et exactement vérifiée. Supposons qu'un point entre en collision avec une surface. On fait le bilan des forces appliquées au point de la collision, et on annule la composante de la force qui est normale à la surface si cette composante est orientée dans la même direction. Cela revient à considérer que de la surface part une force de réaction empêchant l'interpénétration du point au travers de la surface. L'accélération de la particule est donc cantonnée au plan tangent à la surface tandis que l'on force la vitesse normale à 0. C'est la méthode couramment employée dans les systèmes physiques où l'on fait le bilan des forces. Le problème s'exprime en général sous la forme : minimiser l'énergie $f(a)$ avec a l'état du système en tenant compte de la contrainte $g(a) = 0$. La contrainte est donc immédiatement vérifiée, mais l'obtention d'une solution n'est possible exactement que pour des cas très simples et exige la plupart du temps une méthode d'optimisation (convergence vers le minimum, le long du chemin donné par la contrainte).

Une mise en pratique simple des contraintes est proposée par [Kam93]. Il considère le cas simple d'un point entrant en collision avec un plan. Une fois la collision détectée, on considère la vitesse du point avant le choc. Cette vitesse est décomposée en deux composantes normale $\vec{v}_N$ et tangentielle $\vec{v}_T$. La force après l'impact devient :

$$\vec{v}_{t+\Delta t} = \lambda \vec{v}_T - \mu \vec{v}_N$$

où λ est un coefficient de friction et μ est un coefficient d'élasticité du choc. Ce modèle, même s'il est éloigné de la réalité, est très simple à mettre en oeuvre et à contrôler lorsqu'on désire créer une animation donnée.

b) Méthodes à pénalités

Ces méthodes n'ont pas pour but d'empêcher les interpénétrations, mais de pénaliser le système lorsqu'une interpénétration est apparue. Une force est alors ajoutée au système qui tend à le ramener vers un état cohérent.

Une mise en œuvre simple de cette idée est facile à réaliser : sitôt que deux objets entrent en collision ou sont très proches l'un de l'autre, un ressort est ajouté créant une force tendant à éloigner les deux corps. Cette méthode a été employée par [MW88]. Elle présente deux inconvénients principaux. Tout d'abord, il est difficile de régler la constante de raideur du ressort. Quand on l'augmente, on ajoute une autre fréquence au système d'où des éventuels problèmes de convergence des équations. Si on la diminue, les chocs sont lents, peu énergétiques et l'interpénétration dure longtemps. Le second inconvénient est un mauvais traitement des contacts prolongés. Ces derniers ne sont pas stables et tremblent à cause des ressorts sur la zone de contact.

Les méthodes à pénalités repose sur un formalisme précis [PB88]. Résoudre le système dynamique lors d'une interpénétration revient à minimiser l'énergie du système en respectant une contrainte. Mathématiquement, on cherche à minimiser $f(a)$ avec a l'état du système sachant que $g(a) = 0$. On transforme ce problème contraint en un problème non contraint en minimisant la fonction $f(a) + kg(a)^2$. Le terme $kg(a)^2$ est le terme de pénalisation qui augmente l'énergie total du système tant que la contrainte n'est pas respectée. Cette méthode est très attractive car simple à implanter. Elle autorise la définition de plusieurs contraintes et offre la garantie de toujours obtenir une solution. Cette méthode trouve la meilleure solution approchant les contraintes au sens des moindres carrés et de ce fait ne garantit pas de respecter exactement toutes les contraintes. L'inconvénient de cette méthode est de rajouter des hautes fréquences dans le système et de ce fait d'exiger des pas de temps très fin lors de la collision pour obtenir des résultats cohérents.

c) Contraintes de réaction

Dans cette méthode, on suppose le solide modélisé sous la forme de points massiques. On suppose que l'on est dans un état où la contrainte n'est pas vérifiée. On fait alors le bilan des forces appliquées au point. A cette force, on supprime la composante tendant à s'éloigner encore plus de la contrainte. Pour éviter l'interpénétration, cela revient à retirer la composante de la force qui est normale et orientée en direction du plan.

On ajoute alors de façon arbitraire une force calculée pour vérifier la contrainte. La force ajoutée est appliquée à une variable de déviation obéissant à une équation différentielle du premier ordre, dont la constante de temps est fixée par l'utilisateur :

$$\vec{F}_{contrainte} = k\vec{D} + c\frac{d}{dt}\vec{D}$$

Notons qu'il faut tenir compte pour l'autre corps de la force ajoutée selon le principe de la réaction-contre réaction. L'inconvénient de la méthode réside dans le fait qu'une seule force de contrainte peut être ajoutée pour un point massique donné. Cependant, on est assuré de vérifier la contrainte exactement.

d) Multiplicateurs de Lagrange

On se place dans le cadre d'un problème d'optimisation, comme pour les méthodes à pénalité. A la fonction d'énergie $f(\vec{x})$ que l'on cherche à minimiser on ajoute le terme $\lambda \times g(\vec{x})$, où $g(\vec{x}) = 0$ est la contrainte et λ est appelé *multiplicateur de Lagrange*. A l'état solution, la force est nulle donc le gradient de l'énergie est nul, d'où :

$$\overrightarrow{grad}(f(\vec{x})) = -\lambda\overrightarrow{grad}(g(\vec{x}))$$

Les auteurs proposent une méthode adaptée de la descente de gradient afin de trouver les minima de la fonction d'énergie. Ainsi, on fait évoluer l'état du système vers le minimum en posant :

$$\dot{x}_i = -\frac{\partial}{\partial x_i}f(\vec{x}) - \lambda\frac{\partial}{\partial x_i}g(\vec{x})$$

$$\dot{\lambda} = g(\vec{x})$$

L'algorithme peut être étendu à des contraintes multiples, avec possibilité de contraintes par inégalité. En combinant la méthode des multiplieurs de Lagrange et la méthode à pénalité, on obtient la méthode des *contraintes de Lagrange augmentées* (ALC).

4.3.4. Modèles d'interactions dépendant du modèle de corps déformables

Les traitements que nous avons exposés auparavant ne dépendent pas ou peu du modèle de corps déformables sous-jacent. Il existe par contre des modèles de corps déformables étudiés pour leur facilité à traiter les interactions. Nous en présentons deux exemples.

a) Modèle de Gascuel

On a présenté au paragraphe *3.2.3* le modèle de [Gas89]. On suppose que l'on a détecté un nœud et une facette entrant en collision. Si la facette appartient à un objet rigide, la position du nœud est alors fixée au point d'intersection entre la trajectoire du nœud et la facette. Si les deux objets sont déformables, il s'agit de trouver une position intermédiaire. Soit k_0 la constante de raideur du ressort attaché au nœud et k_1, k_2 et k_3 les constantes de raideurs des trois ressorts dont les extrémités définissent le triangle de la facette en intersection. On pose :

$$k = \frac{k_0}{k_0 + (\alpha k_1 + \beta k_2 + \gamma k_3)}$$

Où les coefficients α, β et γ sont les coordonnées barycentriques du point de collision dans le triangle de la facette. Le nœud en collision est alors placé à la position :

$$M = kP + (1-k)I$$

où P est l'emplacement courant du nœud et I le point d'intersection sur la facette. On fait ensuite en sorte que le plan du triangle passe par M en résolvant une équation géométrique. La déformation induite est considérée comme manifestation de la force du choc. D'où cette force est ensuite appliquée à la composante rigide du corps.

b) Les interactions dans Cordis/Anima

Le système Cordis/Anima que l'on a déjà présenté, offre un point de vue intéressant pour le traitement des collisions [JL92]. Le système n'est constitué que de particules en interaction. On ne cherche pas ici à créer des facettes avec ces masses, on préfère les doter d'une sphère d'influence. Cette sphère est donnée par un potentiel qui va fournir une force permettant aux particules soit de se rapprocher, soit au contraire de se repousser si elles sont trop rapprochées. De telles forces sont courantes et bien connues en physique (par exemple la *force de Lennard-Jones*). Le minimum du potentiel donne en quelque sorte le diamètre de la sphère de la particule. Cette vision particulaire permet d'implanter des forces de collision et de contact de façon simple en tant de forme d'interaction particulière. Plusieurs mises en œuvre sont basées sur ces principes comme l'étude d'un véhicule se déplaçant sur des terrains de propriétés diverses [JLR91] ou l'étude de le formation de tas de sable [LHM95].

4.3.5. Les problèmes spécifiques des contacts

Les contacts sont un problème à part entière dans les phénomènes d'interaction. La loi de Coulomb que nous avons décrite s'applique bien lorsqu'un seul point est en contact avec l'autre objet. Cependant, la plupart du temps, ce sont des zones continues qui sont en contact et qui demandent, de ce fait, des traitements particuliers. De plus, la stabilité d'un système dépend complètement de la zone qui est en contact. Ainsi, certains points peuvent quitter la surface en contact, lors d'un basculement pas exemple. C'est ce genre de problèmes que nous abordons dans cette partie.

a) Approche géométrique

L'étude de [BV91] suppose que la simulation se fait entre objets rigides, polyédriques, dans un environnement sans friction. Dans cette méthode, les points de contact sont gérés à l'aide de liaisons temporaires. et un certain montant d'interpénétration est autorisé. On définit une *zone de contact* par les extrémités du polygone définissant la région où il y a contact. La gestion se fait à base d'évènements modifiant l'état des points. Ces derniers peuvent être :

- Actifs : ils définissent le polygone de contact ;
- Passifs : il sont en contact, mais ne sont pas utilisés pour la définition de la zone de contact ;
- Morts : ils ont alors une vitesse telle qu'il s'éloignent du contact.

Durant la simulation, les évènements possibles sont :

- un point actif a quitté son support ;
- un nouveau point de contact est détecté ;
- un point de contact essaie de pénétrer dans le support ;
- un point actif quitte le support.

Durant la simulation, on détecte les transitions d'état des points, ce qui permet à chaque instant de connaître la zone en contact.

b) Approche analytique

Calcul des forces de contact

[Bar89] simplifie la zone de contact entre deux objets polyèdriques en considérant qu'elle est décrite par un nombre fini de points qui sont les sommets du polygone de cette zone. On cherche les forces de réaction en chaque point. Les quatre conditions à vérifier sont :

- (1) Les forces de contact empêchent les corps de s'interpénétrer ;
- (2) Les forces de contact peuvent pousser, mais pas tirer ;
- (3) Les forces de contact n'apparaissent qu'en des points de contact lors d'un contact. La séparation entraîne l'annulation de la force ;
- (4) Les forces de contact sont continues dans le temps.

On suppose que la zone de contact est définie par n points. Chaque point est caractérisé par une normale $\vec{n}_i$, une intensité de force de contact f_i. On regroupe toutes valeurs de force en un vecteur appelé $\vec{f}$.

On considère la quantité :

$$\chi_i(t) = \vec{n}_i(t) \cdot \left(\vec{p}_i^{\,a}(t) - \vec{p}_i^{\,b}(t) \right)$$

Cette quantité mesure la distance relative entre les deux objets c'est à dire entre les deux points $a \in A$ et $b \in B$. En la dérivant une fois on obtient la vitesse relative, puis une seconde fois, on obtient l'accélération relative. On montre que cette accélération relative en chaque point de contact dépend linéairement de toutes les intensité des forces de contact. Si on veut respecter la condition (1), cela revient à dire que l'accélération relative est positive, on a donc, on condensant l'écriture sous forme matricielle :

$$\bar{\bar{A}}\vec{f} \geq \vec{b} \qquad (4.20)$$

où $\bar{\bar{A}}$ et $\vec{b}$ contiennent les coefficients de linéarité entre les accélérations relatives de tous les points de contact et les forces de contact. Notons que la matrice $\bar{\bar{A}}$ est symétrique, positive et semi-définie. La condition (2) entraîne que la force soit toujours dirigée dans le sens de la normale, d'où :

$$\vec{f} \geq \vec{O} \qquad (4.21)$$

Trouver $\vec{f}$ vérifiant *(4.20)* et *(4.21)* est un problème d'optimisation linéaire. La condition (3) implique deux possibilités. En effet, si l'accélération est strictement positive, c'est que le contact prend fin et la force doit être nulle. Par contre, si le contact se prolonge, l'accélération est nulle et la force de contact empêche l'interpénétration. Ces circonstances se résument simplement en affirmant que, en un point de contact, soit l'accélération, soit la force de contact est nulle, d'où le produit entre les deux est toujours nul. Afin d'obtenir une expression matricielle, on choisit non pas de considérer ce terme nul pour chaque point de contact, mais on en fait la somme afin d'obtenir un produit scalaire qui est nul car somme de termes nuls. Soit :

$$\vec{f}^T \left(\bar{\bar{A}}\vec{f} - \vec{b} \right) = 0$$

Cette dernière équation transforme le système en un problème quadratique, *NP dur* en général. Une solution heuristique est alors employée qui tente de deviner si les points de contact quittent la surface ou non (on fixe leur accélération à 0). Une fois le comportement des points déterminés, le problème redevient linéaire. Par contre, le choix des points quittant la surface se fait par tâtonnement. La résolution n'étant pas satisfaisante, [Bar94] optimise l'algorithme précédent. Pour cela, la condition (3) n'est plus regroupée sous la forme d'un produit scalaire, mais on

garde les égalités pour chaque point de contact (le produit de l'accélération au point de contact par la force de non interpénétration est nul). Le problème s'écrit alors :

$$\forall i,\ a_i \geq 0,\ f_i \geq 0 \text{ et } f_i a_i = 0$$

L'algorithme fonctionne alors par étape, c'est à dire que l'on suppose une solution pour k-1 points de contact et on calcule une solution pour un point de contact supplémentaire, en aboutissant donc à une solution pour k points de contact, et ainsi de suite jusqu'à trouver la solution pour tous les points. Notons que cet algorithme fonctionne toujours dans le cas de contacts sans forces de frottements. La terminaison de l'algorithme n'est pas prouvée dans le cas des contacts avec frictions statiques. Selon l'auteur, il semble que l'algorithme est praticable. Enfin, le cas des contact glissants avec frictions dynamiques est traité partiellement (dans ce cas, le fait de devoir ajouter des impulsions est gênant).

[BW92] met en œuvre l'algorithme, dans le cas de corps déformables avec le modèle de [WW90]. Lorsqu'un seul point de contact est détecté, on calcule facilement l'impulsion en cherchant le coefficient λ tel que l'impulsion soit $\lambda\vec{n}$ où $\vec{n}$ est la normale au point de contact. On démontre que, en se servant du même formalisme qu'au paragraphe *3.2.2* :

$$\lambda = \frac{-(\vec{n} \cdot \dot{R}(t)\vec{p}(x))}{\vec{p}(x)^T M^{-1} \vec{p}(x)}$$

Par contre, sitôt que plusieurs points sont en contact, le problème est plus délicat. La méthode exposée précédemment pour les corps rigides peut toujours s'appliquer [Bar90] mais elle suppose un nombre fini de point de contact, ou au plus, un polygone dont on considère les sommets. Tout d'abord, il faut éviter les déformations quadratiques ou plus, car elles nécessitent d'échantillonner la structure : une approximation de la surface en triangle est proposée. Au contraire, les modes linéaires sont plus simples et l'échantillonnage n'est pas nécessaire. Mis à part ce problème de forme de la zone de contact, le traitement est similaire pour les corps déformables. On profite en outre que l'accélération est une fonction linéaire de $\ddot{R}(t_0)$.

Traitement des frictions

La prise en compte des frictions est un problème délicat. [Bar91] montre que la résolution n'est en général pas praticable car la décision de la consistance des forces de frictions est *NP-complète*, et le calcul des forces de contact est *NP-dur*. De plus, si on se contente de résoudre les systèmes avec frictions dynamiques avec des forces de contact, on se retrouve devant des cas d'inconsistance (non existence de forces de contact permettant la non interpénétration) ou des cas d'indétermination (plusieurs solutions possibles, avec impossibilité de les discriminer). L'auteur préconise alors l'utilisation d'un modèle plus proche de la réalité, c'est à dire que l'on considère de très faibles interpénétrations entre objets. Cette méthode repose donc sur le principe des pénalités. A l'aide d'impulsions, on règle les problèmes d'inconsistance et d'indétermination. L'expression du problème est très proche du traitement des forces de non interpénétration détaillé dans le paragraphe précédent.

La méthode décrite fonctionne bien, mais n'obéit plus au principe suivant : on utilise des impulsions, seulement si on ne peut pas faire autrement. Dans la méthode, des forces d'impulsions sont facilement utilisées même si on peut se contenter de forces de contact.

c) Cas des contacts entre plusieurs objets

Ce problème correspond aux cas où plusieurs objets sont en contact simultanément. Ce cas apparaît par exemple dans les modèles de corps articulés. L'approche de [Hah88] est plus générale et on choisit donc de ne présenter que cette solution. On considère un graphe des contact : chaque nœud est un objet et les liaisons indiquent les objets en contact. Un nœud particulier indique les objets inamovibles. Lors du calcul de la réponse aux collisions, on traite d'abord les objets en contact avec les objets inamovibles. L'algorithme revient donc à faire un parcourt du graphe en lar-

geur d'abord en partant du nœud des objets inamovibles. Il faut prendre alors garde aux cycles du graphe et faire des tests tout le long du cycle.

4.3.6. conclusion

Pour traiter les interactions, deux choix se présentent. Soit on choisit un point de vue pragmatique. La simulation se comporte relativement bien dans une grande partie des cas, mais peu parfois exhiber des comportements étranges, à la limite du modèle théorique. Souvent, il va falloir autoriser une certaine interpénétration pour pouvoir traiter les contacts, à moins qu'il ne soit possible, vu les hypothèses de la simulation, de vérifier exactement et analytiquement les contraintes. Les contacts prolongés sont calculés pour chaque point séparément, l'effet de la zone de contact est négligé.

Soit on choisit une approche précise, analytique, permettant de reproduire des cas plus généraux, mais extrêmement coûteuse. Dans un contexte de simulation temps réel, ce choix ne semble pas envisageable.

Enfin, la plupart des traitements autorise une certaine interpénétration, que ce soit pour calculer la force de non interpénétration ou les frictions. Pour les gestes chirurgicaux, la précision du contact est très importante. Il ne faut pas que le médecin pense, lors du retour visuel, avoir découpé l'organe qu'il opère. Il faut donc, dans la mesure du possible, faire usage de méthodes de vérification exacte de contraintes.

4.4. Les découpes et déchirements

La modélisation des corps rigides ou mous, les interactions entre de tels corps sont des sujets clefs en animation et simulation ce qui explique la variété des modèles que nous avons présentée. Les découpes et les déchirements, nécessaires à la modélisation de la plupart des interventions chirurgicales, apparaissent comme des compléments spécifiques de certaines animations. Peu de choses ont été proposées de nos jours, et c'est sous un angle plus général de modélisation avec modification dynamique des modèles qu'il faut aborder ce problème. Ce sont des domaines différents, mais ayant des points communs avec les déchirures et les découpes qui ont été étudiées. Ainsi, les phénomènes de fractures sont intéressants pour la création de certaines animations et des propositions ont été faites. Par ailleurs, le phénomène de fusion a également été modélisé et nous pensons que certaines idées peuvent être appliquées à la modélisation des déchirements.

4.4.1. Modélisation de fractures

L'idée de base pour la simulation des fractures est de supprimer des interactions et ainsi, rendre des parties indépendantes les unes des autres. Par exemple, [TF88] remarquent que leur modèle est compatible avec les fracture en annulant des coefficients intervenant dans l'énergie de déformation du corps.

L'approche de [NTBG91] est complète sur ce problème. A la base, on trouve un modèle masse-ressort. Chaque ressort, en plus de sa longueur au repos, son élasticité et son amortissement, est également caractérisé par un seuil, défini comme le taux de dilatation maximum de sa longueur par rapport à sa longueur au repos. Lorsque la longueur du ressort dépasse ce seuil, la liaison cède. Le taux de variation de longueur permet de déterminer les endroits plus faibles qui se fractureront facilement. Un booléen a pour rôle d'indiquer si une liaison a été brisée ou non. La géométrie du modèle est assurée à la fois aux niveaux macroscopique et microscopique. La structure est divisée en cubes : les arêtes ainsi que les diagonales du cube sont occupées par des ressorts. Ajouter des ressorts aux diagonales des côtés n'est pas utile selon les auteurs. Afin de ne pas fournir de formes trop anguleuses, on permet aux cubes d'être légèrement déformés.

La gestion des fractures nécessite la résolution de deux difficultés : la détection des morceaux qui se séparent et la conservation de la cohérence du volume.

D'une part, lorsque l'objet se fracture en plusieurs morceaux, il faut distinguer ces parties afin de les traiter comme des objets différents dans la suite de la simulation. Pour cela, on utilise le graphe du modèle à ressort. On parcourt le graphe en profondeur d'abord, ce qui nous permet de trouver tous les nœuds appartenant à la partie parcourue. Les autres nœuds font partie d'un ou de plusieurs autres morceaux. Une fois tous les morceaux connus, on localise par une subdivision spatiale, les noeuds de composantes différentes, tels que la distance qui les sépare est inférieure à un seuil. On repousse alors les noeuds les uns des autres (par une méthode de pénalité).

D'autre part, la simple suppression d'une liaison ne suffit pas à provoquer des fractures. Si on supprime une liaison de la diagonale d'un ressort, il reste les liaisons sur les autres diagonales et les arêtes du cube. En supprimant le ressort, on a uniquement annulé une résistance au cisaillement. Il est nécessaire d'introduire la notion de cellules pour pouvoir gérer les effets des fractures. Lorsqu'on supprime une liaison qui a cédé, la cellule permet de déterminer les autres liaisons à supprimer afin que la fracture ait effectivement lieu.

