



HAL
open science

Modélisation, suivi et simulation d'objets articulés et déformables. Application au modelage réel d'une argile virtuelle

Guillaume Dewaele

► **To cite this version:**

Guillaume Dewaele. Modélisation, suivi et simulation d'objets articulés et déformables. Application au modelage réel d'une argile virtuelle. Interface homme-machine [cs.HC]. Institut National Polytechnique de Grenoble - INPG, 2005. Français. NNT: . tel-00584946

HAL Id: tel-00584946

<https://theses.hal.science/tel-00584946>

Submitted on 11 Apr 2011

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

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

Institut National Polytechnique de Grenoble

pour obtenir le grade de
DOCTEUR DE L'I.N.P.G.
Spécialité "Imagerie, Vision et Robotique"

Modélisation, suivi et simulation d'objets articulés et déformables. Application au modelage réel d'une argile virtuelle.

présentée et soutenue publiquement par
Guillaume Dewaele
le 7 décembre 2005

préparée au laboratoire GRAVIR-IMAG, à l'INRIA Rhône-Alpes,
au sein des équipes Perception et Évasion,
sous la direction de Radu Horaud et Marie-Paule Cani

Jury :

Valérie Perrier	Présidente proposée
Christophe Chaillou	Rapporteur
Pascal Fua	Rapporteur
Radu Horaud	Directeur de thèse
Marie-Paule Cani	Co-directrice de thèse
André Crosnier	Examineur
Frédéric Devernay	Examineur

Sommaire

Introduction	1
I Sculpture	5
1 Modélisation géométrique : état de l'art	7
2 Modèles de matériau pour la sculpture	39
3 Un modèle d'argile virtuelle	57
4 Enrichissement du modèle	105
II Capture de gestes par vidéo	125
5 La capture de gestes	127
6 Modèle articulé et déformable de la main	143
7 Suivi robuste d'objets rigides	169
8 Suivi d'objets articulés et déformables	213
Conclusion	231
Annexes	235
A Éléments d'analyse vectorielle	235
B Pseudo-distance	241

II

SOMMAIRE

Bibliographie

243

Table des figures

257

Table des matières

263

Introduction

Contexte

Encouragée, entre autres choses, par les progrès réalisés dans les domaines des images de synthèse et de la vision, la réalité virtuelle a pris une importance croissante ces dernières années. L'interaction entre les mondes réel et virtuel est un domaine d'étude prometteur, les moyens le permettant, notamment, restant encore, pour une large part, à inventer. C'est dans ce cadre qu'ont été menés ces travaux de thèse.

Notre propos était d'étudier et de développer quelques outils qui permettraient, par exemple, à un sculpteur de travailler une argile virtuelle. Nous souhaiterions qu'il fût possible, à l'aide de mouvements parfaitement naturels de la main, exécutés face à des caméras vidéo, d'agir sur un matériau virtuel de la même façon que l'on pourrait le faire avec son pendant réel.

Motivation et difficultés

Pour parvenir à ce résultat, il nous faut résoudre plusieurs problèmes. Le premier tient à l'interaction proprement dite. Une interface basée sur la vision présente des avantages indéniables. D'une part, elle est particulièrement riche. Le mouvement de la main fait intervenir une trentaine de degrés de liberté, bien plus que n'en mesurent la plupart des interfaces de réalité virtuelle. Elle ne gêne aucunement l'utilisateur, au contraire, par exemple, de gants haptiques. D'autre part, elle est parfaitement naturelle, puisque l'on peut directement utiliser ces mouvements pour agir sur le monde virtuel.

Pour qu'une telle interface soit utilisable, elle doit cependant être robuste. La position de la main doit être estimée avec autant de précision que possible, on doit pouvoir corriger d'éventuelles erreurs. Le suivi d'un grand nombre de degrés de liberté n'est pas une tâche facile, en particulier lorsqu'il faut tenir compte d'inévitables occlusions. Le suivi de la main a généralement été envisagé au moyen d'une unique caméra, ce qui a rendu les choses encore plus difficiles. Peu de travaux ont tenté de profiter d'informations tridimensionnelles obtenues par l'utilisation de plusieurs caméras.

Le second problème réside quant à lui dans le monde virtuel. Une argile virtuelle que l'on puisse manipuler comme on le ferait dans la réalité représenterait un outil de modélisation

3D simple et intuitif, qui demanderait peu d'apprentissage à l'artiste. Les possibilités offertes par une argile en mesure de se déformer à petite et à grande échelle, de changer naturellement de topologie, en feraient, dans le monde virtuel, un moyen simple d'ébaucher des objets de forme quelconque.

Il n'existe cependant pas à cette heure de modèle permettant de simuler le comportement d'un tel matériau, et d'interagir avec celui-ci en temps réel. L'argile est, on le verra, un matériau dit viscoplastique, situé entre les fluides visqueux et les solides plastiques. On dispose de nombreux modèles, correspondant à une large gamme de matériaux, mais aucun n'est en mesure pour le moment, comme nous le verrons, de reproduire l'ensemble du comportement d'une argile.

Contributions

Dans cette thèse, nous présentons un modèle d'argile virtuelle, offrant l'essentiel des caractéristiques que l'on attend : empreintes, déformations, changements de topologie, etc. L'utilisateur peut interagir avec ce matériau virtuel au moyen d'un nombre quelconque d'outils, de la forme qu'il souhaite. Pour ce faire, nous avons associé plusieurs couches de simulation, qui collaborent pour reproduire les différents comportements attendus.

Nous proposons également une méthode de suivi des mouvements d'un objet déformable, que nous appliquons au cas particulier de la main. Cette méthode utilise plusieurs caméras filmant les mains de l'utilisateur, et combine différentes approches pour fournir un suivi robuste du mouvement. Une observation du déplacement des points d'intérêt, localisés à la surface de la main, permet d'obtenir une première estimation, qui est ensuite corrigée lors d'une étape d'ajustement du modèle sur les données extraites des séquences vidéo.

Organisation de ce document

Dans une première partie, ce mémoire traite des travaux que nous avons menés dans le dessein de créer un matériau virtuel, semblable à une argile, qui puisse être manipulé en temps réel par un utilisateur.

- Le chapitre 1 présente un tour d'horizon des différentes approches qui ont été suggérées, par le passé, dans le cadre de la modélisation géométrique de formes tridimensionnelles.
- Dans le chapitre 2, nous nous penchons sur les avantages d'un travail sur une argile réelle par rapport aux outils de modélisation précédents, et nous essayons de cerner les différents aspects du comportement d'une argile qu'il nous semble important de conserver, dans le cadre d'une argile virtuelle destinée à la sculpture. Nous examinons également les modèles existants permettant la simulation de matériaux susceptibles de nous suggérer des pistes pour notre modèle d'argile virtuelle.
- Le chapitre 3 détaille le modèle d'argile virtuelle que nous avons développé, un modèle composé de trois couches de simulation distinctes afin de permettre d'obtenir l'essentiel des comportements de l'argile que nous cherchions à mettre en évidence.

- Le chapitre 4 enrichit ce modèle de diverses façons, en introduisant, entre autres choses, des matériaux non homogènes, en couleur, et en proposant une interaction plus riche entre l’argile et les outils. Il s’attarde également sur quelques questions d’implémentation et de rendu.

Dans une seconde partie, nous discutons de la possibilité de suivre le mouvement de la main d’un sculpteur face à un jeu de caméras vidéo, afin de permettre une interaction plus riche entre l’artiste et la machine que ne le permettent la plupart des interfaces conventionnelles.

- Le chapitre 5 résume les travaux ayant déjà été menés dans le domaine du suivi de mouvements par vidéo, en s’attardant tout spécialement sur les méthodes de suivi des mouvements de la main sans marqueur ni dispositif particulier.
- Le chapitre 6 est l’occasion pour nous de présenter un modèle de main, utilisé ensuite pour le suivi. Ce modèle comprend une description cinématique, sur laquelle nous avons greffé une représentation volumique déformable ainsi qu’une description des mouvements de la peau à la surface de la main. Nous introduisons au cours de ce chapitre plusieurs outils qui nous sont utiles par la suite.
- Au cours du chapitre 7, nous détaillons notre approche du suivi de mouvement, mettant en œuvre tout à la fois une mise en correspondance de points 3D entre deux instants successifs (afin d’estimer le déplacement entre ces deux instants) et un ajustement du modèle sur les données acquises par les caméras. Nous appliquons cette stratégie au cas particulier d’un solide indéformable, et nous comparons nos résultats avec les méthodes couramment utilisées, afin de voir dans quelle mesure notre approche mixte nous permet de gagner en précision et en robustesse.
- Le chapitre 8, qui clôt cette seconde partie, voit l’application de notre schéma de suivi à un objet indéformable, et plus particulièrement à la main humaine au moyen du modèle développé plus tôt. Quelques exemples de gestes, synthétiques et réels, nous permettent enfin de discuter des forces et faiblesses de notre approche.

Ce mémoire s’achèvera par un bref inventaire des difficultés qui restent à surmonter afin de proposer un outil de modélisation 3D contrôlé par gestes, et des pistes qui nous ont semblé prometteuses.

Première partie

Sculpture

Chapitre 1

Modélisation géométrique et sculpture virtuelle : état de l’art

1.1 Introduction

De par l’importance que revêt la création de formes 3D dans le domaine de l’imagerie de synthèse, les outils de modélisation sont nombreux et variés. Afin d’en cerner les caractéristiques et les limitations, nous allons parcourir les diverses méthodes qui ont été proposées pour l’élaboration d’objets tridimensionnels, depuis les méthodes les plus simples et les plus anciennes éditant directement le maillage des modèles jusqu’aux approches plus récentes que l’on qualifie parfois de sculpture virtuelle.

Nombre de travaux ont été proposés sur le sujet, et il ne nous sera pas possible, dans le présent document, de nous pencher sur l’ensemble de ces derniers. Nous nous efforcerons donc de présenter quelques-uns des articles parmi les plus marquants, illustrant la grande variété des approches étudiées.

Nous nous intéresserons principalement aux problèmes de “modélisation”, c’est-à-dire aux moyens de représenter, de réaliser et de déformer une forme tridimensionnelle. Les méthodes présentées dans ce chapitre ont essentiellement un caractère géométrique. Nous aborderons dans le chapitre suivant le cas de quelques modèles physiques qui peuvent servir à la modélisation d’un objet.

Nous regarderons, pour ces différentes méthodes, dans quelle mesure il est possible de réaliser des objets de forme quelconque. Toutes les méthodes présentées ici ne sont en effet pas adaptées à la réalisation de n’importe quel volume. En particulier, certaines approches permettent plus ou moins facilement des changements de topologie de l’objet. D’autres, au contraire, sont limitées à des corrections de faible amplitude. L’intuitivité est un autre aspect sur lequel nous nous attarderons, il n’est en effet pas rare qu’une certaine compétence soit requise de la part de l’utilisateur.

1.2 Les débuts

1.2.1 Manipulation de volumes simples

Les premières formes tridimensionnelles qui ont pu être manipulées par ordinateur furent naturellement les objets mathématiques les plus simples : cubes, sphères, cylindres et autres cônes. En assemblant ces diverses primitives, il est possible d'obtenir des objets de forme plus complexe. Afin d'enrichir la gamme de formes que l'on pouvait obtenir de cette façon, une démarche s'est rapidement imposée : la Constructive Solid Geometry, plus connue sous son acronyme, CSG.

Le principe mis en œuvre est simple : il utilise des opérations ensemblistes sur les primitives précédemment citées (prismes, sphères, cylindres). Les principales opérations utilisées sont l'union, la soustraction, et l'intersection. Une forme complexe peut être décrite comme un arbre, dont chaque nœud est l'une des opérations ensemblistes précitées, et chaque feuille est une primitive simple. On pourra par exemple réaliser un trou circulaire dans un objet en lui soustrayant un cylindre, ou obtenir un chanfrein de façon similaire. L'intersection permet quant à elle d'obtenir un objet à partir de plusieurs de ses profils, par exemple.

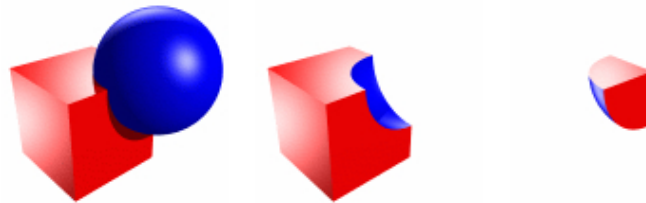


FIG. 1.1 – Les trois opérations combinatoires de la Constructive Solid Geometry : union, différence et intersection.

Bien que très simpliste, cette méthode permet d'obtenir une gamme assez large d'objets, de façon relativement simple et systématique. Elle reste encore la méthode la plus utilisée aujourd'hui dans de nombreux domaines. On pourra citer par exemple l'architecture, qui se prête généralement bien à une description au moyen de formes simples, ou encore la conception industrielle. De nombreux logiciels dits de CAD (Computer Assisted Design) reposent essentiellement sur ce principe, d'autant que bien souvent la réalisation de la pièce elle-même repose généralement sur les mêmes principes techniques, partant d'un bloc sur lequel on va généralement retirer de la matière jusqu'à obtenir la pièce désirée. En outre, les idées introduites par cette méthode (réalisation d'un objet tridimensionnel en ajoutant ou retirant des éléments), certes simples et naturelles, sont restées au cœur de bien des méthodes de modélisation 3D, ainsi que nous allons le voir.

Une autre méthode importante et très usitée pour créer un volume tridimensionnel consiste à transformer une courbe planaire en une surface en la déplaçant dans l'espace. La surface correspond au lieu géométrique occupé par les points de la courbe lors de son déplacement. Cette méthode dite de *sweeping* est particulièrement utile car on dispose d'une gamme relativement importante d'outils mathématiques permettant d'obtenir une large gamme de courbes planes. Parmi les déplacements les plus employés, on dispose évidemment de la translation

rectiligne (ainsi, un simple cercle permet d'obtenir un cylindre, un polygone conduisant à un prisme droit) et de la rotation autour d'un axe, qui correspond en quelque sorte au travail du tourneur : à partir d'un contour on forme un volume de révolution. Nous verrons plus tard que ces idées sont reprises et étendues par les surfaces d'optimisation.

1.2.2 Vers des maillages polygonaux

La réalisation d'un volume 3D, par exemple avec les méthodes précédentes, n'aurait guère d'applications s'il n'était possible de le visualiser. Pour ce faire, il s'est développé de très nombreuses techniques. Par exemple, on peut utiliser des méthodes dites de lancer de rayon, ou *raytracing*. L'écran (ou le dispositif d'affichage) va se comporter comme une caméra qui filmerait la scène virtuelle. A chacun des pixels constituant l'image, on peut associer une direction dans l'espace virtuel où se trouve la scène à afficher. Les algorithmes de lancer de rayon cherchent à déterminer si ces directions croisent la surface d'un des objets de la scène.

Cette méthode permet de travailler avec des objets mathématiques tels que ceux déjà présentés. Mais la recherche d'une intersection est un problème délicat d'un point de vue mathématique, coûteux en terme de temps de calcul, et ce d'autant plus que la scène est complexe. Il s'est rapidement avéré que pour afficher rapidement un volume, il faudrait procéder différemment. Le problème s'est révélé être essentiellement une question de projection d'objets 3D dans un plan. Le triangle est un objet dont la projection est simple, puisqu'il suffit de projeter ses trois sommets. On a donc trouvé utile de transformer toute surface en un ensemble de triangles. Il était ainsi bien plus aisé de les projeter et de les afficher. Ces différents triangles sont généralement jointifs, de façon à recréer une surface continue ; on parlera généralement de maillage.

Bien évidemment, dans la plupart des cas, ce n'est qu'une approximation, puisque les objets sont alors composés d'une multitude de facettes planes, et qu'il n'est plus possible d'obtenir de véritables surfaces courbes. Cependant, dans la mesure où les triangles sont de taille suffisamment petite (et moyennant quelques précautions sur l'éclairage que nous ne détaillerons pas ici), l'illusion est amplement suffisante. Ainsi, pour des raisons techniques, la représentation des surfaces et volumes sous la forme d'un ensemble de triangles (éventuellement de polygones) est devenue peu ou prou un standard, et l'est encore aujourd'hui. La plupart des outils travaillent donc avec cette représentation des surfaces, bien qu'elle ne soit pas la seule employée, ainsi que nous allons le voir.

B. Naylor a montré que les opérations de CSG pouvaient très bien être menées sur des modèles polygonaux [Nay90]. À partir de huit formes simples, pouvant comporter des attributs de couleur ou de texture, et des opérations de base présentées précédemment, il propose un outil de modélisation 3D baptisé SCULPT permettant la création de formes raisonnablement complexes. Dans cet environnement, on parle de sculpture et d'outils. L'outil est l'une des huit formes de base proposées à l'utilisateur. Celui-ci va pouvoir le positionner, puis selon son choix réaliser une soustraction ou une addition du volume de l'outil à l'objet sculpté. Il lui est également donné la possibilité de réaliser ces mêmes opérations en permanence durant le déplacement de l'outil.

L'article reprend et étend des idées plus anciennes, et s'attarde particulièrement sur des

questions d'interface, tel le contrôle de l'outil dans l'espace au moyen d'une simple souris 2D, de même que les questions techniques que pose la détection des collisions entre deux objets, nécessaire au calcul des différentes opérations d'édition. Proposition est faite d'utiliser à cette fin des arbres de partitionnement binaire (ou *BSP-trees*) qui permettent une localisation rapide des objets les uns par rapport aux autres, et facilitent les calculs de collision.

1.3 Édition de maillages polygonaux

1.3.1 Manipulation explicite du maillage

Puisque la plupart des volumes sont représentés sous la forme de maillages polygonaux, il est naturel d'envisager toutes sortes d'outils permettant l'édition ou la modification de ces derniers. On peut ainsi espérer altérer leur forme, et obtenir des objets qu'il est difficile, voire impossible d'obtenir avec les démarches précédentes. On imagine aisément, par exemple, pouvoir obtenir la forme irrégulière d'un fruit en partant du maillage d'une sphère, que l'on déformerait et modifierait à souhait.

La solution immédiate au problème de l'édition d'une surface constituée par un ensemble de triangles consiste à déplacer chacun des sommets desdits triangles. En pratique, cette solution ne saurait être vraiment satisfaisante, d'autant qu'il n'est pas rare, aujourd'hui, qu'un objet soit composé de dizaines de milliers, voire de plusieurs millions de sommets, afin que la surface obtenue soit suffisamment lisse. Le déplacement des sommets un à un demandera bien évidemment beaucoup de travail à l'artiste, qui de plus aura beaucoup de difficultés s'il souhaite que la surface obtenue conserve son caractère lisse et ses éventuels détails.

Plutôt que de déplacer les sommets isolément, on préférera donc les manipuler par groupes. La sélection d'un ensemble de sommets que l'on manipulera de façon rigide ne résout pas la plupart des problèmes soulevés précédemment, il a donc été proposé d'utiliser des fonctions d'influence : lorsqu'un sommet est manipulé par l'utilisateur, les sommets voisins du maillage seront également déplacés, sans que l'utilisateur ait spécifiquement besoin de les sélectionner. La fonction d'influence permet d'amortir le déplacement des différents sommets avec la distance au sommet édité, de sorte que la surface ne change pas lorsque l'on s'éloigne de la zone d'édition.

L'idée d'utiliser de telles fonctions a brièvement été introduite par R. Parent [Par77] dans un système de création d'objets tridimensionnels qui introduit bien des idées, qui, nous le verrons, seront reprises par la suite. Les travaux de L. Moccozet et P. Kalra [MK93] reprennent et développent cette idée de fonctions d'influence permettant le contrôle d'un ensemble de sommets d'un maillage polygonal.

1.3.2 Vers une abstraction du maillage

Les approches précédentes ne sont pas pleinement satisfaisantes pour l'utilisateur. Il n'est pas possible d'oublier le maillage sous-jacent au modèle, et l'artiste doit spécifiquement aller choisir l'un des sommets dudit maillage. J.R. Bill et S.K. Lodha proposent dans [BL95] un

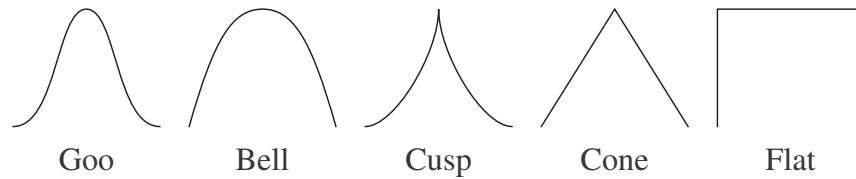


FIG. 1.2 – Quelques exemples de fonctions d’influence proposées par J. R. Bill dans [BL95]

moyen d’abstraire cette sélection, en réintroduisant le concept d’outil. Ces derniers sont des solides (plus précisément des hyperquadriques, pour des raisons d’efficacité tant pour l’affichage que pour la détection des collisions) qui vont réaliser ce travail de sélection à la place de l’utilisateur.

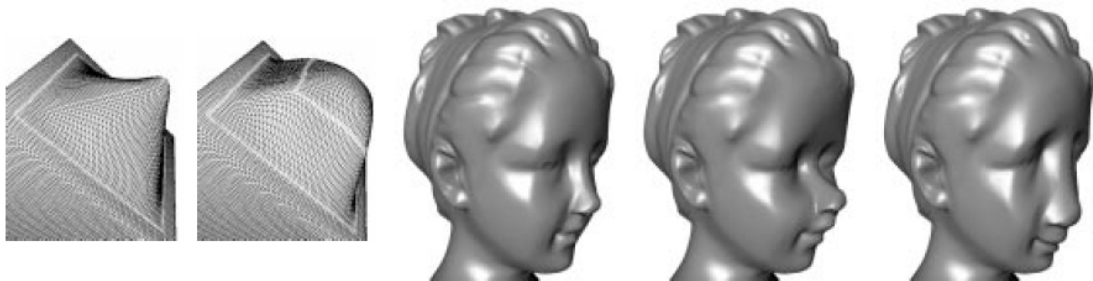


FIG. 1.3 – Deux courbes contrôlent les déformations du maillage dans [KCVS98]. ‘A droite, différents exemples de déformations obtenues avec cette approche.

Celui-ci se voit proposer de pousser ou de tirer les points du maillage au contact de l’outil. Dans le premier cas par exemple, on recherche d’éventuels points du maillage se trouvant à l’intérieur du volume de l’outil après un déplacement de ce dernier. Si c’est le cas, ces derniers sont repoussés jusqu’à se retrouver à l’extérieur de l’outil. Bien évidemment, il est possible d’employer les fonctions d’influence précédemment citées afin de transmettre une partie de ce mouvement aux points voisins dans le maillage. Plus l’influence se fera sentir à longue distance, plus la surface semblera élastique et caoutchouteuse.

Plutôt que d’utiliser un outil, la méthode proposée par L. Kobbelt *et al* dans [KCVS98] propose à l’utilisateur de délimiter la zone qui devra être déformée par une courbe fermée, dessinée directement sur la surface. Les modifications du maillage seront donc restreintes à l’intérieur de ce maillage. La déformation proprement dite sera dirigée par le déplacement d’une seconde courbe, fermée ou non, tracée au sein de la première. L’influence du déplacement varie ensuite continûment entre la courbe servant de contrôle et celle délimitant la zone de déformation.

Afin de préserver les détails les plus fins de la surface, il est également proposé un schéma de simplification pour le maillage. L’algorithme construit une série de maillages de plus en plus simples, en supprimant itérativement arêtes et sommets. La position des points supprimés est stockée sous la forme de ses coordonnées dans un repère local du maillage correspondant à l’itération suivante. Les déformations sont alors appliquées au maillage le plus grossier, puis

répercutées ensuite sur les maillages les plus fins.

1.3.3 Autres méthodes de manipulation

De nombreuses autres méthodes de manipulation et de déformation du maillage ont été suggérées. Plutôt que d'explicitement tirer ou pousser certains des sommets du maillage, on peut par exemple le laisser évoluer selon diverses équations. Les sommets du maillage peuvent être soumis à des forces qui provoqueront leur déplacement. On peut par exemple imaginer une surface qui se dilate ou se contracte localement, en fonction de sa courbure, de ses normales, ou d'autres critères.

J. Lawrence et T. Funkhouser proposent ainsi un système de modélisation dans lequel l'utilisateur peint l'objet [LF03]. Les zones peintes vont alors se dilater ou se contracter, dans une direction donnée ou selon leurs normales, éventuellement en fonction de la courbure de la surface au point considéré. On peut ainsi déformer assez librement les maillages, ainsi que le montre la figure 1.4. Ce genre de déformation ne gère cependant pas les éventuelles auto-collisions, et les changements de topologie auxquels ils conduisent. Ainsi que l'indiquent les auteurs, les surfaces *levelset*, plus généralement connues sous le nom de *surfaces implicites* dans le domaine de l'infographie, résolvent cette limitation en autorisant les changements de topologie, mais au prix de la perte de l'interactivité. Nous reviendrons ultérieurement plus en détail sur ce type de représentation.

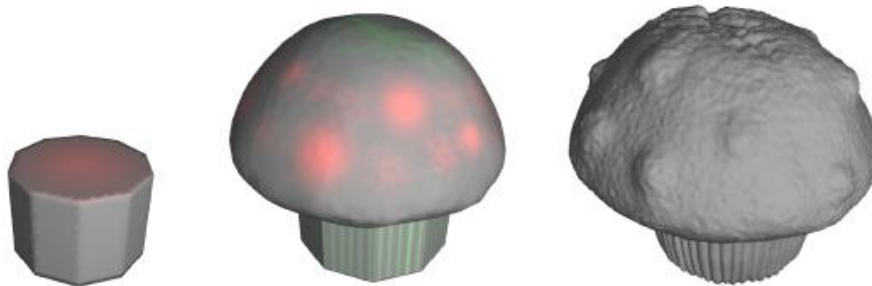


FIG. 1.4 – Une peinture de l'objet permet de contrôler la déformation de son maillage dans [LF03].

1.3.4 Rectification automatique du maillage

Bien que l'utilisateur soit notablement aidé dans les problèmes de sélection, il ne peut toutefois pas totalement oublier la représentation sous-jacente de l'objet. Par exemple, s'il étire trop une zone de l'objet, il peut arriver que les triangles constituant le maillage soient trop étirés pour que l'objet reste lisse. Pour corriger ces effets, il est mis à sa disposition différents outils permettant d'agir sur le maillage, l'affinant par l'adjonction de triangles supplémentaires, le lissant, etc. Lorsqu'il s'agira de percer un trou dans la surface, là encore, ce sont les triangles qui seront retirés, à l'utilisateur de s'assurer que le maillage permet de le faire dans de bonnes conditions.

Les modèles précédent supposent que l'on travaille sur des maillages suffisamment détaillés, et proposent généralement des techniques de subdivision des polygones lorsque ceux-ci sont trop déformés. Il faut constamment créer de nouveaux sommets et réordonner les arêtes si l'on veut que le maillage garde de bonnes propriétés, en particulier visuelles (de courbure notamment) après la déformation. Ce genre de problème apparaît tout particulièrement lorsque l'on essaie de déplacer latéralement une partie du maillage, le long de la surface, par exemple pour changer l'emplacement d'un détail. H. Suzuki et al montrent dans [SSKK98] comment supprimer, modifier et créer des triangles de façon à rendre ces déplacements possibles.

1.3.5 Déformations par squelette

Les approches présentées ci-dessus avaient pour principal objectif de déformer, plus ou moins localement, le maillage polygonal. Elles permettent aisément l'adjonction de détails à l'objet, ainsi que diverses corrections. Toutefois, il n'est pas toujours aisé de déformer une partie complète de l'objet (lorsque c'est nécessaire, d'autres approches sont employées, par exemple des volumes dits FFD, sur lesquels nous reviendrons ultérieurement, dans les travaux de L. Moccozet et P. Kalra [MK93]).

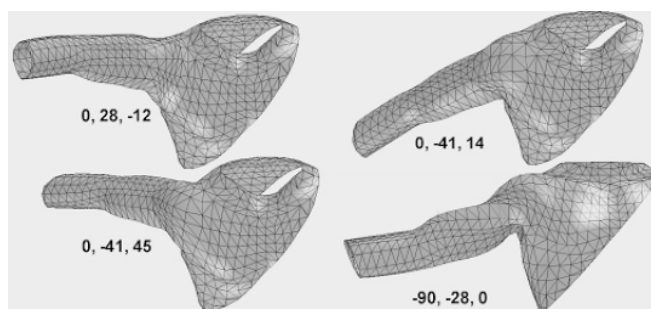


FIG. 1.5 – Déformations d'un bras au moyen d'un squelette [Blo02].

Par exemple, on peut avoir besoin de déformer un maillage représentant un personnage, pour replier un bras ou bien une jambe. *A fortiori*, dans le cas d'un travail d'animation, il n'est bien évidemment pas question de repartir de zéro pour chacune des poses que devra prendre le modèle. Outre la quantité importante de travail que ce serait, il serait difficile d'assurer précisément la cohérence temporelle des différents détails du modèle, et le résultat ne serait pas satisfaisant.

La déformation d'une surface peut généralement être considérée comme le problème d'un déplacement cohérent et intuitif d'un ensemble de points. Comme nous venons de le voir, il est souvent très utile de proposer à l'utilisateur d'agir non pas directement sur le maillage, mais plutôt sur un objet abstrait (un outil, une courbe...). On peut envisager par exemple que l'utilisateur déforme une structure indépendante de l'objet, et que cette déformation est traduite, de façon automatique et transparente, en un mouvement pour les différents points du maillage ou les points de contrôle de l'objet édité.

C'est précisément cette méthode qui est souvent utilisée par les animateurs : plutôt que de déformer leurs modèles 3D, ils agissent sur un squelette sous-jacent. Chacun des points du modèle est fixé à un ou plus généralement plusieurs éléments du squelette, et sont déplacés

en même temps que ce dernier. Cette idée n'est pas récente puisqu'on la trouve déjà dans les travaux de R. Parent [Par77], et a bénéficié entre-temps de nombreux développements [AMH02] [LCJ94] [LCN00].

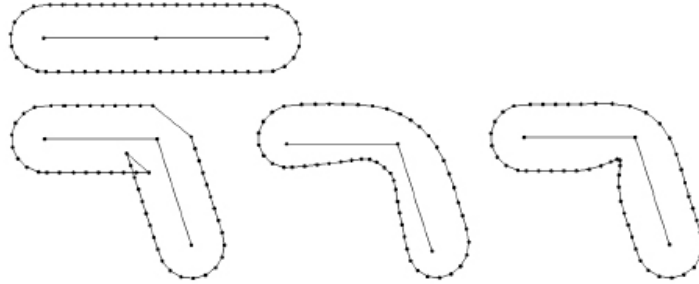


FIG. 1.6 – Différentes fonctions de pondération pour une déformation par squelette. À gauche, seul l'élément le plus proche contribue au déplacement ; au centre, les poids sont fonction de l'inverse du carré de la distance ; à droite, la solution proposée dans [Blo02].

Le mouvement des points de l'objet est généralement choisi comme une combinaison linéaire du mouvement des éléments sous-jacents du squelette. Cette solution est particulièrement populaire à l'heure actuelle, notamment grâce au support, sur les cartes graphiques récentes, de fonctions permettant la combinaison du mouvement de plusieurs points appelées vertex blends, partie importante des opérations permises par les vertex shaders dans les cartes vidéo grand public actuelles [AMH02].

La principale difficulté concerne le choix, pour chacun des points, des poids respectifs de chacun des éléments du squelette. Il arrive (généralement dans le cadre de l'animation) que ces poids soient spécifiquement définis par l'utilisateur. C'est bien évidemment malaisé, surtout lorsque les objets possèdent des maillages très détaillés. Les poids sont donc généralement calculés de façon automatique, principalement sur des critères de distance aux différentes parties du squelette. De nombreuses solutions ont été envisagées, les poids étant généralement choisis selon une fonction de l'inverse de la distance à chacune des parties du squelette. Il arrive généralement cependant que ces poids conduisent à des déformations maladroitement, et qu'il faille les corriger à la main. Cet inconvénient, longtemps considéré comme inévitable, a été corrigé par J. Bloomenthal, à partir de méthodes basées sur la convolution [Blo02].