Pour assurer la convergence et une bonne cohérence des forces, les forces de déformations doivent être du même ordre que les celles de collision. L'énergie cinétique ne doit pas trop varier afin d'éviter une totale décomposition du modèle.

4.4.2. Modélisation de fusion

[TPF89] modélise la fusion en ajoutant aux équations du modèle déformable, les équations de propagation de la chaleur. Les équations se simplifient lorsque l'on se place dans le repère des axes principaux de propagation de la chaleur. Dans un modèle discret, la raideur de chaque lien est fonction de la température. Au delà d'un certain seuil de température, la raideur de la liaison décroît jusqu'à devenir nulle. La liaison se coupe alors. Plus la température augmente, plus on s'approche d'un état où il n'y a plus de liaisons du tout. Ce phénomène est appelé la *thermo-élasticité*. Les lois de Duhamel-Neumann sont utilisées : en deçà d'une certaine température, la liaison est purement élastique. Au delà, la raideur devient fonction de la moyenne des températures aux extrémités de la liaison et enfin, la raideur s'annule. Sitôt que la particule est libre, elle obéit aux lois d'interaction de *Lennard-Jones* (on aboutit à un modèle discret macro-moléculaire de fluide).

4.4.3. Bilan

Le concept des découpes n'est étudié que depuis peu de temps, en relation avec les applications médicales. Le concept de déchirement est lui à rapprocher des phénomènes de fractures. On peut alors se baser sur des critères de distance, où l'on se fixe un seuil maximum d'extension, ou sur des critères de forces où la liaison se coupe dès que les forces exercées sur un point sont trop élevées. L'idéal dans le cadre de nos travaux est de considérer les déchirements et les découpes sous le même formalisme, mais il faut encore démontrer si cela est possible.

On a enfin mis en évidence que dès qu'il y a modification de la structure, quelqu'en soit la cause (découpe, déchirement, fracture, fusion...), il faut que les modifications de géométrie soient faites rapidement et simplement dans la structure : annuler les paramètres d'une liaison par exemple. [Tur96] a proposé plusieurs solutions pour les découpes dans le cadre des éléments finis. Enfin, il faut prendre garde à ce que la structure ne se désagrège pas dès qu'une liaison a été rompue.

Conclusion

Nous avons dressé un panorama des méthodes proposées dans la littérature pour les trois points clefs que sont la détection des collisions, l'accélération de cette détection, le calcul de la réponse. Nous nous sommes également intéressés aux cas extrêmes, les découpes et déchirements. La détection est coûteuse : les calculs reposent soit sur des intersections en 3D, des intersections de trajectoire en 3D, 4D ou par calcul analytique. Dans un cadre de simulation, on désire éviter les algorithmes à itérations, ce qui nous fait préférer les méthodes analytiques sous réserve de pou-

voir les simplifier. Les diverses implantations proposées montrent que la phase de détection doit obligatoirement être précédée d'une phase d'accélération afin de réduire le nombre de calculs exacts de collisions. Les méthodes d'accélération sont diverses, leur choix dépend des conditions de la simulation. C'est pourquoi les comparaisons entre ces méthodes sont rares : il faudrait pour cela définir des classes d'environnements (dense ou pas, statique ou dynamique...) afin de déterminer quelles sont les techniques les plus appropriées. Une fois la détection terminée, il faut déterminer la réaction à la collision c'est à dire différencier les chocs et les contacts. La plupart des méthodes repose sur le concept de contraintes décrites par un potentiel et autorise généralement un certain degré d'interpénétration ce qui n'est pas souhaitable dans notre contexte. Enfin, il faut ajouter des critères sur la structure permettant de détecter quand a lieu une découpe ou un déchirement.

On remarque donc que toute la chaîne de traitement est d'une complexité nettement supérieure à la modélisation des corps mous et c'est généralement cette étape qui nécessite le plus de temps de calcul. De plus, un bon traitement des collisions devrait travailler à un pas de temps plus petit, que ce soit pour la détection ou le calcul de la réponse. La collision est une non linéarité qui augmente considérablement la fréquence d'un système et peut, si les conditions de résolution des équations sont trop strictes, faire diverger le système.

Partie 3

Notre proposition

Introduction

Nous avons fait dans la partie précédente une analyse des solutions actuelles applicables dans notre contexte. On a vu que la conception de notre modèle dynamique interactif temps-réel nécessite la modélisation du corps déformable et la prise en compte des interactions.

Dans cette partie, nous nous appuyons sur l'étude précédente pour concevoir un modèle répondant de façon optimale à notre objectif. Cette présentation s'articule en deux chapitres. Dans le premier, nous exposons les bases théoriques de notre modèle suivies, dans le second, de sa mise en œuvre. Cette mise en œuvre a été validée par la simulation de diverses situations physiques. Elle a également permis de vérifier que notre modèle permet des simulations en temps-réel. Les perspectives d'utilisation sont ensuite exposées ainsi que les remèdes qu'il convient encore de trouver afin d'atteindre un réalisme correct.

5 Proposition de modèle déformable avec interactions

Introduction

Nous exposons dans ce chapitre notre proposition de corps déformables pour la simulation d'interventions chirurgicales. Ce travail est la première phase d'une étude plus générale sur les corps déformables dont le but, à terme, est de déterminer dans quelles situations et pour quels types d'organes, il faut appliquer tel ou tel modèle de corps mous. Les travaux présentés ici n'ont donc pas comme objectif de trouver le modèle idéal[1]. Nous tâchons au contraire d'étudier de façon exhaustive un modèle particulier, d'en extraire les avantages et les limites et d'en déduire les domaines d'applications possibles.

Ce chapitre présentent successivement nos solutions aux trois problèmes que sont la prise en compte des lois de la mécanique, le traitement des interactions (constitué de la détection, l'accélération et enfin le calcul de la réponse) et enfin la mise en œuvre des découpes et déchirements. Notre corps déformable est basé sur les maillages masses/ressorts, que l'on munit d'une composante non déformable décrivant les mouvements rigides du corps. Les interactions sont détectées par un calcul analytique entre les trajectoires des corps. La réponse à la collision est traitée analytiquement elle-aussi, mais en ne respectant pas totalement les lois de la physique. Quelques solutions pour les découpes sont avancées.

5.1. Hypothèses préliminaires

5.1.1. Schéma de la simulation

Le schéma global de simulation, décrit sur la *figure 5.1*, est une version simplifiée du système de [LJFC91] : des capteurs mesurent les déplacements de l'outil, transmettent ces déplacements au modèle de l'organe qui se déforme en conséquence. On utilise le même principe de [GTT89] déjà détaillé au paragraphe *3.2.1*, section *a*. Deux sortes de noeuds apparaissent alors sur le corps déformable. Certains sont dans la zone de contact et pour ceux-là, la position est donnée par l'outil et la force exercée en ces points est inconnue. Elle est calculée en évaluant la force de déformation en ces noeuds et le résultat est envoyé au système à retour d'efforts. La position des autres noeuds est inconnue mais peut être calculée en sommant les forces s'exerçant sur eux et en intégrant la résultat. Le modèle apparaît comme une boîte noire avec en entrée des valeurs de positions et en sortie d'une part la déformation de l'organe pour l'affichage et d'autre part les valeurs des forces de déformation au points de contact, pour le retour d'effort.

1. Ce modèle est utopique, comme nous le montrerons dans la conclusion.

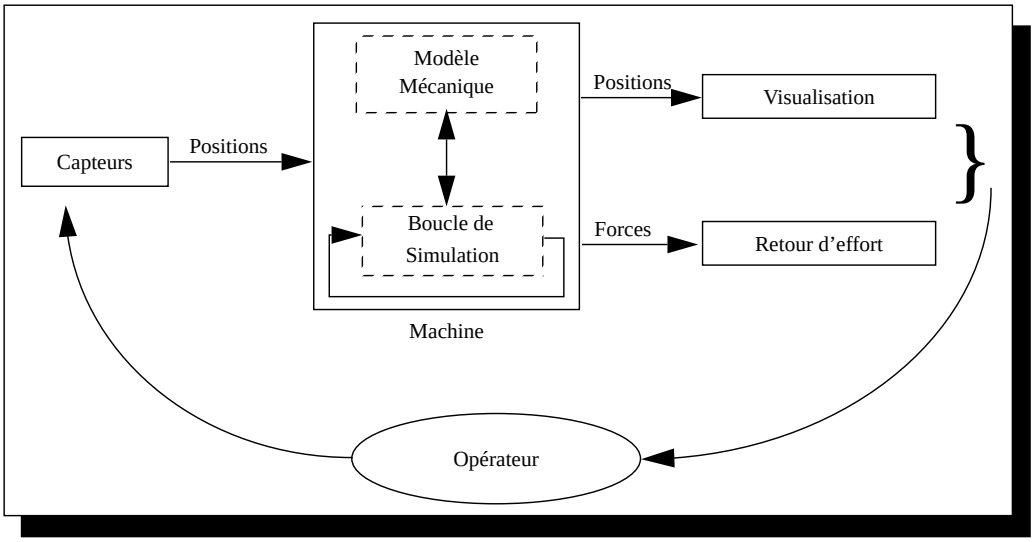

Figure 5.1. *Schéma global de la simulation*

5.1.2. Hypothèses

On cherche avant tout à simuler le comportement de l'organe lors d'une intervention sur celui-ci. On se place de ce fait dans le cadre d'une interaction simple entre un outil rigide et un organe déformable. Ce type d'interaction est donc celui que l'on étudie précisément. Les interactions entre corps déformables correspondent aux contacts entre les organes. Ces interactions peuvent être traitées de façon plus approximative. Pour les déformations, on suppose que les organes sont élastiques.

Enfin, on fait le choix d'employer un modèle polyédrique à faces triangulaires pour la visualisation. En effet, on désire utiliser des cartes 3D accélérées. Ces cartes affichent directement des primitives planes. L'usage de formes gauches ou à grand nombre de côtés surcharge le processeur qui doit découper ces surfaces en triangles.

5.2. Modèle à ressorts avec composante rigide

5.2.1. Pourquoi un modèle masses/ressorts ?

Parmi tous les modèles exposés au *chapitre 3*, les deux éléments les plus adaptés aux simulateurs médicaux sont les modèles éléments finis et les modèles discrets. Ces deux modèles reposent complètement sur la physique et sont capables de simuler une large gamme de corps mous pour des déformations globales et locales. Si on compare la complexité de ces deux méthodes, on constate que la FEM revient, dans le plus simple des cas (éléments linéaires en élasticité linéaire et petits déplacements), à la résolution d'un système linéaire, soit une complexité minimum en $o(n^2)$ où n est le nombre de nœuds. Au contraire, les modèles masses/ressorts ou plus généralement les système de particules interconnectées semblent offrir un meilleur compromis. Dans le cas d'un graphe complètement connecté (chaque masse est reliée à toutes les autres), un modèle masses/ressorts géré par des équations dynamiques et intégré par des méthodes explicites est en $o(n^2)$ (pour n masses, le calcul des forces en chaque masse est de complexité $o(n)$, et on applique l'intégration pour chaque masse). Cette complexité peut être réduite en diminuant le nombre de liaisons. En résumé, la complexité maximale d'un maillage de ressorts (complètement connecté) est la même que la complexité minimum d'un maillage éléments finis (éléments linéaires en élasticité linéaire), tout en étant compatible avec les grands déplacements et en restituant les régimes transitoires. Notre étude porte donc sur les modèles discrets.

5.2.2. Construction du modèle

a) Composante surfacique

On a choisi de construire un modèle à base d'éléments discrets. Pour réduire la complexité du modèle, les deux méthodes suivantes sont employées : intégrer explicitement les équations et réduire le nombre de liaisons.

La première méthode impose l'utilisation de méthodes d'intégrations peu stables numériquement. Notre choix se justifie cependant par la considération suivante : dans un contexte médical, les gestes sont très mesurés, posés, peu violents et donc relativement compatibles avec les limites en fréquences imposées par Shannon.

La seconde méthode cherche à minimiser le nombre de ressorts du modèle. En effet, les équations pilotant le comportement de ressorts d'axe non fixe sont non linéaires et exigent le calcul d'une racine carrée. Le coût du traitement d'un trop grand nombre de ressorts est rédhibitoire. Une idée intéressante est de se contenter d'une modélisation uniquement surfacique de l'objet par un maillage de nœuds massiques reliés entre premiers voisins par des ressorts amortis. Ce maillage a l'avantage de ne traiter que les points de la structure ayant une utilité pour le contact et la visualisation. Un tel modèle est très simple car le nombre de ressorts est linéaire en fonction du nombre de points, si on considère un nombre maximum de premiers voisins par point. Il manque cependant de consistance volumique : le simple fait de poser ce maillage sur un plan et de soumettre les points à la gravité fait s'écrouler l'objet sur lui-même. Un maillage surfacique n'est donc pas suffisant. Il faudrait ajouter des ressorts entre seconds voisins, et/ou des ressorts angulaires et de torsion, ce qui n'est pas compatible avec notre objectif d'utiliser un petit nombre de liaisons. Par ailleurs, les équations régissant les ressorts angulaires sont très coûteuses car elles nécessitent le calcul d'un angle.

b) Composante solide

Une composante surfacique offre la complexité la plus réduite, mais ne suffit pas. Comme on l'a exposé dans le paragraphe précédent, il faut donner une consistance volumique au maillage. Une manière très pratique de traiter les modèles déformables soumis à des déplacements rigides est de considérer que le corps en question est formé d'un noyau rigide, responsable des mouvements de rotation et de translation de l'objet, entouré d'une composante déformable. Des modèles basés sur ce procédé ont été exposés dans le *chapitre 3*. Le modèle de [Gas89] se révèle malheureusement mal adapté à notre cas, car il est incompatible avec les découpes. Le calcul des déformations se fait, en effet, par combinaison linéaire des conservations de volumes et de surfaces, le maillage ne peut donc pas être modifié car il est le garant de la cohérence des déformations. De plus, la présence d'une composante rigide à l'intérieur de la structure ne correspond pas aux propriétés mécaniques des organes que l'on souhaite simuler.

Notre approche utilise une composante rigide que l'on qualifie de *virtuelle*. On signifie par ce terme qu'elle n'est présente que pour les calculs de la dynamique, elle n'a donc pas de réalité physique. On choisit de relier cette composante rigide à un maillage surfacique de ressorts amortis (qui forment la partie visible de l'objet, c'est à dire celle utilisée lors du rendu), par l'intermédiaire de ressorts de longueur au repos nulle. Chaque nœud du maillage surfacique est relié à un homologue sur la composante rigide par l'intermédiaire d'un seul ressort. Cela permet de garder un nombre de ressorts proportionnel au nombre de nœuds dans le maillage.

5.2.3. Comportement du modèle

L'idée de base pour le comportement de notre modèle est simple : sans interaction, les composantes rigides et déformables sont confondues; par contre elles se dissocient lors d'un contact. La composante rigide indique donc la position qu'auraient les noeuds du maillage s'il n'y avait pas d'interaction, ce qui permet à la structure de retrouver sa forme originale lorsque l'interaction cesse. La composante rigide apparaît ainsi comme un simple outil mathématique car elle ne peut

pas entrer en interaction avec les autres objets. Démontrons-le par l'absurde : si la composante rigide entrait en interaction avec l'environnement, l'objet aurait au repos un comportement parfaitement solide puisqu'il n'y aurait aucune déformation de l'objet déformable.

La composante rigide ne subit donc que les forces volumiques comme la gravitation. Lors d'une interaction, d'un choc ou d'un contact, il y a séparation des deux composantes. Par contre, dès que l'interaction cesse, les deux composantes se rejoignent à nouveau. La *figure 5.2*illustre l'état au repos du modèle. La composante rigide n'entre pas en collision avec le plan, donc continue sa chute. Cependant, la composante surfacique étant en collision avec le plan, elle est retenue par ce dernier. La composante rigide est donc retenue par la composante déformable. Cette composante est alors déformée par son propre poids et le poids de la composante rigide. La composante rigide quant à elle garantit que la composante déformable ne s'affaisse pas sous son poids.

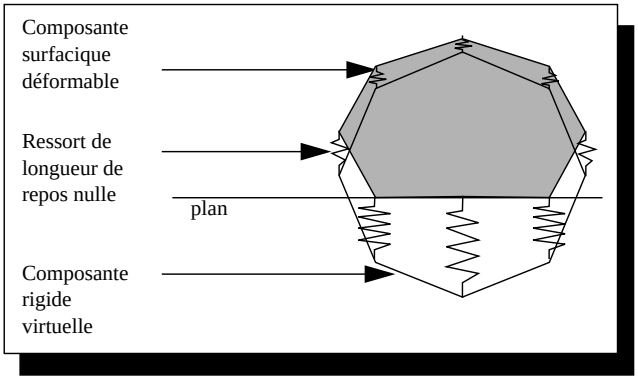

Figure 5.2. *Séparation des composantes solides et déformables lors d'un contact*

Notre modèle est proche de celui de [TW88]. Ce dernier décompose les équations du mouvement en une partie concernant le déplacement en rotation et translation de l'ensemble du système et une autre partie qui exprime les déformations du corps par rapport à une composante non déformée appelée composante de référence. Cette décomposition résulte d'un mécanisme mathématique consistant à exprimer la position des points du solide par rapport à sa position originale sur le corps non déformé. Au contraire, notre formulation repose sur une idée intuitive : la composante rigide indique la position qu'occuperait le corps s'il n'était pas en interaction. Les comportements des deux modèles sont fondamentalement différents. Dans une configuration similaire à la *figure 5.2*, par exemple, la composante non déformée et la composante déformable du modèle de [TW88] auraient leurs centres de masse confondus, la composante non déformable ne dépasserait pas d'une telle distance le plan support, mais resterait autour de la composante déformable.

5.2.4. Intégration des équations

Pour les nœuds du maillage, on fait le bilan des forces extérieures $\vec{F}_{ext}$, c'est à dire la gravité, les forces de viscosité dans l'air, des forces de contact ou collision $\vec{F}_c$ et des forces internes élastiques $\vec{F}_{int}$ puis on calcule l'accélération $\vec{a}_i$ de l'objet, puis par intégration la vitesse $\vec{v}_i$ et la position $\vec{x}_i$ de la particule i :

$$\begin{aligned} \vec{a}_i(t) &= \frac{\sum\left(\vec{F}_{int}(t) + \vec{F}_{ext}(t) + \vec{F}_c(t)\right)}{m_i} \\ \vec{v}_i(t) &= \vec{v}_i(t-\Delta t) + \Delta t \times \vec{a}_i(t) \\ \vec{x}_i(t) &= \vec{x}_i(t-\Delta t) + \Delta t \times \vec{v}_i(t) \end{aligned} \tag{5.1}$$

Avec $\Delta t = 1ms$ le pas d'échantillonnage de notre implantation. Pour la composante rigide, on fait le bilan des forces appliquées qui sont les forces extérieures (gravitation, viscosité de l'air) et les forces de liaisons à la composante déformable. L'accélération et la vitesse du centre d'inertie

est calculée par la même méthode que ci-dessus (la matrice d'inertie inverse peut être calculée une fois pour toutes). Pour l'orientation du solide, on calcule les couples de forces $\vec{T}$, on en déduit le moment, puis la vitesse de rotation instantanée $\vec{\omega}$ par intégration pour aboutir à un axe $\vec{u}$ et un angle de rotation θ instantané. Le tout est ensuite cumulé dans un quaternion représentant l'orientation du solide :

$$\begin{aligned}
\frac{d}{dt}\left(I\vec{\omega}\right)(t) &= \sum\left(\vec{T}_{int}(t) + \vec{T}_{ext}(t)\right) \\
\vec{\omega}(t) &= \vec{\omega}(t-\Delta t) + \Delta t \times \frac{d}{dt}\vec{\omega}(t) \\
\theta(t) &= \theta(t-\Delta t) + \Delta t \times \omega(t) \\
\vec{u} &= \frac{\vec{\omega}}{\omega}
\end{aligned} \tag{5.2}$$

5.2.5. Analyse du modèle

a) Avantages

- Le nombre de ressorts est limité, linéaire en fonction du nombre de noeuds dans le maillage ;
- La composante rigide assure les déplacements en rotation et translation de l'objet, d'où un bon comportement rigide de l'objet ;
- Les ressorts surfaciques permettent de contrôler les déformations en surface de l'objet : on peut définir simplement les forces de résistance à la découpe par exemple ;
- Les ressorts reliant la composante rigide à la composante surfacique déterminent plutôt la résistance à l'enfoncement. En faisant varier la constante de raideur de ces ressorts on est capable de simuler des objets élastiques simplement en surface et sans cohérence volumique (un sachet en plastique, par exemple, ou une corde élastique pour le cas à une dimension) ou alors au contraire des objets presque rigides ;
- La composante rigide assure une seule position d'équilibre des noeuds quelle que soit la topologie du maillage. De plus, le maillage peut être complexifié par l'ajout de ressorts angulaires ou de liaisons avec les seconds voisins... De même, si la topologie du maillage est modifiée durant la simulation, par une découpe, par exemple, la composante rigide assure une certaine cohérence volumique de l'objet. En quelque sorte, la composante rigide assure la robustesse du maillage ;
- Comme pour [TF88], par modification de la composante rigide, on peut simuler des phénomènes de plasticité ;
- Le maillage surfacique fournit des polygones pouvant servir directement à l'affichage de la structure dans les machines accélératrices actuelles. Ces polygones définissent le contour de l'objet pour le traitement des collisions ;
- Les masses du réseau surfacique et la masse de la composante rigide et sa matrice d'inertie sont indépendantes. Cela permet de choisir la partie de la masse de l'objet que l'on attribue à la surface et celle que l'on attribue à la partie volumique. Cette liberté est également valable pour les valeurs de la viscosité du milieu ambiant attribuées aux nœuds et à la composante rigide.

b) Limites du modèle

- Le modèle restitue les régimes transitoires et de ce fait, la synchronicité est critique dans le modèle. Comme cela a déjà été exposé, si la synchronicité n'est pas obtenue, la vitesse de tout mouvement de l'utilisateur est démultipliée ce qui rend instable le modèle ;

- Afin d'avoir un modèle très simple, ce dernier ne gère que des déformations locales. En effet, chaque déplacement est calculé indépendamment du déplacement des autres points. Le comportement global est fixé par la composante rigide. Les transformations globales ne sont donc pas calculées. Le modèle ne peut pas facilement se courber par exemple. Une courbure ne pourra être obtenue qu'en composant des déformations locales ou en courbant la composante rigide elle même ;
- Le contrôle des déformations est très limité. Des contraintes telles que la conservation de volume ou de la surface sont difficiles à transformer en forces que l'on peut intégrer. Il est cependant possible de rajouter des bornes pour limiter les déformations ;
- Les déformations restent parfois trop locales à l'endroit où s'exerce la contrainte, ce qui aboutit à de grandes déformations des ressorts avoisinants et peu de perturbations pour les ressorts plus éloignés. Si la force de contact s'intensifie, cela donne un mouvement de toute la structure et non pas un simple agrandissement de la zone d'interaction ;
- La méthode d'intégration est relativement mal conditionnée. En pratique, il faut parfois augmenter l'amortissement, c'est à dire la viscosité du milieu ambiant de façon importante pour éviter la divergence des solutions. Le modèle est alors très lent, comme plongé dans une substance dense ;
- Le nombre de nœuds dans le maillage doit rester petit pour deux raisons. La première est due au temps réel : le calcul des forces créées par un ressort étant important, on garde le nombre de ressorts petit, et donc, le nombre de masses. La seconde raison est liée aux conditions de l'intégration. En effet, la masse de l'organe et plus précisément son enveloppe surfacique étant constante, plus il y a de noeuds dans le maillage, moins la part de ces noeuds dans la masse totale est importante, donc plus la fréquence propre de ces noeuds devient élevée (la fréquence propre évolue comme $\sqrt{\frac{k}{m}}$), et plus le pas de temps doit diminuer. Au plus le pas de temps diminue, au plus il est difficile d'assurer la synchronicité, car la machine n'est plus suffisamment puissante.

La méthode de [Pro95] consistant à fixer des bornes à la déformations que l'on a déjà exposée précédemment peut permettre de contrer les problèmes de divergence et de déformations trop locales mais cette méthode ne garantissant pas la convergence, elle induit des oscillations auto-entretenues dans le modèle.

5.2.6. Paramétrisation du modèle

Le modèle tel que nous venons de le décrire possède un nombre important de paramètres. Décrire un objet physique revient à trouver les valeurs des paramètres permettant d'approcher au mieux le comportement de l'objet. La nature discrète de notre modèle nous empêche de nous baser uniquement sur la physique et au contraire nous oblige à aborder de façon empirique la paramétrisation du modèle [Lou94].

Tout d'abord, il s'agit de déterminer la part de la masse comprise dans la composante rigide. Le reste de la masse totale est alors compris dans la composante surfacique. La masse à affecter à chaque nœud du maillage est fonction de la densité de masse de l'enveloppe extérieure de l'objet. Les valeurs des coefficients de raideur des ressorts sont fixées par l'ordre de grandeur de la résistance à l'enfoncement et à la découpe que l'on souhaite obtenir.

Les valeurs d'amortissement et de viscosité sont très délicates à déterminer. En effet, elles conditionnent tout d'abord la convergence des équations, ce qui entraîne des bornes inférieures en deçà desquelles le modèle peut diverger si la fréquence d'échantillonnage n'est pas suffisante. Ensuite, dans la mesure où le temps est discrétisé, des bornes supérieures apparaissent ([JB94] montrent qu'avec la discrétisation qu'ils emploient, le coefficient de viscosité est borné par $m/\Delta t$).

Ces deux remarques prouvent que si la base de temps est mal choisie, il est impossible d'obtenir un modèle qui converge, même en augmentant l'amortissement. En d'autres termes, si on se

fixe une base de temps à ne pas dépasser sur la machine de simulation, alors seule une certaine gamme d'objets pourra être simulée.

5.3. Traitement des interactions

Nous proposons pour notre simulation un modèle de traitement des interactions. Il comprend les trois points que nous avons développés tout au long du *chapitre 4*, mais que l'on reprend ici dans l'ordre chronologique de leur utilisation : l'accélération de la détection, la détection entre primitive puis le calcul de la réponse. Les méthodes développées dans cette partie sont indépendantes du type de modélisation que nous venons de présenter. Plus encore, elles sont générales et applicables à un grand nombre de modèles polyédriques.

Les techniques de détermination d'intersection 3D ou de calcul analytique d'intersection de trajectoire sont les deux méthodes les mieux adaptées au contexte temps réel. La première a cependant le double désavantage d'une part de ne pas détecter certaines collisions et d'autre part d'exiger une recherche dichotomique lorsque l'on souhaite connaître les circonstances, le lieu et l'instant exact de la collision. Ces données sont nécessaires pour le calcul de la réponse et doivent donc être fournies par notre routine de détection de collision. En temps réel, des algorithmes itératifs comme la recherche dichotomique sont généralement peu employés car ils ne s'exécutent pas à temps borné. Notre choix s'est donc porté sur le calcul analytique des intersection de trajectoires, sous réserve qu'il soit possible et sans itérations. Une fois la collision détectée, on extrapole la situation afin d'obtenir un nouvel état du système sans interpénétration. Les effets de la collision sont alors calculés pour le pas de temps suivant.

5.3.1. Accélération de la détection

On suppose que l'on se trouve dans un contexte avec de nombreux objets. La détection de collision au plus bas niveau revient à tester chaque primitive des objets entre elles. La complexité est alors quadratique. Si on ne s'intéresse qu'aux interactions entre l'outil et les divers organes, cette complexité doit être multipliée par le nombre d'organes. Si de plus, il faut tester les organes entre eux, cette phase est également quadratique en fonction du nombre d'organes.

On s'intéresse donc à un procédé permettant de détecter rapidement les objets susceptibles d'entrer en collision. Une première étape intéressante est la méthode de [Rob95] exposée au paragraphe *4.2.2*, section *c*, en considérant cette fois-ci des boîtes englobant la trajectoire des objets (afin d'être compatible avec notre algorithme d'intersection de trajectoires). Cette méthode n'est intéressante que lorsqu'elle est appliquée à des environnements avec de nombreux objets : elle n'a donc pas été implantée pour nos premiers essais (où seuls deux ou trois objets sont définis par scène). Par contre, elle sera une étape incontournable pour l'application du modèle dans les simulateurs.

Nous nous sommes principalement intéressés à la phase suivante de l'accélération. On cherchent à savoir si deux objets sont susceptibles d'entrer en collision. On a vu au paragraphe *4.2.2* que les boîtes englobantes offrent des accélérations importantes. Par contre, les méthodes proposées ne cherchent que les intersection 3D, sans tenir compte du temps. De plus, les méthodes les plus rapides sont également celles les moins adaptées aux corps déformables. Notre approche est la suivante : on suppose qu'à l'instant t_0 et à l'instant $t_0 + \Delta t$, on calcule des boîtes englobantes alignées sur les axes pour tous les corps. Pour les corps déformables, cette étape se fait en parcourant chaque nœud et en déterminant quelles sont les plus petites et les plus grandes valeurs pour chaque coordonnée x, y et z. Pour les corps rigides, on peut simplement considérer une boîte englobant la boîte englobante du corps dans son repère local. Comme les trajectoires entre deux instants sont supposées rectilignes uniformes, on considère le tronçon de pyramide (en 4D) dont les deux bases sont les deux boîtes englobant l'objet aux deux instants. La *figure 5.3* montre, pour une seule dimension comment on détecte la collision probable de deux objets.

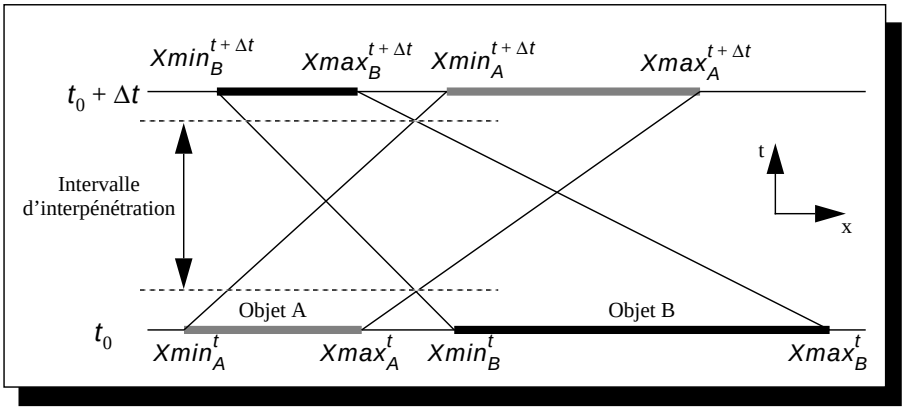

Figure 5.3. *Intersection de trajectoires de boîtes englobantes*

On retrouve le même type de schéma pour y et z. Afin d'alléger les notations, et sans perte de généralité, on pose $t_0 = 0$. De plus, on note $\Delta Xmin_A = Xmin_A^{\Delta t} - Xmin_A^0$, de même pour $\Delta Xmax_A$, et pour l'objet B. A un instant t, la boîte de A est définie par la borne inférieure suivante :

$$Xmin_A^t = Xmin_A^0 + t \times \Delta Xmin_A$$

De façon analogue on obtient sa borne supérieure. A l'instant t, les deux boîtes ne s'intersectent pas si l'une des deux conditions suivantes est vérifiée :

$$Xmin_A^t > Xmax_B^t \text{ ou } Xmin_B^t > Xmax_A^t \qquad (5.3)$$

En exprimant ces termes en fonction de t, et en effectuant les mêmes calculs pour les trois dimensions, on obtient six expressions linéaires en t qui restent positives tant qu'il n'y a pas d'intersection. Par exemple :

$$(\Delta Xmin_A - \Delta Xmax_B)\, t + \left(Xmin_A^0 - Xmax_B^0 \right) \geq 0 \qquad (5.4)$$

La mise en œuvre de l'algorithme est simple. La résolution de chaque inégalité donne un instant correspondant soit à l'entrée en collision, soit à la séparation des deux corps. Par exemple, dans l'équation *(5.4)*, on trouve que :

$$t_{sol} = \frac{Xmin_A^0 - Xmax_B^0}{\Delta Xmin_A - \Delta Xmax_B}$$

Les deux cas se distinguent par le signe de $\Delta Xmin_A - \Delta Xmax_B$. Si l'expression est positive, cela signifie que la condition de non collision *(5.3)* est valable pour $t \geq t_{sol}$ ce qui signifie que la collision prend fin à l'instant calculé, sinon, elle débute à cet instant là.

On obtient donc suivant les cas, une borne supérieure ou inférieure de l'intervalle de temps où se produit l'interpénétration. On parcourt donc les inégalités, chaque inégalité une fois résolue modifie la borne supérieure ou inférieure de l'intervalle d'interpénétration. Sitôt que l'intervalle devient vide ou qu'il n'est plus compris entre 0 et Δt, on conclut qu'il n'y a pas d'intersection et l'algorithme est terminé. En moyenne, il suffit donc de parcourir trois inégalités sur les six.

L'algorithme décrit ci-dessus est un bon moyen de détection intermédiaire, afin de déterminer rapidement si deux objets sont susceptibles de s'interpénétrer ou non. Cependant, il ne suffit pas car, si une collision possible est détectée, il va falloir tester toutes les primitives géométriques des objets entre eux. Notre accélération doit être complétée par une méthode comblant la différence entre les tests entre objets et les tests entre primitives de base. La méthode proposée peut être hiérarchique, mais cette hiérarchie est coûteuse dans le cas des corps déformables.

5.3.2. Détection des collisions

a) Interaction corps solide/corps déformable

Dans les hypothèses préliminaires, on a précisé que l'interaction que l'on étudiait finement était celle intervenant entre objet déformable et un objet solide. Nos objets sont polyédriques, les cas de collisions se réduisent alors aux situations suivantes :

-Nœud du corps déformable contre face du corps solide ;

-Arête du corps déformable contre arête du corps solide ;

-Nœud du corps solide contre face du corps déformable ;

On étudie donc ces trois cas successivement, par ordre de complexité calculatoire. A chaque fois on considère les trajectoires des éléments (nœuds, face ou arête) et on détermine s'il existe un instant où les objets sont à la même position de l'espace. Cependant, un choix conditionne les calculs suivants. En effet, la position du corps rigide, c'est à dire l'outil, n'est mesurée par les capteurs qu'à des instants discrets. Les positions intermédiaires entre ces instants ne sont donc pas connues. En pratique, la trajectoire peut être quelconque. Pour le calcul, il faut considérer une trajectoire particulière arbitraire. On choisit la plus intéressante d'un point de vue complexité à savoir la trajectoire linéaire uniforme (vitesse de déplacement constante).

- Collision point du corps déformable / face du corps rigide

En se plaçant dans le repère de la facette rigide, on peut ramener le calcul à celui d'un point passant par un plan fixe. En effet, si $\vec{\omega}$ est la vitesse de rotation instantanée du solide et $\vec{V}_G$ la vitesse instantanée du centre d'inertie alors on pose que la vitesse $\vec{V}_A$ au point A du corps déformable devient : $\vec{V'}_A = \vec{V}_A - \vec{V}_G + \overrightarrow{GA} \wedge \vec{\omega}$ (ce qui revient à étudier la vitesse relative du point A par rapport à la facette, en faisant une approximation de la rotation). Ensuite, on cherche s'il existe un instant t compris entre 0 et Δt, tel que : $A + t\vec{V'}_A \in \text{facette}$. Ce test s'effectue en injectant l'expression $A + t\vec{V'}_A$ dans l'équation algébrique de la facette, qui est une équation du premier degré. On obtient alors une équation linéaire en t. La valeur de t obtenue est égale à la distance minimale du point A au plan rigide divisée par la composante normale au plan de la vitesse relative $\vec{V}_A$. Il est intéressant d'utiliser dans une phase préliminaire l'orientation de la composante normale de la vitesse relative afin de déterminer si le point s'éloigne de la facette. Dans ce cas, il est inutile de vouloir calculer une intersection. Ce critère est à rapprocher des critères cinématiques d'accélération de la détection de collision exposés au paragraphe *4.2.4*. Si la vitesse est bien orientée, on calcule la solution puis on vérifie qu'elle est bien entre 0 et Δt, on teste alors si le point d'intersection obtenu se trouve effectivement à l'intérieur de la facette (en effectuant une série de produits mixtes).

- Collision arête du corps déformable / arête du corps rigide

De la même façon que le calcul ci-dessus, on se place dans le repère du côté rigide *[CD]* et on transforme les vitesses des extrémités A et B du côté déformable comme exposé dans le premier cas, afin d'obtenir les vitesses relatives $\vec{V'}_A$ et $\vec{V'}_B$. On trouve alors qu'il y a collision si à un moment donné un point de l'arête déformable se trouve sur l'arête rigide. Mathématiquement, on cherche t entre 0 et Δt, u et w entre 0 et 1 tels que $A + t\vec{V'}_A + u\left(B + t\vec{V'}_B - A - t\vec{V'}_A\right) = C + w\overrightarrow{CD}$. Cette expression se sépare en trois équations respectivement en x, y et z. On emploie l'une des équations afin de supprimer l'inconnue w, ce qui nous donne un système de la forme :

$$\begin{cases} a_0 t + a_1 u + a_2 tu + a_3 = 0 \\ a_4 t + a_5 u + a_6 tu + a_7 = 0 \end{cases}$$

u s'exprime alors comme une fonction homographique en t :

$$u = \frac{\alpha t + \beta}{\gamma t + \delta}$$

Une fois u éliminé, on obtient une équation de degré 2 en t. Il reste ensuite à vérifier que les valeurs de t, puis u puis w se trouvent dans leur bon domaine de définition.

- Collision point du corps rigide / face du corps déformable

Ce cas a fait l'objet d'études auparavant. En effet, le problème est identique à celui développé par [MW88] et que nous avons détaillé au paragraphe *4.1.2* section *c*, pour le calcul de l'intersection entre deux corps déformables. La méthode proposée alors aboutissait à la résolution d'une équation de degré 5 nécessitant un algorithme de résolution itératif. Nous préférons traiter le problème de la manière suivante : si D est le point du corps rigide et A, B et C les sommets de la facette déformable, il y a intersection si et seulement si à un instant donné le point A appartient à la facette. Plutôt que d'exprimer l'équation du plan de la facette de façon paramétrique, on préfère l'exprimer sous forme algébrique. En d'autres termes, on cherche t compris entre 0 et Δt tel que : $\left(\overrightarrow{AB}(t) \wedge \overrightarrow{AC}(t)\right) \cdot \overrightarrow{AD}(t) = 0$, avec $\overrightarrow{AX}(t) = X + t\vec{V}_X - A - t\vec{V}_A$ pour $X = B$, C et D. Cette équation est du troisième degré en t. On cherche alors le plus petit t compris entre 0 et Δt, tel que le point d'intersection soit effectivement contenu par la face.

Dans les trois cas, on obtient un polynôme dont on peut trouver les racines en temps constant. La détection est alors immédiate : on teste chaque point du corps déformable avec chaque face du corps rigide, puis chaque arête de l'un avec chaque arête de l'autre et enfin chaque point du corps solide contre chaque face du corps déformable. La complexité de l'ensemble de la détection est donc quadratique.

b) Collision entre corps déformables

La collision entre corps déformables apparaît dans deux situations : lorsque deux organes entrent en contact ou lors d'une collision d'un objet sur lui-même. Ce second cas n'a pas encore été étudié, mais devra être traité car le phénomène d'auto-collision est fréquent sitôt qu'il y a découpe. Sous l'hypothèse que les interactions entre les organes ne nécessitent pas autant de précision qu'entre un organe et l'outil du chirurgien, on peut se contenter du cas le plus fréquent de collision, c'est à dire un nœud de l'un des organes passe au travers d'une face de l'autre organe. Dans ce cas, on néglige les collisions entre arêtes, ce qui n'est pas gênant en pratique, car la forme des organes est généralement arrondie, peu anguleuse. La détection de ce type d'interaction est strictement identique au cas d'un nœud du corps solide sur une face du corps déformable. Cependant, ce calcul exige la résolution d'une équation linéaire du troisième degré. La détection des interactions entre corps déformables est donc aussi coûteuse que dans le cas solide/déformable.

5.3.3. La réponse à la collision

Nous traitons dans cette partie toutes les opérations effectuées une fois la collision entre deux primitives détectée. Le traitement dépend du type de collision. Le premier cas correspond à la collision d'un nœud du corps déformable sur une face du corps rigide. Ce cas est traité par l'algorithme classique de contact entre un point et un plan. Par contre les deux autres cas (collision entre arêtes et collision d'un nœud du corps rigide sur le corps déformable) exigent des traitements particuliers. Ce sont en effet des cas où la résolution du maillage du corps déformable a une incidence visible sur la déformation au niveau de la zone de contact. On propose de traiter ces deux derniers cas en subdivisant le maillage. Cette subdivision permet non seulement d'affiner le maillage autour de la zone de contact, mais en plus de réduire ces deux cas à un seul traitement de la collision : celle d'un nœud du corps déformable contre un ensemble d'arête du corps rigide.

a) Collision point du corps déformable / face du corps rigide

Cette collision correspond à des situations courantes : par exemple lorsque l'organe est posé dans un environnement relativement plan. Une autre situation est celle où le chirurgien utilise un outil pour pousser l'organe. Soit un nœud du corps déformable. Durant la détection; ce nœud est testé par rapport à toutes les faces du corps rigide. A chaque détection, la course du nœud est réduite afin qu'il n'y ait plus interpénétration mais seulement contact. Un même nœud peut en effet entrer en collision avec plusieurs faces durant le même intervalle de temps. A la fin du parcourt, le nœud est en contact avec la première face qu'il a rencontrée sur son trajet.

La réponse à la collision se fait alors de façon analytique tout en restant pragmatique. Les modèles permettant de modéliser les frottements sont utilisés : au niveau cinématique, la nouvelle vitesse est calculée par la méthode exposée au paragraphe *4.3.3*, section *a*, à savoir :

$$\vec{v}_{t+\Delta t} = \lambda \vec{v}_T - \mu \vec{v}_N$$

En ce qui concerne les forces, le modèle de Coulomb (voir paragraphe *4.3.1*) est employé. On repère dans la force animant le point les composantes normale et tangentielle au plan de la collision. La force normale est annulée et à la force tangentielle se rajoutent les forces de frottements. On garantit ainsi que le point ne pourra plus pénétrer dans le plan. La fin du contact ne nécessite pas de traitement particulier. En effet, la détection suppose que la vitesse du point est orientée vers le plan. Si ce n'est pas le cas, aucune collision n'est détectée, le point garde sa trajectoire initiale, et donc, quitte le plan.

b) Les autres cas de collision

La collision arête/arête est essentielle pour notre simulation car elle se produit, par exemple, lorsqu'un outil tranchant est employé pour une découpe. La collision point solide sur face déformable apparaît lorsqu'un outil pointu tire sur l'organe. Le traitement de ces deux cas de collision dépendent complètement de l'échantillonnage surfacique du corps déformable.