Le squelette peut enfin être spécifiquement construit par l'utilisateur, mais il peut être également construit de façon automatique à partir d'un modèle tridimensionnel, ainsi que le montrent J. Bloomenthal et C. Lim dans [BL99]. La forme du squelette est constituée des différents axes médians du volume (c'est-à-dire le lieu géométrique des points correspondant aux centres des sphères inscrites dans le volume, illustré sur la figure 1.7), et les articulations sont déduites des changements d'orientation de ces courbes médianes. Une fois le squelette construit, il est possible de déformer l'objet en agissant sur le squelette lui-même, et obtenir diverses déformations, torsions et élongations de l'objet.

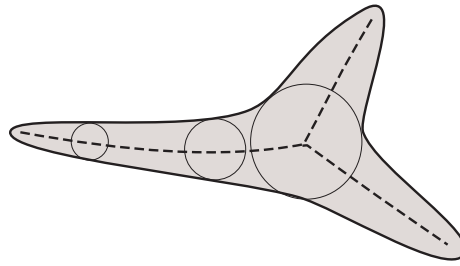


FIG. 1.7 – L’axe médian d’un volume correspond au lieu géométrique constitué des centres des sphères inscrites dans le volume.

1.3.6 Bilan

Les différentes approches présentées au cours de ce chapitre se prêtent bien à des corrections de la forme d’un objet. En revanche, elles ne permettent pas la réalisation aisée d’un objet de forme quelconque. Par exemple, tout changement de topologie va supposer un remaillage de l’objet, qui doit être géré par l’utilisateur lui-même.

Il est de façon générale difficile de s’affranchir de la représentation sous-jacente du matériau. Ainsi, des déformations excessives peuvent conduire à un maillage très étiré, ce qui peut se révéler gênant, tant pour l’affichage que pour la précision des modifications ultérieures. Il faut en outre que le maillage soit suffisamment fin pour que les déformations soient de bonne qualité. Pour contourner ces écueils, diverses méthodes mettent en place un remaillage automatique de la surface.

La manipulation directe des éléments d’un maillage a cependant conduit à l’introduction, particulièrement importante dans le cadre de la sculpture virtuelle, du concept d’“outil”. L’outil permet généralement d’oublier partiellement, pour un temps, le maillage proprement dit, pour travailler à un niveau d’abstraction supérieure. Diverses règles vont retraduire les mouvements de l’outil en changements au niveau du maillage. Un outil qui pousse ou tire une partie du maillage, le restant étant géré comme un matériau élastique, par exemple, permet de mieux comprendre quels effets nos actions auront sur l’objet.

1.4 Surfaces lisses de forme libre

Les surfaces polygonales ont toutefois un grave défaut. Constituées de morceaux de plans, elles sont incapables de modéliser des surfaces lisses et courbes. Lorsque la modélisation a pour but l’affichage, on peut contourner ce problème en utilisant une grande quantité de triangles de petite taille pour approcher au mieux la surface courbe. Ce n’est toutefois pas acceptable si l’on souhaite une définition précise de la surface, et par ailleurs, manipuler une grande quantité de polygones peut se révéler délicat.

On pourrait songer à utiliser des morceaux de cône, de cylindre ou des calottes sphériques pour compenser le manque des surfaces polygonales. Ce n’est hélas pas suffisant pour s’adapter

à n'importe quelle situation. La courbure est en effet uniforme ou varie linéairement sur ces éléments, et certaines surfaces, en particulier les surfaces organiques, peuvent montrer des courbures variant de façon complexe. De nombreuses suggestions ont été faites pour remplacer les polygones dans le cas de surfaces courbes, nous allons en voir quelques-unes, ainsi que les méthodes qui permettent d'agir sur la forme de ces surfaces.

1.4.1 Les surfaces Splines paramétriques

Présentation

Une première suggestion a été faite par l'ingénieur Pierre Bezier en 1972. Celui-ci, travaillant chez Renault, avait besoin de descriptions de surfaces courbes précises, exactes, et faciles à manipuler, dans le but de créer des pièces automobiles, puis de commander les machines qui les fabriqueraient. Son idée repose sur le principe des *splines*. Les splines sont des courbes qui s'efforcent d'interpoler un ensemble de points, dits points de contrôle. Il existe plusieurs variétés de splines, celles auxquelles nous nous intéresserons plus particulièrement sont qualifiées de B-splines.

Les B-splines sont des courbes définies et construites récursivement, à partir du principe du barycentre. Par exemple, dans le cas d'une B-spline de degré 3, qui comporte 4 points de contrôle, on construit pour chaque valeur d'un paramètre k variant entre 0 et 1 les deux segments dont les extrémités correspondent aux barycentres de chaque paire de points de contrôle affectés des coefficients k et $1 - k$, puis un segment supplémentaire de la même façon, et enfin un point de la B-spline, toujours de façon barycentrique. Cette construction est illustrée sur la figure 1.8 pour des valeurs de k égales à $1/3$ et $2/3$.

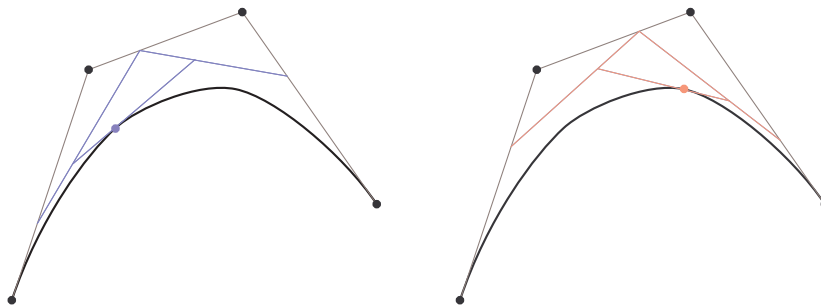


FIG. 1.8 – Construction d'une B-spline d'ordre 3.

On obtient ainsi une courbe fermée, lisse, dont on peut choisir la forme en plaçant convenablement les points de contrôle. Les extrémités de la B-spline passent précisément par les points de contrôle, et leurs tangentes coïncident avec les segments terminaux. Les surfaces de Bézier sont la traduction surfacique de cette idée, avec cette fois-ci une grille $n \times n$ de points de contrôle. Il existe d'autres variantes, que nous n'aborderons pas ici, qui sont notamment décrites dans [BFK84]. Ce type de courbe est fréquemment utilisé dans la plupart des logiciels de CAO, dont *unisurf*, le logiciel de modélisation 3D créé par Pierre Bezier lui-même.

Notons au passage que beaucoup d'outils parmi ceux définis précédemment dans le cas de

surfaces polygonales peuvent également convenir aux surfaces de Bezier, il suffira en effet de les appliquer aux points de contrôle plutôt qu'aux points du maillage.

Des points de contrôle à la surface

L'un des inconvénients que présentent les surfaces de Bezier est que l'utilisateur contrôle la surface par le biais de points de contrôle, et non directement par des points de la surface proprement dite. Il faut donc un peu d'habitude lorsque l'on souhaite créer une forme précise. L'une des premières tentatives visant à résoudre cette difficulté est à mettre au crédit de D. Forsey et R. Bartels, qui introduisent dans [FB88] les H-Splines. Celles-ci permettent une manipulation plus directe de la surface, en remplaçant les points de contrôle par d'autres points, certes toujours prédéfinis, mais situés sur la surface proprement dite. Ces points sont choisis de façon à correspondre à un maximum d'influence sur l'un des points de contrôle, et sont ceux que l'utilisateur emploiera pour déformer la surface.

Cette approche est développée et étendue par B. Fowler dans [Fow92]. Il est alors possible d'utiliser n'importe quel point de la surface comme point de contrôle, la déformation de la surface étant ensuite déduite automatiquement du mouvement du point choisi. Il est en outre permis de librement modifier les tangentes en ce point. Elles sont modélisées par un élément carré tangent à la surface en ce point. En inclinant ce carré, il est possible de fixer la direction des tangentes en ce point, et même d'obtenir des effets de torsion en le faisant tourner autour de sa normale. La modification de la taille du carré permet également de jouer sur la courbure de la surface au point sélectionné. Ces divers effets sont illustrés sur la figure 1.9.

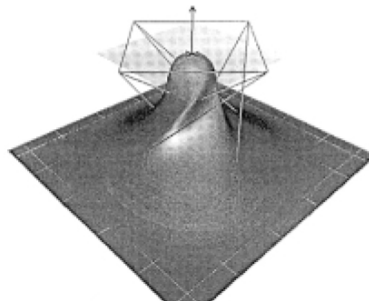


FIG. 1.9 – La déformation de la surface B-Spline dans [Fow92] se fait à partir d'un polygone tangent à la surface, permettant, en plus du positionnement, différents effets de déformation et de torsion.

Problèmes de connexité

Lorsque l'on utilise des carreaux paramétriques pour représenter une surface, on peut parfois rencontrer également des problèmes de continuité entre les différents patches (continuité de la surface, mais aussi de ses dérivées si l'on veut une surface lisse). Pour que la surface puisse être continue, il faut en principe que deux patches adjacents partagent leurs points de contrôle. Raffiner localement un objet conduit alors souvent à une explosion du nombre de points de contrôle sur l'ensemble de l'objet. Si en outre l'on veut des tangentes, voire

des courbures continues au niveau de la jointure, il faudra introduire des points de contrôle supplémentaires, rendant la manipulation de la surface encore plus malaisée.

Un autre apport des H-splines de [FB88] est l'introduction d'*overlays*. Ce sont des B-Splines de petite taille, posées sur la surface, qui seront interprétées comme des corrections qui viendront préciser localement la forme de la surface. Cela permet d'introduire des détails supplémentaires sans avoir à raffiner davantage les éléments initiaux. On peut combiner un nombre quelconque de ces surfaces supplémentaires, pour créer des détails aussi fins que nécessaires. Cette approche a cependant quelques défauts, elle n'est en particulier pas définie de manière unique, et peut devenir assez délicate à manipuler lorsque les raffinements se sont multipliés.

C. Gonzales-Ochoa et J. Peters trouvent d'autres défauts à cette représentation, citant entre autres l'impossibilité de gérer autre chose que des surfaces en "damier". Ils proposent dans [GOP99] d'utiliser directement un maillage comme polygone de contrôle, la surface elle-même étant construite à partir de triangles de Bezier construits sur les polygones du maillage. Il est par ailleurs possible, lorsque l'on veut introduire localement des détails supplémentaires, de raffiner localement le maillage de la surface. Cette méthode présente en outre l'avantage de pouvoir travailler avec n'importe quel type de maillage, que l'on peut éditer au besoin. Ainsi, il est possible, en corrigeant manuellement le maillage, d'obtenir des changements de topologie, ainsi que le montre la figure 1.10.

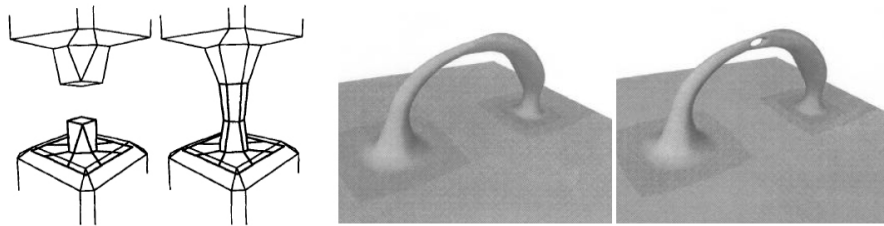


FIG. 1.10 – Les changements de topologie sont possibles dans [GOP99], mais gérés par l'utilisateur lui-même qui doit rectifier le maillage.

1.4.2 Les surfaces d'optimisation

Principe

Les surfaces d'optimisation sont une extension naturelle des surfaces de forme libre, comme les surfaces splines. Pour créer ces surfaces, l'utilisateur définit d'abord un ensemble de contraintes qu'elles devront vérifier. Ces contraintes peuvent par exemple être des courbes spatiales sur lesquelles la surface devra prendre appui, des tangentes, etc. L'idée consiste à préciser les endroits les plus intéressants de la surface, comme des bords, des singularités, des arêtes ou bien encore des sommets. Le reste de la surface sera calculée de façon à minimiser un critère, une énergie, qui prend généralement en compte l'élongation (on cherche une surface qui joint les contraintes de la façon la plus directe possible) et la courbure (on recherche avant tout des surfaces lisses).

H. P. Moreton et C. H. Sequin arguent dans [MS92] qu'il conviendrait plutôt de minimiser

la variation de la courbure, de préférence à la courbure proprement dite, afin de pouvoir librement obtenir des formes simples comme des cylindres, des cônes ou des tores dont la courbure n'est pas nulle. Leur modeleur travaille par ailleurs plutôt sur un maillage que sur des contraintes discrètes.

Déformation de la surface

Si la surface ne répond pas exactement aux attentes, il est éventuellement possible de jouer sur les différents termes présents dans la fonctionnelle à minimiser. Afin de rendre cette modification suffisamment intuitive pour l'utilisateur, G. Celniker propose d'ajouter des poids et des forces supplémentaires [CG91, CW92] qui feront évoluer la surface. Dans ces articles sont introduits plusieurs outils de sculpture, qui permettent d'agir librement et directement sur la surface.

La difficulté, dans ces approches, tient parfois à la description choisie pour la surface. Dans [CG91], la solution est calculée à partir d'éléments finis sur un maillage triangulaire. Dans [CW92], c'est au contraire une surface B-spline qui est utilisée. Afin de pouvoir générer des déformations de toutes tailles sans difficulté, un schéma de raffinement adaptatif similaire à celui des *overlay* de [FB88] est suggéré dans [WW92]. Dans ce dernier article, ce sont directement les contraintes (points et courbes) qui sont déplacées par l'utilisateur lorsqu'il souhaite déformer sa surface.

1.4.3 Les surfaces de subdivision

Difficile de ne pas dire également un mot des surfaces de subdivision, qui, en raison de leur souplesse et de leur capacité à construire des formes complexes et néanmoins lisses, connaissent un succès certain ces dernières années. Les surfaces de subdivision reposent sur un principe simple : on part d'un maillage polygonal d'un objet, de topologie et de connectivité quelconques, et on va le raffiner de façon itérative jusqu'à obtenir une forme lisse.

De nombreux schémas de subdivision ont été proposés, répertoriés pour la plupart par D. Zorin *et al* dans [ZSD⁺00]. Le schéma le plus utilisé est peut-être celui de *Catmull-Clark*, détaillé par exemple dans [HKD93]. La subdivision est réalisée en trois étapes. Dans un premier temps, de nouveaux sommets sont créés au centre de chacune des faces du maillage polygonal. Puis, chacune des arêtes est coupée en deux via l'adjonction d'un nouveau sommet. La position des sommets originaux, enfin, est mise à jour en fonction de ces nouveaux points. Le processus est ensuite réitéré, et converge vers une surface lisse.

Très heureusement, l'évaluation directe de la surface est possible, et il n'est pas indispensable de procéder, en pratique, à la subdivision récursive de la surface. Dans le cas d'un modèle formé de polygones à quatre côtés, on retrouve en fait une surface correspondant à une B-spline cubique. J. Stam a montré dans [Sta98] que les surfaces de subdivision de Catmull-Clark pouvaient être directement évaluées même en présence de sommets extraordinaires de valence différente de 4.

Le schéma usuel des surfaces de subdivision tend à former des objets très lisses, et il est

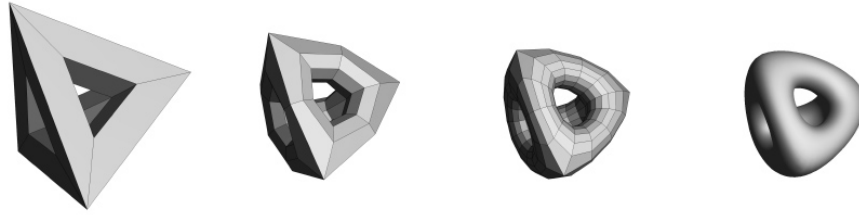


FIG. 1.11 – Subdivisions successives d'un maillage polygonal, présentées dans [DKT98].

quasi impossible d'obtenir des plis marqués sur la surface. Pour contourner cette difficulté, T. DeRose introduit dans [DKT98] des règles spéciales de subdivision, et associe une grandeur à chaque arête une valeur n . Durant les n premières étapes de subdivision, cette arête appliquera les règles spéciales de subdivision, plutôt que les règles habituelles, ce qui conduira à un pli d'autant plus marqué que n est grand, tout en conservant une surface parfaitement lisse (excepté dans le cas particulier où n est choisi infini). Les résultats sont présentés sur la figure 1.12.

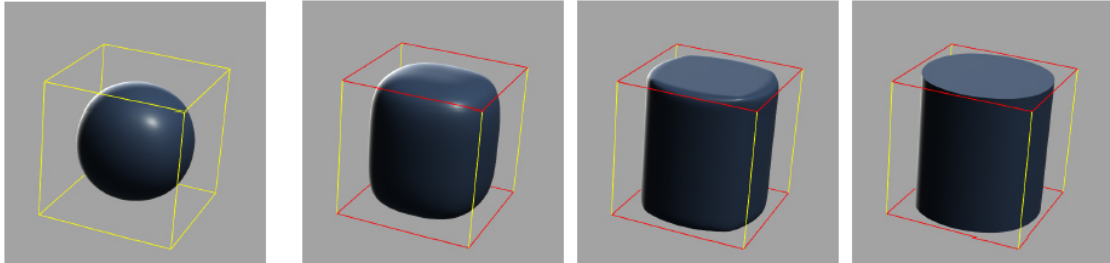


FIG. 1.12 – À gauche, le résultat usuel obtenu par la méthode de Catmull-Clark appliquée à un cube. À droite, les arêtes du haut et du bas se sont vues affecter des poids croissants selon [DKT98].

Les surfaces de subdivision présentent des avantages incontournables lorsque l'on souhaite créer des surfaces lisses et courbes. Elles se prêtent par ailleurs particulièrement bien à des schémas d'édition multi-résolution. A. Khodakovsky propose par exemple dans [KS99] une méthode permettant d'ajouter des détails fins, sous la forme d'extrusion de courbes tracées sur la surface de subdivision. Les surfaces de subdivision reposent toutefois toujours sur la manipulation d'un polygône de contrôle, et c'est à l'utilisateur de comprendre comment les modifications apportées à ces polygones vont se traduire sur la surface. Pour cette raison, les surfaces de subdivision ne correspondent pas à ce que nous recherchons, et nous ne nous pencherons pas plus avant sur les nombreux développements auxquels elles ont donné lieu.

1.4.4 Bilan

Les surfaces de forme libre et les surfaces de subdivision permettent donc de dépasser les limitations intrinsèques aux modèles polygonaux : il est possible, sans avoir besoin de multiplier la complexité des modèles, d'obtenir des surfaces parfaitement lisses et courbes. Plus coûteuses à utiliser dans un cadre temps réel, elles ne remplaceront pas de sitôt les modèles polygonaux pour tous les usages. Cependant, dans les domaines où le rendu n'a pas besoin d'être très rapide, comme par exemple le cinéma, elles sont devenues incontournables. Les splines ont également répondu à un besoin de l'industrie de conception qui avait besoin de pouvoir définir aisément des surfaces courbes avec un nombre limité de paramètres. Cependant, ces surfaces dépendent directement d'un polygone de contrôle, qu'on ne peut ignorer au moment de l'édition. De sorte qu'il n'est pas possible de modifier directement la forme d'une surface. Les éventuels changements de topologie, notamment, doivent être gérés manuellement, et si possible prévus dès le début.

Les surfaces d'optimisation sont un peu plus souples de ce côté, puisque l'on ne manipule pas des points de contrôle, mais des éléments directement placés sur la surface : courbes, plans tangents, etc. On peut ainsi modifier assez simplement la forme de la surface en déplaçant ces éléments de contrôle. Cependant, la forme de la surface entre ces éléments est définie automatiquement, et on ne peut agir dessus que de façon indirecte. Certes, il est possible d'ajouter des contraintes pour déplacer un point précis de la surface, mais les modifications ainsi occasionnées ne sont pas nécessairement intuitives, dans la mesure où la surface correspond à une optimisation globale d'un critère. L'ajout de contraintes supplémentaire rend cette optimisation plus difficile, mais peut aussi avoir des conséquences inattendues sur la forme de la surface à d'autres endroits plus éloignés. Cela demande donc un minimum d'habitude pour obtenir exactement la forme souhaitée.

1.5 Modélisation par surfaces implicites

1.5.1 Présentation des surfaces implicites

Il existe, dans le domaine des mathématiques, essentiellement deux moyens de décrire une surface (et par extension un volume). On peut employer une description paramétrique de la surface, dans laquelle un groupe d'équation permet d'obtenir l'ensemble des points constituant la surface en faisant varier deux paramètres. Les surfaces composées d'un ensemble de triangles, ou bien d'éléments de surface de Bezier, correspondent à cette description. Mais on peut aussi choisir, en mathématiques, d'employer une description algébrique d'une surface. Dans cette description, la surface correspond au lieu géométrique des points vérifiant une certaine équation, de type $f(x, y, z) = \text{constante}$.

Bien que courante et très usitée dans le domaine des mathématiques, cette description n'a été adoptée qu'assez tardivement dans le domaine de l'informatique graphique. Elle présente en effet un défaut important : la surface n'étant pas définie explicitement, son affichage pose quelques difficultés techniques. Quelques méthodes de rendu sont cependant venues à bout de ce problème, ce qui a permis aux surfaces implicites de devenir populaires. Nous revien-

drons ultérieurement sur ces méthodes de rendu, nous nous intéresserons ici uniquement aux questions de modélisation.

1.5.2 Expressions possibles du champ scalaire

Les surfaces (qui délimitent un volume fermé) sont décrites dans ce type de représentation comme une iso-surface d'une fonction scalaire f définie sur l'ensemble de l'espace. Reste à définir cette fonction f , ce qui peut être fait de plusieurs façons distinctes.

- La première est d'utiliser une forme explicite pour f , utilisant des fonctions de bases (nurbs, etc). Si le calcul de f est simple, il est en revanche plus délicat de manipuler directement la surface, puisque les modifications locales de la forme de la surface ne peuvent que rarement être exprimées simplement en fonction des paramètres des fonctions de base.
- Une seconde approche possible reprend et étend le concept de CSG. L'inconvénient des opérations de CSG est leur manque de continuité. On souhaiterait pouvoir assembler deux éléments en ayant une jointure avec de bonnes propriétés de continuité. Sous certaines conditions, il est possible d'opérer une sorte de "fusion" de deux volumes implicites i et j simplement en sommant leurs fonctions f_i et f_j . On peut donc construire simplement des surfaces complexes en fusionnant (ou bien en soustrayant) des formes de base, tels des ellipsoïdes ou des sphères. Cette approche permet très aisément de modifier la topologie d'un objet.
- Une dernière approche consiste à échantillonner la fonction f en divers points d'une grille régulière (le plus souvent sous la forme d'un maillage rectangulaire ou d'un octree), et d'interpoler les valeurs de f entre ces divers points d'échantillonnage. Cette représentation présente l'avantage de se prêter très bien aux modifications locales de la forme de la surface, de même qu'aux changements de topologie sans conduire à une augmentation du coût d'évaluation de f , comme c'est le cas de la fusion progressive de primitives.

1.5.3 Isovolumes exprimés directement

L'utilisation d'une expression explicite pour le champ f est *a priori* difficilement utilisable. On peut aisément obtenir quelques formes simples, comme des hyperquadriques, ou plus complexes, dont on peut librement choisir les paramètres. Réaliser un objet de forme complexe se révèle cependant beaucoup plus difficile, la base de fonctions à utiliser et les paramètres adéquats étant difficiles à établir.

On trouve cependant quelques outils de modélisation qui s'efforcent de procéder de cette façon. A. Raviv et G. Elber proposent par exemple de modéliser des objets décrits dont le champ scalaire est décrit par une somme de patchs constitués de fonctions B-splines à trois variables [RE99], décrites par une équation de type :

$$q(u, v, w) = \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n P_{i,j,k} B_i(u) B_j(v) B_k(w) \quad (1.1)$$

Un outil, de forme quelconque, peut venir ajouter ou retirer de la matière en agissant sur les différents coefficients $P_{i,j,k}$ pondérant chacune des fonctions B-spline. On peut ainsi librement créer un objet de forme et de topologie quelconques. C'est toutefois à l'utilisateur que revient la tâche délicate de créer et placer les différents patches, ce qui n'est pas toujours facile. C'est également à lui que revient la tâche de sélectionner le patch dont les coefficients seront mis à jour par l'action des outils.

K. Mc Donnell reprend dans une série d'articles [MQ00, MQW01, MQ02] le concept des B-splines triparamétriques et des surfaces de subdivision pour décrire un objet implicite, et propose des moyens d'édition plus variés pour déformer la surface. Ainsi, il est possible d'utiliser par exemple des outils physiques pour tirer ou repousser une partie de l'objet sculpté : des ressorts, attachés à la grille définissant les B-Splines, permettent de modéliser les déformations lorsque l'on tire sur un point de la surface, qui à son tour déplacera des sommets du maillage. Il est par ailleurs proposé un retour haptique aux opérations de sculpture. Pour changer la raideur des ressorts, et ainsi jouer sur le comportement physique de l'objet virtuel, les auteurs proposent simplement à l'utilisateur de peindre la surface.

D'autres outils sont également proposés : inflation locale de l'objet, peinture, etc. L'extrusion est rendue possible par l'ajout explicite de cellules, de même que les changements de topologie. Les limitations de cette approche tiennent à la présence d'un polygone de contrôle, même s'il est autant que possible caché à l'utilisateur. Ce dernier est cependant régulièrement amené choisir des cellules ou des facettes, par exemple durant les opérations d'extrusion ou de changement de topologie. Cette opération peut être délicate si le modèle n'a pas été conçu au départ pour présenter des arêtes aux bons endroits, ou lorsque le réseau a été très déformé.

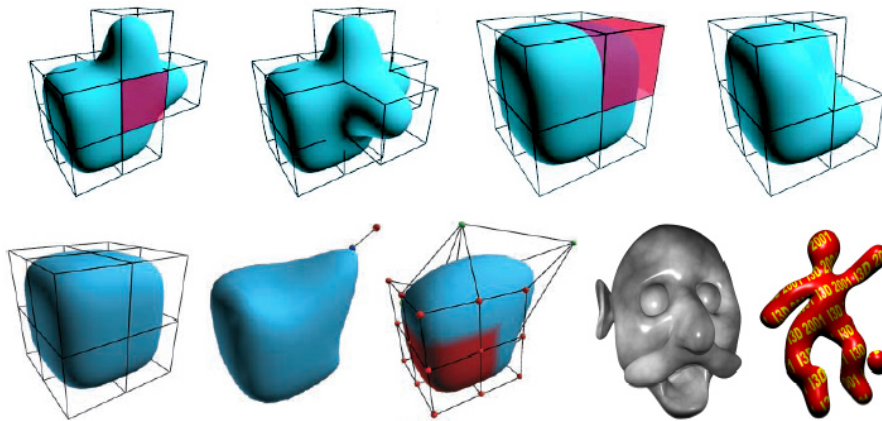


FIG. 1.13 – En haut, des opérations d'extrusion (ajout de cellule) et d'érosion (suppression) proposées par K. Mc Donnell *et al* dans [MQ02]. En bas à gauche, deux exemples de déformations d'un objet présentées dans le même article. En bas à droite, quelques réalisations obtenues au moyen du système proposé.

1.5.4 Fusion de volumes élémentaires

L'une des capacités intéressantes des surfaces implicites réside, on l'a dit, dans leur capacité à fusionner de façon lisse des volumes élémentaires. J. Blinn a été le premier à utiliser cette possibilité intéressante pour permettre le rendu de molécule complexes (en l'occurrence, des hélices d'ADN) [Bli82]. Son idée consiste à fusionner des sphères (une pour chaque atome de la molécule), décrites de façon implicite par une équation de la forme :

$$f_i(\mathbf{r}) = a e^{-b|\mathbf{r}-\mathbf{r}_i|^2} \quad (1.2)$$

où r_i correspond au centre de l'atome considéré.

En sommant ces différentes fonctions f_i , on crée très simplement une surface lisse consistant en la fusion de ces éléments, comme le montre la figure 1.14. Les paramètres a et b permettent de jouer à la fois sur le rayon de la primitive et sur la distance à partir de laquelle deux primitives vont commencer à fusionner. Les volumes ainsi reconstruits ont été qualifiés de “blobby models” ou “blobs”. La démarche a été rapidement adoptée par de nombreux modélisateurs, car elle permet de créer des volumes courbes et lisses de façon relativement simple, même si déterminer la position et les paramètres de chacun des éléments pour obtenir une surface déterminée est loin d'être quelque chose d'aisé.

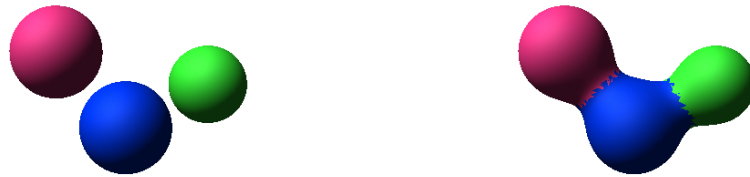


FIG. 1.14 – Exemple de surface implicite (à droite) obtenue par la fusion de trois éléments de base (à gauche).

Blobbies, Metaballs et Soft objects

Cette description présente dans de nombreux domaines l'inconvénient d'avoir une contribution non nulle de chacun des éléments de l'objet à une distance quelconque, de sorte que lorsque le nombre de primitives utilisées augmente, il en est de même pour le temps de calcul. Diverses variantes ont donc été proposées. Une première amélioration de cette fonction f_i combine deux morceaux de parabole afin de créer une fonction de forme similaire, mais qui s'annule au-delà d'une certaine distance.

$$f_i(\mathbf{r}) = \begin{cases} a(1 - \frac{3d^2}{b^2}) & \text{si } 0 \leq d \leq \frac{b}{3} \\ \frac{3a}{2}(1 - \frac{d}{b})^2 & \text{si } \frac{b}{3} \leq d \leq b \\ 0 & \text{si } b \leq r \end{cases} \quad \text{où } d = |\mathbf{r} - \mathbf{r}_i|. \quad (1.3)$$

L'idée générale est similaire, et s'est vue baptisée du nom de “Metaball”. Seule change en fait la façon de déduire f de la distance à la primitive. Une seconde suggestion a été faite

par les frères Wyvill afin d'éviter le calcul de r qui nécessite l'emploi d'une racine carrée. On parle alors de "Soft objects" [WMW86, WW89] :

$$f_i(\mathbf{r}) = \begin{cases} a(1 - \frac{4d^6}{9b^6} + \frac{17d^4}{9b^4} - \frac{22d^2}{9b^2}) & \text{si } d \leq b \\ 0 & \text{si } b \leq d \end{cases} \quad \text{avec toujours } d = |\mathbf{r} - \mathbf{r}_i|. \quad (1.4)$$

Les possibilités concernant les fonctions f_i sont nombreuses, et ce genre de description est très utile pour modéliser des objets déformables. Nous nous servirons ultérieurement de ce genre de description implicite pour créer un modèle volumique de main, avec des expressions un peu différentes pour les termes f_i , permettant de contourner quelques difficultés sur lesquelles nous ne nous attarderons pas pour l'instant.

Surfaces implicites à squelette

Dans les modèles précédents, les différentes expressions des termes f_i utilisent la distance à un point de l'espace. En fait, comme le suggèrent J. Bloomenthal et B. Wyvill dans [BW90], il est possible de considérer la distance à n'importe quel objet géométrique : un segment, une courbe, une surface... Le potentiel ainsi créé est simplement fonction de la distance entre le point P considéré et cette primitive S : $f_i(\mathbf{r}) = \mathcal{F}(\text{dist}(P, S))$. L'effet obtenu correspond simplement à un enrobage de la primitive. La forme de la fonction potentiel \mathcal{F} joue un rôle important dans le processus de fusion. Plus elle décroît lentement avec la distance, plus la fusion des différents éléments sera douce.