Subdivision adaptative du maillage

On a vu que notre modèle supposait un nombre réduit de masses et de ressorts pour des raisons de temps réel et d'échantillonnage temporel. L'échantillonnage de la surface étant faible, le traitement des contacts devient grossier. A l'instar des méthodes employées pour les déformations de tissus [HPH96], on décide de subdiviser le maillage afin de le rendre plus précis dans les zones de collision. En subdivisant le maillage, nous pouvons traiter les collisions avec une résolution aussi fine qu'on le souhaite. La subdivision apparaît lors de la collision d'un point de l'objet rigide

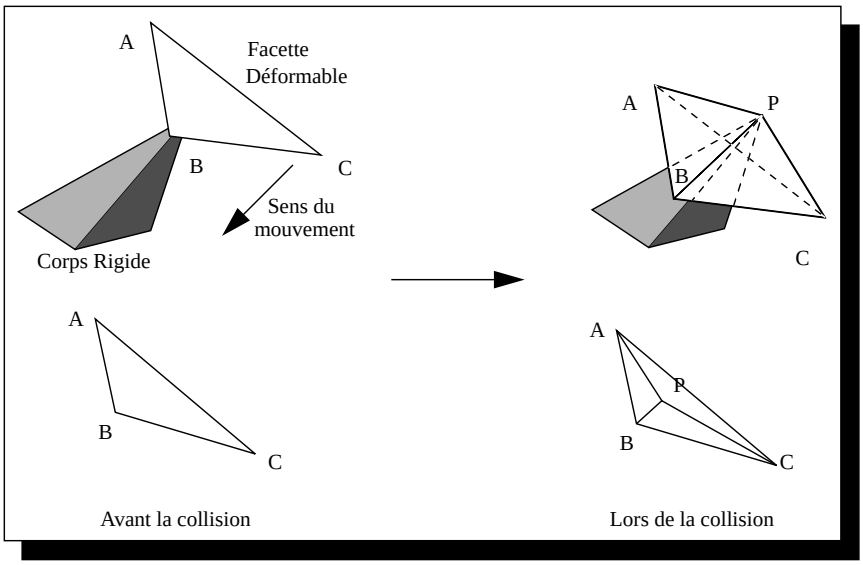

Figure 5.4. *Collision nœud du corps rigide sur facette du corps déformable*

contre un triangle du corps déformable (voir *figure 5.4*) ou lors de la collision entre une arête déformable et une arête rigide. Au point de collision de la surface ou de l'arête, on ajoute un noeud dans le maillage. Le problème revient alors à modifier la densité de points du maillage sans modifier les propriétés physiques de ce dernier.

La conservation des moments d'ordre 0 (la masse totale), d'ordre 1 (le barycentre) et enfin d'ordre 2 aboutit naturellement à la solution triviale d'ajouter une masse nulle. En d'autres termes, il est impossible d'ajouter une masse dans le maillage sans perturber le comportement du système. La méthode de subdivision proposée par [HPH96] aboutit à une solution plaisante, mais qui ne conserve pas la masse : dans un cas interactif, où par exemple le chirurgien tient l'objet, l'augmentation brusque de la masse totale risque de le troubler ! On choisit donc de soulever un minimum de contraintes, afin d'obtenir un système présentant un nombre infini de solutions, dont seules celles donnant des masses positives nous intéressent. On choisit une méthode qui nous garantit de garder la masse totale constante et les masses positives, au détriment des moments d'ordre 2

Nous traitons dans la suite le cas d'une collision d'un point rigide avec un triangle déformable ABC. Ce cas englobe les collisions contre une arête déformable. Cette dernière situation est en effet un cas particulier de la première, avec certains coefficients nuls. La subdivision correspondante est donnée sur la *figure 5.5*.

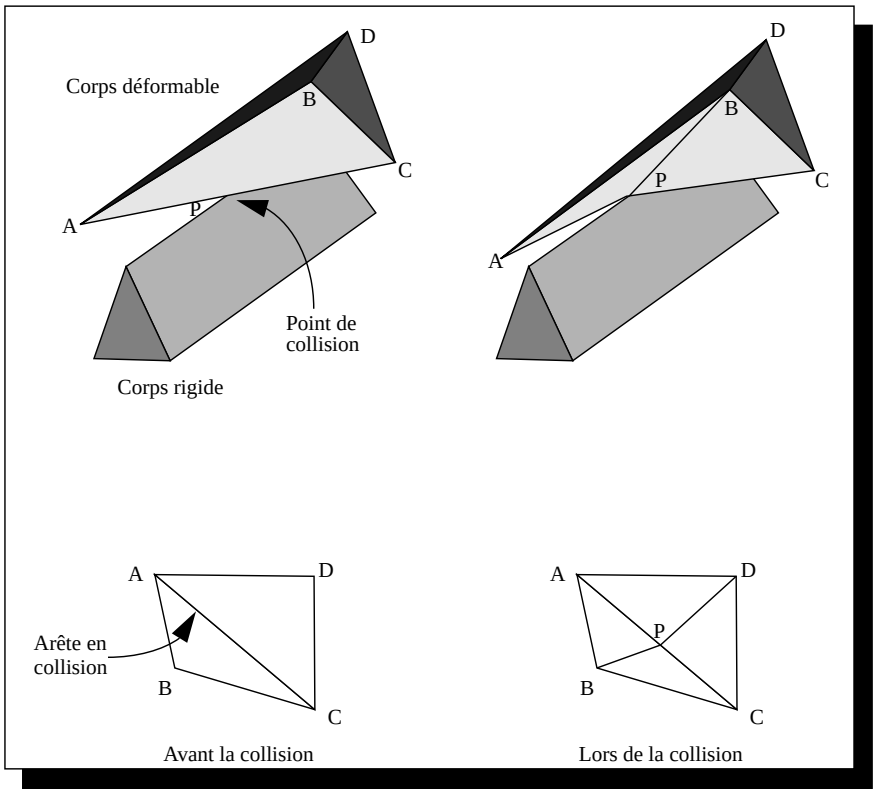

Figure 5.5. *Subdivision lors de la collision arête/arête*

Le point de collision s'exprime de façon unique sous la forme :

$$P = \alpha A + \beta B + \gamma C \text{ avec } \alpha + \beta + \gamma = 1 \tag{5.5}$$

On considère que la part de masse en P est obtenue en prélevant une partie des masse en A, B et C :

$$\overset{P}{t} = a m^{A}_{t-\Delta t} + b m^{B}_{t-\Delta t} + c m^{C}_{t-\Delta t} \tag{5.6}$$

D'où, si on désire conserver la masse totale :

$$\overset{A}{t} = (1-a)\, m^{A}_{t-\Delta t} \quad m^{B}_{t} = (1-b)\, m^{B}_{t-\Delta t} \quad m^{C}_{t} = (1-c)\, m^{C}_{t-\Delta t} \tag{5.7}$$

Les parts des masses allant des nœuds *A*, *B* et *C* vers le nœud *P* dépendent directement de la place de *P* dans le triangle de la façon suivante :

$$\frac{a}{\alpha} = \frac{b}{\beta} = \frac{c}{\gamma} \tag{5.8}$$

Enfin, on considère qu'une fois les masses réparties, on assure une cohérence des nouvelles valeurs en posant que la masse en *P* est telle que :

$$m_t^P = \alpha m_t^A + \beta m_t^B + \gamma m_t^C \tag{5.9}$$

Une fois le système résolu, on trouve que :

$$\frac{a}{\alpha} = \frac{b}{\beta} = \frac{c}{\gamma} = \frac{1}{1 + \frac{\alpha^2 m_{t-\Delta t}^A + \beta^2 m_{t-\Delta t}^B + \gamma^2 m_{t-\Delta t}^C}{\alpha m_{t-\Delta t}^A + \beta m_{t-\Delta t}^B + \gamma m_{t-\Delta t}^C}} \tag{5.10}$$

Tous les termes présents dans l'expression *(5.10)* sont positifs d'où *a*, *b* et *c* positifs. De plus, le dénominateur est toujours supérieur à 1. Le membre de droite de l'expression est donc toujours inférieur à 1. Afin de trouver *a*, *b* et *c*, il faut le multiplier respectivement par α, β ou γ qui sont également inférieurs à 1. On démontre donc facilement que *a*, *b* et *c* sont toujours compris entre 0 et 1, ce qui garantit d'obtenir dans tous les cas des masses positives.

Le même type de calcul est employé afin de déterminer les valeurs de la viscosité dans le milieu ambiant. Il faut tout de même prendre garde à l'ordre de grandeur des masses des nœuds après la subdivision, car la masse des points ajoutés peut être relativement faible et de ce fait incompatible avec les contraintes imposées par Shannon. Il faudrait alors calculer la simulation avec des pas de temps à différents grains, les constantes de temps les plus petites s'effectuant sur un nombre très restreint de nœuds et le pas de temps le plus important sur la composante rigide. Cette solution n'a pas encore été testée car elle nécessite de changer les hypothèses de base de notre simulation (un seul pas de temps pour tout le calcul).

Ensuite, il s'agit de déterminer les constantes de raideurs des nouvelles arêtes formées. La liaison reliant le nouveau noeud à la composante rigide est obtenue par un calcul d'une moyenne pondérée des raideurs des ressorts des nœuds *A*, *B* et *C*. Par contre, pour les nouveaux ressorts de la composante surfacique (ceux reliant le nouveau point aux nœuds *A*, *B* et *C*), les constantes peuvent être déterminées par plusieurs solutions, mais qui ne sont malheureusement pas satisfaisantes. La première consiste à calculer une moyenne dépendant des angles : par exemple le ressort *AP* s'obtient en moyennant les ressorts *AC* et *AB*, en pondérant par les ratios des angles respectivement *BAP* et *CAP* sur *CAB*. Une seconde solution consiste à supposer que la constante d'élasticité sur la distance du ressort au repos est une constante. Toutes ces propositions reviennent à charger les nœuds *A*, *B* et *C* de nouvelles liaisons et entraînent des perturbations dans leur comportement. [TC90] ont proposé une solution permettant de diviser un ressort en deux sans perturber le comportement du système, mais ils ne s'intéressent pas à l'insertion de nouveaux ressorts à l'intérieur d'une maille.

Dans le cas de la collision arête/arête, les nouvelles raideurs pour *AP* et *DP* s'obtiennent de façon analogue, en calculant une moyenne dépendant des angles ou par densité de raideur. On cherche maintenant à déterminer les raideurs de *BP* et *CP*. Soient deux ressorts en série de raideurs k_1 et k_2. Le ressort équivalent a pour raideur :

$$\frac{1}{k} = \frac{1}{k_1} + \frac{1}{k_2}$$

k étant connu, on cherche à trouver les raideurs des deux ressorts. Le point P s'exprime comme :

$$P = \beta B + \gamma C \text{ avec } \beta + \gamma = 1 \tag{5.11}$$

On désire que dans les cas extrême, où *P* est en *B* ou en *C*, la raideur reste inchangée. On trouve alors que :

$$k_{BP} = \frac{k}{\gamma} \text{ et } k_{CP} = \frac{k}{\beta}$$

Cette solution, si elle repose complètement sur la physique, n'est cependant pas intéressante. En effet, les raideurs calculées peuvent prendre des valeurs positives non bornées. La valeur des raideurs a une incidence sur la fréquence du système, on ne peut donc pas, dans ce cas, borner la fréquence d'échantillonnage. L'inconvénient est majeur pour la stabilité du système. Dans notre implantation, on préfère donc poser, arbitrairement :

$$k_{BP} = \beta k \text{ et } k_{CP} = \gamma k$$

Calcul de la réponse

Lorsque l'on a détecté une collision entre arêtes ou entre un point du corps rigide et une face du corps déformable, un nouveau nœud apparaît, issu de la subdivision décrite précédemment. Ce nœud permet de réduire les deux cas à un seul calcul de la réponse : la collision a lieu entre le nouveau nœud du corps déformable et une arête ou un ensemble d'arêtes du corps solide.

Afin de calculer la réponse à ce type de collision, il convient de pouvoir analyser dans un premier temps si la collision prend fin ou si au contraire il faut ajouter des forces empêchant l'interpénétration. Dans le cas de collision vu à la section *a*, ce calcul était simple, puisque seule l'orientation de la vitesse du point par rapport à la facette est à tester. Or dans la collision ici présente, la distinction entre la fin ou non de collision est délicate. En effet, on ne peut pas se contenter d'inspecter le mouvement du nœud déformable et déterminer s'il pénètre dans le corps rigide. La *figure 5.6* illustre un cas d'interpénétration sans interpénétration du nouveau nœud dans la structure. Cette figure est une représentation en coupe, au niveau de la collision de la situation exposée sur la *figure 5.5* La coupe est faite dans un plan orthogonal à l'arête du corps solide. On suppose que lors de la collision, le point P est animé d'une vitesse $\vec{V}_p$. Cette vitesse est telle que le point P ne pénètre pas dans le corps solide. Si on conclut que le point quitte le corps solide, et qu'on empêche pas son mouvement, on obtient une interpénétration de la liaison contenant P.

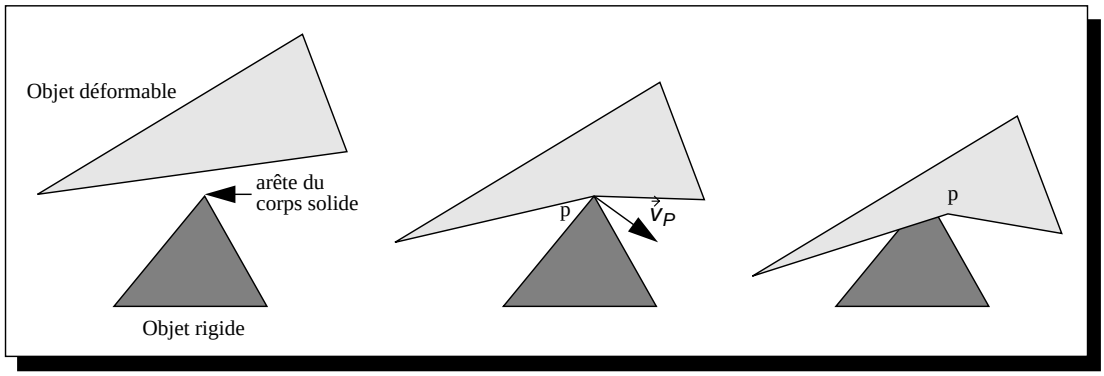

Figure 5.6. *Vue en coupe d'une interpénétration arête/arête*

Ce cas précédent nous permet d'exhiber des zones pour la vitesse du point P où l'on peut certifier s'il y a ou non interpénétration. Par contre, il reste des zones exigeant une étude plus fine (voir *figure 5.7*). Nous choisissons de considérer qu'il y a interpénétration pour les zones indéterminées. Le résultat est qu'une collision est parfois détectée à tort.

Une fois l'interpénétration détectée, on fait en sorte que le point du corps déformable reste sur le corps rigide quels que soient les déplacements respectifs des deux corps. Pour le moment, aucun modèle de calcul de frottements ou de force de contact n'est employé.

5.4. Traitement des découpes et déchirements

Nous n'avons pas encore de solutions définitives concernant les problèmes de découpages et de déchirements, cependant nous présentons dans cette partie nos premiers résultats. Une idée simple pour le traitement des découpes ou déchirements est de vérifier les forces ou de façon équivalente, les déplacements des liaisons de la composante déformable. Sitôt que la force élastique

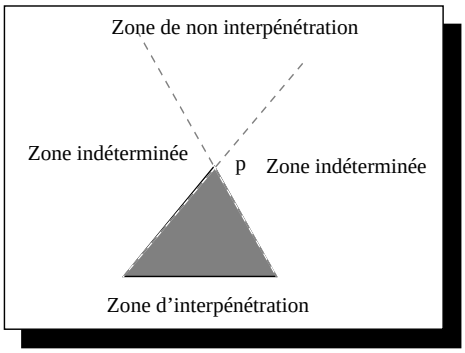

Figure 5.7. *Zones de décision d'interpénétration*

ou le déplacement entre deux nœuds devient supérieur à un seuil, on considère qu'il y a découpe ou déchirement, de façon indifférenciée. Cette méthode, bien qu'attractive pour sa simplicité, est mal adaptée à notre modèle. En effet, la stabilité précaire des solutions due à la méthode d'intégration, fait que les distances entre nœuds peuvent parfois devenir importantes. On peut chercher à borner ces distances pour contrecarrer la divergence. Toutefois, il paraît incohérent de se fixer deux bornes, l'une pour empêcher la divergence et l'autre pour permettre les découpes.

Nous avons cherché une méthode permettant de détecter où ont lieu des découpes provoquées par l'utilisateur. Pour cela, on se place dans le cas des collisions côté/côté puisque c'est ce cas de collision qui survient lors de la manipulation d'un outil tranchant. On désire un test permettant de prendre en compte l'aspect tranchant de l'outil manipulé. On considère que l'on est en présence de la situation où un point du corps déformable est entré en collision avec une arête quelconque de l'outil, comme on le montre sur la *figure 5.8*.

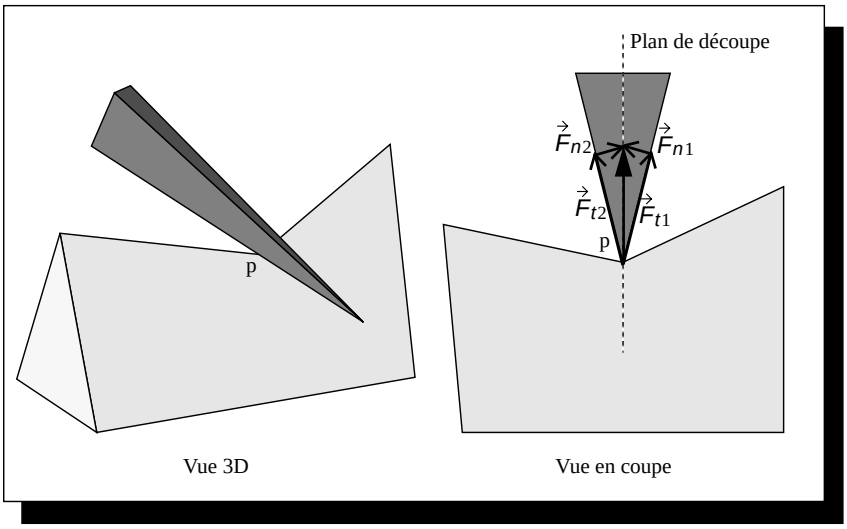

Figure 5.8. *Principe du découpage*

On décide qu'il y a découpe lorsque les composantes tangentielles $\vec{F}_{n1}$ et $\vec{F}_{n2}$ de la force de réaction du point par rapport aux deux plans du corps rigide dépassent un seuil. On peut faire dépendre ce seuil à la fois de l'organe et de l'outil. Cette idée est intuitive : elle signifie simplement que plus l'angle de l'instrument de découpe est aigu, plus la découpe est facile. Au contraire, plus l'angle est obtus, plus les composantes tangentielles de la force sont faibles et plus il faut appuyer pour découper.

On définit alors le *plan de découpe* comme le plan bissecteur des deux faces. Une fois la découpe détectée, on définit une partition de l'ensemble des faces auxquelles appartient le nœud du corps déformable en deux groupes suivant la position du centre (isobarycentre) des faces par rapport au plan de découpe. On sépare alors au niveau de la frontière entre ces deux groupes. La découpe est effectuée, en dupliquant le nœud où s'est effectuée la découpe. La *figure 5.9* illustre ce processus de duplication.

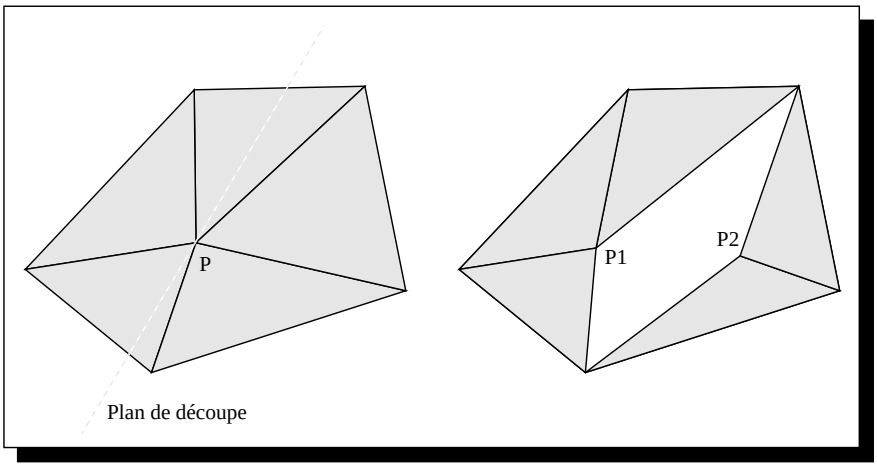

Figure 5.9. *Duplication de nœud lors d'un découpage (vue de dessus)*

Le traitement de la découpe n'est cependant pas achevé et deux problèmes restent à résoudre. Tout d'abord, on a dupliqué le nœud où a eu lieu la découpe. Les deux points que l'on obtient alors ont a priori le même homologue sur la composante rigide. Si on garde la liaison avec la composante solide, sitôt que l'objet est au repos les deux points convergent vers le même point de la composante rigide. La découpe n'est alors plus apparente. Au contraire, si la liaison avec la composante rigide est détruite, les faces de l'objet déformable sont alors uniquement contraintes par deux sommets sur trois. Elles sont alors libres de tourner autour de deux de leur sommets. Le second problème est celui de la visualisation de l'intérieur de la structure : en effet, notre modèle est surfacique, lorsqu'il est découpé, un trou apparaît car aucun affichage n'est effectué dans l'entaille. Il manque un processus permettant de donner une texture au volume du modèle.

On cherche actuellement à résoudre ces deux problèmes de façon unique : la découpe peut créer des nouvelles faces, qui représentent l'intérieur de l'objet. Notre modèle reste alors surfacique, tout en procurant une vue de l'intérieur de la structure et en assurant des contraintes sur les facettes en bordure de la découpe. Une fois cette proposition validée, il reste à résoudre les problèmes liés au changement de topologie de l'objet : les retournement de surface et les divisions d'un objet en deux sont des exemples des problèmes que l'on peut rencontrer et qui restent à étudier.

Conclusion

Nous avons exposé dans cette partie l'ensemble de notre modèle pour la simulation dynamique d'organes. Un grand nombre de questions n'ont qu'une réponse partielle. Cependant, nous avons montré qu'il est possible d'obtenir une complexité minimale permettant de gérer simplement des corps déformables en temps réel. De plus, l'ensemble de notre routine de traitement des collisions est générale et peut donc s'appliquer à tout modèle polyédrique. En outre, nous avons montré qu'il était possible de détecter précisément les divers types de collisions, en temps constant pour chaque primitive, alors que ces détections étaient traitées par itérations auparavant. Les algorithmes d'accélération même s'ils ne sont pas complets offrent déjà des gains importants. Cette affirmation s'appuie sur la mise en œuvre de notre modèle effectuée lors de cette thèse. Cette mise en œuvre est détaillée dans le chapitre suivant.

6 Mise en œuvre

Introduction

Nous présentons dans cette partie, la mise en œuvre de notre modèle. Cette implantation a deux objectifs. D'une part, on cherche à savoir si notre modèle permet une mise en œuvre informatique et offre des simulations réalistes c'est à dire qui simulent de façon réaliste les comportements. D'autre part, on cherche à déterminer sa pertinence dans un contexte temps réel. Le premier point fait l'objet des deux premières parties de ce chapitre où l'on exposera les choix que nous avons faits ainsi que les difficultés rencontrées dans la mise en œuvre d'un tel modèle physique suivi par l'éventail des situations que nous avons simulé. Le deuxième point fait l'objet de la troisième partie où nous présentons les performances en fonction de la complexité du modèle de corps déformable et du traitement des collisions. Nous terminons ce chapitre par une discussion concernant les applications de notre modèle et surtout la comparaison avec son concurrent direct : les éléments finis.

6.1. Détails d'implantation

6.1.1. Environnement choisi

Le moteur de notre simulation a été programmé en *C*, pour ses qualités de rapidité et de portabilité. Il est en outre le langage support de la bibliothèque graphique *OpenGL*. Nous avons choisi cet environnement car il est largement indépendant du matériel. Toutes les machines graphiques, de la *Reality Engine* de *Silicon Graphics* au PC sans accélérateur sont capables à l'heure actuelle d'exécuter des logiciels employant la bibliothèque *OpenGL*. Notre programme a pu être testé sur diverses plates-formes, ainsi les mesures des performances ont pu être effectuées sur deux machines différentes (une *Silicon Graphics Indigo 2* et un *PC Intergraph*).

6.1.2. Optimisation

Notre implantation n'a pas subi d'optimisation particulière, même si nous nous sommes efforcés de réduire les coûts mémoire et l'ensemble des boucles. L'affichage peut être accéléré, grâce à l'emploi des listes d'affichage[1] proposées par *OpenGL*. Les listes d'affichage stockent de façon optimale, mais statique, la position de tous les points d'un objet 3D. S'il est pratique d'employer ces listes pour les objets rigides, c'est à dire ceux dont la géométrie n'est pas modifiée, ces listes ne sont pas du tout adaptées aux objets déformables. En effet, la géométrie de ces objets étant par définition variable et les listes d'affichage étant figées, il faut redéfinir entièrement la liste associée à un objet à chaque déformation. En outre, notre routine de détection de collision exige

1. *display list* en anglais.

de connaître la position de tous les points, il est donc inutile de déporter les calculs de placement des objets sur la carte graphique puisqu'ils doivent également être effectués par le processeur.

Les listes d'affichage n'étant pas exploitables, on envisage un autre procédé d'accélération de l'affichage. Un objet que l'on veut afficher peut être défini sous forme de mailles. Cela permet de ne définir qu'une seule fois un point pour toutes les facettes auxquelles il appartient. La définition des mailles est compatible avec les corps déformables, mais devient difficile à respecter sitôt que le modèle peut être découpé. Nous avons considéré un seul type de maillage, le *FAN* (des triangles disposés autour d'un seul point), au détriment du type *STRIP* (triangles disposés les uns à côté des autres, le long d'un ruban) car il est plus simple à respecter.

Enfin, le processus d'affichage des corps déformables est coûteux car il nécessite de calculer des normales en chaque surface et en chaque nœud sitôt qu'il y a déformation.

6.1.3. Précision des calculs numériques

Nous avons évoqué à de nombreuses reprises le manque de stabilité de l'intégration des équations du mouvement. Ce manque de stabilité est augmenté par les erreurs numériques. Cependant, il est difficile de distinguer les causes des instabilités, les divergences dues à l'intégration et aux erreurs numériques se manifestant de façon identique. En effet, les erreurs numériques peuvent être considérées comme des non-linéarités (au même titre que les collisions) qui viennent perturber les fréquences du système. En pratique, on augmente la précision des réels (passer en *double* voire, en *long double*) afin de vérifier si ce sont les erreurs numériques qui perturbent le système ou non.

De plus, les erreurs numériques peuvent parfois être responsables d'états incohérents du système. Ce phénomène apparaît particulièrement dans les calculs de collision. On a vu que ces calculs reposent sur des égalités. En pratique, on cherche à résoudre le système avec une certaine marge d'erreur, pour cela ces égalités sont transformées en inégalités. La marge d'erreur est alors très délicate à déterminer : trop faible, des collisions peuvent ne plus être détectées car proches des conditions aux limites, trop élevée, des collision peuvent être détectées alors qu'elles n'ont pas lieu. Par exemple, lorsqu'une arête entre en collision avec l'extrémité d'un autre arête, on se trouve face à une condition limite où il est très délicat de différencier les collisions effectives des simples proximités des deux corps.

Nous proposons un embryon de solution en donnant un sens physique à cette marge d'erreur. Pour la collision d'un point avec une surface rigide, nous considérons que cette surface possède une certaine épaisseur. La collision est détectée sitôt qu'un point pénètre dans cette épaisseur. La valeur qui est donnée à cette dernière est par conséquent l'erreur que l'on admet lors du calcul de la collision. Cependant, l'algorithme de la réponse à la collision doit être adapté à la prise en compte de cette épaisseur. En effet, lorsqu'une collision est détectée et que c'est un contact prolongé, il faut ramener le point en dehors de l'épaisseur de la facette, afin que l'état reste cohérent et que le point puisse ensuite quitter la facette. Si le retour du point en dehors de l'épaisseur est mal calculé, cela débouche sur des mouvements spontanés du point qui va se déplacer sur la surface. On remarque ainsi que la prise en compte des erreurs numériques est délicate à mettre en œuvre.

6.2. Exemple de situations simulées

6.2.1. Langage de description

Le modèle que nous proposons est générique et ne fait pas de supposition sur le type de manipulation que l'on effectue. Afin de tester et valider le modèle, il s'agit de lui fournir la possibilité de travailler avec des situations différentes. C'est pourquoi, notre logiciel se base sur un langage simple de description de situation. Le détail des commandes de ce langage est donné en annexes. La situation à simuler est décrite dans un fichier appelé fichier de scène (extension *SCN*). La construction de ce fichier consiste à fournir une liste d'objets que l'on précise rigides ou déformables.

Chaque objet (rigide ou déformable) est caractérisé par sa forme, ainsi que ses propriétés physiques. Ces dernières sont données explicitement dans le fichier de scène. Par exemple, la masse, la raideur des ressorts, les coefficients de frottement... Par contre, la forme des objets est supposée définie par un modeleur, c'est pourquoi elle fait référence à un fichier binaire séparé, de format unique développé pour notre simulation (format *OJT*, donné en annexe). Actuellement un convertisseur permet d'obtenir ce fichier à partir de fichiers de type *NFF* ou équivalent (liste des points de chaque facette avec éventuellement leur normale), on peut ainsi utiliser les données de la majorité des modeleurs du commerce.

Durant la simulation, seul le premier objet rigide déclaré dans le fichier peut être manipulé par l'utilisateur (ce n'est pas une limite du programme, mais une simplification de l'interface). On peut associer un mouvement par défaut de cet objet qui sera le mouvement effectué par le corps tant que l'utilisateur ne prend pas le contrôle du déplacement de l'objet. Le contrôle se fait par l'intermédiaire de la souris.

Enfin, afin de fixer certains degrés de liberté, l'utilisateur peut, en lançant la simulation contraindre certains nœuds du corps déformable à rester à leur position de départ. On dit alors que le nœud est gelé. Dans ce cas, la position du nœud ne peut être modifiée que par la collision avec un objet rigide. Ainsi, l'utilisateur pourra, en manipulant un outil, déplacer le nœud gelé.

6.2.2. Etude d'une situation simple : un objet en chute libre

La première situation étudiée concerne un icosaèdre tombant en chute libre dans un premier temps, puis entrant en collision avec un plan rigide horizontal. La première partie du mouvement montre qu'il n'y a pas de dissociation entre la composante rigide et déformable. L'objet n'est pas déformé par la chute libre. Si un des nœud est gelé, l'objet est alors retenu par ce point et se déforme en se balançant (voir *figure 6.1*).

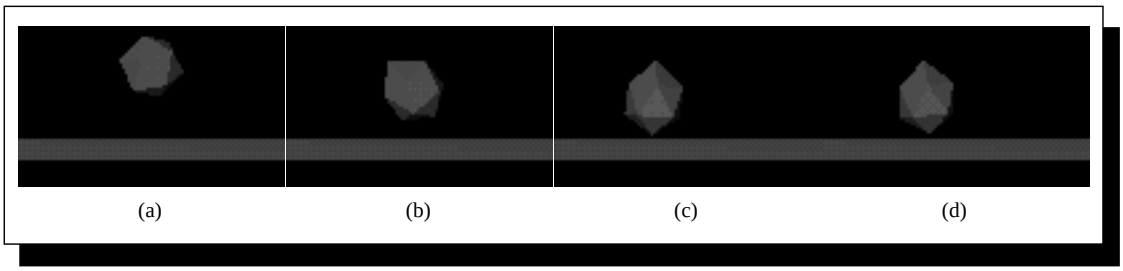

Figure 6.1. *Chute d'un corps autour d'un point fixe*

Si l'objet ne subit aucune contrainte, la simulation sur la *figure 6.2* montre que la collision des nœuds avec le plan est détectée. L'icosaèdre atteint alors une position de repos où il est déformé par son propre poids. Sa déformation est directement fonction de l'élasticité des ressorts.

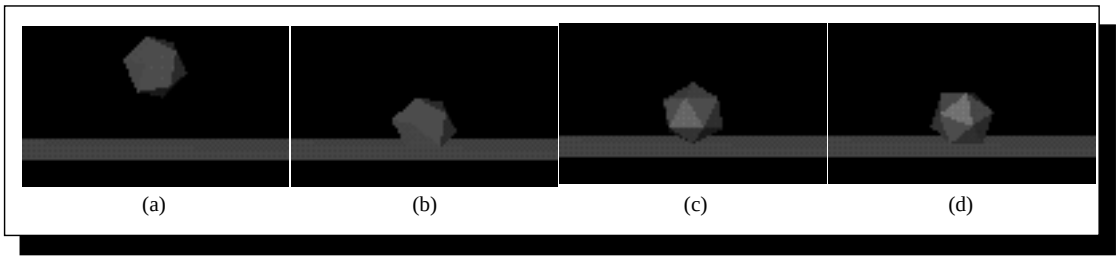

Figure 6.2. *Chute d'un corps sur un plan*

Le test peut alors se compliquer d'une part en donnant un mouvement au solide. Par exemple, une rotation autour de son axe médian ou une translation et observer comment se comporte l'objet déformable. Suivant les frottements entre les deux objets, l'objet déformable subit la même rotation que le plan (frottement élevé), soit au contraire reste immobile en glissant sur le plan en rotation (frottement nul). Ce même test a été effectué en remplaçant l'icosaèdre par une sphère afin d'étudier la simulation dans le cas d'un nombre plus grand de nœuds.

6.2.3. Etude des frottements : cas du corps sur un plan incliné

On considère le cas particulier où l'icosaèdre tombe sur un plan incliné. On fait varier le coefficient de frottement. Il existe une valeur critique (liée à la loi de Coulomb) en deçà de laquelle l'objet glisse le long du plan (voir *figure 6.3*), et au delà de laquelle l'objet reste immobile. La transition entre ces deux cas correspond à la situation où l'objet glisse en décélérant avant d'atteindre sa position de repos.

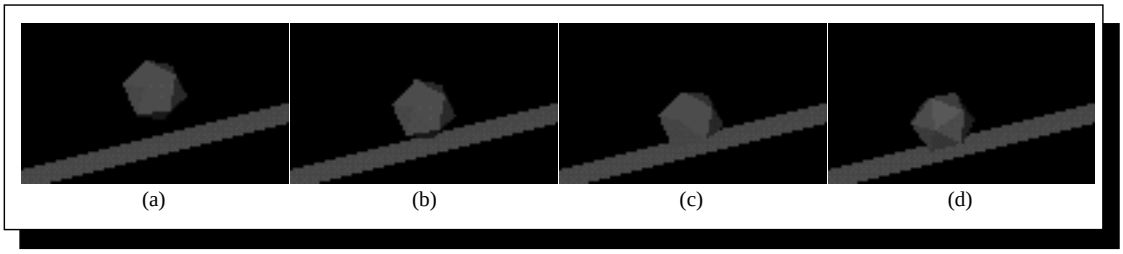

Figure 6.3. *Glissement d'un corps le long d'un plan*

6.2.4. Etude fine des collisions

La collision point rigide sur facette déformable se doit d'être testée dans toutes les configurations de mouvement possibles. On a donc créer une configuration où elle intervient systématiquement. On considère une facette isolée déformable en chute libre entrant en collision avec un sommet d'une facette rigide. La simulation sur la *figure 6.4* montre le résultat escompté, dont la théorie a été présentée dans le chapitre précédent.

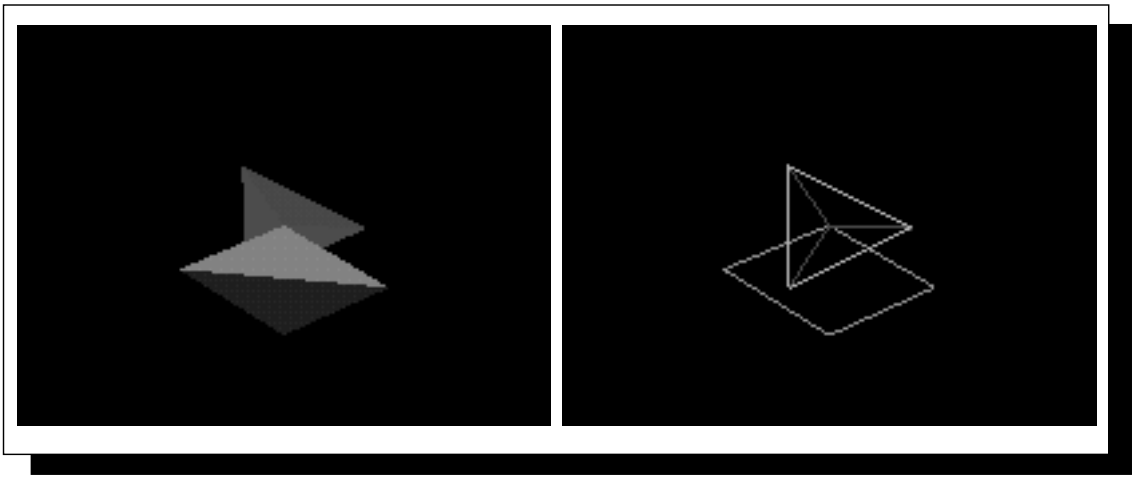

Figure 6.4. *Collision point solide sur facette déformable*

La collision arête/arête est délicate à vérifier car elle n'intervient qu'occasionnellement. Nous avons donc créer des configurations types où ce genre de collisions intervient systématiquement. Par exemple le cas de la simulation présentée sur la *figure 6.5* est celui d'un simple triangle déformable entrant en collision avec le bord d'un plan. On voit sur la figure en fil de fer *(a)*, la nouvelle liaison créée et divisant le triangle en deux.

Enfin la *figure 6.6* montre la subdivision créée lorsque deux collisions arête/arête sont détectée simultanément.

6.2.5. Un cas simple de manipulation

Nous avons crée un fichier décrivant une situation simple où l'utilisateur peut manipuler le corps déformable. La scène contient alors deux objets rigides, un plan support et une pince, et un objet déformable représentant l'organe. Au départ, le corps tombe en chute libre sur le plan et atteint une position stable. L'utilisateur peut alors manipuler la pince pour déformer l'organe. Cette

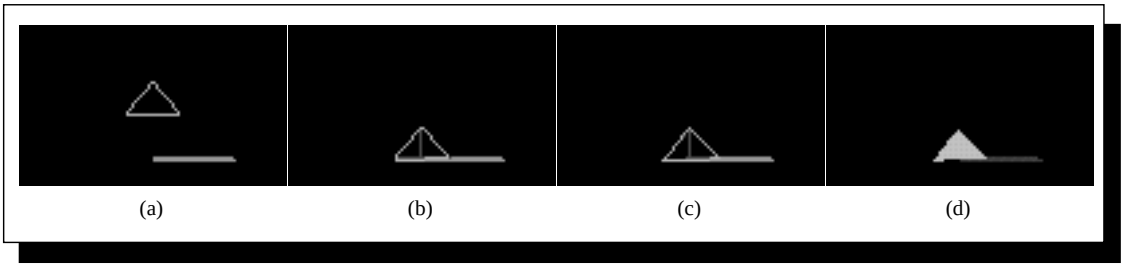

Figure 6.5. *Exemple de collision arête/arête*

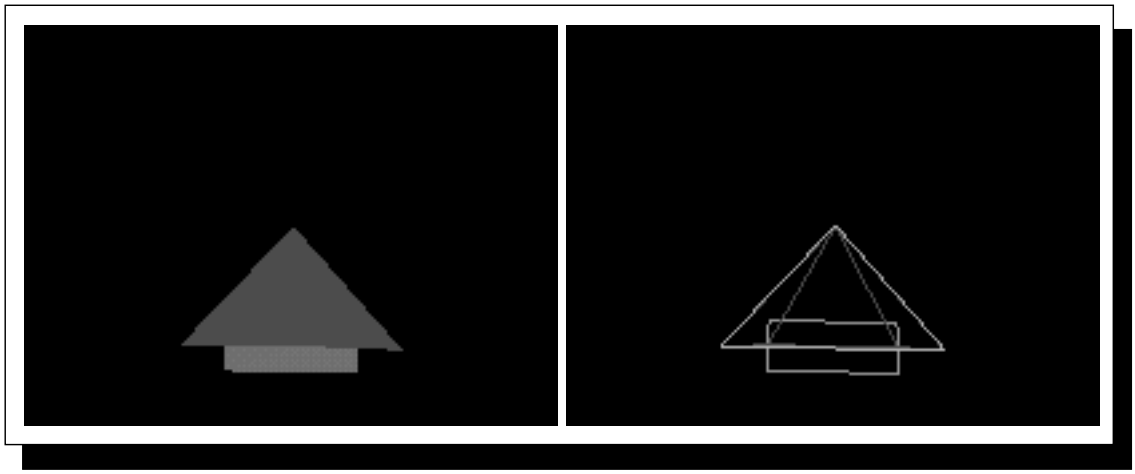

Figure 6.6. *Multiples collisions arête/arête simultanées*

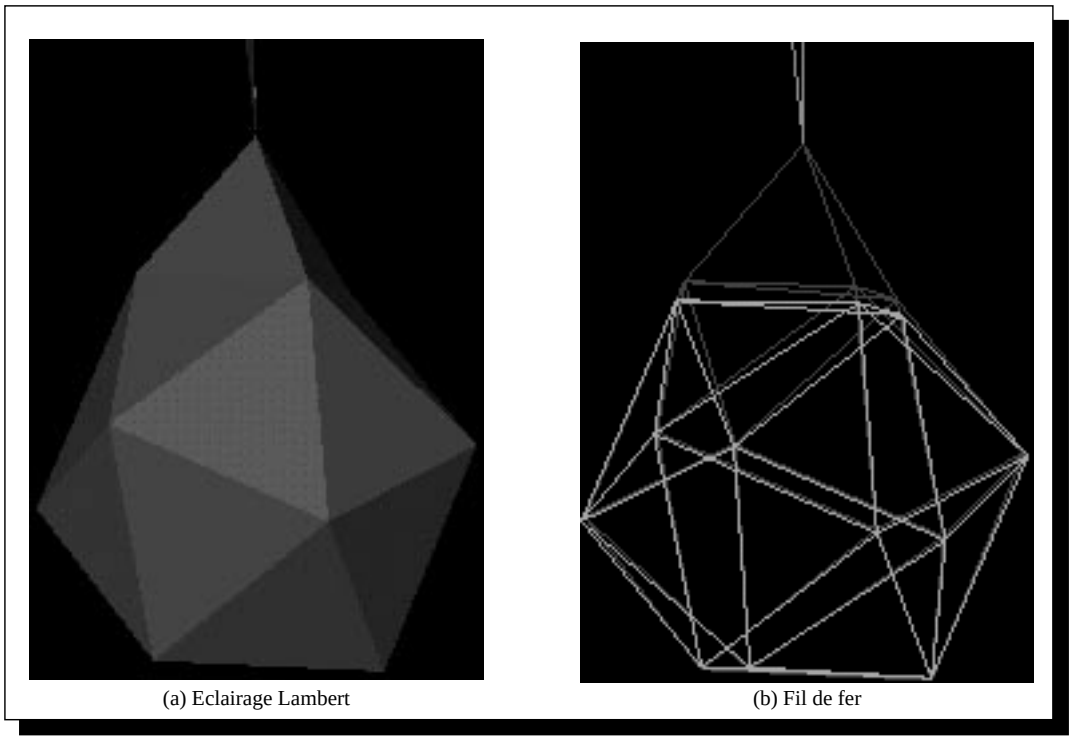

Figure 6.7. *Exemple d'un corps déformable tenu par un outil*
Sur (b), le corps déformable est sombre, la composante rigide est claire

simulation a également pour but de démontrer le résultat obtenu par découpe. La *figure 6.7* montre la situation où le manipulateur tire l'organe avec une outil.