Plutôt qu'une simple somme des différentes contributions f_i , d'autres approches ont été proposées. Si la somme de contributions fournit une fusion lisse d'un ensemble d'éléments, les fonctions *max* et *min* permettent de retrouver les notions d'union et d'intersection propres aux CSG (évidemment, la jonction lorsque l'on utilise de telles fonctions est toujours continue, mais présente des discontinuités des tangentes). Ces différentes approches sont unifiées à travers les R-fonctions définies par A. Pasko *et al* dans [PASS95]. On pourra également citer la technique de mélange procédural introduite par J. Bloomenthal dans [BW90].

L'un des inconvénients de ces approches est que le résultat obtenu dépend du découpage du squelette. L'isosurface obtenue à partir d'un segment n'est pas la même que celle qu'on obtiendrait en combinant les deux moitiés de ce même segment. Pour obtenir cette propriété, J. Bloomenthal suggère de convoluer le squelette par la fonction potentiel \mathcal{F} [BS91]. Dans l'article en question, cette fonction potentiel s'exprime par $\mathcal{F}(d) = \exp(-d^2/2)$, et le champ scalaire obtenu, qui vérifie alors ce principe de superposition, vaut alors :

$$f(\mathbf{r}) = (\mathcal{F} * S)(\mathbf{r}) = \int_S e^{-\frac{|\mathbf{s}-\mathbf{r}|^2}{2}} d\mathbf{s}. \quad (1.5)$$

Sculpture avec des surfaces implicites à squelette

Éditer une telle surface consistera simplement à travailler sur les éléments du squelette. En les déplaçant, les redimensionnant, ou en jouant sur les fonctions potentiel \mathcal{F} utilisées, l'artiste peut librement modifier la forme de la surface. S. Mizuno *et al* propose par exemple de

modifier un objet implicite en ajoutant une série d'ellipsoïdes qui permettront d'en modifier la forme. Des primitives plus variées sont proposées par A. Pasko *et al* dans [PSS01], citons par exemple des modèles polygonaux, des courbes bidimensionnelles ou bien encore des cartes de profondeur.

Ces dispositifs de sculpture virtuelle, utilisant des surfaces implicites à squelette, permettent d'obtenir des volumes lisses assez variés, même s'ils nécessitent une certaine habitude, et sont limités essentiellement à de l'ajout et à de la suppression de matière.

A. Sourin propose également dans [Sou01] de réaliser des opérations similaires à des travaux d'estampage. Les déformations subies par le métal virtuel sont de nature purement géométrique, obtenue par la modification d'une surface implicite en lui ajoutant des termes certes un peu plus complexes que la simple distance à une primitive, mais basés par ailleurs sur la même approche. Quelques résultats sont présentés sur la figure 1.15



FIG. 1.15 – Quelques travaux d'estampage virtuel proposés par A. Sourin dans [Sou01].

Dans [Can93], M.-P. Cani propose une méthode pour modéliser les déformations subies par des objets basés sur des surfaces implicites à squelette lors de collisions. Celles-ci sont interprétées comme des déformations locales appliquées au solide par rapport à leur forme au repos. Elles sont calculées en introduisant des potentiels supplémentaires de collision, permettant de maintenir un parfait contact entre les deux solides implicites lors de la collision, sans interpénétration des deux volumes, tout en dilatant ces mêmes volumes à d'autres endroits afin de maintenir constant le volume global des objets.

On peut ainsi modéliser de façon convaincante les interactions entre un matériau implicite et un outil. Le caractère local des déformations ne permet pas, toutefois, d'envisager la sculpture entière d'un objet grâce à cette approche, mais davantage des corrections locales de la forme d'un objet.

Ces idées sont reprises et étendues dans [OC97], où l'interaction est traitée de façon légèrement différente, afin de diminuer la quantité de calculs à effectuer, et pouvoir prendre en compte la direction du mouvement de l'outil dans le calcul de la déformation. La possibilité d'utiliser ce modèle de déformation dans le cadre d'un outil de modélisation est envisagée. Les surfaces implicites à squelettes n'étant toutefois pas adaptées à des modifications successives et importantes de la forme (le calcul de la fonction potentiel serait de plus en plus difficile à conduire au fur et à mesure que viennent s'ajouter des termes supplémentaires), l'utilisation de surfaces implicites définies sur une grille discrète est alors envisagée.

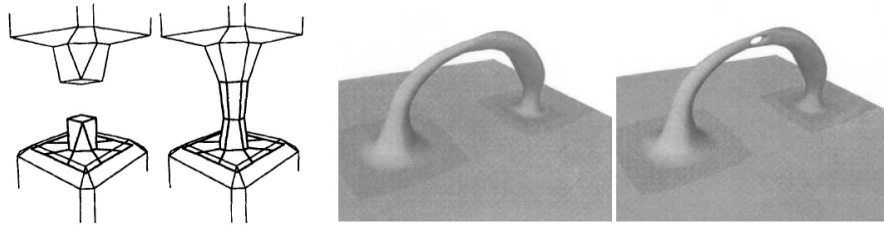


FIG. 1.16 – Déformations d'un volume implicite sous l'action d'un outil arrivant de différentes directions d'après [OC97].

1.5.5 Des surfaces implicites discrétisées

Principe et premières approches

Les descriptions précédentes d'une isosurface ne sont pas faciles à manipuler. Dans le cas de surface implicite à squelette, chaque ajout de matière requiert l'adjonction d'une primitive supplémentaire, alourdissant d'autant les calculs destinés à l'affichage. Ces calculs sont d'autant plus importants que certaines des approches précédentes utilisent par exemple un mélange basé sur les R-fonctions de [PASS95], plutôt qu'une simple somme. Le rendu de la surface nécessite lui aussi une quantité importante de calculs.

Pour lever cette difficulté, T. Galyean et J. Hughes proposent de s'affranchir totalement de la notion de squelette, et échantillonnent le champ scalaire utilisé pour la description de l'isosurface sur une grille régulière [GH91]. Les valeurs du champ entre les différents nœuds de la grille seront simplement interpolés. La sculpture de l'objet consistera à faire évoluer les valeurs du champ stockées sur les nœuds de la grille.

Les auteurs proposent divers outils, qui ont pour effet de faire augmenter les valeurs du champ à l'intérieur du volume de l'outil, conduisant à de l'ajout de matière, ou bien en les faisant décroître, ce qui a pour effet visible de retirer de la matière. Ils proposent également des outils de lissage de la surface obtenue, qui fonctionnent en filtrant les valeurs du champ scalaire. La même démarche est utilisée par S. Wang et A. Kaufman dans [WK95], qui proposent d'autres outils pour interagir avec le matériau, permettant notamment à l'utilisateur d'effectuer des opérations d'extrusion. R. Avila et L. Sobierajski introduisent également leurs propres outils, comme la possibilité de peindre le matériau, dans [AS96]. Ils proposent en outre une démarche pour calculer un retour d'effort lors de la sculpture.

Outre les outils classiques d'ajout et de suppression de matière, E. Ferley propose dans [FCG99] des outils permettant de simuler des empreintes dans le matériau. Ces empreintes ne sont pas le résultat d'une quelconque conservation du volume lors de l'interaction avec un outil solide, mais obtenus de façon purement géométrique, avec des outils qui retirent de la matière à l'intérieur tout en faisant croître les valeurs du champ à proximité immédiate, créant ainsi des bourrelets de matière.

Autre interprétation du champ scalaire

Dans [PF01a], R. Perry et S. Frisken donnent une autre signification au potentiel scalaire, celui de la distance euclidienne à la surface de l'objet. Cette description est plus proche de ce que l'on trouve dans le domaine des *levelsets*, qui sont pour l'essentiel équivalents aux surfaces implicites que nous décrivons. Ce choix impose quelques difficultés lors de la mise à jour, qui n'est plus locale comme c'était le cas précédemment. L'article propose plusieurs solutions pour les contourner, en travaillant en particulier essentiellement au niveau de la surface. Par ailleurs, le champ scalaire ainsi obtenu est continu, mais présente des discontinuités dans les dérivées qui peuvent occasionnellement poser problème.

Plusieurs opérations sont possibles pour permettre la sculpture du matériau. Il est d'abord donné à l'utilisateur la possibilité d'agir directement sur la surface. Pour ce faire, la surface est triangulée sur un voisinage de la zone d'intérêt. L'utilisateur agit alors sur les différents triangles, et le champ de distance est mis à jour. L'artiste peut également opérer des opérations directement sur le champ de distance, dont les effets sont globalement similaires à des opérations de CSG.

Une structure multirésolution

La structure régulière de la grille se prête bien à une représentation multirésolution du problème. Cela permettrait en effet de n'affiner la grille qu'aux endroits où le besoin s'en fait sentir. Une première approche basée sur l'usage d'une octree pour ranger les différentes valeurs échantillonnant le champ est suggérée par J. Bærentzen [Bær98]. Il propose également de décrire les outils sous la forme d'un champ scalaire, afin d'éviter les éventuels problèmes d'aliasing causés par le passage brusque de l'intérieur du volume d'un outil à l'extérieur. L'édition se passe cependant toujours au niveau maximal de détail, et la grille ne peut être que raffinée. Il n'y a pas de mécanisme permettant de la simplifier là où les divisions ne sont plus utiles.

E. Ferley reprend cette approche multirésolution pour proposer un outil complet multirésolution de sculpture virtuelle [Fer02]. Seules les cellules occupées sont effectivement stockées, et une table de hachage permet de retrouver aisément les cellules dans lesquelles s'effectuent les ajouts et retraits de matière. L'utilisateur peut ainsi modéliser un objet sans aucune contrainte de taille ou de résolution. Grâce à une application hiérarchique des outils, il est en outre possible d'appliquer un rendu adaptatif de la surface modélisée.

1.5.6 Bilan

L'un des principaux avantages des surfaces implicites dans le cadre de la sculpture virtuelle réside dans leurs capacités à accepter n'importe quel changement de topologie. Il est en effet aisé de fusionner plusieurs morceaux, découper un objet, réaliser des trous, sans avoir à se préoccuper de la représentation sous-jacente de la matière, comme c'est généralement nécessaire. La surface peut être modifiée très simplement au niveau local, par ajout ou suppression de matière, en particulier dans les modèles où la fonction implicite est définie sur

une grille discrète.

La plupart des modèles se limitent cependant à de telles opérations d'ajout ou de suppression de la matière, purement géométriques. Les empreintes proposées par le modèle de E. Ferley [Fer02] fournissent une solution géométrique intéressante pour modéliser les effets de la collision du matériau avec un outil. Il serait par ailleurs envisageable de calculer des déformations locales assurant la conservation du volume. Les modèles présentés ne permettent cependant pas d'obtenir simplement des déformations à grande échelle, correspondant par exemple au repliement d'une partie de l'objet.

1.6 Sculpture par déformation de l'espace

Nous avons vu précédemment qu'il existait de nombreuses façons de décrire des surfaces et des volumes. Nous allons à présent nous pencher sur des outils qui permettent de déformer un objet sans se préoccuper de la représentation choisie pour décrire celui-ci. Dans la suite, nous nous pencherons sur des approches qui permettent de déformer un objet de façon indépendante de sa représentation.

L'idée générale des sculptures par déformation de l'espace consiste à créer des transformations sur tout l'espace qui représenteront les déformations subies par les objets. Ces transformations définissent plus précisément, pour tout point de l'espace, la nouvelle position qu'il devra occuper. On choisit généralement des transformations bijectives, ce qui permet de s'assurer que les modifications subies par l'objet sont réversibles. Si ce n'est pas le cas, deux points distincts de la surface de l'objet peuvent être amenés en un même point de l'espace, et ne pourraient plus être séparés, ce qui peut se révéler gênant.

1.6.1 Utilisation d'opérateurs de déformation

Une première approche sur ce principe a été proposée par A. H. Barr [Bar84]. Elle consiste à déformer l'espace au moyen d'opérateurs simples, permettant toute une gamme de flexions, torsions et déformations tant au niveau local que global. La manipulation de matrices pour les déformations n'est cependant pas des plus aisées pour l'utilisateur. Il n'est pas toujours facile de construire l'opérateur qui permettra l'extrusion ou la modification d'un endroit précis de l'objet.

De façon à ce que l'utilisateur ait davantage de contrôle, il est utile de "localiser" ces déformations. P. Borrel et A. Rappoport proposent par exemple à l'utilisateur de sélectionner un point dans l'espace, et un rayon d'influence pour la transformation [BR94]. La translation du point sélectionné est transmise aux points situés dans la sphère d'influence centrée autour de ce point. Elle est modulée (par une fonction B-spline) lorsque l'on s'écarte du point, jusqu'à s'annuler aux limites de la sphère, afin que la transformation obtenue soit continue.

Bien entendu, comme le montre B. Crespin dans [Cre99], on n'est pas limité à la seule translation d'un point. On peut envisager dans ce même volume d'influence d'autres transformations, comme des rotations. Le volume d'influence, quant à lui, n'a pas à avoir de forme particulière. N'importe quelle fonction d'influence, permettant de différencier une partie de



FIG. 1.17 – Quelques objets sculptés par déformations de l'espace, présentés dans [Bar84].

l'objet, peut être employée. Si de plus l'influence varie de façon continue, la transformation de l'espace le sera également, et la déformation préservera la continuité de l'objet.

1.6.2 Les volumes FFD (Free Form Deformations)

Les volumes FFD ont apporté une solution similaire et intéressante, un peu plus systématique. On enveloppe l'objet d'un volume parallélépipédique, qui servira de volume de contrôle. Ce volume sera ensuite déformé par l'utilisateur pour appliquer des modifications à l'objet. La déformation à l'intérieur du parallélépipède peut directement être déduite du mouvement de ses sommets.

Contrairement au cas d'une déformation obtenue par le mouvement d'un squelette, le fait que chaque point de l'objet édité est entouré par huit sommets qui l'entourent permet de s'affranchir du délicat problème des poids. La solution la plus simple consiste à prendre, pour chaque point à l'intérieur du volume de contrôle, un mouvement qui sera une interpolation linéaire des mouvements des sommets du volume de contrôle, les poids étant directement définis par les coordonnées du point à l'intérieur du volume, ou par l'intermédiaire des polynômes de Bernstein [SP86]. On obtient ainsi des volumes paramétriques de Bezier, qui permettent de déformer notre modèle de façon similaire à ce que l'on peut faire avec une surface spline.

Diverses extensions de ce principe ont également été proposées. S. Coquillard remarque dans [Coq90] qu'il peut être utile de choisir des volumes de contrôle autres que des parallélépipèdes rectangles. Ceci permet d'avoir plus de souplesse dans la réalisation des déformations, qui sont réalisées en deux étapes distinctes. Dans un premier temps, l'utilisateur construit un réseau dont la forme est adaptée au type de déformation qu'il souhaite appliquer à l'objet, et positionne ce réseau là où il souhaite appliquer les modifications à l'objet. Puis l'objet est effectivement lié au réseau de déformation, et l'utilisateur peut utiliser les points de contrôle de ce dernier pour appliquer des modifications à l'objet sculpté.

La gamme des effets qu'il est possible d'obtenir de cette façon est bien plus riche, comme on peut le voir sur la figure 1.19. Il est possible de créer des réseaux de déformation de topologie complexe, permettant d'obtenir d'autres types de déformations, moyennant quelques

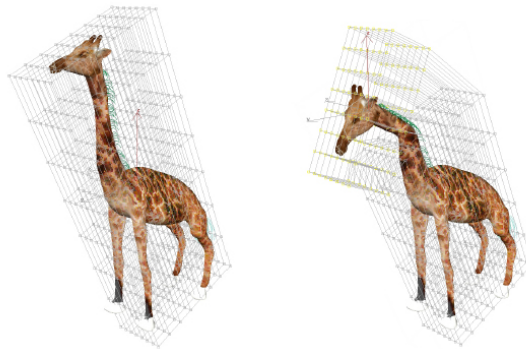


FIG. 1.18 – Déformation d'un modèle de girafe au moyen des FFD [Deb00].

précautions [MJ96]. Il n'est en revanche pas possible de modifier la topologie de l'objet au moyen de ces techniques, qui induisent des transformations continues de la surface.

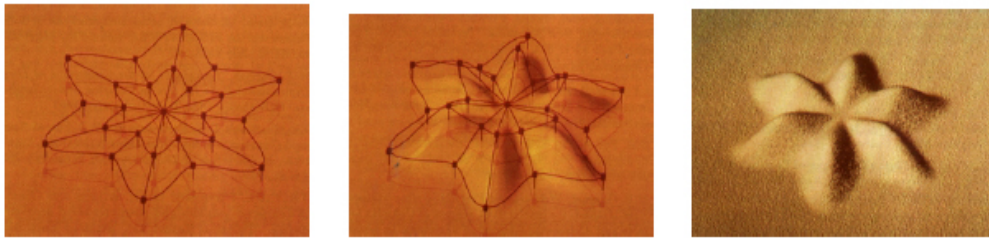


FIG. 1.19 – Un exemple de déformation FFD basée sur un réseau non parallélépipédique [Coq90].

L. Moccozet et P. Kalra vont plus loin en proposant une méthode qui rend inutile la création du réseau lui-même [MT97]. Plutôt que de le définir explicitement, l'utilisateur choisit simplement librement dans l'espace un certain nombre de points de contrôle qui serviront à déformer l'objet. Les déformations sont alors gérées de façon automatique. Pour chaque point de la surface qu'il faudra déplacer, on calcule la contribution des différents points de contrôle. Les points de contrôle effectivement utilisés seront ceux qui participent à la création d'un volume de Voronoï autour du point étudié. Son mouvement sera une combinaison linéaire du mouvement des points de contrôle, dont les poids sont les coordonnées de Stibon. Les auteurs montrent dans l'article comment définir un simplexe de Bezier dont les coordonnées coïncident avec les coordonnées de Stibon, permettant ainsi un calcul aisé des déformations de la surface à partir du mouvement des points de contrôle.

La manipulation des points de contrôle, enfin, présente le même inconvénient que dans le cas de surfaces de Bezier : il n'est pas toujours facile de déformer une surface à partir de points situés à quelque distance de celle-ci. W. M. Hsu propose donc, comme pour le cas des surfaces de Bezier, de directement agir sur des points de la surface proprement dite [HHK92]. Lorsque l'utilisateur déplace un point situé à la surface, le mouvement de chacun des points de contrôle du volume FFD est calculé de façon à obtenir le déplacement requis. Bien évidemment, le problème du déplacement des points de contrôle admet une infinité de solutions, les auteurs privilégiant celle pour laquelle leur déplacement est minimal.

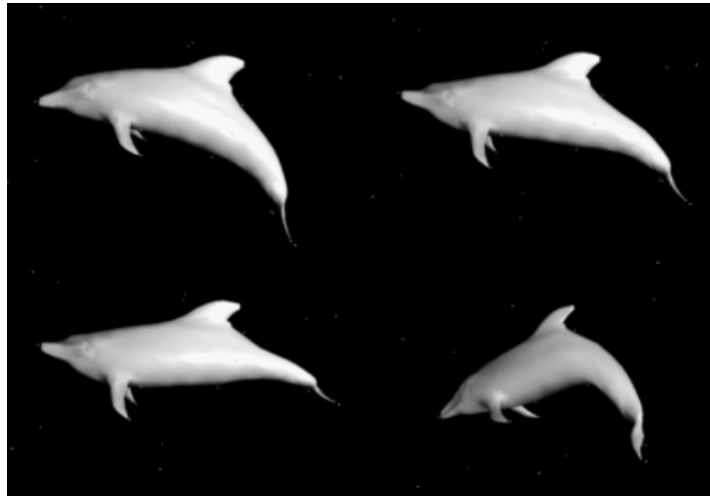


FIG. 1.20 – Quelques déformations d’un modèle, présentées dans [MT97].

Une approche similaire, baptisée du nom de “*wires*”, a été proposée par K. Singh *et al* dans [SF98]. Elle consiste à déposer sur la surface de l’objet des “fils” plutôt que des points de contrôle. Le déplacement de ces fils, ensuite, entraîne la surface qui reste toujours en contact avec ces derniers. Il est possible, à travers plusieurs paramètres, de mieux contrôler la déformation proprement dite. Cette méthode est à rapprocher, dans une certaine mesure, des surfaces d’optimisation, dans la mesure où la déformation de l’objet doit impérativement passer par ces courbes, mais elle est appliquée ici au principe des déformations.

1.6.3 Déformations à volume constant

D’un point de vue intuitif, pour la sculpture, la conservation du volume est une caractéristique intéressante. Les auteurs de [SP86] précisent également qu’il est envisageable d’estimer les variations de volume induites par les FFD, et envisagent le cas de déformations conservant le volume de l’objet. Pour ce faire, il faut que la divergence du champ de déformation soit nulle en tout point, ce qui, en pratique, impose à la matrice jacobienne correspondante d’être de trace nulle.

Des déformations de type FFD préservant le volume ont été proposées entre autre par G. Hirota *et al* dans [HML99]. La conservation du volume est assurée, en pratique, grâce à un mécanisme d’optimisation qui estime les variations de volume lorsque les polygones composant l’objet se déplacent. Un traitement multi-résolution est par ailleurs employé afin de garantir l’interactivité des déformations.

Une autre approche a été proposée par A. Angelidis [ACWK04], dans laquelle des opérateurs de torsion de l’espace sont combinés afin de proposer un outil intuitif de déformation d’une surface tout en préservant son volume : c’est un cas particulier de l’approche initiée par A. H. Barr dans laquelle l’opérateur a une divergence nulle en tout point de l’espace. En pratique, l’utilisateur peut tirer ou pousser sur la surface de l’objet le long d’un chemin, et l’objet est déformé en conséquence.

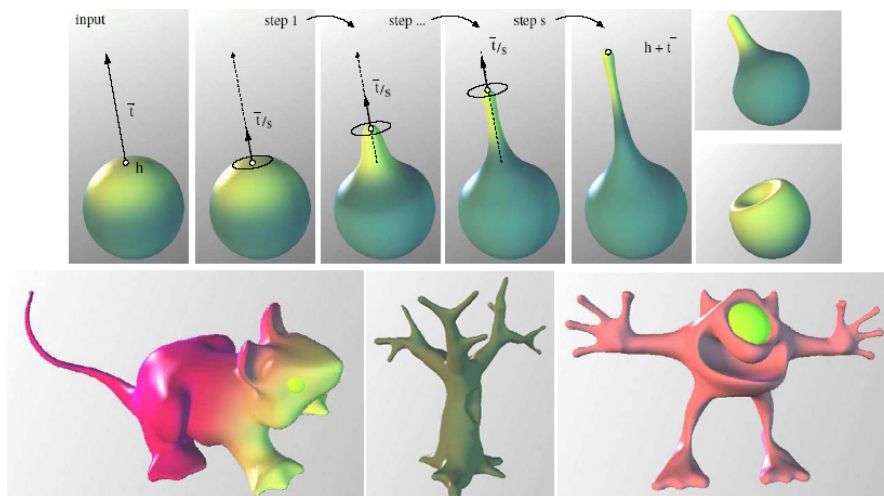


FIG. 1.21 – En haut, l’outil de déformation à volume constant présenté dans [ACWK04]. En dessous, quelques exemples de réalisations obtenues avec cette méthode.

1.6.4 Bilan

Grâce à ces différentes méthodes, il est relativement aisé de déformer un objet de façon quelconque. On imagine ainsi aisément comment on peut changer la pose d’un personnage avec ce type d’approches. Malheureusement, ces méthodes ont aussi de sérieuses limitations.

La première tient à la topologie de l’objet. Ces méthodes évitent en général, par construction, toute collision (ce qui permet par ailleurs de revenir en arrière à tout moment). Il n’est donc pas question de changer la topologie d’un objet. Lorsque des collisions sont possibles, les résultats dépendent la représentation choisie pour l’objet. Dans le cas d’une représentation polygonale, par exemple, on obtiendra des surfaces qui s’interpénètrent, et il faudra détecter ces collisions pour corriger le maillage. Il en est de même pour d’autres représentations, ces problèmes devront être traités au cas par cas.

La deuxième limitation tient au réseau de déformation lui-même, pour les méthodes qui l’utilisent. Il est difficile de construire un réseau de façon automatique qui permette les modifications que l’on envisage pour l’objet, en particulier si sa topologie est complexe. Le réseau doit être suffisamment fin pour prendre en compte les détails de l’objet (on veut par exemple pouvoir déformer de façon indépendante les deux jambes d’un personnage, ce qui suppose qu’aucun nœud n’ait d’influence sur les deux jambes en même temps). Il est fréquent qu’il soit nécessaire de créer ce réseau soi-même, en fonction de la déformation que l’on recherche. Par ailleurs, un réseau très fin suppose beaucoup de nœuds dont il faut préciser le déplacement. Là encore, ce n’est pas un problème simple.

1.7 Modèles à base d'esquisses

1.7.1 Premiers pas

Une dernière approche pour la modélisation de formes tridimensionnelles est basée sur la constatation suivante : nous sommes capables d'imaginer la forme d'un objet à partir d'une ou de quelques projections bidimensionnelles. Si l'ordinateur est capable d'en faire autant, alors peut-être sera-t-il plus facile de créer une forme en dessinant quelques esquisses correspondant à des projection intelligemment choisies.

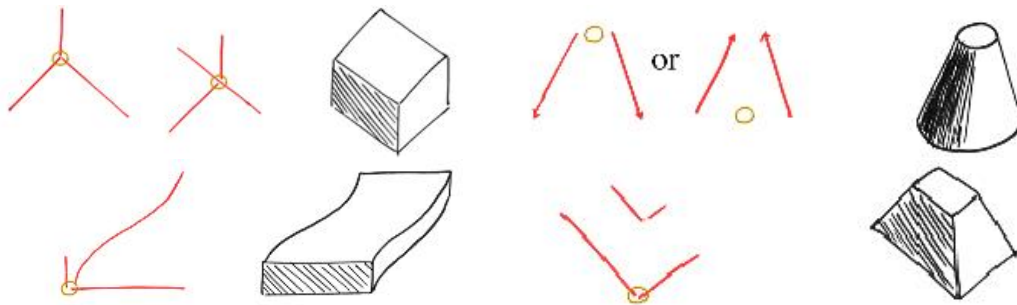


FIG. 1.22 – Quelques signes pour la création de volumes simples avec SKETCH.

L'une des premières implémentations de cette idée est proposée par R. Zeleznik, K. Herdon et J. Hughes [ZHH96]. Leur interface, baptisée SKETCH, propose de créer un ensemble de formes simples (cubes, cônes, sphères) ou extrudées à partir de contours, simplement avec quelques signes d'édition tracés à la souris. Par défaut, les tracés sont des lignes droites alignées sur les axes du repère. Un temps d'arrêt permet de tracer des courbes quelconques, qui sont projetées sur les objets de la scène si l'utilisateur presse la touche "majuscule" durant le tracé. D'autres signes d'édition permettent de manipuler la caméra, déplacer les objets ou encore réaliser quelques opérations basiques d'édition.

Ces bases sont reprises par Igarashi *et al* dans une interface baptisée CHATEAU [IH01]. Là encore, on construit essentiellement des surfaces planes, généralement parallèles aux axes principaux du repère. De très nombreux signes d'édition viennent aider la modélisation, permettant de réaliser des chanfreins, tailler des surfaces ou encore égaliser les proportions des différents éléments.

Ces approches ont l'immense avantage de permettre de réaliser très rapidement et aisément des surfaces simples, dès que l'on en a compris les principes fondateurs et appris les quelques commandes. Évidemment, elles sont restreintes aux formes simples, créées par CSG ou extrudées. En contrepartie de la simplicité de création, on doit faire quelques concessions à la généralité des surfaces et volumes qui peuvent être réalisés.

1.7.2 Création de volumes courbes

T. Igarashi a également proposé une interface nommée TEDDY permettant cette fois de réaliser, avec une approche similaire, des volumes 3D comportant principalement des surfaces

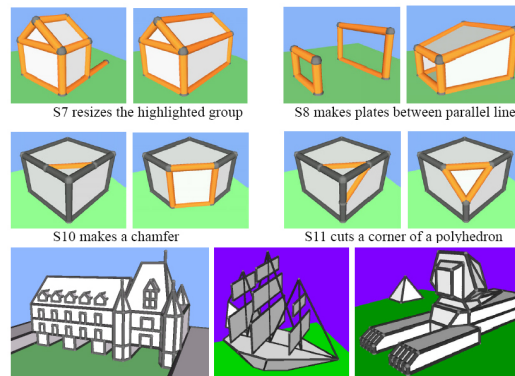


FIG. 1.23 – Divers signes d'édition proposés par CHATEAU, et quelques résultats de modélisation.

non planes [IMT99]. Une courbe fermée permet de créer un premier potatoïde qui servira de base (un carré donnera non pas un cube mais une sorte de coussin, tout se passe comme si la courbe fermée “gonflait”). Ensuite, on développe la forme de plusieurs façons. La principale action consiste à créer des extrusions : dans un premier temps, on trace une courbe fermée sur le volume actuellement à l'écran, qui servira de support. Une seconde courbe, dont les deux extrémités prennent appui sur ce premier contour, permet d'effectivement créer un volume dont on a tracé une silhouette.

De nombreux outils sont proposés pour modifier l'objet : découpes, déformations basées sur un squelette, etc. Toutes sont basées sur des signes d'édition simples. La réalisation des volumes imaginés nécessite une bonne compréhension de la façon dont fonctionne l'interface, et une astucieuse manipulation de la caméra pour choisir le point de vue le plus adapté. Il est parfois difficile de maîtriser l'épaisseur des formes ainsi créées. L'auteur reconnaît également que des volumes inattendus peuvent être obtenus si la courbe d'extrusion est inhabituelle, ou si le contour d'appui n'est pas suffisamment plat.

Une approche similaire a été suivie par O. Karpenko [KJFH02], qui suggère l'utilisation de surfaces implicites pour remplacer les modèles polygonaux utilisés par TEDDY, afin d'obtenir des surfaces de meilleure qualité. La forme de base est toujours un “blob” tracé par l'utilisateur sous la forme d'un contour fermé. L'interface propose de déplacer les blobs s'ils sont mal placés (en utilisant l'ombre de la primitive comme indice supplémentaire pour le placement), ou bien encore de retoucher leur forme si elle est incorrecte par un simple signe d'édition. Les différents blobs sont fusionnés sous la forme d'une isosurface, en respectant une hiérarchie qui est devinée par le système d'après le placement des différents blobs, et peut être au besoin corrigée par l'utilisateur.

A. Alexe propose également un système de modélisation à base d'esquisses utilisant des surfaces implicites [AGB04]. Plutôt que de construire un ensemble de blobs, leur approche consiste à déterminer un squelette à partir des méthodes de Delauney, puis de construire sur ce squelette un volume dont la silhouette correspond au tracé de l'utilisateur. Les auteurs proposent également quelques signes d'édition, et discutent des moyens d'obtenir la surface la plus lisse et agréable possible.

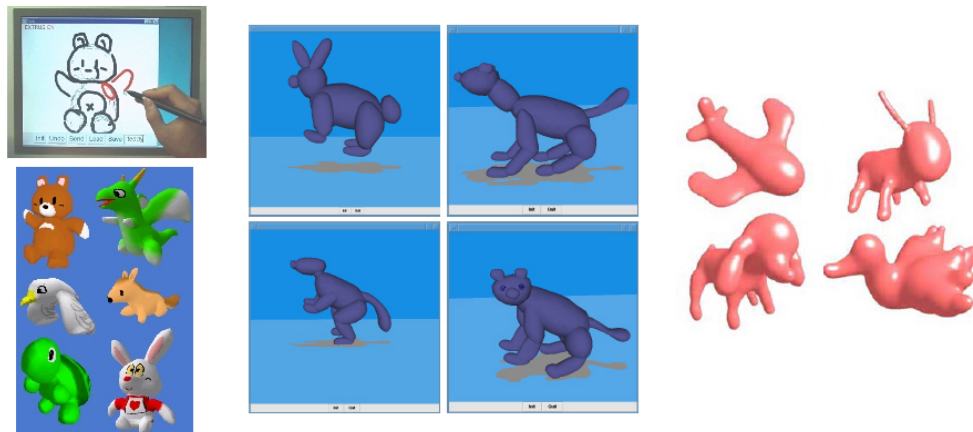


FIG. 1.24 – Quelques modèles 3D obtenus par les systèmes précités. À gauche, l'interface de TEDDY, et quelques réalisations. Au centre et à droite, les travaux de O. Karpenko et de A. Alexe utilisent des volumes implicites plutôt qu'un maillage polygonal.