6.3. Mesures des performances

Les exemples que nous avons détaillés dans la partie précédente ont pour but de démontrer que notre modèle est implanté et donne des résultats pertinents. Nous nous intéressons maintenant à l'aspect interactivité de notre modèle. Pour cela, nous mesurons le temps d'exécution des simulations, en étudiant l'impact du nombre de nœuds et de liaisons du corps déformables, le coût des collisions...

Les mesures de performances ont été effectuées sur diverses architectures. Nous ne cherchons pas ici à comparer ces architectures, mais à déterminer les contraintes liées à la machine, ainsi que le type de matériel actuellement nécessaire à la simulation. Les principaux essais ont été réalisés sur une station *Silicon Graphics INDIGO 2*, possédant un processeur *R4600* tournant à *133Mhz* et un accélérateur *XZ*. Par ailleurs, un *PC Intergraph* modèle *TDZ300*, basé sur le *Pentium Pro 200Mhz* et muni d'une carte accélératrice *GLZ5* compatible *OpenGL* a été employé. Les mesures obtenues sur le *PC Intergraph* sont très intéressantes car elles représentent la puissance qu'auront les micro-ordinateurs personnels dans les mois à venir. Enfin, nous avons pu faire quelques tests de rapidité sur une station *Onyx Infinite Reality*, équipée d'un processeur *R10000* à 194Mhz (attention, les mesures actuellement données ont été effectuées avec un programme sans optimisation pour le *R10000* et sont donc à considérer comme des bornes inférieures).

Précisons que l'ensemble des mesures effectuées ici ne tiennent pas compte de l'affichage, pour des raisons d'interprétation des résultats et des raisons techniques. Tout d'abord il est peu significatif de comparer un programme incorporant à la fois du calcul réel et de la génération d'image sur deux architectures. On ne peut pas dans ces conditions, interpréter les résultats, car on ne sait pas si ce sont les calculs ou la génération de l'image qui conditionne le résultat. Par ailleurs, d'un point de vue technique, les étapes d'affichage ne sont pas traitées de la même façon suivant l'architecture. Par exemple, l'affichage peut parfois être exécuté dans un processus séparé, auquel cas, ce temps n'est pas mesuré dans les tests. Si au contraire l'affichage est traité dans le même processus que le moteur de la simulation, les temps d'affichage apparaîtront dans les mesures, d'où une faiblesse apparente de la machine.

6.3.1. Influence du nombre de points et de liaisons

On choisit trois cas de simulation, avec un nombre de masses et de ressorts croissants. Les mesures sont récapitulées dans le tableau de la *figure 6.8*. La simulation de type 1 est celle d'un corps comportant 12 nœuds, 30 liaisons et 20 faces. Le type 2 correspond à un corps déformable de 120 nœuds, 346 liaisons et 228 faces. Enfin le type 3 représente un corps de 258 nœuds, 768 liaisons et 512 faces. Les trois simulations consistent à laisser tomber le corps sur un plan horizontal. Afin de pouvoir comparer les temps, la procédure d'accélération des collisions a été supprimée lors de ces mesures.

	Indigo2	PC	Onyx
Type 1	630Hz	880Hz	2500Hz
Type 2	36Hz	60Hz	180Hz
Type 3	18Hz	25Hz	60Hz

Figure 6.8. *Fréquence de simulation en fonction de la complexité du modèle*

6.3.2. Etude du temps de calcul dédié aux collisions

On s'intéresse ici à l'impact du temps de détection de collisions sur le temps total du calcul. Avec les mêmes scènes qu'au paragraphe précédent, on considère sur la *figure 6.9* la boucle de simulation sans détection de collision. On mesure la nouvelle fréquence de simulation obtenue. On calcule par ailleurs le pourcentage occupé par la collision, en considérant le rapport de ces mesures avec celles de *figure 6.8*. On obtient un chiffre très intéressant montrant que sur nos exemples de complexité croissante, la détection des collisions demande 90% du temps de calcul.

	Indigo2	PC	Onyx
Type 1	3500Hz/82%	10680Hz/91%	14000Hz/82%
Type 2	330Hz/89%	1300Hz/95%	1450Hz/87%
Type 3	145Hz/87%	610Hz/95%	650Hz/90%

Figure 6.9. *Fréquence de simulation sans détection de collision*

Enfin, sur la *figure 6.10* on évalue l'impact de notre routine d'accélération de la collision. Les chiffres obtenus ne sont pas à considérer entre les types (les objets étant de géométrie, de dimension et de propriétés physiques non équivalentes) mais par rapport aux chiffres donnés pour ce type dans les deux tableaux précédents.

	Indigo2	PC	Onyx
Type 1	640Hz	900Hz	2600Hz
Type 2	40Hz	61Hz	180Hz
Type 3	24Hz	30Hz	67Hz

Figure 6.10. *Fréquence de simulation avec la routine d'accélération des collisions*

6.3.3. Bilan

Sans affichage, les mesures effectuées montrent que sur des machines moyenne gamme, le temps réel est obtenu avec plusieurs centaines de points. Cependant, ces temps montrent que le simulateur n'est pas synchrone, sauf en utilisant le processeur *R10000*. En effet, le pas de simulation choisi étant de *1ms*, il faut obtenir une fréquence d'au moins *1000Hz* pour atteindre la synchronicité. On remarque en outre que les calculs liés à la dynamique ne représentent que *10%* environ du temps total, le reste étant occupé par le traitement des collisions. Il est donc clair que la synchronicité ne pourra être atteinte qu'avec une accélération considérable de la détection de collision.

D'un point de vue pratique, la simulation obtenue est interactive et l'utilisateur interagit avec le modèle de façon naturelle, même lorsque la simulation est affichée régulièrement. Cependant, pour des raisons liées à l'environnement de développement, une image est affichée à chaque pas de simulation (ou tous les *n* pas de simulation, *n* étant choisi). En toute rigueur, il faudrait que l'affichage soit géré par un processus indépendant tournant à une fréquence de *25Hz* par exemple. Cependant, cette réalisation se trouve limitée par le système d'exploitation.

En effet, quand bien même la synchronicité pourrait être assurée avec la puissance disponible, le temps du processeur fourni à l'application n'est généralement pas contrôlable, sur les systèmes d'exploitation multi-tâches comme *Unix* ou *Windows NT*. Ainsi, si on veut pouvoir assurer la

synchronicité, le simulateur doit être porté sur des systèmes d'exploitation temps-réel. Seuls ces systèmes offrent au programmeur l'opportunité de contrôler le temps offert à chaque tâche.

6.4. Applications du modèle : comparaison avec les éléments finis

La mise en œuvre nous a permis de définir les environnements où le modèle est le mieux adapté. Nous montrons dans cette partie les critères qui, de notre points de vue, sont décisifs pour le choix d'un modèle tel que le notre ou au contraire ce qui fera préférer un modèle éléments finis. Ce modèle est en effet, actuellement le plus utilisé dans les simulateurs chirurgicaux.

Tout d'abord, les calculs de la dynamique purs sont très rapides. Les temps de calcul sont inférieurs à la milliseconde là où les éléments finis offrent des temps supérieurs à la seconde. On peut conclure que les éléments finis sont plutôt adaptés aux cas de simulations d'opérations sur un organe unique. Dans ce cas, seul cet organe est découpé en éléments finis, et l'algorithme de résolution ne s'applique qu'aux équations gouvernant le comportement de cet organe. Le résultat obtenu, sous réserve d'employer quelques accélérations pour le rendre temps réel, permettra de restituer fidèlement le comportement de l'organe.

Par contre, dans un contexte comme la cœlioscopie, où de nombreux organes sont susceptibles d'être opérés, il n'est pas concevable de devoir modéliser par éléments finis chaque organe. Outre la lourdeur du temps de calcul associé, il faut prendre en considération le coût mémoire exigé par le stockage des matrices associées à chaque organe. En outre, le résultat d'un calcul élément fini statique est parfois loin d'être suffisant pour certains corps déformables. Par exemple, les trompes des organes génitaux féminins sont très déformables et peuvent être comparées, en première approximation, à des cordes élastiques. Sans dynamique, les éléments finis ne fournissent qu'une position de repos du corps déformable. Or ici, elles peuvent être multiples. De plus, Si on suppose qu'une trompe est tirée par un outil et brusquement relâchée, la chute d'un tel organe n'est pas restituée. Au contraire, un modèle basé sur les ressorts en dynamique simule naturellement un tel phénomène.

6.5. Perspectives

La mise en œuvre du modèle n'est pas complètement achevée. Nous dressons une liste des problèmes restants ainsi que les solutions que nous envisageons pour les résoudre.

- le calcul de la réponse lors du traitement des collisions entre arêtes et lorsqu'une face du corps est déformée reste problématique. Une méthode se basant sur des demi espaces de déplacements autorisés a été implantée mais ne fournit pas encore des résultats attendus dans tous les cas de collisions que nous avons testés. D'autres critères géométriques et cinématiques plus précis devraient donner de meilleurs résultats ;
- la collision entre corps déformables peut être détectée grâce aux routines de collisions que nous avons proposées. Par contre, la réponse à ce genre de collision est plus délicate à résoudre car elle nécessite de prendre en compte des déformations des deux corps entrant en collision. Chaque corps absorbe une partie de l'énergie de la collision en se déformant, la part absorbée par chaque corps doit être évaluée en fonction des propriétés mécaniques des deux corps, de la violence du choc, etc ;
- la subdivision adaptative, en réduisant la valeur des masses et raideurs, modifie la fréquence propre du système. Le niveau de subdivision doit être borné ce qui crée des imperfections dans le traitement des collisions au dernier niveau et enfin la détection de fin de déformation afin de retrouver l'état avant que la subdivision ne soit implantée ;
- le processus de découpages et de déchirures est incomplet. Nous envisageons d'ajouter des faces représentant les parties volumiques devenant visibles ;

- les calculs d'auto-collision sont nécessaires lors des découpes car l'influence de la composante rigide devient faible. Or c'est elle qui empêche, dans une certaine mesure, les autocollisions ;
- l'affichage de la structure : les modèles géométriques et mécaniques sont les mêmes, le nombre de points pour l'affichage n'est donc pas suffisant d'où un aspect anguleux. Nous envisageons une modification de notre modèle permettant de multiplier le nombre de nœuds géométriques. Cependant, un tel choix entraîne un surcoût de la détection de collisions ;
- la modélisation géométrique a une incidence sur la stabilité du système : si l'objet présente des pôles, le modèle est particulièrement instable en ces points. Cependant, de nombreux objets issus des modeleurs possèdent des pôles. Actuellement aucune solution n'est envisagée, nous ne simulons que les objets sans pôle ;
- nous continuons de chercher à contrôler la stabilité du système. La première idée consiste à borner les distances, mais on obtient alors des oscillations du modèle. Une autre solution consiste à détecter les augmentations de l'énergie totale due à la divergence. Cette solution ne nous semble pas intéressante car notre modèle de collision n'est pas basé sur des considérations énergétiques : lors d'une collision, l'énergie totale du système peut augmenter car le processus ne respecte pas complètement les lois physiques. On peut également s'intéresser aux méthodes d'intégration implicites, mais elles sont très coûteuses. Enfin, d'autres solutions issues de l'automatique sont également possibles mais n'ont pas été étudiées pour l'instant ;
- notre modèle est dynamique, la synchronicité doit être assurée. Notre programme doit donc être optimisé ou des architectures plus puissantes doivent être employées.

Conclusion

Le modèle que nous avons proposé a fait l'objet d'un implantation sur machine. Grâce à cette mise en œuvre, nous avons montré que le modèle se comporte de façon attendue. Etant ouvert et se basant sur la physique, nous avons pu vérifier qu'il se comporte naturellement, même en dehors des situations types ayant servi de tests lors de la conception. Cette adaptabilité à la situation contribue à la robustesse du modèle. Malheureusement, les problèmes de stabilité dus à l'intégration en fonction du temps, font diverger les équations très facilement ce qui se traduit par une *explosion* du modèle. Le modèle reste donc relativement instable. Par exemple, un mouvement rapide du manipulateur fait diverger les calculs.

Les mesures des performances montrent que le modèle n'est synchrone à *1000Hz* pour quelques dizaines de points que si l'on dispose d'une machine très haut de gamme. Il est possible d'obtenir des gains en performance en optimisant le programme et en l'exploitant pleinement sur une architecture dédiée aux calculs vectoriels (une grande partie des opérations sont effectuées sur des vecteurs). Par ailleurs, il n'est pas exclu de porter le programme sur une architecture parallèle, mais cela demande une étude particulière visant à montrer l'intérêt de la parallélisation pour notre simulateur : le niveau de parallélisation (objet, groupe de primitives, primitives) ainsi que la répartition des données pour les collisions sont à déterminer.

De façon prioritaire, il faut trouver une technique permettant de limiter le nombre de collisions que l'on calcule. Pour cela, il manque à notre simulation une étape permettant de déterminer rapidement quelles sont les primitives à tester. C'est l'étape intermédiaire située entre la détection de collisions au niveau objet et au niveau primitive.

Sous certaines contraintes, notre modèle est toutefois exploitable tel qu'il est. En effet, les problèmes restants ne gênent pas son utilisation dans des applications médicales, car les mouvements sont très mesurés. La fréquence retenue pour la simulation peut également être réduite à 200 ou 300Hz. Nous prévoyons ainsi, durant le premier semestre 1997, de l'implanter dans le simulateur de cœlioscopie que nous avons décrit au paragraphe *2.2.3*.

7 Conclusion - Discussion

Nos travaux se placent dans le contexte de la simulation chirurgicale à but pédagogique. Les études sont effectuées dans le cadre du Groupement d'Intérêt Scientifique Lillois sur les simulateurs médicaux. Ces travaux répondent à une demande du corps médical qui cherche des outils pour se former de façon rapide et efficace aux nouvelles techniques thérapeutiques. Les projets en cours appartiennent à des domaines aussi variés que l'ophtalmologie, la cœlioscopie, l'écho-endoscopie et l'amniocentèse.

Les simulateurs médicaux font l'objet d'études dans le monde depuis le milieu des années 1980. Au LIFL, les premières études remontent à 1993. La première partie de nos travaux a été consacrée à l'étude d'un simulateur de photocoagulation par laser. Ce simulateur est achevé depuis février 1995 et a fait l'objet d'une étude en site clinique dont les premiers résultats sont très prometteurs.

Les simulateurs nécessitent la prise en compte des interactions mécaniques entre les organes et les instruments, ce qui fait leur spécificité et explique les difficultés de conception et de réalisation. Afin de simuler les interactions mécaniques de façon réaliste, il faut s'appuyer sur les lois de la physique et plus particulièrement sur la mécanique des corps déformables. Les études en animation ont débouché sur plusieurs modèles de corps déformables, mais peu d'entre eux sont utilisables dans un contexte temps-réel. Les deux modèles généralement utilisés pour les simulateurs médicaux sont les éléments finis et les maillages masses/ressorts.

En outre, simuler une opération chirurgicale nécessite de prendre en compte les interactions entre les instruments et les organes. Le traitement des collisions regroupe les trois étapes que sont la détection des collisions, l'accélération de cette détection ainsi que le calcul de la réponse. En général, dans les simulations physiques, la phase correspondant aux traitements des collisions accapare la plus grande partie du temps de calcul (90%). La phase d'accélération est donc une étape cruciale.

Dans un contexte de simulation chirurgicale, les interactions extrêmes tels que les découpes et déchirures doivent être prises en compte. Le modèle de corps déformable doit donc supporter les modifications de structure.

Notre choix s'est porté sur la modélisation de corps par maillage masse/ressorts. Afin de rendre notre modèle compatible avec le temps réel, nous avons cherché à réduire le nombre de nœuds du maillage ainsi que le nombre de liaisons entre ces nœuds. Le maillage obtenu étant relativement grossier, un procédé de subdivision est employé pour affiner le maillage lors des contacts. Le modèle de collisions employé traite tous les cas de collisions entre un objet polygonal rigide et un objet polygonal déformable. Des solutions encore incomplètes mais prometteuses pour le traitement des collision et la prise en compte des découpes et déchirements sont proposées.

Ce modèle a fait l'objet d'une mise en œuvre qui démontre qu'il est compatible avec le temps réel. La détection des collisions est cruciale car elle nécessite 90% du temps de calcul. Une première solution pour accélérer la détection a été proposée et devra, dans la suite, être complétée par un processus permettant d'affiner la recherche des primitives entrant en collision.

Même s'il reste des problèmes à résoudre, ce modèle sera employé pour construire un simulateur de cœlioscopie.

De l’avenir des simulateurs médicaux...

La recherche sur les simulateurs médicaux a encore de grands jours devant elle. Les prototypes en sont à leurs balbutiements, ce qui se comprend aisément lorsque l’on compare l’âge de la simulation médicale à celui des simulateurs de vol. Cependant, alors que les gouvernements commencent à considérer ce formidable outil d’apprentissage, on peut penser que le début du XXIe siècle verra l’avènement de la simulation pour la chirurgie. Le marché n’est pour l’instant que frémissant. Les idées font néanmoins leur chemin dans le monde médical. La simulation est présentée comme la prochaine révolution de la médecine. Partout dans le monde, des équipes universitaires mais aussi des entreprises se lancent dans cette voie et il ne faut pas douter que les résultats à venir permettront d’opérer sur des patients virtuels.

La puissance demandée par la simulation requiert l’emploi de machines haut de gamme. Il est à espérer que le coût de ces machines devienne suffisamment bas pour que les services hospitaliers puissent y avoir accès. On peut prédire que sous la pression de l’opinion publique, des patients et des gouvernements, le recours à l’apprentissage par simulateur deviendra une étape incontournable dans la formation des médecins.

Cet ouvrage représente notre contribution à la construction de simulateurs chirurgicaux. Il aborde des domaines très variés : nous nous sommes efforcés de présenter, peut-être parfois rapidement, les notions nécessaires. Nous sommes convaincus que c’est grâce à de tels projets multidisciplinaires que la recherche avance le plus.

Dans l’antiquité, les grecs pensaient que la géométrie était la science du philosophe, la mère de toute connaissance. Plus tard, Descartes ajouta que “*toute la philosophie est comme un arbre dont les racines sont la physique, et les branches qui sortent de ce tronc sont toutes les autres sciences qui se réduisent à trois principales à savoir la médecine, la mécanique et la morale*”. Dans cet ouvrage, nous vous avons parlé de géométrie, de physique, de mécanique, ainsi qu’un peu de médecine et de morale. A n’en pas douter, la simulation chirurgicale est un pas considérable de l’Homme vers la sagesse !

Annexe I. Notations mathématiques

Dans cet ouvrage un certain nombre de notations sont employées :

- $\Re$: ensemble des réels ;
- $\vec{u}$: désigne le vecteur u ;
- $\overline{\overline{M}}$: désigne la matrice M ;
- $\dot{q}$: désigne la dérivée exacte de la fonction q par rapport au temps. Le nombre de points représente le nombre de dérivations ;
- $\frac{da}{db}$: désigne la dérivée totale de la fonction a par rapport à la variable b ;
- $\frac{\partial a}{\partial b}$: désigne la dérivée partielle de la fonction a par rapport à la variable b ;
- $\overrightarrow{grad}(f)$: désigne le vecteur gradient de la fonction f.

Annexe II. Les quaternions

Le but de cette annexe n'est pas de faire une présentation formelle des quaternions, mais plutôt de les observer sous un angle pratique et purement calculatoire. Les quaternions sont un moyen suffisant et condensé de traiter le problème des rotations dans l'espace à 3 dimensions. Ils ont été introduits par Sir William Rowan Hamilton en octobre 1843, en tant qu'extensions des nombres complexes. On trouvera dans [Sho85] et [Can86] un résumé de la théorie des quaternions, ainsi qu'une étude exhaustive des rotations en divers dimensions dans [Bai96].

II.I. Définition

II.I.I. Structure d'algèbre

On appelle quaternion un élément de $\Re^4$. On note Q l'ensemble des quaternions. On décompose un quaternion en deux parties, la première dite *scalaire*, composée de la première composante, la seconde dite *vectorielle* constituée des trois composantes suivantes. Ainsi si $q \in Q$ on peut écrire $q = (w,x,y,z)$ ou $q = (w,\vec{x})$. Hamilton a proposé une autre notation, inspirée des complexes :

$$q = w + xi + yj + zk \text{ avec } i^2 = j^2 = k^2 = ijk = -1$$

L'ensemble des quaternions étant isomorphe à $\Re^4$, il est muni de l'*addition* de $\Re^4$, où l'on somme les composantes. Si $q_1 = \left(w_1,\vec{x}_1\right)$ et $q_2 = \left(w_2,\vec{x}_2\right)$ alors :

$$q_1 + q_2 = \left(w_1 + w_2,\vec{x}_1 + \vec{x}_2\right)$$

L'élément neutre est le quaternion nul (toutes les composantes à 0) et l'opposé est l'opposé des composantes.

De la même façon, on peut le multiplier par un réel quelconque :

$$\alpha q = (\alpha w, \alpha\vec{x}) \text{ avec } \alpha \in \Re$$

L'ensemble est également muni d'une *multiplication* non commutative définie ainsi :

$$q_1 q_2 = \left(w_1 w_2 - \vec{x}_1 \cdot \vec{x}_2, w_1\vec{x}_2 + w_2\vec{x}_1 + \vec{x}_1 \wedge \vec{x}_2\right)$$

L'élément neutre de la multiplication est le quaternion de partie réelle 1 et de partie vectorielle nulle.