1.7.3 Des esquisses pour déformer les maillages

Les interfaces basées sur des esquisses présentent d'indéniables avantages d'intuitivité, même si elles sont souvent sous-contraintes. En conséquence, l'utilisateur n'a pas toujours un plein contrôle sur certains paramètres (l'épaisseur des volumes esquissés par exemple, A. Alexe utilise pour cela l'inclinaison du stylet), ou bien encore certaines hypothèses supplémentaires doivent être faites pour pouvoir inférer le volume 3D (privilégier les axes principaux du repère, par exemple).

Ces approches ne sont donc probablement pas adaptées à la réalisation de n'importe quel volume, mais l'approche est intéressante. Le principe des esquisses peut également être utilisé pour déformer intuitivement et rapidement des modèles existants. C'est la suggestion faite par A. Nealen *et al* dans [NSACO05]. L'utilisateur sélectionne une partie du maillage, choisit une vue, et esquisse d'une simple courbe la nouvelle silhouette que devrait avoir la zone sélectionnée.

Là encore, au prix d'une légère concession au contrôle (le système doit faire quelques hypothèses sur un problème sous-contraint), on gagne infiniment en simplicité, et les résultats sont plutôt convaincants.

1.7.4 Bilan

La modélisation par esquisses est une approche récente et nouvelle du concept de modélisation 3D. Elle a pour elle un avantage certain au niveau de la simplicité. Un certain nombre de formes peuvent être réalisées très simplement. D'autres, en revanche, vont nécessiter une bonne compréhension du fonctionnement de l'interface. Avec TEDDY, par exemple, il est facile d'extruder un objet, par exemple pour ajouter un bras à un personnage. En revanche, tracer une anse est particulièrement difficile, comme certains changements de topologie.

De façon générale, ce genre d'approche tente de "deviner" l'information manquante dans une esquisse. En fonction des situations, elle y parvient plus ou moins bien. Divers signes d'édition viennent aider le logiciel à comprendre ce que souhaite l'artiste, même si, au passage, on perd une partie du côté naturel de l'édition.

1.8 Modélisation 3D pour le grand public

La modélisation d'une forme tridimensionnelle n'est pas quelque chose qui a pu être popularisé auprès du grand public, et reste réservée à un public restreint. Il est devenu relativement aisé de se servir d'un traitement de texte, d'élaborer une feuille de calcul, d'éditer un fichier sonore. N'importe qui peut, depuis plus d'une vingtaine d'années, exercer ses talents artistiques avec l'un des nombreux logiciels de dessin disponibles sur le marché. Force est de constater que le domaine de la 3D, même s'il génère un engouement certain, n'est en revanche pas d'abord aussi aisé.

Quelques efforts ont pourtant été faits pour démocratiser ce domaine. On trouve quelques modélisateurs assez intuitifs, résolument destinés au grand public, comme par exemple un logiciel appelé *Polygon Maker*. Les actions disponibles s'apparentent cependant essentiellement à des assemblages de formes basiques et à la manipulation de sommets, et nous avons déjà souligné les limitations de ce genre d'approche dès que l'on souhaite créer des formes complexes, en particulier des formes rondes et organiques. Une suggestion a également été faite, en mai 2004, de l'usage possible d'un écran tactile et d'un stylet pour proposer un outil de modélisation volumétrique relativement simple. Le système s'apparente à un tour, et il est possible d'utiliser le stylet afin de retirer localement de la matière à l'objet.

L'application la plus avancée permettant une modélisation 3D intuitive effectivement proposée au grand public apparaît dans un autre jeu vidéo, *Magic Pengel*, ainsi que dans sa suite, *Graffiti Kingdom*. Ceux-ci mettent en œuvre les idées présentées dans le système TEDDY [IMT99]. Le joueur peut ainsi, en quelques coups de crayon, créer des personnages en 3D auquel le jeu donnera vie. Mais la plupart du temps, les logiciels de modélisation demandent une certaine compétence. Si rien n'empêche aujourd'hui de créer des objets très complexes, rendre cette création simple et intuitive reste un problème délicat, auquel il n'a pas été apporté de solution satisfaisante.

Chapitre 2

Des modèles de matériau pour la sculpture virtuelle

2.1 Introduction

2.1.1 Vers un modèle physique d'argile

Limitations des approches précédentes

La grande variété des outils permettant de créer des formes tridimensionnelles souligne à quel point il est difficile de fournir un outil de modélisation 3D qui réponde à tous les besoins. Parmi ces derniers, on citera par exemple la facilité avec laquelle on peut obtenir la forme recherchée, et la variété des formes pouvant être reproduites.

La plupart des logiciels de modélisation présentés jusqu'ici requièrent une certaine compétence, et un temps plus ou moins long d'apprentissage. Il est fréquent que l'utilisateur agisse indirectement sur la surface, via par exemple des points de contrôle, ou des dispositifs de déformation de la surface. Il n'est pas non plus toujours possible d'oublier la structure sous-jacente décrivant la surface, même si de nombreuses approches, souhaitant tendre vers le concept de sculpture virtuelle, s'efforcent de masquer autant que possible la représentation choisie.

Les approches les plus intuitives introduisent le concept d'"outil" pour interagir avec l'objet. Malheureusement, les effets induits par ces outils ne sont pas toujours intuitifs, et doivent parfois être modulés par divers paramètres. Dans un but de simplicité, il est parfois choisi de s'inspirer du comportement d'un matériau réel, que l'utilisateur a déjà eu l'occasion de manipuler. Ces approches sont généralement plus naturelles que celles purement géométriques.

L'autre difficulté concerne la variété des volumes et surfaces qu'il est possible d'obtenir. La représentation choisie impose parfois quelques contraintes à l'utilisateur. Les modèles surfaciques, par exemple, se prêtent généralement mal à la gestion des collisions, et *a fortiori* des problèmes de changement de topologie, qu'il faut résoudre au cas par cas. Alors même qu'il

est aisé de creuser un trou dans un matériau réel, ou de coller deux pièces, cela peut être très délicat dans le monde virtuel. Ces opérations requièrent souvent que l'utilisateur corrige de lui-même le maillage ou la représentation sous-jacente.

Une argile virtuelle

Dans bien des situations, les artistes préfèrent encore travailler en sculptant un matériau réel, généralement de la glaise ou de l'argile, et digitalisent le résultat au moyen d'un scanner 3D. C'est par exemple le cas dans le domaine de l'automobile où l'aérodynamique des véhicules est d'abord travaillée avec de l'argile, quand bien même les simulations, puis les machines, nécessiteront ultérieurement une version digitale de cette même forme.



FIG. 2.1 – Beaucoup de formes tridimensionnelles sont encore réalisées en deux temps, par digitalisation d'une sculpture réalisée en argile ou avec d'autres matériaux.

Il y a probablement plusieurs raisons à cette préférence. La première est sans doute une meilleure appréciation de la forme, malgré le développement actuel des moyens d'affichages et du retour haptique. Une seconde raison tient à la simplicité et la richesse d'interaction que procure l'utilisation d'une argile. Dans cette étude, nous nous pencherons sur cette seconde raison. L'argile peut être aisément manipulée par n'importe qui avec quelque succès, et permet cependant d'obtenir une grande variété de formes.

C'est la raison pour laquelle nous aimerions pouvoir proposer une contrepartie virtuelle à la sculpture sur argile. Nous allons nous attacher à essayer de définir un modèle de matériau, avec lequel nous pourrions interagir, aussi proche que possible du matériau réel pour que tout un chacun puisse s'en servir sans avoir à en passer par un long et fastidieux apprentissage. Les sculpteurs pourraient, quant à eux, trouver dans l'utilisation d'une argile virtuelle un certain nombre d'avantages, physiques tant que pratiques, par rapport à l'utilisation d'un matériau réel.

2.1.2 Avantages d'une argile virtuelle

Un matériau numérique

Une grande part des avantages procurés par une argile virtuelle, par rapport à sa contrepartie réelle, trouve son origine dans le caractère numérique de celle-ci. On retrouve ainsi

naturellement tous ceux que procure généralement le travail avec un objet virtuel. Parmi ceux-ci, les retours en arrière : si une modification ne répond pas à ses attentes, l'utilisateur peut simplement l'annuler, ce qui n'est pas toujours possible avec un véritable matériau. L'ensemble du processus de modélisation peut être enregistré, au besoin, afin de pouvoir reprendre là où le besoin se fait sentir. C'est par ailleurs une option potentiellement utile pour les personnes qui s'intéressent au processus de création lui-même.

De façon similaire, une même ébauche pourra servir à la réalisation de plusieurs objets, sans qu'il soit nécessaire de répéter les mêmes gestes. On retrouve également le principe du copier-coller : nul besoin de réaliser plusieurs fois le même objet, il suffit de le dupliquer selon ses besoins.

‘A tout moment, il sera possible de suspendre son travail pour le reprendre ultérieurement. Il est donc possible de modéliser un objet en plusieurs séances, sans craindre de voir sécher le matériau. Il est également plus facile de s'échanger des travaux (un simple fichier suffit), de travailler à plusieurs sur un même projet. Plutôt que de partir de rien, il est aisé de reprendre une forme existante (éventuellement extraite par un scanner 3D), afin de la modifier selon ses vœux.

S'affranchir des contraintes physiques

Il devrait être possible, avec une argile virtuelle, de réaliser des formes qui, pour diverses raisons, ne seraient pas envisageables dans le monde réel. Par exemple, la gravité impose des contraintes sur le modèle, notamment de rigidité, qui sont parfois résolues en ajoutant une armature à l'objet sculpté. Il semble très rare que la gravité soit réellement utile à l'artiste. Dans un monde virtuel, il sera éventuellement possible de prendre en compte cette gravité (et simuler les éventuels effondrements de l'objet), mais on peut tout aussi bien s'en affranchir complètement pour réaliser des formes plus aériennes.

On est également en mesure de résoudre bien des problèmes d'encombrement. Il peut être délicat, avec un objet réel, d'accéder à certains endroits. Par exemple, la partie inférieure de l'objet peut être difficile d'accès, ou bien un élément de la sculpture peut gêner l'accès à un autre. Dans un monde virtuel, ces difficultés ne sont plus : on peut retourner l'objet sans difficulté, ou bien sectionner pendant un temps une partie de la sculpture, afin de faciliter l'accès à un autre endroit.

Dans le cas d'une sculpture réelle, il se pose parfois également des problèmes de taille. Certains objets peuvent être trop grands pour être sculptés aisément, certains détails trop petits pour les outils. Dans un univers virtuel, il est possible de choisir librement la taille de ses outils, de même que l'échelle à laquelle on va travailler (du moins dans certaines limites, imposées par des questions de temps de calcul et de stockage des informations, partiellement résolus par des approches dites multi-résolution). Il sera donc aussi aisé de modifier la forme globale de la sculpture que de s'attarder sur de petits détails.

Un parfait contrôle des paramètres

Enfin, le comportement d'un matériau réel est influencé par un grand nombre de paramètres physiques, voire chimiques. Dans le cas de l'argile, on peut par exemple citer la teneur en eau, ou bien encore la température. Il est parfois difficile pour l'artiste de contrôler précisément ces différents paramètres. Certes, il est possible de sécher localement le matériau, ou bien d'ajouter un peu d'eau pour le rendre plus malléable. Ces opérations sont toutefois parfois délicates à réaliser, et ne peuvent être faites qu'au niveau de la surface du matériau. En outre, l'argile va naturellement sécher, voire se craqueler, sans intervention de la part de l'utilisateur.

Dans le cas de la pâte à modeler, la température influe notablement sur son comportement. Le matériau va naturellement chauffer lorsque l'on va le manipuler, et certaines contraintes externes peuvent également jouer un rôle. Dans le cas de films d'animation où les personnages sont des sculptures en plasticine (citons par exemple la série *Wallace et Gromit*, ou encore le film *Chicken Run*, des mêmes studios), les éclairages nécessaires aux prises de vues posent parfois de réels problèmes. Avec un matériau virtuel, il est possible de contrôler aisément ces différents paramètres. Une argile virtuelle ne sèchera qu'à la demande de l'artiste, qui pourra également augmenter la teneur en eau là où il en a besoin. Le matériau virtuel se révèle donc plus souple à l'usage pour l'artiste.

2.2 Comportement et caractéristiques d'une argile

2.2.1 À mi-chemin entre les solides et les liquides

Dans la suite, nous nous pencherons sur le cas de matériaux tels que l'argile, la glaise ou encore la plasticine. Leur comportement est complexe, et nous allons d'abord nous attacher à mettre en évidence les différents aspects de ce comportement qui nous intéressent, et que nous souhaitons reproduire virtuellement. Il est d'usage de distinguer trois grands types de comportements physiques : liquide, solide et gazeux. Il apparaît rapidement que les matériaux que nous étudierons n'entrent pas du tout dans l'une de ces catégories, mais se situent plutôt quelque part à mi-chemin entre les solides et les liquides.

Les "solides"¹ dont la glaise, l'argile et la plasticine sont les plus proches sont ceux présentant un comportement de type plastique. Lorsque l'on exerce une contrainte suffisante sur ces derniers, il est possible de les déformer de façon permanente. De la même façon, si l'on déforme un objet en argile, les modifications seront persistantes. On constate toutefois plusieurs différences notables. En particulier, il n'est pas possible de fusionner deux morceaux de matériau plastiques simplement en les pressant l'un contre l'autre. Il est parfois possible de le faire en chauffant les matériaux, qui, à chaud, prennent un comportement pâteux quelque peu similaire à celui des matériaux qui nous intéressent.

¹En théorie, la phase solide se caractérise par une forme propre, indéformable. Il n'existe pas *stricto sensu* de véritables solides dans la vie courante, tous se révélant au moins légèrement élastiques en pratique, et même un bloc d'acier s'écoulera... sur une échelle de plusieurs millions d'années, cependant. La frontière entre liquides et solides est largement arbitraire, et nous considérerons le terme solide dans son acception la plus générique.

Lorsque l'on presse de l'argile contre un objet solide (un moule par exemple), on constate qu'il se déforme de façon à occuper tout l'espace à sa disposition. S'il a à sa disposition un interstice, il s'écoulera à travers celui-ci. Lorsque l'on presse un échantillon de ces matériaux à l'intérieur du poing, on s'attend à ce qu'il s'échappe entre les doigts. Dans certaines conditions, on peut même constater un écoulement spontané de certains d'entre eux, sous l'effet de la gravité par exemple.

Ce sont des comportements que l'on associe davantage aux liquides. Toutefois, les contraintes à appliquer à ces matériaux afin de les voir s'écouler sont plus importantes que celles nécessaires à un liquide usuel, et les temps caractérisant les écoulements sont également notablement plus importants. Le comportement de l'argile ou de la glaise se rapproche du comportement que présenterait un liquide pourvu d'une très grande viscosité. Mais là encore, l'analogie n'est pas parfaite.

Argile, glaise et plasticine ont un comportement que l'on peut qualifier de "pâteux", situé quelque part entre les liquides visqueux et les solides plastiques. Ces matériaux sont généralement qualifiés de *viscoplastiques*. On peut les décrire comme des fluides très visqueux qui ne s'écoulent que lorsque s'exerce une contrainte. Dans la suite, nous utiliserons principalement le terme d'argile pour désigner les matériaux que nous étudions, mais ce terme recouvrira l'ensemble des matériaux précités.

2.2.2 Les caractéristiques recherchées

Il n'est probablement pas absolument nécessaire, dans le but de créer un outil de modelage virtuel, de décrire de façon très précise, d'un point de vue physique, ces matériaux viscoplastiques. Ce que nous recherchons, c'est un matériau virtuel qui présentera l'essentiel des caractéristiques que l'on attend d'une argile. Nous dresserons ici une liste des comportements que nous souhaitons reproduire, afin de mieux cerner le modèle qui conviendrait à nos besoins.

D'une part, nous avons dit qu'il était relativement aisé, avec les matériaux qui nous intéressent, d'assembler deux objets. C'est une partie intéressante du modelage, puisque dans bien des cas il est pratique de créer séparément chacun des éléments qui composent l'objet que l'on réalise, puis de les assembler. Un sculpteur pourra, par exemple, lorsqu'il crée un vase, modeler séparément des anses, qu'il fixera par la suite sur l'objet.

De façon plus générale, il est possible et aisé, avec de l'argile, de retirer mais aussi d'ajouter de la matière à la surface d'un objet. La encore, c'est un point important du modelage avec de l'argile, puisque cela permet de rajouter de l'argile où il en manque, afin par exemple de créer un détail supplémentaire, contrairement au cas de la sculpture sur pierre ou sur bois, où il est uniquement possible de retirer de la matière.

Nous avons vu de nombreux outils de modélisation permettant d'assembler des objets, et de les sculpter en ajoutant ou retirant de la matière [FCG01]. Il leur manque toutefois un pan tout entier des caractéristiques de l'argile pour pouvoir être considérés comme tels : les déformations. Sous l'effet de contraintes, il est en effet possible de produire un vaste ensemble de déformations d'un objet en argile, qui elles aussi sont importantes dans le cas du modelage

d'une argile. On distingue de nombreux types de déformations.

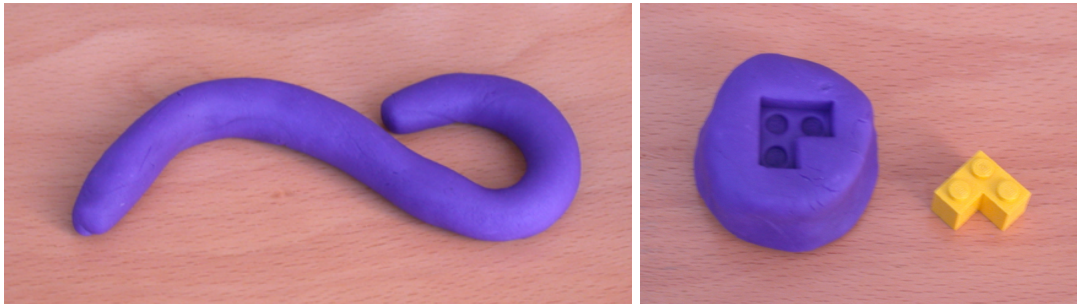


FIG. 2.2 – L'argile, ou ici la plasticine, permettent des déformations, ou la création d'empreintes dans l'objet modelé.

- D'abord, les nombreux effets d'empreinte. Lorsque l'on presse un objet rigide sur de l'argile, il va y laisser sa forme. L'argile présente un creux d'une forme correspondant à l'objet, et généralement, la matière déplacée par l'objet vient former des bourrelets sur son pourtour. A l'inverse, il est possible de presser de l'argile dans un moule, afin de donner à celle-ci la forme dudit moule. La pâte à modeler, tout particulièrement, exploite beaucoup ce principe de moulage.
- Ensuite, il est possible de déformer un objet en argile à une plus grande échelle. Par exemple, un artiste voudra créer un personnage bras et jambes écartés, et lui faire prendre une pose définitive une fois celui-ci terminé, en pliant, déformant et orientant ses membres.
- Avec de l'argile, il est courant de changer la topologie de l'objet au cours de son modelage. En effet, il est fréquent de percer un trou dans l'objet, d'assembler des éléments en les fusionnant, etc. Certaines parties du matériau pourront également entrer en collision.
- L'argile, enfin, est un matériau incompressible. Son volume, lorsque le sculpteur travaille, ne change pas (en dehors, bien évidemment, des ajouts et des retraits de matière directement consentis par l'utilisateur). Cette conservation du volume est une caractéristique importante si l'on souhaite pouvoir se faire une idée intuitive des déformations qui vont être occasionnées lorsqu'une partie est tirée ou poussée.

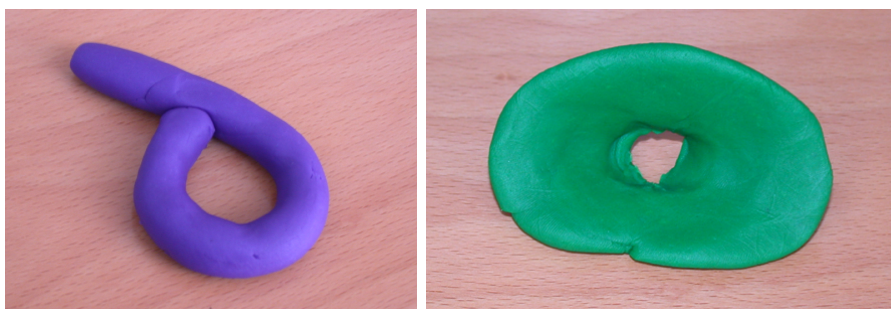


FIG. 2.3 – Les changements de topologie (ici un recollement et un trou) font partie des caractéristiques importantes du comportement des matériaux étudiés.

Toutes ces caractéristiques, nous aimerions les retrouver dans le matériau virtuel que nous allons développer. Ainsi que nous l'avons dit, nombre de modèles proposés précédemment

présentaient certains des comportements présentés ci-dessus, mais pas l'ensemble de ceux-ci. La plupart des modèles à base de déformations de surfaces polygones ou splines, par exemple, faisaient montre de difficultés avec la gestion des collisions, et les changements de topologie devaient généralement être gérés par l'utilisateur lui-même. 'A l'inverse, les modèles utilisant des volumes implicites, très à leur aise sur les changements de topologie, ne proposent pas de véritables outils de déformation.

2.3 État de l'art sur les modèles physiques de matériau pour l'animation

Nous avons vu qu'il n'existait pas de modèle géométrique adéquat imitant suffisamment bien une argile pour l'utilisation que nous envisageons. Pour cette raison, nous nous sommes tournés vers des modèles physiques. Il n'existe pas, à l'heure actuelle et à notre connaissance, de modèle simulant un matériau viscoplastique tel que l'argile qui réponde entièrement à nos besoins, et présente l'ensemble des comportements précédemment identifiés. De nombreux modèles permettent cependant de simuler des corps élastiques ou plastiques, des fluides visqueux, voire pâteux. Ces modèles pourraient nous servir de base. Nous allons examiner quelques-uns de ces modèles afin d'en cerner les forces et les limitations.

2.3.1 Les matériaux élastiques et plastiques

Les réseaux masses-ressorts

Les premières tentatives de simulation de solides déformables, présentant généralement un comportement élastique, ont été basées sur des modèles dits *masses-ressorts*. Le matériau y est modélisé par un ensemble de masses ponctuelles, reliées entre elles par des ressorts. Ce genre de description nodale est courante dans le domaine de la simulation physique de matériaux déformables. Le mouvement de quelques points particuliers est calculé, le mouvement du reste du matériau étant déduit, généralement par interpolation, du mouvement de ces points. Les premiers travaux sur ce genre de système sont à mettre au crédit de S. Platt et N. Badler, qui ont utilisé ce genre de réseau pour animer des visages [PB81].

Ce modèle a par la suite été extensivement utilisé, avec diverses variations. Des termes d'amortissement ont été souvent ajoutés au modèle, afin d'éviter l'apparition d'éventuelles oscillations non souhaitées. Chadwick a utilisé un réseau masse-ressorts contrôlant une grille FFD pour modéliser les déformations d'un matériau, afin d'animer la chair de personnages [CHP89]. Miller a quant à lui proposé d'ajouter des contraintes de conservation du volume afin de modéliser des matériaux incompressibles [Mil88].

Parmi les difficultés rencontrées dans l'utilisation de ces modèles, on trouve quelques problèmes liés aux questions d'intégration numérique. Il est parfois difficile d'assurer la stabilité de ces modèles. Pour ce faire, il a été suggéré d'adapter le pas d'intégration, en se basant sur des critères énergétiques pour détecter les éventuels problèmes [Jou96], ou bien encore d'utiliser des méthodes d'intégration implicite, comme le proposent Baraff et Witkin dans

[BW98] pour permettre la simulation de ressorts de grande raideur.

En jouant sur la topologie du réseau et la raideur des différents ressorts, il est possible de simuler un grand nombre de comportements élastiques, en fonction des besoins, même si l'identification des paramètres est un problème délicat, généralement résolu par l'expérience. Les matériaux ainsi modélisés sont élastiques, ce qui ne convient pas exactement à nos propres besoins. On peut cependant envisager d'adapter ce genre de simulation à des solides plastiques, simplement en changeant la longueur à vide des ressorts, après déformation. Cela permet de figer le matériau dans sa nouvelle position. Cependant, les déformations s'accumulant, le réseau sera de plus en plus déformé, et la simulation sera alors de moins en moins précise. En outre, ces modèles ne tiennent pas compte d'éventuelles autocollisions, et *a fortiori* de possibles changements de topologie.

Matériaux continus

Une description plus précise et correcte des matériaux, utilisant principalement des termes énergétiques, a été proposée dans les travaux de D. Terzopoulos, qui a proposé de nombreux modèles physiques de matériaux déformables. Ces travaux débutent en 1987 avec la proposition d'un modèle permettant de simuler le comportement de matériaux parfaitement élastiques [TPBF87]. Le matériau simulé (qui peut se présenter sous la forme d'une courbe, d'une surface ou bien encore d'un volume) répond à travers une énergie de déformation aux différentes contraintes qui lui sont appliquées. Ces contraintes peuvent être de différents types. Des interactions de contact, par exemple lorsqu'un objet entre en collision avec le matériau comme sur la figure 2.4, ou lorsqu'un fluide visqueux est au contact de l'objet. Le modèle proposé gère également les déformations du matériau sous l'effet de son propre poids.

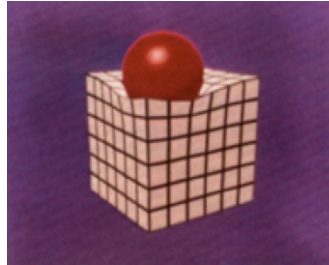


FIG. 2.4 – Déformation d'un cube élastique sous le poids d'une sphère, d'après le modèle de [TPBF87].

Ce modèle est étendu dans [TF88] aux comportements viscoélastiques et plastiques, plus proches de ce dont nous aurions besoin pour simuler le comportement d'une argile, qui se comporte comme un corps viscoplastique. D. Terzopoulos identifie quelques comportements élémentaires :

- parfaitement élastique, la déformation e est proportionnelle à la force appliquée F , et disparaît lorsque cesse la contrainte,

$$e = kF$$

- parfaitement visqueux, la déformation augmente de façon constante et permanente,

d'autant plus vite que les forces appliquées sont importantes,

$$e = \int_0^t \frac{1}{\eta} F dt$$

- parfaitement plastique, où le matériau s'oppose à toute déformation en-dessous d'une certaine contrainte.

Ces différents comportements élémentaires sont combinés pour simuler des comportements plus complexes, principalement viscoélastiques ou élastoplastiques. Dans ce modèle, la position d'un point du matériau est décrite comme la somme d'une position de référence \mathbf{r} et d'un déplacement supplémentaire dû aux déformations du matériau \mathbf{e} . Cette formulation modélise donc ces déformations par rapport à une forme de référence, et permet de les calculer en utilisant les divers comportements précités. Une partie des déformations \mathbf{e} sont répercutées, au travers d'une loi d'évolution simple, sur la forme de référence, permettant ainsi une meilleure modélisation des effets plastiques du matériau.

Ces simulations requièrent cependant beaucoup de calculs, rendant leur utilisation dans un environnement temps-réel impossible. En outre, s'il est possible de modéliser des fractures, en déchirant le matériau là où les déformations, et par conséquent les contraintes, sont les plus fortes, la possibilité de recollement n'est pas envisagée. Le solide conserve un souvenir de sa forme originale. Les relations entre deux points voisins du solide peuvent être détruites, mais il ne peut être créé de nouvelles relations.

Imposition de contraintes aux matériaux

Une grande part du comportement des solides déformables peut être explicitée sous la forme de contraintes. Le contact avec un objet impose des contraintes de mouvement à la surface du matériau, le caractère incompressible de ce dernier implique quant à lui une invariance du volume. Le respect de ces contraintes est une étape clé dans la simulation du comportement de solides déformables.

J. Platt et A. Barr proposent une série de méthodes pour résoudre ces contraintes, en fonction de leur type. La première méthode, qualifiée de *Reaction Constraints*, la plus immédiate, consiste à introduire des forces supplémentaires qui viendront assurer que la contrainte est bien vérifiée [PB88]. Elle est toutefois, comme le signalent les auteurs, relativement limitée dans son usage, essentiellement au cas de particules ponctuelles, et ne s'applique pas au cas d'objets continus. La seconde démarche au contraire s'applique bien au cas des matériaux dont le comportement est simulé par l'optimisation d'une énergie.

Deux approches sont usuellement envisagées dans cette situation. Une méthode par pénalité, qui ajoute un terme à l'énergie lorsque la contrainte n'est pas vérifiée. Les contraintes ainsi traitées ne sont pas précisément respectées, et le poids du terme favorisant le respect de la contrainte dans l'expression de l'énergie est difficile à choisir. Une seconde méthode, en utilisant les multiplicateurs de Lagrange, permet un respect exact de la contrainte, mais pose quelques problèmes de résolution. La méthode proposée par J. Platt et A. Barr, baptisée *Augmented Lagrangian Constraints* mélange ces deux approches pour permettre un respect de la contrainte tout en simplifiant la résolution. Les auteurs utilisent cette formulation pour

simuler le comportement de différents solides élastiques et plastiques en assurant une parfaite conservation de leur volume.

Vers une intégration temps-réel

Bien évidemment, ces différentes approches sont très coûteuses en temps de calcul, et ne sont pas, pour la plupart, utilisables dans une application temps-réel. Diverses solutions ont été suggérées pour résoudre ce problème. Pentland et Williams suggèrent dans [PW89], comme c'est souvent le cas en physique, de rechercher les modes propres de vibration du matériau. De ces modes, ils ne conservent que ceux de plus basse fréquence. Cette décomposition en modes propres permet, grâce au précalcul, d'obtenir des déformations parfaitement interactives. L'élimination des modes propres de fréquence élevée garantit par ailleurs une meilleure stabilité lors de l'intégration. Pentland et Williams supposent en outre que la forme de l'objet n'intervient que peu dans ses modes propres, et l'assimilent à un simple parallélépipède de mêmes dimensions lors du calcul de ses modes d'oscillations. Les résultats ainsi obtenus sont corrects, mais relativement grossiers.

G. Debonne utilise quant à lui une description hiérarchique de l'objet par des éléments finis, afin de permettre la simulation de déformations élastiques en temps réel [Deb00, DDCB01]. Si l'interaction est cette fois-ci de bien meilleure qualité, elle repose toujours sur une grande quantité de précalculs qui autorisent la simulation des déformations en temps réel. Le matériau a donc une structure qu'il conserve au cours du temps. En particulier, sa topologie ne peut évoluer, ce qui ne convient pas pour simuler des matériaux comme de l'argile dont la forme et la topologie évoluent au cours du temps.

2.3.2 Les modèles de fluides

Modèle physique de fluides : l'équation de Stokes

Outre les solides plastiques, l'argile peut être comparé à un liquide dont la viscosité serait élevé. Nous nous sommes donc également penchés sur les travaux qui ont été menés dans le cadre de la simulation de fluides. Le comportement de la plupart des fluides incompressibles est remarquablement bien décrit par le système d'équations dit de *Navier-Stokes*. Il découle directement de l'application du principe fondamental de la dynamique à des éléments infinitésimaux de fluide, et relie l'évolution temporelle des vitesses en tout point du fluide \mathbf{u} (description eulérienne) au terme de pression p et aux forces extérieures exercées sur le fluide \mathbf{F} . ρ désigne la masse volumique du fluide considéré, et ν sa viscosité cinématique η/ρ . Le dernier terme de l'expression, $(\mathbf{u}.\mathbf{grad})\mathbf{u}$, provient du choix d'une description Lagrangienne du problème, et exprime le gradient des vitesses dans le fluide le long d'une ligne de champ. Nous ne reviendrons pas ici sur la justification de ces équations, qui pourra être trouvé dans l'ensemble des ouvrages traitant de mécanique des fluides (voir par exemple [LL86]).

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho} \mathbf{grad} p + \nu \Delta \mathbf{u} - (\mathbf{u}.\mathbf{grad})\mathbf{u} + \mathbf{F} \\ \mathit{div}(\mathbf{u}) = 0 \end{cases} \quad (2.1)$$

Le caractère incompressible du fluide, matérialisé ici par la condition de non-divergence des

vitesse $\operatorname{div}(\mathbf{u}) = 0$, est assez bien vérifié dans le cas de la plupart des liquides. Les matériaux qui nous intéressent (argile, glaise, plasticine) entrent également sans difficulté dans le cadre de cette approximation. On trouve bien évidemment d'autres modèles, par exemple pour décrire le comportement des gaz, sur lesquels nous ne nous attarderons pas².

Nécessité d'une résolution numérique

Ce type d'équation peut être résolue dans une foule de cas particuliers, mais le cas général est inextricable, et il n'existe pas de solution analytique qui conviendrait à n'importe quelle situation. De sorte que l'équation de Stokes est la plupart du temps, dans le domaine de la mécanique des fluides, résolue de façon numérique. On sectionne usuellement l'espace selon une grille la plus fine possible, et l'on calcule à chaque pas de temps l'évolution de la vitesse et de la pression (éventuellement de la température) du fluide dans chacune des cellules de la grille.

Les modèles utilisés en mécanique des fluides font montre d'une précision et d'une stabilité souvent remarquables, mais ne nous sont guère d'un grand secours, dans la mesure où ils requièrent énormément de ressources et de puissance de calcul, et ne permettent pas le temps réel qui nous est nécessaire pour une manipulation interactive. Les plus puissants calculateurs dont on dispose actuellement³ sont souvent dédiés à ces tâches, et montrent pourtant leurs limites. Il n'est donc pas possible de les utiliser directement, et il faudra admettre certaines simplifications des modèles.

Simplification des équations

Plusieurs personnes ont travaillé sur la simplification des équations de mécanique des fluides, en particulier des équations de Stokes, afin de permettre leur usage en infographie. Il n'est en effet pas utile, dans ce domaine, d'obtenir des résultats rigoureusement exacts d'un point de vue numérique, tant que visuellement ils demeurent convaincants. L'important étant d'être en mesure d'extraire l'essence des mouvements du fluide (en gardant à l'esprit, cependant, qu'en général ce sont les plus infimes détails, les éclaboussures ou les gouttes projetées, qui font la différence entre une simulation convaincante et une qui l'est moins).

En 1996, N. Foster et D. Metaxas ont utilisé un schéma numérique d'intégration des équations de Navier-Stokes dans le cas d'un fluide newtonien incompressible [FM96], proposé en 1965 par F. Harlow et J. Welch [HW65]. Le problème est discrétisé dans une grille régulière, comme c'est l'usage, et résolu de façon itérative pour de petits pas de temps, les vitesses à l'instant $t + 1$ étant calculées à partir des contraintes extérieures et des vitesses à l'instant t .

La principale difficulté dans la résolution des équations de Navier-Stokes provient du fait qu'il y a un couplage fort entre les termes de vitesse et celui de pression. En outre, le caractère

²Dans le cas de matériaux compressibles, on assure simplement la conservation de la quantité de matière à travers la relation $\operatorname{div}(\rho\mathbf{u}) = 0$. Dans un matériau incompressible, ρ est constant, ce qui permet de retrouver l'expression précédente.

³La grappe d'ordinateurs de Météo France se trouve, par exemple, être actuellement le plus puissant outil de calcul de l'Hexagone.

incompressible du fluide impose, à travers la seconde équation $\text{div}(\mathbf{u}) = \mathbf{0}$, des conditions à la forme du champ de vitesse.

L'idée suggérée consiste en fait à déterminer un champ de pression p qui, lorsqu'on l'injecte dans le système d'équation, permet d'obtenir un champ de vitesse compatible avec ces conditions. C'est une idée fréquemment utilisée dans le domaine de la mécanique des fluides, où l'on utilise usuellement la résolution d'une équation de Poisson pour calculer le champ de pression correspondant. N. Foster propose plutôt d'employer directement la divergence du champ de vitesse pour en déduire un champ de pression.

La résolution est conduite en alternant deux étapes : un calcul du champ de vitesse à l'instant $t + 1$, qui ne respecte pas nécessairement la condition de non-divergence, puis une évaluation d'un champ de pression p qui permettra de s'opposer à ces problèmes (la pression augmente dans les cases où la divergence du champ de vitesse calculé est négative, traduisant une accumulation du fluide dans la cellule, et diminue lorsque au contraire la divergence est positive). Ce champ de pression permettra d'obtenir une nouvelle estimation du champ de vitesse à l'instant $t + 1$.

Cette méthode donne des résultats très convaincants, mais souffre généralement d'instabilité lorsque les vitesses sont élevées (ce qui est probablement une conséquence de l'usage de différences finies au niveau des cellules de la grille), ou, plus gênant dans le cas qui nous intéresse, lorsque la viscosité est importante. En outre, la simulation reste gourmande en terme de temps de calcul, rendant son utilisation dans une application interactive malaisée. L'article se penche également sur divers problèmes liés à l'affichage : utilisation de marqueurs pour déterminer précisément la position de la surface du fluide à tout instant, considérations sur le rendu proprement dit.

Améliorations des modèles

Pour contourner les problèmes d'intégration, d'autres modèles plus poussés ont été développés, utilisant par exemple des schémas d'intégration plus fins (avec des pas de temps de taille variable), ou bien encore d'autres méthodes pour calculer les dérivées (Méthode de *Crank-Nicholson* par exemple). La plupart s'efforcent de produire le meilleur rendu visuel possible pour le fluide, et s'efforcent en particulier de prédire le comportement le plus réaliste possible de sa surface, la partie du fluide qui apparaît immédiatement à l'œil.

Ainsi S. Chen a montré qu'on pouvait simuler avec davantage de finesse le comportement d'un fluide en s'intéressant au mouvement de particules situées au niveau de l'interface entre le fluide et l'air [CJRF97]. Des modèles hybrides combinant particules et simulation volumique ont rapidement fait l'objet de l'essentiel de l'attention. N. Foster introduit en 2001 de nombreuses notions permettant d'améliorer le comportement du fluide [FF01]. Entre autres, ils combinent une surface implicite et des marqueurs pour un meilleur comportement de la surface du fluide, et proposent une nouvelle approche nettement plus efficace pour le calcul du champ de pression. Un schéma d'intégration temporel plus complet est également proposé pour réduire les erreurs éventuelles. Plus important dans le cas qui nous intéresse, l'interaction du fluide avec des objets en déplacement a été étudiée et simulée.

La plupart des outils pour interagir avec un fluide sont alors disponibles. Un modèle

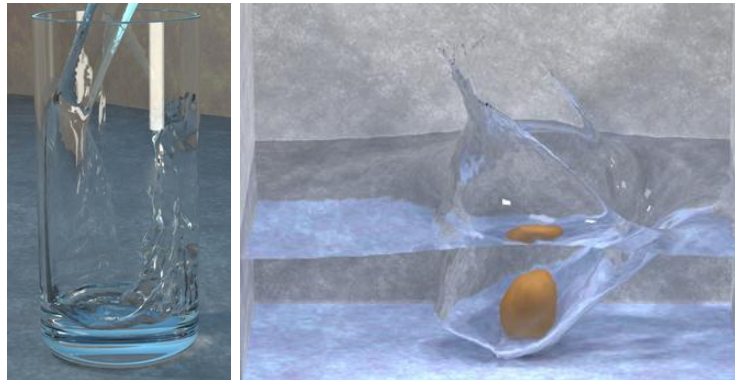


FIG. 2.5 – Des exemples de simulations de fluides présentés dans [EMF02].

donnant des résultats substantiellement meilleurs est proposé par Douglas Enright l'année suivante [EMF02], soulignant l'intérêt pour le domaine, et les rapides progrès qui sont effectués. Là encore, l'essentiel du travail a porté sur la simulation du mouvement de la surface. Un système de particules permet d'en modéliser finement le mouvement, la surface étant ensuite reconstruite à partir de la position des particules sur le principe des surfaces implicites. Les effets visibles à la surface (action du vent sur un plan d'eau, stabilisation de la surface, etc.) peuvent par ailleurs être plus facilement contrôlés.

Une meilleure interaction entre fluides et solides est proposée ensuite par Mark Carlson [CMT04]. L'idée clé de leur méthode consiste à simuler les solides comme s'ils étaient des fluides, en oubliant le fait qu'ils sont indéformables. Le champ de vitesse est ensuite corrigé, de façon à ce qu'il corresponde effectivement à un mouvement rigide aux endroits où se trouvent les solides (d'où le nom de "*rigid fluids*" porté par la méthode). Le mouvement des solides dans ce modèle n'est cependant pas imposé, il résulte simplement de la réaction du fluide, les résultats étant particulièrement convaincants d'un point de vue visuel.

Ces nombreux modèles présentent toutefois deux inconvénients majeurs pour l'usage que nous souhaiterions en faire. Conçus pour donner le meilleur résultat possible d'un point de vue graphique, le temps de calcul nécessaire n'a pas été pris en compte, et ces modèles, en particulier les derniers, ne sauraient être utilisés dans le cadre d'une simulation en temps réel. En outre, la stabilité de ces modèles est hasardeuse lorsque la viscosité devient élevée (comme c'est le cas avec des matériaux de type argile), excepté si l'on réduit drastiquement les pas d'intégration, ce qui se traduit par un accroissement important du temps de calcul.

Stable fluids

Quelques modèles ont cependant privilégié la vitesse de calcul, au prix de quelques sacrifices sur la qualité graphique et la précision du résultat, et peuvent simuler le comportement d'un fluide en temps réel. Le principal obstacle se situe au niveau de la stabilité des algorithmes d'intégration. Des méthodes différentes évitant ces problèmes ont pu être développées dans des cas précis. Les *stable fluids* de Jos Stam [Sta99], par exemple, permettent la simulation d'un fluide eulerien en temps réel, avec peu de ressources de calcul. Malheureusement, pour ce faire, d'importantes hypothèses sur l'espace de travail doivent être faites. Le fluide



FIG. 2.6 – interaction entre fluide et solides d’après le modèle de [CMT04].

doit en effet occuper l’ensemble d’un espace de travail de forme cubique, alors que dans le cas de l’argile nous travaillons avec une surface quelconque dépendant du temps.

Il est certes proposé, dans l’article en question, de déformer un volume de fluide de forme quelconque. En fait, dans ce cas, le voisinage de ce volume est également constitué de fluide, avec les mêmes propriétés de densité et de viscosité. Ce genre de simulation correspond au cas de fumées (un gaz plongé dans un autre gaz de densité similaire), ou bien encore dans le cas de colorants dans un fluide (là encore, deux “zones” sont visibles, mais les propriétés physiques des deux zones sont similaires). Cela ne correspond pas au cas qui nous intéresse d’un objet fluide qui serait entouré d’air. On s’exposerait donc, en reprenant ce modèle, à des comportements surprenants et peu naturels, dans la mesure où ce n’est pas de l’air ou du vide qui entoure l’objet : pour fusionner deux volumes, par exemple, il faudra chasser le fluide qui les sépare, fluide dont le comportement diffère de façon importante de l’air. Avant même d’entrer en contact, les deux volumes risquent notamment de s’aplatir.

Animation de fluides visqueux

Le schéma d’intégration d’Euler entre pour une bonne part dans l’instabilité des modèles précédents lors de la simulation de fluides visqueux. En particulier, plus le fluide est visqueux, plus les pas de simulation doivent être brefs pour que le résultat soit correct. Or, dans notre cas, les matériaux étudiés sont particulièrement visqueux. M. Carlson a récemment proposé d’utiliser le schéma d’intégration dit d’*Euler implicite* afin de contourner ces problèmes de stabilité, et ainsi simuler le comportement de fluides visqueux sans un surcoût déraisonnable en termes de calculs [CMBT02]. Ils proposent dans le même article une correction pour corriger le comportement de certaines particules fluides, notablement freinées par le schéma choisi.

Cette méthode permet la simulation de matériaux très visqueux, avec un seul pas d’intégration pour des temps allant jusqu’au dixième de seconde, en des temps presque interactifs (sur un ordinateur moderne, ces temps sont inférieurs à la seconde). Bien qu’étant une solution possible dans un avenir proche, le modèle n’est pas suffisamment rapide pour correspondre à nos besoins de simulation interactive.

La stabilité de telles simulations de fluides visqueux n’est cependant assurée sous réserve



FIG. 2.7 – Fonte d'un bunny obtenue par simulation d'un fluide visqueux d'après [CMBT02].

qu'une condition nommée *Courant-Friedrichs-Lewy* soit vérifiée [PFTV93]. Cette condition impose des restrictions sur la vitesse à laquelle l'information se propage d'une cellule à l'autre. Lorsque cette condition est vérifiée, il est cependant impossible d'obtenir des effets de déformation à grande échelle comme nous le souhaiterions. Il est par conséquent impossible pour ces différents modèles de simulation de fluides d'obtenir l'ensemble des effets que nous recherchons. Au mieux, seuls les effets locaux pourront être rendus par ces modèles.

Des fluides viscoplastiques

Une simulation de fluides viscoplastiques a très récemment été proposée par T. Goktekin *et al* [GBO04]. Ces fluides présentent des différences notables avec les fluides représentés par les équations de Navier-Stokes. A l'instar des matériaux plastiques, par exemple, il leur faut vaincre une résistance à l'écoulement. Pour tenir compte de ces différences, T. Goktekin a introduit dans les équations de Navier-Stokes un terme supplémentaire, représentant les effets élastiques, qui fait intervenir les déformations subies par le matériau.

Les résultats obtenus permettent de simuler le comportement de très nombreux types de matériaux visqueux et viscoplastiques. Certains semblent relativement proches de matériaux tels que l'argile. La méthode est pour l'instant encore coûteuse en terme de temps de calcul (une demi-heure par seconde d'animation sur une machine récente d'après les auteurs), et ne peut donc servir, pour l'instant, de matériau virtuel pour une utilisation interactive. Par ailleurs, l'argile, lorsqu'elle est relativement sèche, a un comportement quasi-rigide qui est généralement mal reproduit par les modèles de fluides, celui-ci n'y faisant vraisemblablement pas exceptions. Cela reste toutefois une solution très prometteuse pour la simulation d'un matériau tel que l'argile.

Des réseaux aux systèmes de particules

Pour représenter des matériaux davantage visqueux et pâteux, et dont la structure et la topologie évoluent au cours du temps, d'autres approches ont été proposées. D. Terzopoulos étend dans [TPF91] le principe de rupture développé pour les matériaux plastiques. Le matériau est initialement décrit sous la forme d'un réseau masse-ressorts, mais ces derniers peuvent être détruits lorsque la température dépasse une certaine température. Les masses deviennent alors des particules indépendantes, qui réagissent grâce à divers potentiels de type

Lennard-Jones, couramment rencontrés dans le domaine de la physique, qui combinent une répulsion à courte distance et une attraction à longue distance.

La répulsion à courte distance permet d'assurer une certaine conservation du volume du matériau, tandis que l'attraction va permettre aux particules d'interagir entre elles, modélisant ainsi des effets de viscosité. Les particules peuvent en outre réagir avec leur environnement, par exemple par l'intermédiaire des *Reaction Constraints* de J. Platt et A. Barr [PB88]. Les modèles à base de particules n'imposent aucune contrainte topologique, et ont été largement utilisés pour décrire le comportement de fluides visqueux.

Interaction entre particules

Divers modèles d'interaction entre les particules ont été proposés [LC86, Ton98]. Ils peuvent être arbitrairement complexes, et il est possible d'obtenir une large gamme de comportements de fluides, pouvant faire preuve d'une grande viscosité, ou bien montrer quelques effets plus exotiques. Les interactions induisent cependant essentiellement des effets locaux, il est nettement plus difficile d'obtenir des effets cohérents à grande échelle. La simulation du matériau consiste simplement dans le calcul du mouvement de chacune des particules, ce qui se révèle généralement assez coûteux, le nombre de particules requis pour la simulation de la plupart des matériaux étant rapidement élevé. Quelques résultats temps-réel ont pu être obtenus pour les cas les plus simples [Gui02], mais généralement, ces modèles ne sont pas adaptés à une manipulation interactive.

Il se pose en outre divers problèmes de visualisation de la surface. Généralement, elle est obtenue par la construction d'une isosurface basée sur la position des différentes particules, ce qui pose occasionnellement des problèmes d'imprécision lors des contacts avec le monde extérieur. Ces problèmes sont naturellement exacerbés par la contrainte d'un affichage en temps réel.

M. Desbrun propose d'utiliser un modèle en couches pour obtenir un bon rendu de la surface, en particulier lors des collisions, et pour assurer la conservation du volume de l'objet [DC95]. Ce modèle, à l'instar de certains modèles de fluides présentés précédemment, combine des particules pour calculer les déplacements à grande échelle de la matière, et une surface implicite élastique capable de se déformer localement et de conserver son volume. Le résultat est un bon exemple de matériau visqueux avec lequel il est possible d'agir, cependant pas de façon interactive.

Les modèles SPH

Plutôt que de considérer l'interaction directe entre les particules deux à deux, les modèles SPH (*Smoothed Particles Hydrodynamics*, issus de la recherche de la dynamique stellaire, préfèrent une approche différente. Ils étudient le mouvement de particules (d'étoiles) dans un champ créé par l'ensemble des autres particules. Ces particules représentent un échantillonnage des caractéristiques de leur voisinage. Ce formalisme respecte plus finement la physique sous-jacente du problème, et de plus peut se prêter à une simulation multi-résolution [Des97].

Pour un temps réservés aux cas des gaz et des matériaux très peu rigides, pour des raisons de temps de calcul, [DC99], les modèles SPH ont ces derniers temps fait une entrée remarquée dans le domaine de la simulation des fluides. Ainsi, M. Müller et S. Premoze proposent tout deux des modèles de fluides basés sur des systèmes de particules, capables d'interagir avec des parois fixes [PTB⁺03, MCG03]. Pour des cas raisonnablement simples, utilisant quelques milliers de particules, le temps de calcul d'une image est de l'ordre de la seconde, ce qui permet d'espérer la simulation de fluides en temps réel dans un avenir relativement proche.

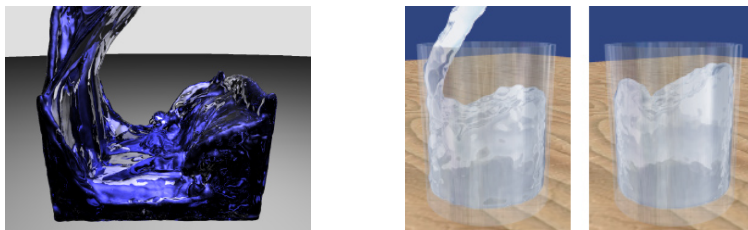


FIG. 2.8 – Exemples de simulations de fluides selon [PTB⁺03] et [MCG03].

Des modèles à base d'automates cellulaires.

Récemment, la suggestion a également été faite d'utiliser des automates cellulaires pour modéliser des matériaux déformables, dans le but de proposer des matériaux virtuels de type argile. Les premiers travaux de ce type sont à mettre au crédit de H. Arata *et al* dans [ATTY99]. Le matériau proposé permet une interaction avec des outils de forme quelconque qui vont "pousser" la matière. On peut ainsi obtenir des déformations locales sans difficulté, autorisant d'éventuels changements de topologie. La figure 2.9 illustre le type de résultats que l'on peut obtenir avec ce système.

Stricto sensu, le modèle précédent ne repose en fait pas sur des bases physiques. Plus récemment, S. Druon a proposé un modèle plus complet, dans lequel il justifie par ailleurs l'origine des actions effectuées par l'automate cellulaire grâce aux équations de la mécanique des fluides [Dru04]. Des résultats intéressants sont obtenus pour un fluide dénué de viscosité. On peut ainsi aisément effectuer des empreintes dans le matériau comme on le voit sur la figure 2.9.

Toutefois, en ne tenant pas compte des effets de viscosité, seules des déformations locales peuvent être obtenues, et le matériau est un peu trop fluide pour ressembler exactement à de l'argile. Pour palier cet inconvénient, S. Druon propose ensuite une résolution d'une équation de Poisson afin de tenir compte des effets de viscosité. Le cas bidimensionnel donne des résultats encourageants.

Nous ne détaillerons pas ici plus avant ces modèles cellulaires, qui sont relativement proches de ce que nous allons utiliser pour simuler le caractère fluide de notre argile virtuelle. Nous y reviendrons donc plus en détail dans le chapitre suivant.

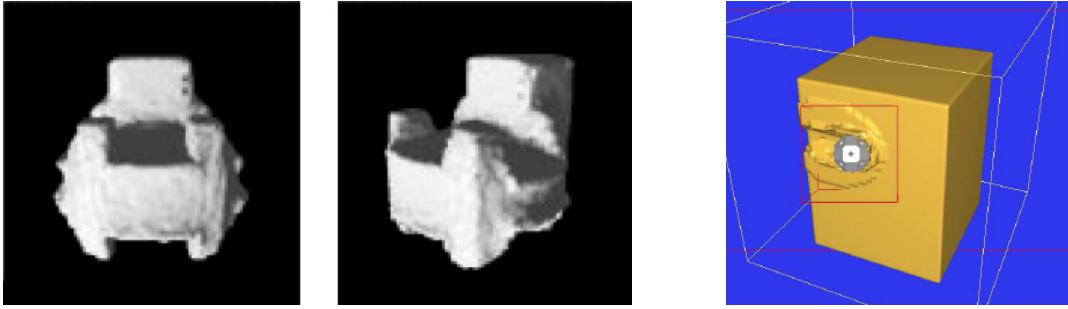


FIG. 2.9 – Déformations locales d’un matériau calculées grâce à un automate cellulaire. À gauche, un exemple de réalisation obtenue grâce aux travaux de H. Arata *et al* [ATTY99], à droite, un outil sphérique repousse une argile virtuelle d’après S. Druon [Dru04].

2.3.3 Bilan

Les différentes tentatives de simulation du comportement d’un fluide ont montré qu’il était difficile de simuler des fluides suffisamment visqueux pour l’usage que l’on souhaite en faire, même si de récents travaux ont montré que l’ajout d’un terme supplémentaire aux équations de Navier-Stokes permet de simuler des fluides dont le comportement est celui d’objets viscoplastiques. La condition *Courant-Friedrichs-Lewy* pose cependant généralement une limitation très forte sur le transfert des mouvements pour qu’une simple simulation de fluide permette d’obtenir un matériau visqueux en mesure de subir des déformations à grande échelle.

Les modèles à base de particules interagissantes permettent de simuler des matériaux proches de l’argile. Ils ne parviennent toutefois pas à imiter la totalité des effets recherchés. Les modèles SPH, qui à l’heure actuelle sont presque en mesure de simuler le comportement de fluides en temps réel, ouvrent des perspectives intéressantes. Ces modèles, cependant, tendent naturellement à s’écouler et ne maintiennent pas leur forme au cours du temps.

A l’inverse des modèles de fluides, les simulations de matériaux élastiques ou plastiques permettent très aisément d’obtenir des déformations à grande échelle. Ce sont plutôt les déformations locales, dues à la fluidité du matériau, occasionnant des effets comme la création d’empreintes, qui ne sont pas disponibles. Et surtout, pour assurer une réponse temps-réel de ces matériaux, la structure sous-jacente servant à la simulation est précalculée, éventuellement raffinée ou modifiée par la suite. L’une des conséquences néfastes de ce précalcul est qu’il est difficile, pour ne pas dire impossible, de gérer d’éventuels changements de topologie avec ces modèles.

Aucune approche ne peut donc nous donner pleinement satisfaction. Les plus proches de le faire sont sans doute les simulations de fluides visqueux (les systèmes de particules étant à proscrire de par leur coût en temps de calcul). La principale difficulté à lever va consister à trouver un moyen de contourner la condition *Courant-Friedrichs-Lewy* pour permettre la simulation d’effets quasi immédiats sur de grandes échelles. Les modèles basés sur des automates cellulaires fournissent de bons résultats dans le cadre de la simulation des déformations locales, et permettent divers effets d’empreintes. En revanche, ils ne permettent pas de procéder à des déformations à grande échelle du matériau.

Chapitre 3

Un modèle d'argile virtuelle

3.1 Un modèle multi-couches

3.1.1 Retour sur l'équation de Stokes

Effets des différents termes de l'équation de Stokes

Rappelons à toutes fins utiles les équations de Navier-Stokes avec lesquelles nous allons travailler :

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho} \mathbf{grad} p + \nu \Delta \mathbf{u} - (\mathbf{u} \cdot \mathbf{grad}) \mathbf{u} + \mathbf{F} \\ \operatorname{div}(\mathbf{u}) = \mathbf{0} \end{cases} \quad (3.1)$$

Attardons-nous un instant sur cette expression afin de voir quels effets trouvent leur origine dans chacun de ses termes. Ceci nous permettra d'avoir une idée des termes les plus importants pour nous permettre d'obtenir les différentes caractéristiques que l'on souhaite pour notre matériau. Oublions un temps l'action des forces extérieures, sur lesquelles nous reviendrons ultérieurement, pour nous pencher plus avant sur les trois autres termes.

Le premier terme est orienté depuis les zones de haute pression vers les zones de pression plus faible. Lorsque la vitesse du fluide est uniforme, en particulier lorsqu'elle est nulle partout, ce terme va faire apparaître des courants qui auront pour effet de transporter la matière vers ces zones de basse pression. De cette façon, la pression au sein du fluide aura tendance à s'uniformiser.

Ce comportement est à rapprocher des effets acoustiques : lorsque la membrane d'un haut-parleur s'avance, elle crée localement une surpression, l'air se situant juste devant ladite membrane étant brusquement comprimé. Cet excès de pression va disparaître tandis que l'air se déplace vers l'avant, créant l'onde de choc que constitue une onde sonore. On cherche un comportement similaire pour notre matériau : lorsque l'on presse l'outil sur ce dernier, on s'attend à ce que la matière soit poussée, jusqu'à atteindre des zones libres où elle pourra se déverser, généralement aux abords de l'outil, créant de ce fait les boursouffures et bourrelets

⁰Ce modèle d'argile peut être retrouvé dans [DC03] ou bien dans [DC04a].

de matières attendus.

Cet effet est essentiellement local : dès que l'on atteint une zone uniforme en termes de pression, le mouvement n'est plus transmis. En outre, la transmission ne sera assurée que dans la direction du gradient de pression, autrement dit, elle sera dirigée principalement dans la direction suivie par l'outil. Il n'est pas possible de transmettre des effets latéraux, les couches de fluide pouvant glisser les unes sur les autres. Ce terme gouverne un comportement très fluide du matériau.

Le deuxième terme, quant à lui, est nul lorsque les vitesses sont homogènes ou varient linéairement dans le fluide, mais prend de l'importance lorsque l'on observe des variations spatiales plus complexes des vitesses. Il est d'autant plus important que la viscosité cinématique ν du fluide est élevée. Ce terme va tenter de lisser le champ de vitesse, et c'est celui-ci qui pose les plus ardues problèmes aux différentes simulations. Nous verrons plus tard qu'il consiste en une diffusion de la quantité de mouvement dans le fluide. C'est grâce à ce terme, notamment, que les mouvements peuvent être transportés dans le fluide sur une distance importante, et ce également de façon transverse, un comportement issu de la nature visqueuse du fluide considéré.

Le troisième terme, enfin, fait apparaître le gradient du champ de vitesse mesuré le long d'une ligne de courant. Il trouve son origine dans la description lagrangienne du problème. En effet, si les particules fluides en amont de la ligne de courant, c'est-à-dire celles qui, très bientôt, parviendront au point considéré, ont une vitesse plus grande que celles, en aval, qui s'en éloignent (auquel cas le gradient est négatif), il est naturel de voir une vitesse locale augmenter. C'est un terme inertiel qui exprime le fait que les particules qui arrivent, qu'elles soient plus rapides ou plus lentes que celles qu'elles remplacent, tendront à conserver leur vitesse le long de la ligne de flux.

Importance relative des différents termes

Il est d'usage, en mécanique des fluides, de pondérer les contributions relatives des deuxième et troisième termes dans l'équation 2.1. La raison principale est que le troisième terme, non-linéaire, est délicat à manipuler lorsque l'on souhaite une résolution analytique du problème. Montrer qu'il est négligeable devant un autre terme, usuellement le second, permet alors de s'en affranchir, les calculs se révélant alors beaucoup plus aisés.

On compare donc ces deux termes d'un point de vue dimensionnel. Leur rapport $\frac{\mathbf{u} \cdot \text{grad}(\mathbf{u})}{\nu \Delta(\mathbf{u})}$ est homogène à un terme de la forme $\frac{v \cdot l}{\nu}$ où v est une vitesse caractéristique de l'écoulement, l en est une dimension, et ν correspond à la viscosité cinématique du fluide. C'est l'expression du nombre de Reynolds. Dans notre cas, les vitesses mises en jeu sont très faibles (le mouvement des outils ne dépassant guère quelques centimètres à la seconde). La viscosité cinématique des matériaux qui nous intéressent ont une viscosité dépassant les 10^6 poiseuilles, et ces matériaux sont adaptés à des sculptures dont la taille va de quelques centimètres à quelques mètres, avec des outils de quelques centimètres.

Autrement dit, le nombre de Reynolds est, dans le cas le plus défavorable, inférieur à 10^{-7} , autrement dit très inférieur à 1. Le terme inertiel est donc parfaitement négligeable dans notre

cas. On ne gardera donc que les deux premiers termes (ainsi que le terme faisant intervenir des forces extérieures, mais encore une fois, nous y reviendrons plus tard). On qualifie parfois les écoulements de ce genre de fluides de “rampants”. Ils sont donc représentés par le jeu d’équations suivant :

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho} \mathbf{grad} p + \nu \Delta \mathbf{u} \quad (+\mathbf{F}) \\ \text{div}(\mathbf{u}) = \mathbf{0} \end{cases} . \quad (3.2)$$

3.1.2 Solution proposée : un modèle en trois couches

Comme on l’a dit, aucun modèle actuellement disponible (fluide, élastique, plastique...) ne saurait nous donner entière satisfaction. Nous venons de voir que dans l’équation de Stokes, deux termes sont important pour décrire le mouvement d’un matériau comme l’argile. Le premier est généralement bien simulé par les modèles de fluide, tandis que le second pose davantage problème.

Notre approche sera donc un peu différente. Plutôt que de simuler l’ensemble des comportements du matériau avec un seul modèle, nous allons en combiner plusieurs afin de rendre compte de l’ensemble des comportements du matériau. Nous allons construire une simulation comportant plusieurs couches qui vont collaborer.

Une première couche simulera, avec des modèles inspirés de la mécanique des fluides, le comportement fluide, généralement local, du matériau (en particulier ceux induits par le premier terme de l’équation de Stokes). Il s’assurera également que le volume du matériau est bien constant. Une seconde couche de simulation ajoutera les effets à grande échelle qui font défaut à notre modèle fluide. Nous introduirons également une troisième couche, simulant une tension de surface, afin de palier des défauts de ces deux modèles conduisant à un émiettement du matériau.

La résolution d’un problème en le découpant en des problèmes plus simples, et en superposant les solutions ainsi obtenues, est une démarche courante dans les sciences physiques. Elle peut d’ailleurs se justifier lorsque les équations sont linéaires. C’est une démarche qui a naturellement été adoptée, également, dans le domaine de l’animation. J. Chadwick *et al* suggéraient par exemple d’utiliser un modèle en couche pour animer un personnage [CHP89]. Au-dessus d’un squelette articulé, les auteurs modélisent une couche de muscles et une peau pour obtenir un personnage. Nous réutiliserons nous-même, un peu plus tard, ce type de représentation pour établir un modèle de main. Les différentes couches peuvent ainsi répondre à des objectifs différents. Dans le modèle proposé par M. Desbrun dans [DC95], un système de particule permet d’obtenir les déformations générales subies par le matériau, mais c’est une surface déformable qui va, indépendamment, s’assurer de la conservation du volume au moyen de déformations locales. C’est donc cette idée que nous allons reprendre.

3.1.3 Une représentation sous-jacente commune

Ces différents modèles vont collaborer pour déformer le matériau. Tous deux vont déterminer des déplacements de fluide, qui vont être combinés afin de décrire toute la gamme de comportements possibles du matériau. Pour ce faire, ces modèles vont devoir travailler sur une même

structure de données. Nous avons choisi d'utiliser une description implicite du matériau. Ces descriptions permettent en effet de traiter aisément le cas des changements de topologie qui nous sont chers.