Toutes ces opérations procurent à l'ensemble des quaternions une structure d'*algèbre*.

Par ailleurs, afin de ne traiter que la *partie scalaire* d'un quaternion, on notera :

$$[q] = w$$

Par abus de langage et commodité d'écriture, tout vecteur est considéré comme un quaternion de partie scalaire nulle et tout scalaire comme un quaternion de partie vectorielle nulle. Par exemple, on pourra multiplier un quaternion par un vecteur sans préciser qu'auparavant, ce vecteur est projeté dans l'espace des quaternions. De la même façon, la multiplication d'un quaternion par un réel apparaît comme la multiplication entre deux quaternions, dont l'un a une partie vectorielle nulle.

II.I.II. Norme du quaternion

On appelle conjugué d'un quaternion, le quaternion de même partie scalaire, mais de partie vectorielle opposée. On le note :

$$\bar{q} = (w, -\vec{x})$$

La partie scalaire apparaît donc comme :

$$[q] = \frac{q + \bar{q}}{2}$$

On remarque que la partie vectorielle du produit entre un quaternion et son conjugué est nulle. Le scalaire obtenu est appelé carre de la norme du quaternion. On définie donc la norme comme :

$$\|q\| = \sqrt{q\bar{q}} = \sqrt{w^2 + \vec{x} \cdot \vec{x}}$$

La norme du quaternion correspond à la norme euclidienne dans $\Re^4$. On appelle *quaternion unitaire* tout quaternion de norme 1. Notons que cette définition de la norme nous permet d'exprimer l'inverse d'un quaternion pour la multiplication. En effet, comme :

$$\frac{q\bar{q}}{\|q\|^2} = 1$$

On a :

$$q^{-1} = \frac{\bar{q}}{\|q\|^2}$$

II.II. Représentation des rotations par des quaternions

II.II.I. Quaternion de rotations

Une rotation dans l'espace est définie par un angle et un axe orienté. On définit par $(\theta, \vec{u})$ la rotation d'axe $\vec{u}$ et d'angle θ. On peut lui associer le quaternion unitaire défini par :

$$\left(\cos\frac{\theta}{2}, \sin\frac{\theta}{2}\vec{u}\right)$$

Ainsi, la composition de deux rotations, la première de quaternion q_1 suivi de la seconde de quaternion q_2 est égale au produit $q_2 q_1$. L'image d'un vecteur la rotation de quaternion q s'écrit :

$$\vec{v}^* = q\vec{v}\bar{q}$$

II.II.II. Equivalence entre matrices et quaternions

La représentation classique des rotations 3D en synthèse d'images est la matrice. Cette dernière n'est pas véritablement pratique. D'une part, il n'est pas facile de construire la matrice d'une rotation quelconque alors que le quaternion associé est très facilement obtenu. De plus, la composition de rotations exige alors une multiplication matricielle dont le coût est de 3 multiplications et deux additions pour chaque terme de la matrice soit 27 multiplications et 18 additions. En comparaison, le coût d'une multiplication entre quaternions est de 16 multiplications et 12 additions. Il est donc plus efficace de multiplier deux quaternions que deux matrices.

Par contre, le calcul de l'image d'un vecteur par les matrices est de 9 multiplications et 6 additions. Par les quaternions, le coût est celui de deux multiplications de quaternions, soit 32 multiplications et 24 additions. On peut alléger ce coût en tenant compte du fait que la partie réelle du vecteur est nulle, mais cela ne fait que supprimer 4 multiplications et 4 additions. Pour calculer l'image d'un vecteur, il est donc plus intéressant d'employer les matrices.

En conclusion, l'usage des quaternions est à recommander pour la définition des rotations et leurs compositions. De plus, il est facile de reconstituer l'axe et l'angle de la rotation à partir du quaternion (cela nécessite un arc-cosinus et un sinus ou une racine carré, suivant le calcul choisi). Cela est très pratique, car les librairies graphiques 3D actuelles acceptent l'entrée de rotations quelconques. Il est ainsi plus pratique de fournir l'axe et l'angle de rotation, cela permet d'être indépendant de la représentation des matrices interne à la librairie.

Par contre, sitôt que l'on veut calculer l'image d'une série de points ou de vecteurs, il vaut mieux dans ce cas revenir à une expression matricielle des rotations. On cherche alors à exprimer la matrice représentant la même rotation qu'un quaternion $q = (w,x,y,z)$. Cette matrice est :

$$\begin{bmatrix} 1-2y^2-2z^2 & 2xy-2wz & 2xz+2wy \\ 2xy+2wz & 1-2x^2-2z^2 & 2yz-2wx \\ 2xz-2wy & 2yz+2wx & 1-2x^2-2z^2 \end{bmatrix}$$

Un algorithme optimisé pour le calcul de cette matrice est proposé dans [Sho85] qui permet de n'avoir recours qu'à 9 multiplications et 15 additions.

Annexe III. Les formats de fichiers employés

III.I. Fichier *SCN*

La situation simulé par notre moteur est définie dans un fichier de description de la simulation. Ce fichier que l'on appellera fichier *scène* (par analogie avec les fichiers de description d'environnement employés en synthèse d'images). Son extension est *.SCN*. Sa structure globale est une suite de définitions d'objets rigides ou déformables, données dans un ordre quelconque. Les définitions sont bâties sur le modèle suivant :

```
DEFORM nom1
    attributvaleurs
    attributvaleurs
    (...)

SOLIDE nom2
    attributvaleurs
    attributvaleurs
    (...)

SOLIDE nom3
    attributvaleurs
    attributvaleurs
    (...)

DEFORM nom4
    attributvaleurs
    attributvaleurs
    (...)
...
```

Chaque objet est défini par un nom faisant référence à un fichier (*nom.OJT*) décrivant la géométrie de l'objet. Les valeurs nom1, nom2... peuvent donc être identiques, si deux objets ont la même géométrie. Ce fichier est décrit dans le paragraphe suivant. Les attributs communs aux deux types d'objet sont :

- *scale val* : mise à l'échelle de facteur *val* ;
- *trans tx ty tz* : translation de vecteur *(tx,ty,tz)* ;
- *rot rx ry rz* : rotations (non implanté).

Dans le cas des corps déformables, les attributs possibles sont :

- *masse val :* masse totale du corps déformable ;
- *densite val* : part de la composante déformable dans la masse totale ;
- *raidvois val* : valeur de la raideur entre voisins dans le maillage surfacique ;
- *raidrep val* : valeur de la raideur des ressorts entre les composantes déformables et rigides ;
- *amortbary val* : coefficient de viscosité de l'air pour la composante rigide ;
- *amortnoeud val* : coefficient de viscosité de l'air pour les masses du maillage ;
- *amortlia val* : coefficient d'amortissement des ressorts entre les masses.

Pour les corps solides les paramètres sont :

- *frottements val* : coefficient de la loi de Coulomb ;
- *collis val1 val 2* : coefficient de friction et d'élasticité pour le calcul de la vitesse après collision ;

- *epaisseur val* : épaisseur des faces pour déterminer l'erreur numérique autorisée lors des collisions.

III.II. Fichier OJT

Un format de fichier binaire a été employé permettant de définir une grande variété de types de données, sans nécessiter de longs temps de chargement. Ce format se veut ouvert et permet par exemple de donner des normales aux nœuds différentes suivant la facette, ou par exemple donner des attributs de couleurs aux nœuds, sans normale.

```
Format de Fichier OBJET v1.2


Type   NBOctets      But
CHAR    4x1      Chaine "OBJT" : Caracteristique du fichier
ULONG   4        Nombre de points
ULONG   4        Nombre de triangles
ULONG   4        Nombre d'aretes
FLOAT   3x4      Position du centre de masse
FLOAT   3x4      Valeur R, V, B, entre 0 et 1 de l'ambiant des faces "front"
FLOAT   3x4      "    "    "      "    du diffu des faces "front"
FLOAT   3x4      "    "    "      "    du speculaire des faces "front"
FLOAT   4        "    entre 0 et 1 du coef speculaire des face "front"
FLOAT   3x4      Valeur R, V, B entre 0 et 1 de l'ambiant des faces "back"
FLOAT   3x4      "    "    "      "    du diffu des faces "back"
FLOAT   3x4      "    "    "      "    du speculaire des faces "back"
FLOAT   4        Valeur entre 0 et 1 du coef speculaire des faces "back"

ULONG   4        Caracteristiques des points :
                       bit 0: si 1, les points ont une normale
                       bit 1: si 1, les points ont une couleur

[   -------- Liste des noeuds --------
FLOAT   3x4      Position x,y,z des noeuds
FLOAT   3x4      Valeur des normales au points si besoin
FLOAT   3x4      Valeur R,V et B de la couleur au point si besoin
]

[   --------- Liste des facettes -----
ULONG   3x4      Numero de noeuds des sommets du triangle oriente
                 (commence par 0)
USHORT  2        Type de maillage :
                       0 -> Fin de liste
                       1 -> Triangle simple ou Fin de maillage
                       2 -> Triangle strip
                       3 -> Triangle Fan
...
USHORT 2         Caracteristiques des normales donnees ensuite
                       bit 0 -> Si 1, Normale de la surface
                       bit 1 -> Si 1, Normale pour chaque noeud du triangle
FLOAT   3x4      Normale au triangle si besoin
FLOAT   3x3x4    Normale en chaque noeud du triangle si besoin
]

[   ------ Liste des aretes --------
ULONG   2x4      Extremites de l'arete en num de noeud (Num1<Num2)
```

Bibliographie

Afin de clarifier cette bibliographie, nous avons choisi de la séparer en donnant d'une part les application dédiées au domaine médical et d'autre part celles dédiées à l'animation et à la synthèse d'images en général.

Applications médicales

[Bai96] Bainville E., "Modélisation Géométrique et Dynamique d'un Geste Chirurgical" *Thèse de doctorat*, Université Joseph Fourier, Grenoble, mars 1996.

[BC96] Bro-Nielsen, M., et Cotin, S., "Real-time Volumetric Deformable Models for Surgery Simualtion using Finite Elements and Condensation" *Proceedings of the EUROGRAPHICS'96 conference, Computer Graphics Forum*, 15, 3, Poitiers, 28-30 Août 1996, pp 57-66.

[BCCL95] Bady, J., Couvet, S., Coville, M., Levaillant, J.M., Mares, P., et Zarca, D., "Simulateur d'Entraînement aux Gestes Opératoires : l'Exemple de l'Endoscopie Gynécologique" *Gynécologie - Obstétrique pratique*, 80, décembre 1995, pp 1-4.

[BFO92] Bajura, M., Fuchs, H., et Ohbuchi, R., "Merging Virtual Objects with the Real World : Seeing Ultrasound Imagery within the Patient" *SIGGRAPH'92 Conference Proceedings, Computer Graphics*, 26, 2, Chicago, 26-31 juillet 1992, pp 203-210.

[BG93a] Boissonat J.D., et Geiger B., "Modélisation 3D et Simulation de Mouvements" *Premières Journées AFIG-GROPLAN*, Labri Bordeaux 1993, pp 187-191.

[BG93b] Boissonat, J.D., et Geiger, B.,"3D Simulation of Delivery" *Proceedings of Visualization'93*, pp 416-419.

[BL93] Bady, J., et Levaillant, J.M., "Simulateur de Geste Opératoires et Endoscopie Gynécologique" *Gynécologie Obstétrique*, 13 (53), 1993.

[Bro95] Bro-Nielsen, M., "Modelling Elasticity in Solids Using Active Cubes - Application to Simulated Operations", *Proceedings of the CVRMed'95 Conference*, Nice, 3-5 avril 1995, pp 535-541.

[CDBA96] Cotin, S., Delingette, H., Bro-Nielsen, M., Ayache, N., Clément, J. M., Tasseti, V., et Marescaux, J., "Geometric and Physical Representations for a Simulator of Hepatic Surgery" *Proceedings of the Medecine Meets Virtual Reality IV*, San Diego, California, 17-20 janvier 1996.

[CDCT96a] Cotin, S., Delingette, H., Clémet, J.M., Tassetti, V., Marescaux, J., et Ayache, N., "Simulation de Chirugie Hepatique avec Système de Retour de Forces" *Les Interfaces des Mondes Réels et Virtuels*, Montpellier, France, 1996.

[CDCT96b] Cotin, S., Delingette, H., Clémet, J.M., Tassetti, V., Marescaux, J., et Ayache, N., "Volumetric Deformable Models for Simulation of Laparoscopic Surgery" *Computer Assisted Radiology*, Paris, France, 1996, pp 793-798.

[CDKM94] Chaillou, C., Dubois, P., Karpf , S., Meseure, P., et Rouland, J.F., *Dispositif et Procédé de Simulation d'un Examen ou d'une Opération Chirurgicale Effectuée sur un Organe Simulé*, Dépôt de brevet no 9405487.

[CDTL92] Cinquin, P., Demongeot, J., Troccaz, J., Lavallée, S., Champleboux, G., Brunie, L., Leitner, F., Sautot, P., Mazier, B., Perez, A., Djaid, M., Fortin, T., Chenin, M., et Chapel,A., "IGOR:

Image Guided Operating Robot. Methodology, Medical Applications, Results" *Innov. Tech. Biol. Med.*, 13, 4, 1992, pp 373-393.

[CEO93] Cover, S.A., Ezquerra, N.F., O'Brien, J., Rowe, R., Gadacz, T., et Palm, E., "Interactively Deformable Models for Surgery Simulation" *IEEE Computer Graphics & Applications*, 13, 6, novembre 1993, pp 68-75.

[DCH95] Daane, S., Constantinou, P. et Hesselroth, G., "A $100 Surgical Simulator for the IBM PC" *Proceedings of the Medecine Meets Virtual Reality III*, San Diego, California, 19-22 janvier 1995.

[Del96] Delaby, F., *Simulateur de Cœlioscopie Gynécologique : Mise en Place d'un Algorithme de Détection de Collisions*, Diplôme d'Etudes Appronfondies d'Informatique, Université des Sciences et Technologies de Lille, juillet 1996.

[DGCL93] Delmotte, J.S., Gay, G., Chirol, P.L., Liénard, S., et Rey, J.F., "Simulation par Images Virtuelles et Analogiques d'Actes Chirurgicaux Complexes et Pathologie Bilio-Pancréatique" *Troisième séminaire du groupe de travail "Animation et Simulation"* Lille, octobre 1994.

[DRMK95] Dubois, P., Rouland, J.F., Meseure, P., Karpf, S., et Chaillou, C., "Simulator for Laser Photocoagulation in Ophtalmology", *IEEE transactions on Biomedical Engineering*, vol 42, 7, juillet 1995, pp 688-693.

[DSCP94] Delingette, H., Subsol, G., Cotin, S. et Pignon, J., "A Craniofacial Surgery Simulation Testbed" *Rapport de recherche INRIA* n°2199 février 1994.

[GK95] Geiger, B., et Kikinis, R., "Simulation of Endoscopy" *Proceedings of the CVRMed'95*, Nice, 3-6 avril 1995, Spinger Verlag, pp 277-281.

[GW87] Gillies D.F., et Williams C.B., "An Interactive Graphic Simulator for Teaching of Fibrendoscopic Techniques" *Proceedings of the EUROGRAPHICS'87 Conference*, Amsterdam, 24-28 août 1987, pp 127-138.

[HGW92] Haritsis A., Gillies D., et Williams C.B. "Realistic Generation and Real Time Animation of Images of the Human Colon" *Proceedings of the EUROGRAPHICS'92 Conference, Computer Graphics Forum*, 11, 3, Cambridge, 7-11 septembre 1992, pp 367-379.

[HTB95] Henry, D., Troccaz, J., et Bosson, J.-L., "Virtual Echography : Simulation of an Ultrasonic Examination" *Proceedings of Computer Assisted Radiology CAR'95*, berlin 21-24 juin 1995.

[Jam96] Jambon, A.C., *Mise en Place d'un Simulateur de Cœlioscopie à Visée Pédagogique*, Diplôme d'Etude Approfondies de Génie Biomédical, Faculté de Médecine d'Amiens, septembre 1996.

[Jou93] Jouve, F., *Modélisation de l'œil en élasticité non linéaire*, Masson, Paris, 1993.

[KHDS95] Kaplan, K., Hunter, I., Durlach, N., Shodek, D., et Rattner, D., "A Virtual Environment for a Surgical Room of the Future", *Proceedings of the "Virtual Reality World"* 1995, pp 159-164.

[KKKH95] Kühnapfel, U. G., Krumm, H. G., Kuhn, C., Hübner, M., et Neisius, B., "Endosurgery Simulations with KISMET : A Flexible Tool for Surgical Instrument Design, Operation Room Planning and VR Technology Based Abdominal Surgery Training" *Proceedings of "Virtual World"* 1995, pp 165-171.

[KKKN96] Kuhn, Ch., Kühnapfel, U., Krumm, H.G., et Neisius, B., "A Virtual Reality Based Training System for Minimally Invasive Surgery" *Computer Assisted Radiology*, Paris, France, 1996, pp 764-769.

[LBLG95] Lasko-Harvill, A., Blanchard, C., Lanier, J., et McGrew, D., “A Fully Immersive Cholecystectomy Simulation” *Proceedings of the Medecine Meets Virtual Reality III*, San Diego, California, 19-22 janvier 1995.

[Mes93] Meseure, P., *Définition et Réalisation d’un Simulateur de Photocoagulation Laser*, Diplôme d’Etudes Approfondies d’Informatique, Université des Sciences et Technologies de Lille, septembre 1993.

[MKCD95] Meseure, P., Karpf, S., Chaillou, C., Dubois, P., et Rouland, J.F., “Low-Cost Medical Simulation: a Retinal Laser Phtocoagulation Simulator” *Proceedings of the Graphics Interface ’95*, Québec City, Québec, 17-19 mai 1995, pp 155-162.

[MMRM95] Merril, J.R., Merril, G.L., Raju, R., Milman, A., Meglan, D., Preminger, G.M., Roy, R., et Babayan, R., “Photo-Realistic Interactive Three Dimensional Graphics” *Proceedings of the Medecine Meets Virtual Reality III*, San Diego, California, 19-22 janvier 1995.

[MRDK95] Meseure, P., Rouland, J.F., Dubois, P., Karpf, S., et Chaillou, C., “Sophocle: a Retinal Laser Photocoagulation Simulator - Overview” *Proceedings of the Conference on Computer Vision, Virtual Reality and Robotics in Medecine (CVRMed’95)*, Nice France, 3-5 avril 1995, Springer Verlag 905, pp 105-114.

[Peu96] Peugnet, F., *Evaluation du simulateur de photocoagulation rétinienne SOPHOCLE*, Diplôme d’Etudes Approfondies de Génie Biomédical, Faculté de Médecine d’Amiens, septembre 1996.

[PTE95] Peuchot, B., Tanguy, A., et Eude, M., “Virtual Reality as an Operative Tool during Scoliosis Surgery” *Proceedings of the CVRMed’95*, Nice, 3-6 avril 1995, Spinger Verlag, pp 549-554.

[Sat95] Satava, R. M., “Telesurgery, Virtual Reality and the New World Order of Medicine”, *Proceedings of the “Virtual Reality World”* 1995, pp 173-177.

[SBM94] Sagar M. A., Bullivant D., Mallinson G. D., Hunter P. J, et Hunter J. W., “A Virtual Environment and Model of the Eye for Surgical Simulation” *SIGGRAPH’94 Conference Proceedings, Computer Graphics annual conference series*, Orlando, 24-29 juillet 1994, pp 205-212.

[SPHL95] Sinclair, M. J., Peifer, J.W., Haleblian, R., Malcolm, N.L., Green, K., et Hull, D.S., “Computer Simulated Eye Surgery - A Novel Teaching Method for Residents and Practitioners” *Ophthalmology* 1995 (102), pp 517-521.

[VCDM97] Varlet, E., Chaillou, C., Dubois, P., et Maunoury, V., “Etude d’un Simulateur pour l’Apprentissage de l’Echo-Endoscopie”, *Innovation et Technologie en Biologie et Médecine*, (*à paraître*).

[ZMFG95] Ziegler, R., Mueller, W., Fisher, G., et Goebel, M., “A Virtual Reality Medical Training System” *Proceedings of the CVRMED’95*, Nice, 3-6 Avril 1995, Springer Verlag, pp 282-286.

Synthèse d’images / Animation

[AD92] Arnaldi, B., et Dumont, G., “Vehicle Simulation Versus Vehicle Animation” *Eurographics Workshop on Animation and Simulation*, Cambridge, 7-11 septembre 1992.

[Bar84] Barr, A.H., “Global and Local Deformations of Solid Primitives” *SIGGRAPH’84 Conference Proceedings, Computer Graphics*, 18, 3, Minneapolis, 23-27 juillet 1984, pp 21-26.

[Bar89] Baraff, D., “Analytical Methods for Dynamic Simulation of Non-Penetrating Rigid Body” *SIGGRAPH’89 Conference Proceedings, Computer Graphics*, 23, 3, Boston, 31 juillet-4 août 1989, pp 223-232.

[Bar90] Baraff, D., “Curved Surfaces and Coherhence for Non-Penetrating Rigid Body Simulation” *SIGGRAPH’90 Conference Proceedings, Computer Graphics*, 24, 4, Dallas, 6-10 août 1990, pp 19-28.

[Bar91] Baraff, D., “Coping with Friction for Non-Penetrating Rigid Body Simulation” *SIGGRAPH’91 Conference Proceedings, Computer Graphics*, 25, 4, Las Vegas, 28 juillet-2 août 1991, pp 31-40.