L'objet sculpté sera donc décrit par un champ scalaire $f(\mathbf{r})$, défini sur tout l'espace de travail. Son volume proprement dit correspondra au lieu des points pour lesquels ce champ se situe au-dessus d'une isovaleur I_{obj} . La visualisation de ce volume consistera donc à rendre l'isosurface vérifiant la condition $f(\mathbf{r}) = I_{obj}$. Ce champ sera discrétisé sur une grille cubique régulière, ses valeurs étant interpolées entre les nœuds de la grille. On dispose d'ores et déjà d'outils permettant de travailler avec un tel champ, notamment pour ajouter et retirer de la matière à l'objet [FCG01]. Notre objectif sera donc d'y ajouter la possibilité d'obtenir divers effets de déformation à petite et grande échelle.

Il n'est généralement pas donné de sens particulier aux valeurs prises par le champ scalaire, hormis parfois la distance à l'isosurface. Nous allons au contraire lui donner un sens particulier. La valeur prise par le champ en un point de la grille cubique correspondra directement à la quantité de matière contenue dans une cellule entourant ce point. Comme le matériau est incompressible, ladite cellule peut contenir une quantité limitée de fluide. Les valeurs prises par le champ scalaire devraient donc en permanence être comprises entre 0 (la cellule est vide) et une valeur limite, que l'on choisira égale à 1 dans la suite, signifiant que la cellule est pleine d'argile.

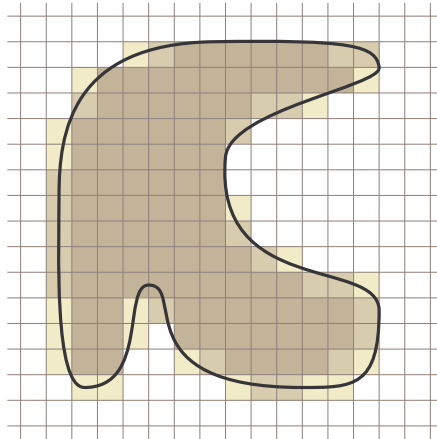


FIG. 3.1 – La matière constituant l'objet est stockée dans une grille régulière. Les cellules les plus foncées correspondent à celles contenant le plus de fluide. Le volume est visualisé par une isosurface de la densité.

Chacun des modèles que nous allons développer va donc travailler sur cette même représentation du matériau. La couche fluide et la couche de déformation vont essentiellement déplacer de la matière d'une cellule à une autre. Ils pourront ainsi travailler en collaboration. Le déplacement de matière se fait simplement en diminuant la valeur du champ dans une cellule, et en l'augmentant de la même quantité dans une autre cellule. Cette action correspond très exactement à un transfert de matière d'une cellule à l'autre. Par construction, la quantité totale de matière dans l'espace de travail restera toujours inchangé.

Il est utile de noter que beaucoup d'outils de sculpture déjà développés peuvent être

aisément employés avec cette représentation. Divers outils permettant d'ajouter ou d'enlever de la matière, ou bien encore de déformer géométriquement la surface, pourront être utilisés. Dans la suite, nous nous développerons ce modèle en couche nous permettant de produire des déformations intuitives et naturelles de l'objet, à petite et grande échelle, comme si ce dernier se trouvait être un matériau de type argile.

Dans le reste du présent chapitre, nous présenterons successivement les principes gouvernant chacune des couches de simulation de notre modèle. Le chapitre suivant abordera quant à lui plusieurs améliorations et enrichissement de ce modèle en couche, permettant d'agrandir notablement la gamme des effets disponibles.

3.2 Première couche : un modèle de matériau fluide

3.2.1 Comportement du matériau

Rappelons avant toute chose que nous allons nous efforcer, dans la suite, de construire un modèle empirique de matériau qui permette de reproduire l'essentiel des effets qui nous intéressent, effets que l'on espère aussi naturels que possible. Ce modèle ne constituera en aucune façon une véritable résolution des équations de *Navier-Stokes* pour le matériau considéré, même de façon approchée. Les discussions qui vont suivre sur la forme prise par les différentes équations, s'appuyant sur des cas physiques, n'ont pas vocation à démontrer de leur justesse, mais s'efforceront plutôt d'expliquer pourquoi leur expression est naturelle. Encore une fois, nous ne voulons pas simuler la réalité physique du problème, mais simplement en reproduire l'essentiel des effets.

Nous l'avons d'ores et déjà dit, c'est le terme faisant intervenir le gradient de pression, $-\mathbf{grad}(p)$ qui va nous intéresser pour l'instant. On ne peut cependant totalement ignorer le terme faisant intervenir le laplacien du champ de vitesse \mathbf{u} . En effet, les matériaux comme l'argile, de par leur viscosité très importante, se caractérisent par un amortissement très fort de tout mouvement à l'intérieur du fluide.

Dès que les contraintes cessent, tout mouvement à l'intérieur du matériau cesse pratiquement immédiatement. C'est justement ce laplacien, qui tend à uniformiser les vitesses, qui empêche toute inertie du fluide. Toutefois, aussitôt qu'une contrainte est imposée au matériau, de très fortes pressions apparaissent à l'intérieur de ce dernier, et le terme $-\mathbf{grad}(p)$ va mettre le fluide visqueux en mouvement. Du terme comprenant un laplacien, nous ne garderons que l'amortissement des vitesses dès que cessent les contraintes. Nous cherchons donc à modéliser un matériau dont le comportement serait décrit par :

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho} \mathbf{grad} p \\ \mathit{div}(\mathbf{u}) = \mathbf{0} \end{cases} . \quad (3.3)$$

Lorsque le fluide est au repos, en l'absence de contrainte, la pression est uniforme à l'intérieur, de sorte qu'aucun mouvement de matière n'est initié. Sous l'action de contraintes de la part des outils, la pression p va évoluer, et le fluide se mettra en mouvement de façon à égaliser à nouveau la pression. On peut écrire la pression p sous la forme $p(\mathbf{r}) = p_0 + \delta p(\mathbf{r})$ où $\delta p(\mathbf{r})$ correspond aux augmentations locales de la pression, dues à l'afflux de matière.

On peut donc réécrire :

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho} \mathbf{grad} \delta p. \quad (3.4)$$

3.2.2 Bilan pour une des cellules de la grille

Considérons le cas d'une cellule de notre grille, comme représentée sur la figure 3.2. Tout mouvement du fluide va se traduire par des échanges de matières avec ses six voisines. Nous allons nous intéresser aux échanges de matière entre la cellule (i, j, k) et sa voisine $(i + 1, j, k)$. Compte tenu de l'équation précédente, les échanges de matière entre les deux cellules sont proportionnels à la différence de pression existant entre les deux cellules. C'est en effet l'expression que prendra le gradient dans ce cas discret.

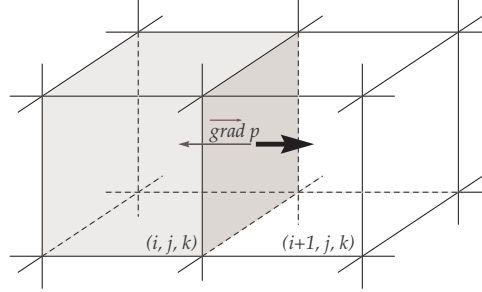


FIG. 3.2 – Les échanges de matière entre une cellule et sa voisine à travers leur facette commune (ici de la gauche vers la droite) s'opposent au gradient de pression.

La quantité de matière δn traversant l'interface séparant ces deux cellules (qui sera positive lorsque la matière passe en majorité de la cellule $(i + 1, j, k)$ à la cellule (i, j, k)), durant un instant δt , est directement proportionnelle à la différence de pression entre les deux cellules. Nous pouvons ainsi écrire¹ :

$$\delta n \propto \delta t (\delta p_{i+1,j,k} - \delta p_{i,j,k}). \quad (3.5)$$

En prenant en compte les échanges avec les six voisines, on trouve

$$\begin{aligned} \delta n \propto \delta t & ((\delta p_{i+1,j,k} - \delta p_{i,j,k}) + (\delta p_{i-1,j,k} - \delta p_{i,j,k}) \\ & + (\delta p_{i,j+1,k} - \delta p_{i,j,k}) + (\delta p_{i,j-1,k} - \delta p_{i,j,k}) \\ & + (\delta p_{i,j,k+1} - \delta p_{i,j,k}) + (\delta p_{i,j,k-1} - \delta p_{i,j,k})). \end{aligned} \quad (3.6)$$

On peut reconnaître dans la formule précédente l'expression du laplacien exprimée sur une grille discrète (on trouvera dans l'annexe A les formulations discrètes des différents opérateurs vectoriels). Le terme δn se simplifie donc, et s'exprime en tout point de la grille selon :

$$\delta n \propto \delta t \Delta(\delta p). \quad (3.7)$$

¹Le sigle \propto signifie ici "proportionnel à". Le coefficient de proportionnalité, qui comprend notamment le terme $1/\rho$, nous importe peu pour la suite.

Dans la mesure où l'excès de pression δp dans une cellule peut être considéré comme proportionnel à l'excès de matière δn , cette relation peut être réécrite $\delta n \propto \delta t \Delta(\delta n)$.

Nous allons donc résoudre la question de la réaction du matériau à une sollicitation extérieure de façon similaire à la démarche suivie par N. Foster dans [FM96] : les actions des outils sur le fluide vont conduire à des déplacements de matière qui ne sont pas compatibles avec le caractère non compressible du fluide (qui interdit à la densité de matière dans une cellule de dépasser son maximum, que nous avons fixé à 1). Pour corriger ces effets indésirables, nous allons appliquer une démarche itérative qui va déplacer le fluide jusqu'à ce que les conditions d'incompressibilité soient vérifiées. Pour ceci, nous allons utiliser la loi d'évolution pour les excès de densité δn que nous venons d'établir :

$$\frac{\partial(\delta n)}{\partial t} = \Delta(\delta n). \quad (3.8)$$

C'est en fait une opération très simple de relaxation entre les cellules : si la matière est présente en trop grande quantité dans une cellule, l'excès est transmis à ses voisines immédiates. A la différence de ce qui se passe en mécanique des fluides, on appliquera cependant ce schéma uniquement dans les cellules où la densité n dépasse effectivement le maximum autorisé. En effet, seuls les excès de pression (lorsque le terme δp est positif) vont effectivement induire des mouvements de matière. Si n est inférieur au maximum, la matière peut trouver suffisamment de place dans la cellule de façon à annuler le terme de surpression δp .

La relaxation va s'achever dès que la matière en excès aura pu être évacuée vers l'extérieur de l'objet, là où il y a de la place vacante. Si l'on revient à l'image acoustique présentée précédemment, l'idée clé est la propagation, la diffusion des excès de pression à l'intérieur du matériau. Il n'est donc pas étonnant de retrouver ici une équation caractéristique des phénomènes de diffusion.

Cette équation de relaxation permettant de déplacer la matière de façon à imposer la condition de non-compressibilité du matériau, a été établie de façon similaire à celle présentée ici dans les travaux de S. Druon [Dru04]. Il est précisé à juste titre que cette équation d'évolution ne garantit pas de façon absolue la conservation de la quantité de matière dans le matériau, mais la mise en œuvre pratique de cette équation de relaxation, telle qu'elle est proposée dans [Dru04] ou comme elle le sera dans la suite, permettra de s'assurer que la quantité globale de matière dans l'espace de travail reste globalement inchangée.

3.2.3 Un automate pour simuler la diffusion

Principe

Reste à présent à implémenter cette relaxation de la façon la plus efficace possible. L'idée qui nous est venue, et qu'ont également choisi de suivre H. Arata [ATTY99] et S. Druon [Dru04], est d'utiliser pour ce faire un automate cellulaire. La façon dont les données sont rangées dans une grille régulière s'y prête à merveille, d'autant que les différentes opérations d'échange de matière entre deux cellules se font localement, entre cellules voisines.

Les automates cellulaires utilisent des opérations généralement simples, répétées sur chacune des cellules de la grille et leurs voisinages respectifs, pour élaborer des comportements complexes. Ils peuvent généralement être implémentés de façon très efficace, et parallélisés au besoin. Cela devrait donc nous permettre d'obtenir une simulation de notre matériau à une vitesse suffisante pour permettre l'interaction avec l'utilisateur.

Opération élémentaire sur une cellule de l'automate

Nous allons nous pencher à présent plus avant sur le comportement d'une cellule de la grille. Elle contient une quantité n de matière. Cette quantité peut, ainsi que nous l'avons dit, librement varier entre 0 (la cellule est vide) et 1 (l'espace est alors entièrement rempli de fluide). Lorsque n est supérieur à 1, il y a un excès de matière $n - 1$ dont la cellule va chercher à se débarrasser. L'action élémentaire de notre automate consistera simplement à retirer cette quantité en excès de la cellule, et la répartir entre ses différentes voisines. Peu importe que les voisines en question soient pleines ou vides, notre automate n'en tiendra pas compte, du moins pour l'instant : sa seule action consiste à répartir la matière en excès dans les cellules voisines. Elles déverseront elles-mêmes leur excédent, le cas échéant, quand viendra leur tour...

Si l'on souhaite imaginer un peu mieux ce qui se passe, on peut se figurer un bac à glaçon placé sous un filet d'eau. L'eau coule dans un des compartiments, et le remplit progressivement. Lorsque celui-ci est plein, il déborde, et l'eau en trop se retrouve dans les quatre compartiments voisins, qui commencent à se remplir. Quand eux-mêmes seront remplis, l'eau ira de proche en proche remplir des compartiments de plus en plus loin. C'est une bonne image, dans le cas bidimensionnel, de ce que notre automate va chercher à faire. Ce type d'opération est parfois également rencontrée dans les simulations de comportement des tas de sable, généralement dans des problèmes à deux dimensions [ON03].

On peut montrer qu'un automate appliquant cette règle élémentaire correspond bien, sur l'ensemble de la grille, à l'équation 3.8 de propagation précédemment établie. En effet, une fois notre opération élémentaire aura été effectuée sur l'ensemble des cellules de la grille, on peut faire le bilan des gains et pertes de matière pour notre cellule. Celle-ci aura perdu une quantité de matière $n_{i,j,k} - 1 = \delta n_{i,j,k}$, correspondant à l'excès de matière contenue dans la cellule. En contrepartie, elle aura reçu de ces six voisines un fraction d'un sixième de leurs excès respectifs $\delta n_{i',j',k'}$. Soit :

$$\delta n_{i,j,k}^{t+1} - \delta n_{i,j,k}^t = -\delta n_{i,j,k} + \frac{\delta n_{i+1,j,k} + \delta n_{i-1,j,k} + \delta n_{i,j+1,k} + \delta n_{i,j-1,k} + \delta n_{i,j,k+1} + \delta n_{i,j,k-1}}{6} \quad (3.9)$$

On peut reconnaître dans cette équation un laplacien exprimé sur une grille discrète. L'évolution de l'excès de matière dans notre cellule correspond donc bien à l'expression 3.8 :

$$\delta n_{i,j,k}^{t+1} - \delta n_{i,j,k}^t = -\Delta \delta n_{i,j,k} \quad (3.10)$$

Un automate cellulaire réalisant l'opération de déversement que nous venons de présenter est donc en mesure de réaliser l'opération de relaxation qui nous intéresse. Seules les cellules présentant un excès de matière effectueront réellement une opération, ce qui limite d'autant le

nombre d'actions à réaliser, une bonne part des cellules de la grille ne vérifiant heureusement pas cette condition.

Il est important de noter que l'opération élémentaire de notre automate ne fait que *déplacer* de la matière d'une cellule vers d'autres cellules. La quantité totale de matière dans l'ensemble de la grille est donc parfaitement conservée, même s'il n'était pas aisé de s'en assurer à partir de l'équation 3.8. La conservation de la quantité totale de matière est pour nous d'une importance cruciale. En effet, il n'y aura pas d'apparition ou de disparition de matière, seulement des mouvements de celle-ci, correspondant à des déformations de l'objet. La densité du fluide étant également fixée (au travers de la limite sur la quantité de matière que peut contenir une cellule), la conservation de la masse se traduit également par une conservation du volume global de l'objet, moyennant quelques légères fluctuations dues à la répartition de la matière sur les bords de l'objet, où la densité ne vaut ni 0, ni 1.

Autres opérations élémentaires envisageables

L'action élémentaire que nous proposons n'est pas la seule en mesure de réaliser la relaxation correspondant à l'équation 3.8. S. Druon propose dans [Dru04] d'utiliser plutôt un automate dit *moyenneur*. L'évolution des valeurs dans la grille s'exprime selon :

$$\delta n_{i,j,k}^{t+1} = \frac{1}{\text{card}(\mathcal{V})} \sum_{(i',j',k') \text{ dans } \mathcal{V}} \delta n_{i',j',k'}. \quad (3.11)$$

Dans cette expression, \mathcal{V} désigne un ensemble de cellules (i', j', k') entourant la cellule considérée (i, j, k) (celle-ci étant incluse dans cet ensemble \mathcal{V}). La solution la plus simple consiste à choisir un voisinage cubique autour de la cellule, mais S. Druon suggère plutôt l'utilisation d'un voisinage de Margolus. Ce type de voisinage, illustré sur la figure 3.3, change à chaque itération, mais présente l'avantage de faire intervenir un nombre nettement plus restreint de cellules par rapport à un voisinage plus classique.

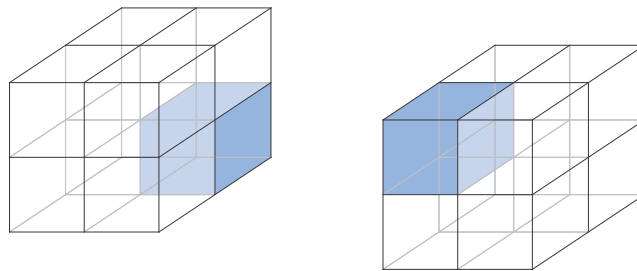


FIG. 3.3 – Les voisinages de margolus d'une cellule, correspondant aux itérations paires et impaires.

Il nous est difficile de discuter des avantages et inconvénients de ces opérations élémentaires. Tandis que notre approche vise plutôt à chasser les excès vers l'extérieur, un automate moyenneur cherche plutôt à extraire ces mêmes excès depuis l'extérieur. Des critères de performance seraient sans doute le meilleur arbitre, mais il semblerait que cela dépende beaucoup de la situation considérée. Dans la suite, nous garderons le principe d'un automate "déverseur",

qui nous paraît un peu plus intuitif et nous permettra de procéder à quelques optimisations intéressantes, mais l'essentiel de la méthode pourrait être adaptée à l'usage de ce genre d'automate "moyenneur".

Mise en oeuvre pratique

Dans un automate cellulaire classique, on applique les règles tour à tour à chacune des cellules. Ici, l'espace de travail peut être vaste, et procéder de cette façon pourrait se révéler être un vaste gâchis de puissance de calcul, puisqu'un nombre relativement limité de cellules seront susceptibles d'éventuellement présenter un excès de matière. Nous allons donc procéder différemment et utiliser une file d'attente, contenant une liste des cellules susceptibles de présenter un excès de matière. Les résultats obtenus seront peu ou prou identiques, mais bien des calculs inutiles seront de ce fait évités.

Procéder de cette façon présente un second avantage. En général, on travaille avec deux grilles distinctes : l'une contient les valeurs initiales pour chaque cellule, l'autre les valeurs mises à jour. Travailler avec deux grilles n'est pas un inconvénient important, c'est en revanche rendu nécessaire car les résultats d'une opération élémentaire sur une cellule pourraient influencer ceux obtenus ensuite sur une cellule voisine. Si l'on travaillait avec une seule grille, la matière se propagerait plus rapidement dans les directions de parcours de la grille, comme le montre la figure 3.4.

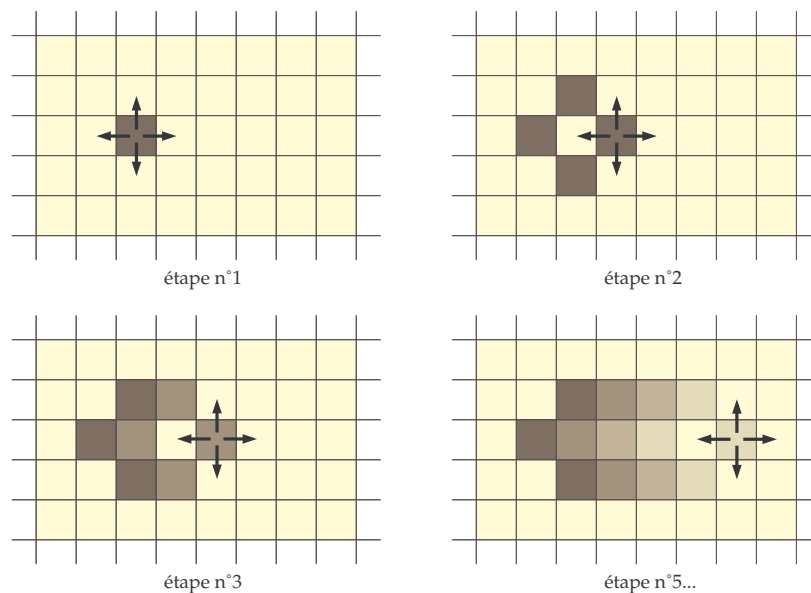


FIG. 3.4 – Un parcours de la grille de gauche à droite accélérerait le mouvement de la matière dans cette direction, par rapport aux directions opposées.

En utilisant une file, la plupart de ces effets ne seront pas notables (nous verrons un peu plus tard que ce n'est pas tout à fait exact, mais qu'il existe des moyens simples d'en

atténuer largement les conséquences). Avec notre approche, nous travaillerons donc sur une grille unique. Certes, on ne respecte peut-être plus à la lettre l'équation de relaxation 3.8, mais l'esprit en est intact. Notons aussi que cette simplification est rendue possible par le fait que le caractère temporel de la relaxation ne nous intéresse pas, mais seulement son résultat une fois la relaxation achevée.

À chaque étape, on récupère la cellule au sommet de la file, et l'on regarde si effectivement excès il y a, auquel cas il sera réparti entre les six cellules voisines. Ces six cellules, avec le supplément de matière qu'elles viennent de recevoir, peuvent voir leur densité excéder la densité critique, et seront donc toutes les six placées à la fin de la file, pour examen ultérieur. On pourrait éventuellement vérifier tout de suite si le seuil est dépassé, de façon à ne pas placer dans la file des cellules dont on se rendra compte ultérieurement qu'elles ne sont pas pleines, mais en pratique, c'est une légère perte de temps. En effet, une cellule pourra recevoir de la matière de plusieurs directions différentes, et ne dépasser le seuil qu'au second, ou au troisième apport. Ne tester le dépassement du seuil que lorsque la cellule arrive en tête de file permet de ne pas multiplier ces tests.

Notons que l'on souhaite éviter qu'une même cellule apparaisse à plusieurs endroits dans la file de traitement. En effet, l'excès de matière sera déversé dans ses voisines dès l'apparition de la cellule en tête de file, et lorsque adviendra le tour des autres occurrences de cette même cellule, il est probable qu'il n'y ait rien de plus à faire. Pour cela, on munit donc chaque cellule d'un drapeau, que l'on lève lorsque la cellule entre dans la file, et que l'on rabaisse lorsqu'elle arrive en tête de file. De cette façon, on sait à chaque instant si une cellule est déjà présente dans la file et le cas échéant qu'il est inutile de l'y faire entrer une seconde fois.

L'opération consistant à examiner si la cellule en tête de file présente un excès de matière, et, si c'est le cas, à le répartir entre ses voisines, sera répétée jusqu'à ce que la file soit complètement vidée de ses cellules. À ce moment, nous saurons que plus aucune cellule ne fait montre d'un quelconque excès de matière. Bien évidemment, le temps nécessaire à une vidange complète de la file est variable, et peut, en théorie, être long (en particulier lorsque des excès apparaissent profondément à l'intérieur d'une pièce, où la matière devra parcourir beaucoup de chemin avant de trouver de la place libre, ce qui arrive heureusement assez rarement lorsque l'on sculpte un objet).

Afin de garder une interactivité avec le matériau, on peut également poser un nombre d'opérations limites au delà duquel on s'arrêtera, laissant en plan l'étape de relaxation. Certes, certaines cellules présenteront encore probablement des excès de matière, et par conséquent la conservation du volume de l'objet n'est alors pas tout à fait assurée (en fait, on observe un rétrécissement du volume, dans la mesure où il est localement plus dense qu'il ne devrait). Cependant, en gardant la pile en l'état, on s'assure que ces excès seront traités dès que l'on disposera à nouveau de temps de calcul pour achever la relaxation. On peut donc forcer une fréquence de travail, au prix éventuel, pour les machines les plus lentes ou les cas les plus complexes, d'un léger retard des effets. Les bourrelets, par exemple, au lieu d'apparaître instantanément, le feront progressivement en l'espace de quelques cycles de la simulation, sans que cela soit particulièrement gênant.

3.2.4 Subtilités de la méthode

Complétion et tolérances

Quelques subtilités concernant l'algorithme précédent méritent d'être mentionnées. Tel qu'il a été présenté, il n'a aucune chance de se terminer un jour. Prenons en effet deux cellules voisines, que nous noterons a et b . La cellule b est pleine, quant à la cellule a , elle contient un excès de matière. L'excès de matière de a sera réparti entre ses six voisines, parmi lesquelles on trouve b , laquelle se trouve alors elle-même avec un excès de matière. La cellule b déverse à son tour cet excédent dans son voisinage, de sorte que a repasse à nouveau la barre critique.

À supposer que les deux cellules soient entourées de cellules pouvant accueillir librement leurs déversements, l'excès de a diminuera à chaque échange de façon géométrique, d'un facteur $1/36$. Il sera donc très rapidement négligeable, mais jamais rigoureusement nul. Il nous faut donc instaurer une tolérance supplémentaire : si l'excès de matière dans une cellule se trouve en deçà de cette tolérance, il ne sera pas déversé, de façon à ce que l'algorithme puisse se terminer. Comme cette tolérance influera sur la conservation du volume de l'objet (la densité des cellules pleines pouvant varier dans cet intervalle de tolérance), il faut généralement la choisir assez faible. C'est un choix à faire entre la précision de la conservation du volume et les calculs engendrés. En pratique, une tolérance de un ou deux pour cent n'est pas visible à l'œil, et permet d'arrêter assez rapidement l'algorithme.

Surrelaxation

Lorsqu'une cellule déverse un excès de matière dans ses voisines, elle peut s'attendre à voir souvent revenir jusqu'à $1/6$ de la matière évacuée, dès l'étape suivante, et même un peu plus ultérieurement. En se basant sur cette constatation, on peut envisager une "sur-relaxation", et déplacer davantage de matière que l'excès constaté dans la cellule. Certes, celle-ci se retrouvera avec une densité strictement inférieure à celle du seuil critique, et donc ne sera plus pleine. Mais cet état de fait est temporaire, dans la mesure où la cellule sera remplie à nouveau par les apports de ces voisines.

À l'intérieur de la matière, cette approche permet de grandement accélérer la vitesse de la méthode. On accède à l'état d'équilibre bien plus rapidement. Il faut cependant s'armer de précautions : si le retour attendu de la matière n'a pas lieu, on se retrouvera avec une matière dont la densité n'est plus la densité maximale, et on constatera une dilatation, quelquefois importante, de l'objet. On peut certes commencer à examiner si les cellules voisines retourneront ou non tout ou partie de la matière évacuée, mais cela nuirait à la simplicité de l'algorithme, et de ce fait grèverait de façon importante ses performances.

Une première réponse consiste à modérer cette sur-relaxation, et déverser une quantité de matière certes un peu supérieure à l'excès mesuré, mais sans toutefois compter sur un retour de chacune des voisines pour combler le manque de matière ainsi créé. C'est une solution qui, en pratique, fonctionne de façon relativement acceptable, permettant un gain notable en termes de performances, sans impact visible sur le comportement du matériau.

Plus astucieusement, on peut réfléchir aux cellules qui ne renverront pas de matière “à l’envoyeur”. Ce sont les cellules qui ne sont pas pleines, principalement celles se situant sur les bords de l’objet. S’il est possible de différencier les cellules internes à l’objet de celles en contact direct avec les cellules, proches des bords, qui ne sont pas pleines, on pourra n’appliquer la “sur-relaxation” qu’aux premières, ou du moins le faire de façon plus importante. Or ce marquage peut être fait (et mis à jour) de façon aisée : chaque fois qu’une cellule apparaissant en tête de file n’est pas pleine, on peut marquer ses six voisines.

Ce marquage permettra d’éviter d’appliquer une pleine sur-relaxation dans les cellules concernées (on peut au choix utiliser un marqueur unique par cellule, ou bien un marqueur spécifique pour chacune des directions, au choix). Ces marqueurs seront à l’inverse retirés pour des cellules pleines en tête de file. Cette détection des cellules à moitié remplies aux bords de l’objet sera d’ailleurs potentiellement utile ultérieurement, lorsque l’on voudra traiter d’effets de surface, ou bien encore simplement afficher l’objet sculpté.

Propagation vers l’extérieur

Ce qui nous importe, dans cette étape de diffusion, c’est que la matière en excès se déplace vers l’extérieur de l’objet, là où elle pourra trouver de la place pour se déverser. Évidemment, le temps passé à déplacer la matière en boucle à l’intérieur de l’objet est du temps perdu. Une solution couramment utilisée dans ce genre d’application consiste à privilégier les déplacements de matière dans la direction opposée au gradient de densité au niveau de la cellule. Dans beaucoup d’applications, le gradient du champ scalaire qui sert à définir l’objet est en effet une bonne approximation de la direction dans laquelle l’isosurface est la plus proche (plus précisément, dans la direction opposée au gradient lorsque l’on se trouve dans l’isovolume, dans la direction de ce même gradient lorsque l’on est à l’extérieur de ce même volume).

En pratique, cette solution ne fonctionne pas correctement dans notre cas, de part le sens particulier que l’on a donné au champ scalaire. En effet, en temps normal, l’intérieur de l’objet est caractérisé par un champ scalaire uniforme, toutes les cellules étant pleines. Le gradient y est donc nul, il est alors impossible de savoir quelle direction nous pourrions privilégier. Ensuite, et plus gênant encore, lorsqu’une cellule contenant un excès de matière le déverse dans ses voisines, elle crée un minimum local du champ scalaire, et si nous utilisons le gradient, nous risquons de privilégier le retour de la matière dans cette même cellule.

Si l’on veut bénéficier du même effet, il faudra donc procéder différemment, et conserver d’une façon ou d’une autre la direction dans laquelle on peut espérer trouver des cellules qui ne soient pas pleines. Pour ce faire, nous ajoutons à chaque cellule une caractéristique supplémentaire $d_{i,j,k}$, qui figurera une distance la séparant des plus proches cases qui ne sont pas vides. Bien évidemment, ces distances devront être mises à jour en permanence. Pour ce faire, nous rajoutons simplement une règle supplémentaire dans l’automate. Lorsque l’on s’occupe de la cellule (i, j, k) , si celle-ci n’est pas pleine, alors nous avons $d_{i,j,k} = 0$. Dans le cas contraire, la distance $d_{i,j,k}$ peut être considérée comme le minimum des distances de ses six voisines augmenté d’une unité.

L’algorithme, au début, va se caractériser par un manque d’information : les déversements n’ayant pas encore trouvé d’emplacement libre, les distances vont simplement augmenter un

peu partout au sein du matériau. Toutefois, dès que la propagation va atteindre l'un des bords de l'objet, il va se mettre en place un gradient qui pourra guider la matière vers la sortie. En effet, les cellules sur les bords de l'objet voient leur distance $d_{i,j,k}$ fixée à 0, et toutes les cellules qui leur sont attachées vont rapidement voir cesser l'augmentation de leur quantité $d_{i,j,k}$.

La connaissance de ces coefficients $d_{i,j,k}$ permet de favoriser les déplacements de matière vers l'extérieur : lorsqu'une cellule a un excès de matière à déverser, elle peut le faire uniquement dans ses voisines (i', j', k') vérifiant $d_{i',j',k'} \leq d_{i,j,k}$. Les résultats obtenus sont subtilement différents visuellement, mais la différence est mince. On verra moins souvent apparaître un gonflement d'un côté de l'objet produit par un excès de matière au voisinage situé de l'autre côté, comme cela peut arriver en utilisant l'approche présentée tantôt. En fonction des situations, le gain en termes de temps de calcul peut être notable, en particulier lorsque le chemin séparant les excès de matière de la surface libre de l'objet est long.

Du voisinage d'une cellule

On peut également se demander si le choix d'un voisinage en 6-connexité dans notre grille est le meilleur choix possible. C'est assurément le plus logique : les échanges de matière se faisant à travers les faces des cellules, celles-ci peuvent n'échanger de matière qu'avec leurs six voisines immédiates. On pourrait cependant envisager de déverser les excès de matière dans des cellules non immédiatement voisines, en espérant de cette façon accélérer quelque peu la relaxation. Encore faut-il toutefois pouvoir pondérer les différentes contributions pour des cellules dont les rôles ne sont pas équivalents.

La meilleure façon de procéder dans ce cas est de considérer ce qui se passe si l'on applique l'algorithme présenté de façon à mesurer quelle quantité de matière devrait aboutir dans ces cellules. En gros, un dix-huitième de l'excès atteint les cellules partageant une arête avec la cellule centrale, après deux itérations, autrement dit trois fois moins que ses voisines directes². Avec cette idée en tête, on peut choisir de déverser un trentième de l'excès dans lesdites cellules, et alors seulement un dixième dans les voisines directes (on peut aisément vérifier que cela répartit bien la totalité de la matière en excès).

Nous avons testé cette méthode, mais en pratique, le gain en performance n'est absolument pas convaincant, dans la mesure où, dans le même temps, on complique sérieusement chaque étape de l'itération. Quant aux différences en terme de résultat, elles ne sont pas notables visuellement. En outre, cette modification n'est pas compatible avec d'autres aspects de la méthode, et en complique sérieusement d'autres (principalement du fait du nombre accru de voisins à gérer).

Commutativité et propagation parasite

Puisque l'on applique successivement les opérations dans chaque cellule, sans utiliser une seconde grille, l'ordre des opérations se trouve avoir une certaine importance. Si deux cellules

²Les cellules liées par un sommet reçoivent quant à elles, en trois temps, un soixante-douzième de la matière en excès, ce que l'on peut aisément négliger

voisines ont chacune à déverser un excès de matière, l'ordre dans lequel on va les traiter conduira à des résultats notablement différents. En fait, si l'on a une série de cellules alignées, toutes présentant des densités au-delà du seuil critique, on peut voir apparaître un transport parasite de la matière le long de cet alignement de cellules comme illustré sur la figure 3.5.

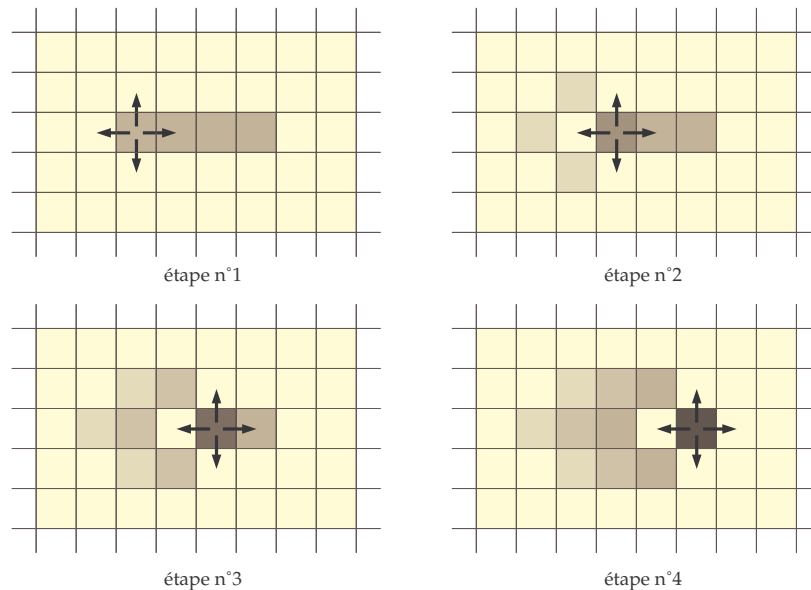


FIG. 3.5 – L'ordre des cellules peut avoir une importance si l'on souhaite éviter des propagations parasites, comme ici vers la droite.

Avec un minimum de précautions, on peut toutefois assez largement s'affranchir de ces problèmes. Les cellules de la grille (dont nous noterons les trois coordonnées i , j et k) peuvent être réparties en deux catégories : celles, que l'on qualifiera de “paires”, dont la somme $i + j + k$ est congrue à 0 modulo 2, et celles, “impaires”, dont la même somme est congrue à 1. Les cellules paires n'ont que des voisines impaires, et réciproquement. En traitant le cas des seules cellules paires, puis celui des cellules impaires, la majeure partie des problèmes rencontrés sera écartée. Notamment, on ne constatera plus les effets de transport parasite précédemment énoncé (en fait, *stricto sensu*, ils existent encore, mais à un ordre infiniment moins gênant).

Reste à savoir comment traiter alternativement les cellules paires et les cellules impaires. En fait, si à un instant, ces deux classes de cellules sont séparées dans la file, elles le resteront tout au long de la relaxation. Prenons par exemple le cas des cellules paires rassemblées en tête de la file, les cellules impaires remplissant le reste de celle-ci. Les cellules traitées sont alors paires, et n'ajouteront en bout de file que leurs voisines, toutes impaires. Évidemment, la réciproque est tout aussi valable. Il suffit donc, lorsque l'on construit la file initiale, de ranger par exemple en premier lieu toutes les cellules paires, et dans un second temps toutes les cellules impaires. Cela se révèle trivial dans le cas d'une file vide, à peine plus complexe pour une file contenant déjà des cellules.

3.2.5 Déversement direct

Plutôt que d'utiliser notre démarche itérative pour évacuer progressivement la matière vers l'extérieur, on pourrait songer à directement rechercher à quel endroit cette matière en excès pourrait être déversée. En pratique, cela consiste essentiellement à chercher les cellules pouvant accueillir de la matière les plus proches de celles où se trouvent les excès. Là encore, on peut chercher ces cellules grâce à un schéma de propagation.

Partons d'une cellule présentant un excès de matière. Ses six voisines immédiates sont à une distance de 1 de la cellule étudiée. En cherchant les voisines de ces voisines, nous trouverons les cellules à une distance de 2. Pour éviter tout retour en arrière (parmi ces "voisines de voisines", il y a la cellule qui nous intéresse, et qui n'est bien évidemment pas à une distance de 2), il suffit de marquer les cellules déjà traitées. En continuant de proche en proche, il est aisé de trouver les cellules à une distance de 3, puis de 4, etc. de la cellule présentant un excès. A chaque étape, il est possible de regarder si certaines des cellules peuvent accepter de la matière.

Si c'est le cas, il est possible de répartir l'excès de matière entre ces différentes cellules. Il faut toutefois faire attention de ne pas y introduire davantage de matière qu'elles ne puissent contenir, et il convient de procéder avec prudence. Notons e l'excès de matière à déverser, et n le nombre de cellules prêtes à en accueillir. On cherche ensuite, parmi les cellules en question, celle pouvant accueillir le moins de matière. Si elle est en mesure d'en accepter une quantité e/n , alors on peut placer une quantité e/n de fluide dans toutes ces cellules. Dans le cas contraire, on remplit la cellule jusqu'au seuil. Il reste alors une quantité e' de matière à déverser, et $n - 1$ cellules. On recherche à nouveau la cellule la plus pleine parmi les cellules restantes, de façon à regarder si elle peut accepter une quantité $e'/(n - 1)$ de matière. Et ainsi de suite. Si le nombre de cellules disponible atteint 0 avant que toute la matière ait pu être déversée, on franchit une étape de plus en cherchant les cellules éloignées d'une unité de plus, et l'on recommence.

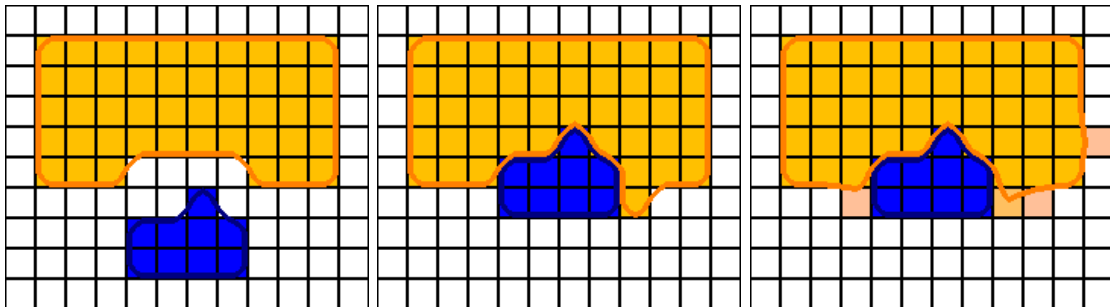


FIG. 3.6 – Résultats en deux dimensions avec la méthode du déversement direct de l'excès de matière. À gauche, la position initiale. À droite, le résultat obtenu après déplacement de l'outil, respectivement pour $\alpha = 1$ et $\alpha = 0.5$.

Selon les cas, cette approche peut fonctionner mieux ou moins bien que la méthode présentée précédemment. Dans certains cas très obstrués, le temps de calcul économisé peut être important. En revanche, il faut appliquer l'algorithme précédent pour chacune des cellules présentant un excès de matière, ce qui peut être rapidement pénalisant si beaucoup de cel-

lules sont concernées. D'autant qu'il est nettement plus délicat (mais néanmoins possible) de différer le traitement de certaines cellules dans l'image suivante. Cela signifierait que certains bourrelets apparaîtraient immédiatement, alors que d'autres ne le feraient qu'ultérieurement, ce qui peut être visuellement dérangeant.

Le nombre de cellules à traiter peut toutefois être nettement réduit si l'on considère non pas les excès d'une seule cellule, mais ceux de huit cellules d'un seul coup. Une erreur d'une case sur le calcul de la zone libre la plus proche ne génère pas d'artefacts très visibles. La démarche devient alors nettement plus exploitable en pratique, même si elle n'est pas toujours idéale si l'on souhaite s'assurer un framerate stable.

Les résultats obtenus par cette démarche sont très légèrement différents, les bourrelets étant davantage "collés" aux outils. La raison tient au fait que la zone libre la plus proche est la seule qui va effectivement recevoir de la matière, les zones plus éloignées étant délaissées. Cela donne une matière plus fluide en apparence, et des résultats qui ne sont pas toujours satisfaisants. Pour atténuer cet effet, plutôt que de déverser l'intégralité de l'excès ϵ dans les premières cellules que l'on trouve, on peut n'en déposer qu'une fraction $\alpha\epsilon$, où α est un paramètre que l'on peut ajuster. Des valeurs situées entre 0.5 et 0.7 permettent d'obtenir des déversements un peu plus doux à proximité des outils, ainsi que le montre la figure 3.6.

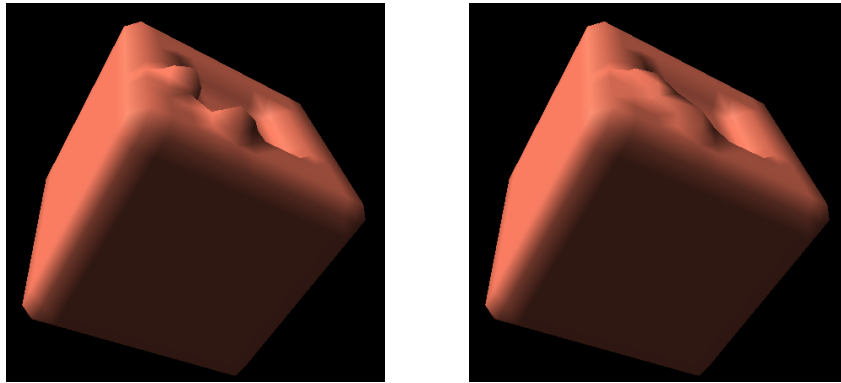


FIG. 3.7 – Comparaison des résultats sur un objet tridimensionnel pour pour $\alpha = 1$ (à gauche) et $\alpha = 0.5$ (à droite). De même que dans le cas bidimensionnel, le paramètre α permet d'obtenir un bourrelet plus doux et plus naturel. L'aspect quelque peu anguleux de l'objet est dû à la faible résolution utilisée pour mieux faire apparaître les détails.

3.3 Interaction de la première couche avec les outils

3.3.1 Introduction

Dans tout ce qui a précédé, nous avons exposé une méthode permettant d'éliminer, par propagation, des excédents locaux de matière dans le matériau. Nous n'avons pas abordé, pour l'instant, les causes ayant pu créer cet excédent de matière. Certes, on peut envisager de directement les créer, au moyen d'un outil ajoutant de la matière dans certaines cellules. Les effets en sont d'ailleurs intéressants : cela produit des excroissances et des gonflements

dont la forme correspond globalement à la géométrie des cellules dans lesquelles on a ajouté de la matière. La forme est d'autant plus adoucie que cet ajout se fait profondément dans le matériau. Une simple cellule, dans un espace de travail vide, dans laquelle on injecterait une grande quantité de matière permet d'obtenir une boule de matière dont le volume est directement lié à la quantité de matière injectée.

On pourrait arguer qu'il est uniquement possible de produire des gonflements de l'objet, par ajout de matière. En effet, retirer localement de la matière ne permettra pas d'obtenir des creux ou des sillons, la matière n'ayant pas de loi particulière, pour l'instant, concernant des cellules dont la densité est inférieure à la densité limite. En fait, il est possible d'obtenir l'effet inverse, permettant de creuser le matériau, simplement en utilisant le "complémentaire" de l'espace de travail : plutôt que de creuser le matériau, nous allons faire enfler le vide qui l'entoure, ce qui donnera le même résultat. En pratique, cela revient à échanger les rôles du matériau et du vide qui l'entoure, donc à remplacer la densité $d_{i,j,k}$ dans tout l'espace par la valeur $1 - d_{i,j,k}$ (1 correspondant à la quantité maximale de matière que peut contenir une cellule).

Cette approche permet de générer des bosses et des creux à la surface de l'objet, et se révèle être un moyen intéressant pour ajouter ou retirer de la matière à l'objet. Cela ne correspond cependant pas à ce que nous souhaitons obtenir : un matériau que l'on puisse réellement déformer à l'aide d'un outil, dans lequel il soit possible de laisser des empreintes en pressant ledit outil sur l'objet sculpté, simplement en repoussant la matière et en s'appuyant sur des considérations de conservation du volume. Nous allons voir comment créer de tels outils dans les paragraphes qui suivent.

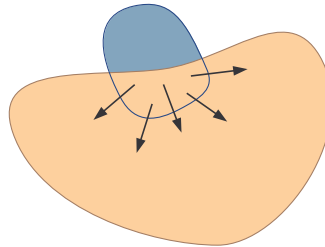


FIG. 3.8 – L'outil interagit avec l'objet sculpté lorsque leurs volumes s'intersectent. La matière est alors chassée hors du volume de l'outil.

3.3.2 Description des outils

Un outil, que ce soit la main de l'artiste ou bien un objet quelconque, est avant tout un objet dans lequel l'argile ne peut pénétrer. C'est une zone de l'espace dans laquelle il ne peut se trouver de matière. Lorsque l'outil se déplace dans l'espace de travail, il peut arriver que de la matière, immobile pour l'instant dans l'espace de travail, se retrouve à l'intérieur du volume de l'outil. La matière va alors simplement être chassée vers l'extérieur de l'outil, comme si celui-ci l'avait repoussée hors de son volume.

La première étape consistera donc à extraire le fluide au sein de l'outil. Ensuite, il est probable que le déplacement du fluide qui vient de se produire ait causé l'apparition d'excès

locaux de matière, comme le déplacement de la membrane d'un haut-parleur peut générer une surpression de l'air. Nous allons pouvoir utiliser la démarche de relaxation présentée tantôt pour traiter le cas de cet excédent. L'excès de matière va s'écouler jusqu'à trouver un peu de place libre pour se déverser.

Chasser la matière de l'intérieur de l'outil peut être fait de différentes façons. La plus immédiate consiste à retirer la matière présente à l'intérieur de l'outil, et la reverser dans les cellules les plus proches situées hors de l'outil. Cette solution présente l'avantage d'être simple, mais il faut prendre garde à certaines choses. En particulier, si l'outil arrive rapidement au contact de l'objet, et que l'on utilise le déplacement de l'outil pour chasser la matière se trouvant à l'intérieur de l'outil, on risque fort de la déplacer trop loin.

Une solution simple consiste à décomposer le mouvement de l'outil en plusieurs mouvements de plus petite amplitude. Éventuellement, de discrétiser le mouvement de l'outil en une succession de déplacements élémentaires sur la grille. De cette façon, les déplacements ne causeront jamais d'éjection indésirable de la matière à une grande distance. Cette approche peut cependant être un peu fastidieuse.

3.3.3 La relaxation pour chasser le fluide de l'outil

Nous avons préféré procéder différemment, et utiliser également le processus de relaxation pour chasser la matière hors de l'outil. Nous avons dit que la matière était en excès dans une cellule si elle dépassait un certain seuil. Dans l'outil, ce seuil est simplement ramené à 0 : toute matière à l'intérieur du volume de l'outil est en excédent, et doit être déplacée. Sur les bords de l'outil, le seuil prendra une valeur différente. En effet, les cellules concernées sont seulement en partie occupées par le volume de l'outil, il reste donc un peu de place pour le fluide.

Cette gradation du volume occupé au niveau du bord de l'outil est importante si l'on s'intéresse à l'état de surface de l'objet sculpté. En effet, si nous imposions que les cellules soient uniquement soit vides, soit pleines, nous nous exposerions à d'éventuels problèmes d'aliasing. La surface obtenue aurait alors un aspect rugueux et anguleux. La gradation de la place disponible dans les cellules frontières assure à la surface de l'objet sculpté un aspect aussi lisse que celui de l'outil utilisé.

3.3.4 Guidage de la relaxation

La démarche présentée précédemment présente un inconvénient important. En effet, la relaxation assure que la matière va sortir de l'outil, mais ne précise pas de quel côté elle va le faire. La majorité de la matière va ressortir là où la surface de l'outil est la plus proche, mais une partie pourra traverser l'outil de part en part et se retrouver derrière lui. Cet effet est bien évidemment gênant, puisque dans la réalité la matière ne traverse pas l'outil, mais est uniquement repoussée par ce dernier.

Afin de corriger cet inconvénient, nous allons guider la matière à l'intérieur de l'outil dans la bonne direction. Plutôt que de déverser les excès de matière selon les six directions, comme

c'est le cas habituellement, nous allons privilégier certaines d'entre elles, et empêcher le mouvement de la matière selon certaines directions. Sur les bords de l'outil, seuls les déversements vers l'extérieur du volume seront autorisés. De cette façon, on empêche également la matière de retourner à l'intérieur de l'outil lors de la relaxation : il fait office d'obstacle au déversement, et le fluide est obligé de le contourner pour trouver de la place libre.

A l'intérieur même de l'outil, on peut utiliser le même principe : les seules directions autorisées sont celles qui amènent la matière vers l'extérieur. Il peut être intéressant de privilégier également les directions qui correspondent au mouvement de l'outil, de façon à ce que la matière soit bien repoussée en aval de ce dernier, plutôt que latéralement. Utiliser uniquement la direction de déplacement de l'outil serait envisageable, mais cela se révèle pénalisant du point de vue du temps de calcul. En effet, très souvent, la matière repoussée par l'outil devra de toute façon par la suite le contourner pour trouver des cellules non remplies. Si de la matière entre profondément dans l'objet, il n'est donc pas choquant qu'une partie soit évacuée directement latéralement plutôt qu'uniquement dans la direction du déplacement.

3.3.5 Définition de l'outil

Volume et espace occupé

Les outils utilisés sont donc essentiellement des volumes munis d'informations supplémentaires indiquant les directions dans lesquelles la matière doit s'écouler. Le volume est défini de la même façon que l'objet sculpté, par une fonction f_o définie sur l'espace qui prend la valeur 1 dans l'outil, décroît rapidement jusqu'à 0 au niveau de sa surface, et vaut 0 partout ailleurs. Il est possible d'utiliser tant des représentations analytiques que discrètes pour décrire la surface implicite de l'outil.

Pour connaître la place disponible dans une cellule quelconque de l'espace de travail, il suffit donc de connaître la valeur v que prend cette fonction f_o au centre de la cellule, qui représente la part de la cellule occupée par l'outil. L'espace restant dans la cellule, à la disposition du fluide, sera donc $1 - v$. Si la quantité de matière dans la cellule dépasse cette quantité $1 - v$, elle sera partiellement chassée lors du processus de relaxation. La répartition de la matière à la surface de l'objet sculpté correspondra donc, après un contact avec l'outil, au complémentaire de la fonction f_o . Ainsi, comme nous l'avons souligné, il est possible d'obtenir des états de surface satisfaisants pour l'objet.

Les méthodes habituelles permettront le rendu de l'isosurface correspondant à f_o , qui figurera le volume de l'outil. Comme l'outil est indéformable, cependant, la surface en question peut aisément être précalculée une fois pour toutes. Le choix de l'isovaleur utilisée est important. Nous souhaiterions, en effet, que lorsque l'outil est en contact avec l'objet sculpté, la surface de celui-ci et celle de l'outil coïncident. Pour ce faire, si l'isovaleur choisie pour rendre la surface de l'objet est I_{obj} , il suffira de prendre pour l'outil une isovaleur de $1 - I_{obj}$. En effet, dans les cellules frontières, les densités de l'objet et de l'outil se complètent, après relaxation, pour atteindre la valeur du seuil, à savoir 1 dans le cas qui nous occupe (le fluide occupant le reliquat d'espace libre laissé par l'outil).

Différents types d'outils

On peut d'ores et déjà envisager toute une gamme d'outils où la fonction f peut être définie de manière analytique. Par exemple, un outil sphérique de taille s placé en \mathbf{r}_0 sera décrit par la fonction

$$f_o(\mathbf{r}) = \begin{cases} 1 & \text{si } |\mathbf{r} - \mathbf{r}_0| \leq s - \epsilon/2 \\ 0 & \text{si } |\mathbf{r} - \mathbf{r}_0| \geq s + \epsilon/2 \\ 1/2 - \frac{|\mathbf{r} - \mathbf{r}_0| - s}{\epsilon} & \end{cases} \quad (3.12)$$

où ϵ est un terme positif qui permet éventuellement de passer doucement et continûment de 1 à 0 pour éviter des problèmes d'aliasing sur l'objet sculpté.

Cela permet déjà de créer une assez grande variété d'outils de tailles et de formes variées. Toutefois, n'importe quelle fonction f répondant aux critères précédents pourra convenir. La fonction f n'a pas besoin d'être analytique, il suffit de pouvoir l'évaluer en tout point de l'espace. On peut donc utiliser comme fonction f un champ scalaire, défini sur les sommets d'un maillage, et interpoler les valeurs du champ entre les sommets du maillage.

Les objets sculptés par notre système entrent eux-mêmes dans cette catégorie. Il sera donc possible, pour l'utilisateur, de sculpter ses propres outils. Par exemple, il peut réaliser un objet en argile, puis s'en servir comme outil (cette fois-ci rigide) afin de faire des empreintes sur une autre sculpture. De façon générale, n'importe quelle surface ou volume peut, moyennant une éventuelle conversion, servir d'outil de sculpture.

Directions pour l'écoulement

Nous avons dit précédemment qu'il était utile, dans le cadre de l'étape de relaxation, de disposer de directions dans lesquelles chasser la matière se trouvant dans de l'outil. La seule donnée de $f(\mathbf{r})$ ne saurait donc suffire, il faut y adjoindre un champ de vecteurs $\mathbf{v}(\mathbf{r})$ précisant en tout point la direction à suivre pour atteindre l'extérieur de l'outil. Peu importe la façon dont on calculera ces directions, qui n'ont pas, par ailleurs, besoin d'être déterminées avec précision. C'est essentiellement un moyen de guider la matière dans la bonne direction. Comme les outils sont indéformables, ces directions peuvent être calculées une fois pour toutes, par exemple lors du chargement de l'outil, et n'auront pas à être mises à jour durant l'utilisation de l'outil.

Il est d'usage, lorsque l'on s'intéresse aux surfaces implicites, d'utiliser comme indication de la direction vers l'extérieur du volume le gradient du champ scalaire f . Toutefois, dans le cas qui nous occupe, ce gradient est généralement nul au sein de l'outil (où la fonction f est homogène).

Pour des formes prédéterminées (comme c'est par exemple le cas de la sphère présentée ci-dessus), ce champ de vecteur n'est guère difficile à établir. Le problème peut venir en revanche des outils importés, par exemple ceux créés par notre propre dispositif de sculpture virtuelle. Il existe cependant de nombreuses façons de résoudre ce problème. Nous allons en proposer une pour le cas d'une fonction f discrétisée sur une grille, comme c'est le cas pour un outil créé par l'utilisateur.

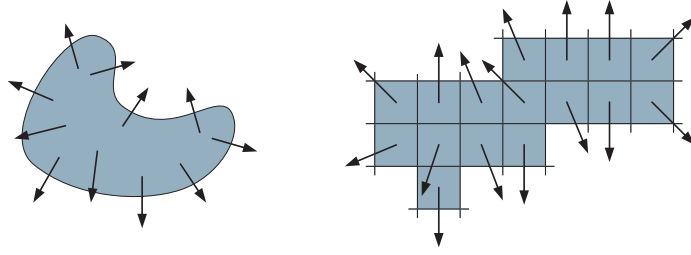


FIG. 3.9 – Que l’outil soit défini de façon analytique ou discrète, une direction privilégiée pour l’écoulement est définie en tout point.

Pour chaque point \mathbf{r}_i de la grille situé à l’intérieur de l’outil, on définira une direction privilégiée pour l’écoulement par

$$\mathbf{U}_i = \sum_j \frac{\mathbf{r}_i - \mathbf{r}_j}{|\mathbf{r}_i - \mathbf{r}_j|^4} \quad (3.13)$$

où la somme est conduite sur l’ensemble des points \mathbf{r}_j de la grille qui vérifient $f(\mathbf{r}_j) = 0$ (point situé complètement à l’extérieur de l’outil) et dont l’un des six voisins immédiats au moins ne remplit pas cette condition. Cela revient essentiellement, en pratique, à chercher la direction dans laquelle l’extérieur de l’outil est le plus proche. De ce vecteur \mathbf{U}_i , on ne conservera que la direction, sous la forme d’un vecteur normé \mathbf{u}_i . Cet ensemble de vecteurs normés, évalués régulièrement dans le matériau, indiquent la direction dans laquelle la matière devra sortir.

Utilisation des directions

Lorsque l’on déplace l’outil dans l’espace de travail, nous déplacerons donc avec lui, de façon rigide, l’ensemble de ces directions. Pour chaque cellule à l’intérieur de l’outil, on peut alors connaître une direction privilégiée pour l’écoulement. On pourra, au choix, prendre le vecteur de direction le plus proche du centre de la cellule, ou bien une interpolation des différents vecteurs entourant cette même cellule (en gardant à l’esprit qu’il faudra renormaliser cette interpolation).

Le vecteur directeur $\mathbf{u}_{i,j,k}$ servira alors à déterminer les directions utilisées. L’écoulement dans une direction est permis si l’angle entre ladite direction et le vecteur directeur $\mathbf{u}_{i,j,k}$ est inférieur à un certain angle choisi à l’avance, ainsi que le montre la figure 3.10. La condition sur l’angle est aisément traduisible en une condition sur son cosinus, et donc en une condition sur le produit scalaire entre $\mathbf{u}_{i,j,k}$ et la direction considérée. Si cette direction est l’une des directions de la grille, le cosinus correspond directement aux différentes composantes de $\mathbf{u}_{i,j,k}$ suivant les trois axes.

Il nous est apparu qu’un angle de 60° donnait des résultats probants. La difficulté est en effet de ne pas trop favoriser une direction par rapport à une autre, tout en limitant quand même les directions autorisées. Les conditions se traduisent alors aisément sur les composantes de $\mathbf{u}_{i,j,k}$. Par exemple, les déversements suivant l’axe x , dans la direction des x positifs, seront admis si et seulement si :

$$\mathbf{u}_{i,j,k}^x > 0.5. \quad (3.14)$$

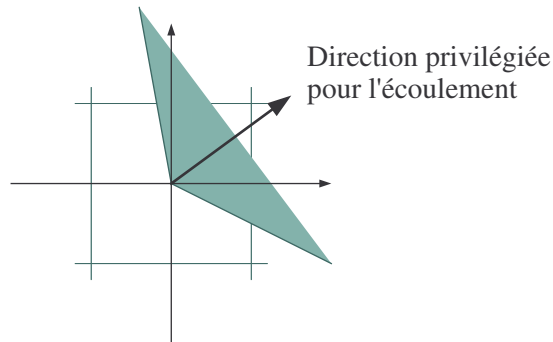


FIG. 3.10 – La matière peut s’écouler dans un cône de demi-angle au sommet de soixante degrés autour de la direction $\mathbf{u}_{i,j,k}$. Sur cet exemple, l’écoulement est possible dans la direction des x positifs et des y positifs.

Il en va de même pour chacune des autres directions. Le nombre de voisines accessibles pour chaque cellule peut être stocké, de façon à pouvoir rapidement calculer la part que chacune d’entre elles recevra d’un éventuel excès de matière.

Lorsque l’outil est en mouvement, on pourrait modifier les directions privilégiées de l’écoulement en fonction du déplacement. A cet effet, plutôt que d’utiliser le seul vecteur $\mathbf{u}_{i,j,k}$ pour déterminer la direction privilégiée, on choisira un terme de la forme $\alpha \mathbf{u}_{\text{outil}} + (1 - \alpha) \mathbf{u}_{i,j,k}$, où $\mathbf{u}_{\text{outil}}$ est un vecteur unitaire indiquant la direction du mouvement de l’outil, et α un paramètre compris entre 0 et 1 que l’on peut ou non faire dépendre de la vitesse de déplacement. Un paramètre α élevé tend à davantage “enfoncer” la matière dans l’objet lors des contacts, ce qui peut occasionner davantage de calculs. Visuellement, les différences n’étaient pas notables pour que nous décidions d’user de cette possibilité.

3.3.6 Limitations de l’interaction

Ce type d’outil donne des résultats satisfaisants. En particulier, on peut obtenir aisément des empreintes convaincantes, et percer sans trop de difficultés des trous dans le matériau, ainsi qu’on peut le voir sur les exemples ci-dessous. Évidemment, seuls les effets dus à la fluidité du matériau sont pour l’instant accessibles, nous verrons, dans la suite du présent chapitre, comment le modèle peut être enrichi.

Il y a cependant deux effets indésirables qu’il convient de noter. D’abord, notre méthode ne fait que chasser, à chaque instant, la matière au sein de l’outil, et ne prend pas en compte les déplacements de l’outil entre ces différents instants. Ainsi, si un outil, situé initialement à l’extérieur de l’objet, est brusquement amené dans de ce dernier, la conséquence sera la création d’une bulle de vide à l’intérieur de l’objet (et d’un gonflement en conséquence, assurant la conservation du volume). Aucune pénétration de l’objet ne sera visible, comme si l’outil avait été directement “téléporté” dans l’objet.

Cela limite la vitesse de déplacement des outils, si l’on veut que les résultats soient de bonne qualité. On peut éventuellement décomposer les mouvements de chaque outil en plusieurs étapes lorsque ceux-ci sont importants, la seule limitation étant bien entendu le surcoût de

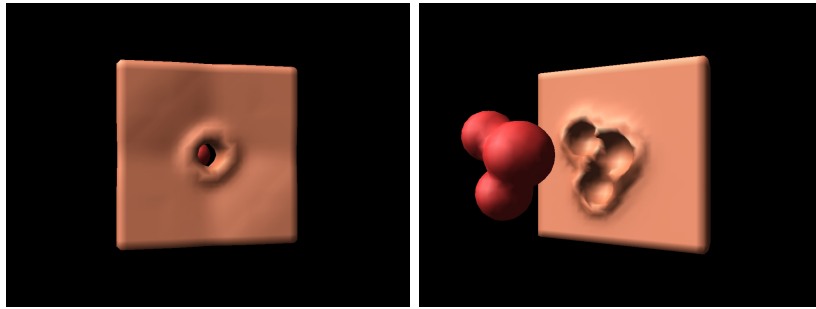


FIG. 3.11 – La première couche de notre argile virtuelle permet le percement du matériau par l’outil, et la création d’empreintes dans l’objet modelé.

calcul que cela engendrerait.