[Bar92] Baraff, D., “Non-Penetrating Rigid Body Simulation” *Eurographics ’93 State of the Art Report*, Chapter 2, Barcelone, 6-10 septembre 1993.

[Bar94] Baraff, D., “Fast Contact Force Computation for Non-Penetrating Rigid Bodies” *SIGGRAPH’94 Conference Proceedings, Computer Graphics annual conference series*, Orlando, 24-29 juillet 1994, pp 23-33.

[Bath82] Bath, K.J., *Finite Element Procedures*, Prentice-Hall 1982.

[Boy79] Boyse, J. W., “Interference Detection among Solids and Surfaces” *Communications ACM*, vol 22, 1, janvier 1979, pp 3-9.

[BT95] Bandi, S., et Thalmann, D., “An Adaptative Spacial Subdivision of the Object Space for Fast Collision Detection of Animated Rigid Bodies” *Proceedings of the Eurographics’95 Conference, Computer Graphics Forum*, 14, 3, Maastricht, 28 août-1 septembre 1995, pp 259-270.

[BV91] Bouma, W.J., et Vanecek, G., “Collision Detection and Analysis in Physically Based Simulation” *Eurographics Workshop on Animation and Simulation*, Vienne, 1991, pp 191-203.

[BW92] Baraff, D., et Witkin, A., “Dynamic Simulation of Non-Penetrating Flexible Bodies” *SIGGRAPH’92 Conference Proceedings, Computer Graphics*, 26, 2, Chicago, 26-31 juillet 1992, pp 303-308.

[Cam90] Cameron, S., “Collision Detection by Four-Dimensional Intersection Testing” *IEEE Transactions on Robotics and Automatons* juin 1990, 6, 3, pp 291-302.

[Can86] Canny, J., “Collision Detection for Moving Polyhedra” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, mars 1986, vol 8, no 2, pp 200-209.

[CC86] Cameron, S.A., et Culley, R.K., “Determining the Minimum Translational Distance between Two Convex Polyhedra” *Proceedings of the IEEE International Conference on Robotics and Automatons*, 1986, pp 591-597.

[CHP89] Chadwick, J.E., Hauman, D.R., et Parent R.E., “Layered Construction for Deformable Animated Characters” *SIGGRAPH’89 Conference Proceedings, Computer Graphics*, 23, 3, Boston, 31 juillet - 4 août 1989, pp 243-252.

[CK86] Culley, R.K., et Kempf, K.G., “A Collision Detection Algorithm Based on Velocity and Distance Bounds” *Proceedings of the IEEE International Conference on Robotics and Automatons*, 1986, pp 1064-1069.

[DG94] Desbrun, M., et Gascuel, M.P., “Highly Deformable Material for Animation and Collision Processing” *Eurographics Workshop on Animation and Simulation*, Oslo, 1994.

[DG95] Desbrun, M., et Gascuel, M.P., “Animation of Soft Substances with Implicit Surfaces” *SIGGRAPH’95 Conference Proceedings, Computer Graphics annual conference series*, Los Angeles, 6-11 Août 1995, pp 287-290.

[DTG95] Desbrun, M., Tsingos, N., et Gascuel, M.P., “Adaptative Sampling of Implicit Surfaces for Interactive Modeling and Animation”, *Eurographics Workshop on Implicit Surfaces*, Grenoble, 18-19 avril 1995, pp 171-185.

[Dwo94] Dworkin P. J., “Efficient Collision Detection for Real-Time Simulated Environments” *Mater thesis, Massachusetts Intitute of Technology,* 1994.

[DZ93] Dworkin, P., et Zelter, D., “A New Model for Efficient Dynamic Simualtion” *Eurographics Workshop on Animation and Simulation*, Barcelone, 4-5 septembre 1993, pp 135-147.

[ESP92] Essa, I.A., Sclaroff, S., et Pentland, A., “A Unified Approach for Physical and Geometric Modeling for Graphics and Animation” *Proceedings of the EUROGRAPHICS’92 Conference, Computer Graphics Forum*, 11, 3, Cambridge, 7-11 septembre 1992, pp 129-138.

[Far93] Farin, G., *Curves and Surfaces for Computer Aided Geometric Design - A Practical Guide*, Academic Press, troisième édition, 1993.

[FDFH90] Foley, J.D., van Dam, A., Feiner, S., et Hughes, J., *Computer Graphics - Principles and Practice*, Addison Wesley, seconde édition, 1990.

[FS91] Forgue, M.C., et Shinya, M., “Real-Time Impact Dynamics Simulation” *Compugraphics* 1991, pp 427-435.

[Gas89] Gascuel, M.P., “OSEA : un Nouveau Modèle de Matière pour Traiter les Collisions entre Objets Déformables” *Actes de PIXIM 89*, pp 309-323.

[Gas90] Gascuel, M.P., “Deformation de Surfaces Complexes : Techniques de Haut Niveau pour la Modélisation et l’Animation” *Thèse de doctorat*, Université de Paris Sud, centre d’Orsay, octobre 1990.

[Gas93] Gascuel, M.P., “An Implicit Formulation for Precise Contact Modeling between Flexible Solids” *SIGGRAPH’93 Conference Proceedings, Computer Graphics annual conference series*, Anaheim, 1-6 août 1993, pp 313-320.

[GLM96] Gottschalk, S., Lin, M.C., Manocha, D., “OBBTree: a Hierarchical Structure for Rapid Interference Detection” *SIGGRAPH’96 Conference Proceedings, Computer Graphics annual conference series*, New Orleans, 4-9 Août 1996, pp 171-180.

[Gou94] Gourret, J.P., *Modélisation d’Images Fixes et animées*, chapitre VI “Assemblage et Animation d’Objets Déformables : Simulation par Eléments Finis”, Masson, Paris, 1994.

[GTT89] Gourret, J.P., Magnenat Thalmann, N., et Thalman, D., “Simulation of Object and Human Skin Deformations in a Grasping Task” *SIGGRAPH’89 Conference Proceedings, Computer Graphics* 23, 3, Boston, 31 juillet - 4 août 1989, pp 21-30.

[GSF94] Garcia-Alonso, A., Serrano, N., et Flaquer, J., “Solving the Detection Problem” *IEEE Computer Graphics and Applications*, Mai 1994, pp 36-43.

[HA90] Haeberli, P., et Akeley, K., “The Accumulation Buffer: Hardware Support for High Quality Rendering”, *SIGGRAPH’90 Conference Proceedings, Computer Graphics*, 24, 4, Dallas, 6-10 août 1990, pp 309-318.

[Hah88] Hahn, J.K., “Realistic Animation of Rigid Bodies” *SIGGRAPH’88 Conference Proceedings, Computer Graphics*, 22, 4, Atlanta, 1-5 août 1988, pp 299-308.

[HBZ90] Von Herzen, B., Barr, A.H., et Zatz, H. R., “Geometric Collisions for Time Dependent Parametric Surfaces” *SIGGRAPH’90 Conference Proceedings, Computer Graphics*, 24, 4, Dallas, 6-10 août 1990, pp 39-48.

[HPH96] Hutchinson, D., Preston, M., et Hewitt, T., “Adaptive Refinment for Mass/Spring Simulations” *Eurographics Workshop on Animation and Simulation*, Poitiers, 31 août - 1 septembre 1996, pp 31-45.

[Hub96] Hubbard, P.M., “Approximating Polyhedra with Spheres for Time-Critical Collision Detection” *ACM Transactions on Graphics*, 15, 3, juillet 1996, pp 179-209.

[JB94] Joukhadar, A., et Baud, C., “Modélisation Physique pour la Planification de Saisie avec une Main Articulée” *Rapport de Recherche INRIA N° 2247*, Avril 1994.

[JBL94] Joukhadar, A., Bard, C., et Laugier, C., “Planning Dextrous Operations using Physical Models” *Proceedings of the IEEE Conference on Robotics and Automaton*, 1994, pp 748-753.

[Jim93] Jimenez, S., “Modélisation et Simulation Physique d’Objets Volumiques Déformables Complexes” *Thèse de doctorat*, Institut National Polytechnique de Grenoble, ACROE, LIFIA, novembre 1993.

[JL92] Jimenez, S., et Luciani, A., “Animation d’Objets en Interaction avec Collisions et Contacts Prolongés Adhérents” *Actes de GROPLAN*, Nantes, novembre 1992, pp 145-151.

[JLR90] Jimenez, S., Luciani, A., et Raoult, O., “Simulation et Contrôle Gestuel en Temps Réel d’une Large Variété d’Objets Physiques, Flexibilité, Plasticité, Rupture, Comportement de Multitude” *Actes de Groplan* 1990.

[JLR91] Jimenez, S., Luciani, A., et Raoult, O., “Physical Simulation of Land Vehicles with Obstacle Avoidance and Various Terrain Interactions” *Proceedings of the Eurographics’91 Conference*, Vienne, 1991, pp 245-261.

[Jou95] Joukhadar, A., “Dynamic Modeling System for Robotics Applications” *Rapport de recherche INRIA*, n° 2543, mai 1995.

[Kam93] Kamat, V.V., “A Survey of Techniques for Simulation of Dynamic Collision Detection and Response” *Computers and Graphics*, 17, 4, 1993, pp 379-385.

[Kar95] Karpf, S., “3D temps réel : Technologies et Produits” *Actes des 3èmes journées de l’Association Française de l’Informatique Graphique*, Marseille, 22-24 novembre 1995, pp 37-40.

[LC86] Luciani, A., et Cadoz, C., “Utilisation de Modèles Mécaniques et Géométriques pour la Synthèse et le Contrôle d’Images Animées” *2ème colloque CESTA*, Nice 1986, pp 704-711.

[LC91] Lin, M.C., et Canny, J.F., “A Fast Algorithm for Incremental Distance Calculation” *Proceedings of the IEEE International Conference on Robotics and Automatons*, avril 1991, Sacramento, California, pp 1008-1015.

[LC92] Lin, M.C., et Canny, J.F., “Efficient Collision Detection for Animation” *Eurographics Workshop on Animation and Simulation*, Cambridge, 7-11 septembre 1992.

[LHM95] Luciani, A., Habibi, A., et Manzotti, E., “A multi-Scale Model of Granular Materials” *Proceedings of the Graphics Interface’95 Conference*, Québec City, mai 1995,pp 136-146.

[LJFC91] Luciani, A., Jimenez, S., Florens, J.L., Cadoz, C., et Raoult, O., “Computational Physics : a Modeler Simulator for Animated Physical Objects” *Eurographics Workshop on Animation and Simulation*, Vienne, 1991, pp 425-437.

[LJFR89] Luciani, A., Jimenez, S., Folrens, J.L., Raoult, O., et Cadoz, C., “Modèles Comportementaux : vers une Approche Instrumentale de la Synthèse d’Images” *actes de Groplan 1989*, parution BIGRE 67, janvier 1990, pp 259-272.

[Lou94] Louchet, J., “Identification Evolutive de Modèles Physiques d’Animation à Base de Masses-Liaisons”. *Troisième séminaire du groupe de travail “Animation et Simulation”* Lille, octobre 1994.

[Mil88] Miller, G., “The Motion Dynamics of Snakes and Worms” *SIGGRAPH’88 Conference Proceedings, Computer Graphics*, 22, 4, Atlanta, 1-5 août 1988, pp 169-173.

[MT92] Metaxas, D., et Terzopoulos, D., “Dynamic Deformation of Solid Primitives with Constraints” *SIGGRAPH’92 Conference Proceedings, Computer Graphics*, 26, 2, Chicago, 26-31 juillet 1992, pp 309-312.

[MW88] Moore, M., et Wilhelms, J., “Collision Detection and Response for Computer Animation” *SIGGRAPH’88 Conference Proceedings, Computer Graphics*, 22, 4, Atlanta, 1-5 Août 1988, pp 289-298.

[NTBG91] Norton, A., Turk, G., Bacon, B., Gerth, J., et Sweeney, P., “Animation of Fracture by Physical Modeling”, *The Visual Computer*, 7, pp 210-219.

[PB88] Platt, J.C., et Barr, A.H., “Constraint Methods for Flexible Models”, *SIGGRAPH’88 Conference Proceedings, Computer Graphics*, 22, 4, Atlanta, 1-5 août 1988, pp 279-288.

[PC81] Potmesil, M. et Chakravarty, I., “A Lens and Aperture Camera Model for Synthetic Image Generation”. *SIGGRAPH’81 Conference Proceedings, Computer Graphics*, 15, 3, Dallas, 3-7 août 1981, pp 297-305.

[Pro95] Provot, X., “Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behaviour” *Proceedings of the Graphics Interface’95 Conference*, Québec City, Québec, 17-19 mai 1995, pp 147-154.

[PW89] Pentland, A., et Williams, J., “Good Vibrations : Modal Dynamics for Graphics and Animation” *SIGGRAPH’89 Conference Proceedings, Computer Graphics*, 23, 3, Boston, 21 juillet - 4 août 1989, pp 215-222.

[Rob95] Roberts, D., “Collision Detection : Getting the Most out of your Collision Tests” *Dr Dobb’s Journal*, 1995 (http://www.ddj.com/ddsbk/1995/1995/05/roberts.htm).

[Sho85] Shoemaker, K., “Animating Rotation with Quaternion Curves”, *SIGGRAPH’85 Conference Proceedings, Computer Graphics*, 19, 3, San Fransisco, 22-26 juillet 1985, pp 245-254.

[SP86] Sederberg, T.W., et Parry, S.R., “Free-Form Deformation of Solid Geometric Models” *SIGGRAPH’86 Conference Proceedings, Computer Graphics*, 20, 4, Dallas, 18-22 Août 1986, pp 151-160.

[SP91] Sclaroff, S., et Pentland, A., “Generalized Implicit Functions For Computer Graphics” *SIGGRAPH’91 Conference Proceedings, Computer Graphics*, 25, 4, Las Vegas, 28 juillet-2 août 1991, pp 247-250.

[SP95] Savchenko, V.V., et Pasko, A.A., “Collision Detection for Functionally Defined Deformable Objects” *Eurographics Workshop on Implicit Surfaces*, Grenoble, 18-19 avril 1995, pp 217-221.

[SWFC93] Snyder, J. M., Woodbury, A.R., Fleisher, K., Currin, B., et Barr, A.H., “Interval Methods for Multi-Point Collisions between Time-Dependent Curved Surfaces” *SIGGRAPH’93 Conference Proceedings, Computer Graphics annual conference series*, Anaheim, 1-6 août 1993, pp 312-334.

[TC90] Thingvold, J.A., et Cohen, E., “Physical Modeling with B-Spline Surfaces for Interactive Design and Animation” *Proceedings of the Symposium on Interactive 3D Graphics, Computer Graphics*, 24, 2, Snowbird, 25-28 mars 1990, pp 129-137.

[TF88] Terzopoulos, D., et Fleisher, K., “Modeling Inelastic Deformation : Viscoelasticity, Plasticity, Fracture” *SIGGRAPH’88 Conference Proceedings, Computer Graphics*, 22, 4, Atlanta, 1-5 août 1988, pp 269-278.

[TPBF87] Terzopoulos, D., Platt, J., Barr, A., et Fleisher, K., “Elastically Deformable Models” *SIGGRAPH’87 Conference Proceedings, Computer Graphics*, 21, 4, Anahaim, 27-32 juillet 1987, pp 205-214.

[TPF89] Terzopoulos, D., Platt, J., et Fleisher, K., “Heating and Melting Deformable Models (From Goop to Glop)” *Proceedings of the Graphics Interface’89 Conference*, London Otario, Canada, 1989, pp 219-226.

[Tur96] Turner, G., *Modélisation de Corps Mous par Eléments Finis en Vue d’Animation Temps Réel*, Diplôme d’Etudes Approfondies d’Informatique, Université des Sciences et Technologies de Lille, juillet 1996.

[TW88] Terzopoulos, D., et Witkin, A., “Physically Based Models with Rigid and Deformable Components” *IEEE Computer Graphics and Application*, novembre 1988, pp 41-51.

[ULC94] Uhl, C., Luciani, A., et Cadoz, C., “Le Simulateur Temps-réel TELLURIS pour le Langage CORDIS-ANIMA : Modèles Physiques, Gestes, Sons, Images”. *Troisième séminaire du groupe de travail “Animation et Simulation”* Lille, 20-21 octobre 1994, pp 59-73.

[Van94] Vanecek, G., “Back Face Culling Applied to Collision Detection of Polyhedra” *The Journal of Visualization and Computer Animation*, vol 5, 1994.

[VT94] Volino, P., et Thalmann, N.M., “Efficient Self-Collision Detection on Smoothly Discretized Surface Animations using Geometrical Shape Regularity” *Proceedings of the Eurographics’94 Conference*, *Computer Graphics Forum*, 13, 3, Oslo, 12-16 septembre 1994, pp 155-166.

[VT95] Volino, P., et Thalmann, N.M., “Collision and Self-Collision Detection : Efficient and Robust Solutions for Highly Deformable Surfaces” *Eurographics workshop on Animation and Simulation*, Maastricht, 2-3 septembre 1995, pp 55-65.

[WW90] Witkin, A., et Welch, W., “Fast Animation and Control of Nonrigid Structures” *SIGGRAPH’90 Conference Proceedings*, *Computer Graphics*, 24, 4, Dallas, 6-10 Août 1990, pp 243-252.

[ZOMP93] Zyda, M.J., Osborne, W.D., Monahan, J.G., et Pratt, D.R., “NPSNET: Real-Time Vehicle Collisions, Explosions and Terrain Modifications” *The Journal of Visualization and Computer Animation*, 4, 1, 1993, pp 13-24.

[ZPG95] Zeiller, M., Purgathofer, W., et Gervautz, M., “Efficient Collision Detection for General CSG Objects” *Eurographics workshop on Animation and Simulation*, Maastricht, 2-3 septembre 1995, pp 66-79.

Résumé

Cette étude s'inscrit dans la simulation d'interventions chirurgicales. Notre objectif est de concevoir des simulateurs à but pédagogique, où le malade est remplacé par un patient virtuel. L'avantage majeur de cette technique est d'autoriser l'entraînement à volonté en supprimant tout risque pour les patients. Par rapport aux classiques simulateurs de conduite, on retrouve les difficultés habituelles liées à la production en temps réel des images, à la qualité du rendu et au réalisme. Cependant, les simulateurs chirurgicaux exigent également la prise en compte des interactions mécaniques entre le chirurgien et la partie du corps humain opérée.

Ce document s'articule en deux parties. La première présente les concepts de la simulation médicale puis l'activité du laboratoire sur les simulateurs pédagogiques. Plusieurs simulateurs sont en cours d'étude, en ophtalmologie, en écho-endoscopie ou encore en cœlioscopie. Parmi les projets présentés, un simulateur de photocoagulation par laser (SOPHOCLE) est détaillé, car il fait partie du travail effectué lors de cette thèse.

La seconde partie aborde les problèmes liés à la prise en compte des interactions mécaniques et constitue le cœur de notre recherche. La simulation des interactions exige d'une part de savoir modéliser le comportement de corps mous que sont les organes opérés. Afin d'obtenir un résultat conforme à la réalité, il faut que le modèle s'appuie sur les lois physiques régissant le mouvement. Les propositions de modèles de ce type sont nombreuses en animation, mais peu d'entre eux s'intéressent véritablement à l'aspect temps-réel de la simulation. D'autre part, le second problème à résoudre concerne le traitement des interactions. Nous abordons cette notion sous les trois aspects que sont la détection, l'accélération de la détection puis le calcul de la réponse à la collision. Ce dernier point consiste non seulement à différencier les chocs des contacts prolongés, mais aussi à prendre en compte les interactions extrêmes comme les découpes ou les déchirures.

Nous proposons un modèle dynamique dont le but est de répondre de façon optimale, en temps réel à ces contraintes. Ce modèle est basé sur un maillage masses/ressort surfacique chargé de gérer les déformations locales et une composante rigide permettant les déplacements solides. Il réagit aux contraintes soumises par un outil contrôlé par le chirurgien grâce à une unité traitant les collisions. Un processus de subdivision permet d'affiner le modèle dans les zones de contact. Cette étude est accompagnée d'une mise en œuvre démontrant que le modèle permet d'atteindre une interactivité en temps réel. Par rapport aux modèles proposés auparavant, il est particulièrement adapté aux environnements incorporant un grand nombre de corps déformables susceptibles d'être manipulés.

Abstract

This work is a contribution to the surgical simulation field. Our aim is to design simulators for teaching, by replacing the patient by a virtual one. This technique allows the young surgeon to practice as often as he wants without compromising the integrity of the patient. This kind of simulation needs the production of high-quality images in real-time like those generated by flight simulators. However, it also requires to take into account the mechanical interaction between the surgeon and the part of the human body which is operated.

This document is composed of two parts. The first part presents the surgical simulation concepts and the simulators which are currently designed in the laboratory, in various fields such as ophthalmology, echo-endoscopy or gynaecologic laparoscopy. A simulator for laser photocoagulation of the eye fundus is presented as part of this work.

The second part deals with the specific questions related to mechanical interactions. The first problem is to simulate the behaviour of organs. To get a simulation as realistic as possible, the model has to rely on the physical laws of motion and deformable bodies mechanics. Though a lot of models have been proposed in animation, only few of them are compatible with real-time. The second problem consists in the computing of the interactions ans is adressed in three points : detection, acceleration of detection and response computation. The last point has to differentiate collisions from contacts and take into account cuttings and tears.

Our model is aimed to be real-time. It is based on a mass/spring surface mesh in charge of local deformations and a rigid compound which is responsible of rigid motions. It reacts to any constraints exerted by a manipulator. It can be refined in the contact zones thanks to a subdivision process. This model has been implemented and allows interactivity in real-time. It is especially adapted to environment with a high number of deformable bodies.