Une autre limitation du modèle actuel est qu’il n’a pas beaucoup de tenue. Il se comporte de façon très similaire à un fluide, et les outils ont tendance à chasser de la matière devant eux. Lorsque l’on perce un trou, par exemple, il reste un peu de matière en aval de l’outil, comme s’il en sortait humide. Cette matière est essentiellement perdue, et se disperse dans l’espace de travail. Une petite partie du volume peut alors disparaître. Cet effet est dû à l’absence de toute tension de surface dans notre matériau, et nous verrons également, à la fin de ce chapitre, comment on peut envisager d’atténuer cet effet.

3.4 Objectifs de la seconde couche

3.4.1 Insuffisance de la seule couche fluide

Le modèle développé jusqu’à présent permet de donc simuler un certain nombre des comportements que nous recherchons. Principalement, on retrouve les conséquences de la conservation de volume provenant du caractère non compressible du matériau : la possibilité de réaliser des empreintes en poussant des outils sur l’objet sculpté, et d’obtenir des bourrelets de part et d’autre de l’outil. Les éventuels changements de topologie peuvent librement être obtenus, grâce au choix d’une représentation implicite pour le matériau : nous avons vu, en effet, qu’il était aisé de percer un trou dans le matériau simplement en poussant un outil à travers ce dernier.

Nombre d’effets ne sont toutefois pas encore disponibles. En particulier, il n’est pas possible de plier une partie de la sculpture en déplaçant une de ses extrémités, ou d’obtenir des déformations globales similaires du matériau. C’est un inconvénient important si l’utilisateur veut par exemple pouvoir plier une partie de sa sculpture. Ces déformations proviennent, nous l’avons dit, de la très grande viscosité du matériau, parfois à la limite d’un corps solide. Ces effets ne peuvent être simulés par les méthodes usuellement utilisées pour modéliser les fluides, mêmes visqueux, à partir des équations de *Navier-Stokes*. D’autres effets manquent également à l’appel, comme par exemple la transmission d’efforts latéraux. Déplacer latéralement un outil le long de la surface de l’objet, par exemple, n’entraînera pas de conséquence particulière : le matériau glisse le long de l’outil sans être emporté. Il a un comportement essentiellement

“fluide”, sans aucun effet de viscosité.

3.4.2 Une seconde couche pour les déformations globales

Ces défauts proviennent naturellement du fait que l’on ait “oublié” pour l’instant, dans notre modèle, le terme $\nu\Delta\mathbf{u}$ dans l’expression de l’équation de *Navier-Stokes*. Ce choix avait été rendu indispensable pour permettre une simulation rapide et interactive de ce comportement fluide. Seul ce dernier nous importait alors, nous avons dit que nous nous pencherions séparément sur les problèmes de viscosité en ajoutant à notre modèle un second jeu de lois de comportement qui les prendrait en compte de façon indépendante. Cette seconde couche apparaît comme indispensable si l’on souhaite avoir un comportement satisfaisant pour notre argile virtuelle. Elle travaillera en collaboration avec la première, et leur combinaison donnera un matériau dont les caractéristiques seront plus proches de notre objectif.

Lorsque les outils entrent en contact avec le matériau et se déplacent, ils entraînent par contact une partie du matériau avec eux. Le fluide à proximité des outils est mis en mouvement, que ce soit par frottement lorsque l’outil se déplace latéralement, par pression, voire par adhérence lorsque l’outil tente de s’éloigner de la surface de l’objet (même si *a priori* le côté “collant” du matériau se trouve être plutôt gênant pour le sculpteur). Ce mouvement sera efficacement transmis à l’ensemble du fluide, grâce aux effets de viscosité.

Oublions un moment les effets locaux de déplacement de la matière, et efforçons-nous de réfléchir au mouvement à plus grande échelle du fluide. Ce que doit simuler la seconde couche de notre modèle, ce sont les déplacements à grande échelle du fluide, en oubliant les éventuels déplacements locaux, par exemple dans le voisinage immédiat des outils, dus à son caractère fluide. Ce dont nous avons besoin, c’est d’un champ de déformation sur l’ensemble du matériau, déduit à partir du mouvement des différents outils.

3.4.3 Utilisation du champ de déformation

Quand nous aurons établi ce champ de déformation, nous pourrons déplacer la matière contenue dans chaque cellule du maillage vers sa destination. Cette dernière est simplement désignée par la valeur prise par le champ de déformation au centre de la cellule. Comme nous souhaitons ne pas perdre de matière, nous allons procéder de la même façon que précédemment : nous prélèverons donc la matière présente dans une cellule, afin de la reverser dans la cellule désignée par le champ de déformation.

Évidemment, le champ de déformation ne désignera généralement pas le centre d’une cellule, mais une destination recouvrant plusieurs cellules. La matière prélevée dans la cellule d’origine sera répartie dans les huit cellules entourant la destination, proportionnellement à leur recouvrement, comme suggéré sur la figure 3.12. Il pourra bien évidemment arriver que le champ de déformation remplisse certaines cellules de plus de matière qu’elles n’en peuvent contenir. Cela arrive en particulier lorsque deux objets entrent en contact, ou bien lorsqu’un objet subit des auto-collisions. La couche fluide présentée au chapitre précédent permettra de traiter ces excès. La couche modélisant les déformations à grande échelle sera donc la première à être simulée.

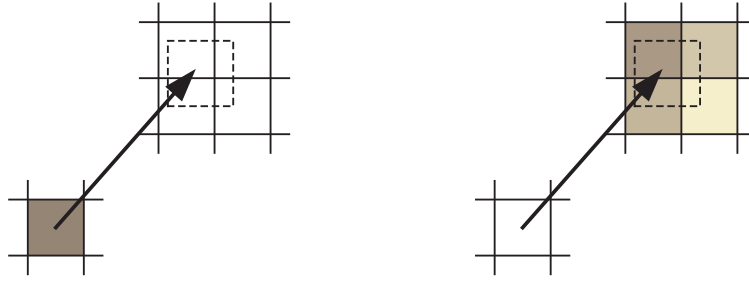


FIG. 3.12 – La matière déplacée par la seconde couche de notre modèle, simulant les déformations élastiques, est répartie dans les cases désignées par le champ de déplacement.

3.4.4 Remarque sur la terminologie

Ce que cherche à construire cette seconde couche, c'est un champ de déformation du matériau, qui constitue en pratique un champ de déplacement régissant le transport de la matière. Toutefois, dans la suite, il sera parfois plus naturel de parler de *champ de vitesse* au sein du fluide. Les équations de mécanique des fluides, notamment, préfèrent travailler avec des vitesses.

La distinction n'en est toutefois pas une. En effet, nous cherchons à évaluer les déplacements entre deux instants successifs t et $t + 1$. Le déplacement des outils entre ces deux instants peut très simplement être converti en une vitesse, par simple division par le temps correspondant à un pas de simulation. Inversement, un champ de vitesse pourra être converti en un champ de déplacement par simple multiplication par ce même temps.

Par la suite, nous utiliserons donc indifféremment les notions de *champ de déformation* ou de *champ de vitesse*, en fonction de ce qui nous semble être le plus naturel. Ces deux notions permettent toutes deux de déduire simplement les déplacements de matière dans la grille qu'il faudra effectuer.

3.5 Une première approche

3.5.1 Un modèle dérivé des solides élastiques et plastiques

Introduction

La toute première idée qui vient, lorsque l'on souhaite calculer un champ de déformation résultant de l'action d'un certain nombre d'outils, c'est de s'inspirer des méthodes développées dans le cas des matériaux plastiques ou élastiques. Nous avons vu qu'un matériau plastique pouvait être considéré, en première approximation, comme un matériau élastique dont les déformations deviennent permanentes à chaque instant. Nous pourrions donc envisager un modèle de déformations élastiques simple, choisi parmi ceux présentés dans le second chapitre,

pour calculer les déformations subies par le matériau entre deux instants successifs.

Discussion de l'adéquation des modèles

Ces modèles présentaient quelques inconvénients et limitations. D'abord, ils sont généralement définis dans la limite des petites déformations. Avec des vitesses de déplacement raisonnables des outils, cette condition sera naturellement vérifiée. En effet, comme nous l'avons dit, nous travaillons avec un matériau présentant un comportement plastique, qui va conserver ses déformations au cours du temps. Les déformations sur un pas de temps peuvent être considérées comme modérées, et la nouvelle position tiendra lieu de position de référence pour l'étape suivante. On reste donc toujours dans le cadre des petites déformations.

La gestion des collisions et des auto-collisions n'est, par ailleurs, pas chose facile. On souhaite en général éviter que deux objets, ou deux parties d'un même objet, s'interpénètrent, ce qui occasionne de nombreux calculs supplémentaires pour détecter ces collisions. Cela ne se trouve pas être, dans notre cas, un problème bien important. Si deux éléments entrent en collision, ils vont simplement fusionner, et à partir de l'étape suivante, ils ne formeront plus qu'un.

Une autre difficulté pourrait résider dans le fait que la plupart de ces modèles ne garantissent pas une conservation du volume, et plus précisément un champ de déformation à divergence nulle. En pratique, cette difficulté sera largement levée par l'action de la couche fluide de notre modèle : si le champ de déformation devait provoquer une accumulation locale de fluide, celle-ci serait naturellement redistribuée vers les cellules vides les plus proches. Il ne peut donc pas apparaître de diminution du volume du matériau du fait des déformations. On verra par exemple apparaître des bourrelets de matière au niveau des plis formés par les déformations. À l'intérieur du coude d'un personnage en argile dont on tenterait de plier le bras, par exemple.

Le cas de champs de déformation à divergence positive est un peu plus gênant. Rien n'empêche en effet les cellules de contenir moins de matière que le maximum permis. Nous verrons à la fin de ce chapitre comment introduire une tension de surface qui tentera d'apporter une réponse à ce problème en s'efforçant de "recompacter" le matériau. Cela ne fonctionne cependant pas aussi bien. Remarquons toutefois qu'on travaille bien plus souvent en poussant le matériau qu'en l'étirant, étape d'autant plus rare que le matériau que nous construisons devrait être aussi peu "collant" que possible. Ce cas devrait donc ne pas se produire trop régulièrement.

Un réseau masses-ressorts

Il semblerait donc, à première vue, que ces modèles devraient répondre assez bien à nos besoins. Il y a toutefois une difficulté majeure à considérer. L'action de la couche fluide va produire divers déplacements de matière dans le matériau. En outre, nous avons déjà signalé que la connexité de l'objet considéré pouvait changer. Enfin, chaque nouvelle position devient *de facto* la position de référence pour la suite des déformations. Il faudra donc remettre à jour à chaque instant la structure (généralement un réseau) qui va permettre de calculer les

différentes déformations.

Beaucoup de modèles de déformation, pour rester parfaitement interactifs, procèdent à une grande quantité de précalculs lorsqu'ils construisent cette structure, calculs conduits avant le commencement de la simulation. Dans notre cas, nous allons devoir construire la structure durant la simulation. Nous ne pouvons donc espérer construire qu'un réseau assez simple pour décrire la matière. Le choix le plus naturel semble être la construction d'un réseau masse-ressorts.

Ces réseaux ne sont plus guère utilisés dans le cas de la simulation de solides élastiques ou plastiques, pour lesquels on préfère utiliser par exemple des modèles d'éléments finis, plus riches et plus complexes, utilisant des tenseurs de déformation [Deb00]. On les retrouve cependant encore régulièrement pour simuler par exemple le comportement de tissus ou de chevelures. Ces réseaux sont relativement simples à construire, et peuvent être animés rapidement.

3.5.2 Construction du réseau de déformation

Forme du réseau

Encore une fois, pas question pour nous de construire un réseau complexe, représentant précisément le volume sculpté. Ce ne serait pas compatible avec une implémentation interactive de notre modèle. Nous avons donc choisi de construire un réseau cubique, dont les axes et la régularité sont étroitement associés à la représentation en grille du matériau.

Les objets avec lesquels nous travaillons peuvent occuper plusieurs milliers de cellules, voire davantage, afin que l'on puisse aisément obtenir des détails au niveau de leur surface. Il n'est pas utile d'associer un nœud à chacune de ces cellules, d'autant que cela serait particulièrement coûteux en termes de temps de calcul, tant pour le créer que lorsque nous en viendrons à l'animer.

Il n'est cependant pas acceptable de trop éloigner ces nœuds. En effet, le mouvement du fluide au sein de l'objet sera interpolé à partir du mouvement des nœuds les plus proches, à la manière des volumes FFD. Un maillage trop grand nous donnerait un résultat complètement erroné de déformation du matériau, le réseau représentant trop imparfaitement le volume sculpté.

Le maximum acceptable pour une taille de maille nous est apparu comme étant d'un nœud du réseau masse-ressort pour quatre cellules, selon chacune des trois directions de l'espace. On a donc un nœud pour un cube de quatre cellules de côté, soit soixante-quatre en tout. Le nœud est placé au centre de ce volume de soixante-quatre cellules. Une sphère d'une vingtaine de cellules de côté donne un réseau comportant une soixantaine de nœuds, ce qui permet d'espérer pouvoir l'animer correctement.

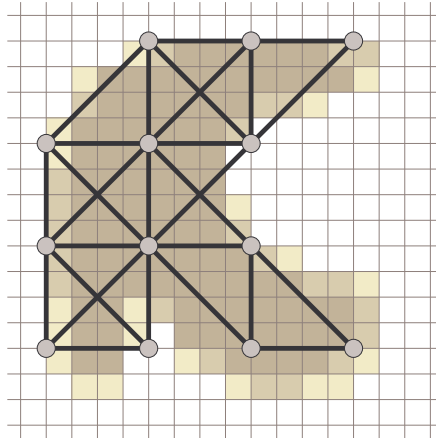


FIG. 3.13 – Construction du réseau masses-ressorts pour le calcul des déformations.

Choix des paramètres

Reste à définir les paramètres du réseau (masse des nœuds, raideur et longueur à vide des différents ressorts) et sa topologie (c'est-à-dire, quels sont les nœuds effectivement liés par un ressort?). La masse des nœuds est le paramètre le plus facile à fixer. Pour chacun d'entre eux, nous mesurons la quantité de matière contenue dans les soixante-quatre cellules qui les entourent. Le nœud n'est créé que si cette quantité dépasse un certain seuil (que l'on peut choisir égal à 0), ce qui permet de ne créer de nœuds qu'à l'intérieur même du matériau et dans son voisinage immédiat. La quantité totale de matière fixe naturellement la masse du nœud.

Reste ensuite à placer les ressorts. On distingue trois types de ressorts dans notre maillage cubique : ceux qui suivent les arêtes du maillage, ceux qui s'orientent suivant les diagonales des facettes, et enfin ceux qui suivent les diagonales des cubes élémentaires constituant le maillage. Les ressorts placés sur ces diagonales sont particulièrement utiles dans la mesure où ceux-ci aident à conserver constant le volume de chaque élément du maillage, de sorte que la divergence du champ de déformation obtenu est proche de 0. Un maillage constitué de ces seuls ressorts est relativement souple, c'est l'adjonction des deux autres types de ressorts qui rigidifiera quelque peu le maillage.

La longueur à vide des ressorts est choisie de façon à ce qu'en l'absence de déplacement d'une masse, le système soit à l'équilibre, donc très simplement la longueur qu'a le ressort initialement, avant toute tentative de déformation du réseau. Quant à leur raideur, on s'attend à ce que les zones avec davantage de matière exercent des contraintes plus fortes que les zones plus pauvres.

Nous souhaiterions naturellement qu'un ressort soit d'autant plus raide que la matière à cet endroit est solide et compacte. Au contraire, ils seront plus souples dans les endroits plus fins, les bords du matériau, les extrémités, bref dans les endroits où la densité est la plus faible. Pour chaque ressort, nous choisirons donc une raideur proportionnelle à la densité de la matière, et plus précisément à la quantité de matière contenue dans un ensemble cubique de huit cellules entourant le centre du ressort. Si cette quantité est inférieure à un certain

seuil (nous avons par exemple choisi un seuil correspondant à 1/3 de la quantité maximale de matière pouvant être contenue dans ces huit cellules), le ressort n'est simplement pas créé.

Limitations

Le réseau ainsi construit donne une approximation plus ou moins bonne de la géométrie de l'objet. Les zones relativement fines (en particulier celles dont l'épaisseur est inférieure à une demi-douzaine de cellules), qu'elles soient planes ou filiformes, n'ont généralement que très peu de tenue, voire pas du tout. Ceci à cause du fait qu'il n'y a généralement dans ce cas qu'une seule couche de masses dans l'objet : en l'absence de ressorts de torsion, ces structures n'ont aucune rigidité. Inversement, dans les zones vides séparant deux objets près proches, il peut arriver que des ressorts apparaissent, et qu'un objet exerce sur le second des efforts qui n'ont pas lieu d'être, et que la fusion des deux objets en soit notablement compliquée.

Ces inconvénients sont dus au caractère simpliste de notre réseau masses-ressorts, et pourraient être résolus avec des réseaux plus détaillés, et surtout avec l'emploi de techniques multi-résolution. Ce n'est malheureusement pas possible à l'heure actuelle si nous voulons garder une simulation interactive du matériau.

3.5.3 Animation du réseau

Action des outils

Les outils vont très simplement accrocher certaines des masses du réseau, et les entraîner dans leur déplacement. Le choix des masses se fera sur un seul critère de localisation : les masses incluses dans le volume d'un outil seront animées du même déplacement que l'outil. Plutôt que de prendre les masses à l'intérieur du volume de l'outil avant qu'il ne se déplace, on préfère prendre celles à l'intérieur du volume de l'outil à mi-chemin de son déplacement.

Ce choix présente plusieurs avantages. D'une part, la couche fluide a tendance à évacuer la matière à l'extérieur des outils, de sorte que, bien souvent, il ne restera plus suffisamment de matière dans les outils pour qu'un nœud apparaisse au sein de leur volume. Ensuite, cela évite certains effets d'adhérence du matériau aux outils lorsqu'on essaie de les écarter du matériau. A mi-chemin de son déplacement, l'outil s'est suffisamment éloigné du matériau pour que plus aucun nœud ne soit présent dans son volume.

Relaxation du réseau

Une fois les nœuds sélectionnés déplacés selon le mouvement des outils, ils sont fixés, et on laisse le reste du réseau évoluer vers un état d'équilibre, en appliquant les lois usuelles de la mécanique. Nous cherchons à déterminer le champ de déformation qu'engendrera à terme le mouvement des masses du réseau manipulées par les outils, et donc à s'approcher autant que possible de la convergence. Le schéma d'intégration dit "*Euler implicite*" est en théorie le plus adapté pour un calcul rapide de cette convergence, puisqu'il permet de travailler avec des pas de temps bien plus grands.

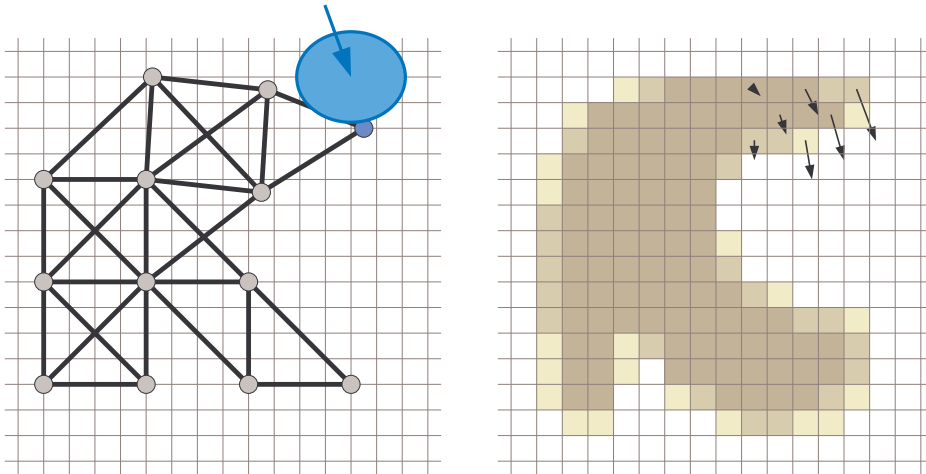


FIG. 3.14 – Le déplacement d'un outil entraîne certains des nœuds du réseau, lequel se déforme en conséquence (le déplacement n'est ici qu'une illustration du principe). A droite, le champ de déformation dans le matériau est déduit du déplacement des nœuds du réseau.

Certaines des déformations obtenues ne semblent cependant pas très naturelles. L'une des raisons tient au fait que les cellules préfèrent rester cubiques, du fait de la présence des ressorts diagonaux (leur absence, ainsi que nous l'avons signalé tantôt, occasionne d'autres problèmes dont une très mauvaise conservation du volume qui rend plus difficile la tâche de la première couche).

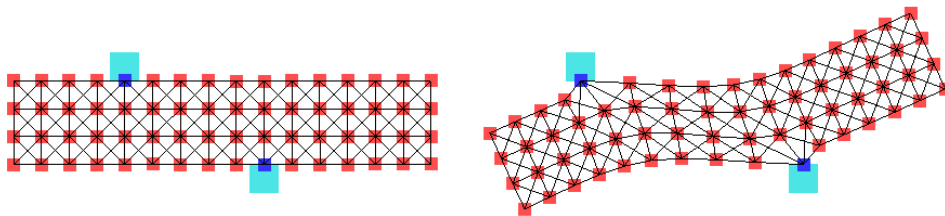


FIG. 3.15 – Le mouvement des outils déforme le réseau en entraînant certains de ses nœuds. Le comportement obtenu est naturel pour un réseau masses-ressorts, mais les effets ne correspondent pas toujours à ce que l'on attend d'une argile virtuelle, comme dans le cas présent où le simple déplacement latéral d'un outil déforme le matériau de façon inattendue.

La façon dont les outils se fixent à la grille peut par ailleurs conduire à des résultats très différents. Si un outil se fixe sur un unique nœud du réseau, il n'y aura pas de contraintes sur la rotation de l'objet autour de ce point, contraintes qui apparaissent lorsque deux nœuds ou plus voient leur déplacement imposé. Bien évidemment, l'utilisateur n'a aucun contrôle sur ce dernier point, ce qui conduit à des comportements du matériau qui peuvent parfois être difficiles à prévoir.

Notons qu'animer le réseau jusqu'à obtenir une convergence complète n'est pas toujours

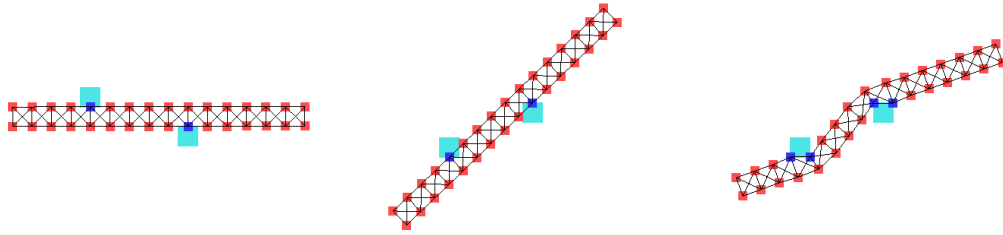


FIG. 3.16 – Ainsi qu’on peut le constater, les déformations obtenues dépendent notablement du nombre de noeuds “accrochés” par les outils. À gauche, la position initiale. Au centre, un seul noeud a été déplacé par chacun des outils, tandis qu’à droite, les outils (de même taille, mais légèrement décalés) ont déplacé deux noeuds chacun.

la meilleure solution. L’état d’équilibre du réseau ne dépend pas des masses des noeuds du réseau, et assez peu des raideurs des différents ressorts. Pour mieux prendre en compte les effets dus aux variations de densité du matériau, il pourrait être intéressant de ne faire qu’un nombre modéré de pas en direction de la convergence (ainsi, les zones souples du matériau, où la raideur des ressorts est plus faible, seront moins déplacées que celles où la matière est bien compacte, et plus rigide). La difficulté réside dans le fait que le nombre de pas à effectuer est très difficile à choisir, et ce choix dépend par ailleurs largement de l’échelle considérée. Nous n’avons guère de solution à proposer à ce problème.

Une autre difficulté que l’on peut rencontrer dans la simulation est l’apparition d’oscillations. En l’absence d’amortissement, le réseau est prompt à vibrer, selon différents modes d’oscillation difficilement contrôlables. Pour peu que la simulation s’arrête au mauvais moment, le résultat peut être une réponse plus importante aux contraintes que les contraintes elles-mêmes ! Paradoxalement, les qualités de stabilité de la méthode *Euler implicite* la rendent particulièrement sensible à ces effets d’oscillation. Des méthodes plus simples (la méthode classique d’Euler, ou celle de Runge-Kutta) donnent de moins bons résultats, mais qui grâce à leurs défauts se trouvent générer moins de situations de ce genre.

3.5.4 Limitations de cette approche

En définitive, nous avons trouvé très difficile, pour ne pas dire impossible, d’obtenir un champ de déformation satisfaisant avec cette méthode, qui pourtant nous avait semblé intéressante à l’origine. La construction d’un nouveau réseau à chaque instant est coûteuse, et le réseau obtenu souvent imparfait, voire inadapté pour représenter correctement le solide.

La relaxation pose quant à elle nombre de problèmes théoriques, et consomme beaucoup de temps de calcul pour un résultat pas toujours convaincant. Si les oscillations du réseau sont généralement recherchées, par exemple dans le cas de la simulation d’un matériau élastique, elles sont un réel problème pour nous. Ces difficultés pourraient être surmontées en cherchant directement une solution statique au problème : calculer directement la position d’équilibre du réseau masses-ressorts (ou bien d’éléments finis) compatible avec le mouvement des outils. Les quelques essais que nous avons faits dans cette direction laissent entrevoir des difficultés en termes de temps de calcul (la construction du réseau est déjà coûteuse, la détermination

de la position d'équilibre requiert quant à elle des calculs matriciels importants). Par ailleurs, nous avons déjà signalé qu'il n'était pas nécessairement préférable d'aller jusqu'à l'équilibre complet du réseau.

Peut-être peut-on espérer trouver une solution en employant plutôt des particules interagissantes, et utiliser certaines techniques de FFD pour en déduire le champ de déformation. Il se posera cependant certains problèmes, comme l'ensemencement du volume avec ces particules (qui pourrait vraisemblablement être réalisé à partir de la grille de densité du matériau), leur mise à jour, et surtout leur façon de réagir avec les outils. Nous n'avons pas poursuivi dans cette voie, mais avons préféré envisager les choses sous un angle différent.

3.6 Retour sur les modèles de fluides

3.6.1 Le second terme de l'équation de Stokes

Nous allons à présent nous pencher plus avant sur l'action du second terme de l'équation de Navier-Stokes, dont nous avons dit qu'il régissait la diffusion des vitesses dans le matériau. En ne gardant que ce terme dans l'équation de *Navier-Stokes*, on obtient :

$$\frac{\partial \mathbf{u}}{\partial t} = \nu \Delta \mathbf{u}. \quad (3.15)$$

Si l'on considère une ligne de courant rapide entourée de couches plus lentes, le laplacien du champ de vitesse n'est pas nul. Le terme qui nous intéresse à présent aura pour effet de freiner la ligne de courant rapide, tout en accélérant les couches adjacentes, plus lentes. Cet effet trouve son origine dans l'échange de particules entre les couches : certaines particules rapides quittent la couche rapide pour les couche lentes, augmentant la vitesse moyenne de ces dernières, et inversement.

C'est donc à tous niveaux un problème de diffusion des vitesses, ainsi que le laissait suggérer la forme prise par l'équation 3.15. Plus le matériau est visqueux (se traduisant par un ν important), plus cette diffusion sera rapide. Les matériaux avec lesquels nous travaillons ont une viscosité cinématique très importante, de sorte que cet effet est pratiquement instantané. Un travail aux dimensions indique en effet que le temps nécessaire pour atteindre le régime permanent est de l'ordre de d^3 :

$$[t] = \frac{[d]^2}{\nu}. \quad (3.16)$$

Les temps nécessaires à l'établissement du régime permanent sont donc bien inférieurs aux pas de la simulation (quelques dizaines de pas par seconde). Nous pouvons donc admettre que

³Il est d'usage, dans le domaine de la mécanique des fluides, d'estimer par exemple les temps caractéristiques d'un écoulement en estimant l'ordre de grandeur de chacune des quantités (voir par exemple [LL86]). Ici, dans l'équation 3.15, nous assimilerons par exemple la vitesse \mathbf{u} à un terme de la forme $\frac{[d]}{[t]}$ où $[d]$ est une taille caractéristique de l'écoulement, et $[t]$ un temps caractéristique. $\Delta \mathbf{u}$ devient quant à lui $\frac{[d]}{[t][d]^2}$, et ainsi de suite. Etant donné qu'on peut envisager plusieurs "distances caractéristiques" (taille de l'objet, taille des détails, taille d'une cellule), nous utiliserons dans l'équation 3.16 le cas le plus défavorable, autrement dit la dimension la plus grande, celle de l'objet tout entier. Même dans ce cas défavorable, le temps caractéristique ainsi obtenu reste bien inférieur de plusieurs ordres de grandeur au pas de simulation.

durant l'essentiel du temps que dure un pas de simulation, le champ de vitesse correspond à celui que l'on obtient une fois que le régime permanent a été atteint. Dans ce régime permanent, les vitesses n'évoluent plus, et le champ de vitesse doit donc vérifier :

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{0} = \nu \Delta \mathbf{u}. \quad (3.17)$$

Le champ de vitesse que nous recherchons devrait donc vérifier $\Delta \mathbf{u} = \mathbf{0}$ partout au sein du matériau, et naturellement satisfaire les conditions aux limites. Celles-ci imposent que le mouvement au voisinage des outils soit semblable au mouvement de ces derniers. En l'absence de toute condition aux limites particulières (par exemple dans le cas d'un objet isolé, ou bien encore manipulé par un seul outil, et en l'absence de toute force extérieure comme son propre poids), on retrouve la solution $\mathbf{u} = \mathbf{cste}$, signifiant que les vitesses sont rapidement toutes identiques à l'intérieur du matériau, qui se comporte alors comme un objet rigide. C'est ce qui nous avait poussé, dans le paragraphe précédent, à annuler les vitesses à l'intérieur du matériau à chaque étape, nous affranchissant ainsi de tout effet inertiel.

3.6.2 Approches pour la résolution

Pour résoudre cette équation $\Delta \mathbf{u} = \mathbf{0}$ et obtenir le champ de vitesse dans du matériau, il existe plusieurs méthodes. La plus simple est de simuler effectivement la diffusion suggérée par l'équation 3.15, et d'appliquer cette formule jusqu'à la convergence. Les vitesses des cellules sous l'influence d'un outil voient leur vitesse imposée comme égale au mouvement de l'outil correspondant, tandis que les autres évolueront selon la loi $\frac{\partial \mathbf{u}}{\partial t} = \nu \Delta \mathbf{u}$.

Évidemment, les choses ne se passent jamais aussi aisément, la convergence n'étant pas aisée à établir. Le choix du bon pas de temps, en particulier, est crucial si l'on veut éviter toute instabilité du modèle. À cause des valeurs importantes que peut prendre la viscosité cinématique ν , les dérivées obtenues sont en effet rapidement très grandes. En outre, pour qu'une vitesse à une extrémité du matériau soit diffusée jusqu'à l'autre côté de celui-ci, il faudra appliquer à l'ensemble du matériau au moins autant d'étapes de simulation qu'il n'y a de cellules entre ces deux extrémités, et vraisemblablement beaucoup plus. On retrouve en fait ici la condition de *Courant-Friedrichs-Lewy* évoquée précédemment, qui limite largement la simulation des fluides très visqueux. Elle plaçait en effet une limite sur la vitesse de propagation de l'information (ici la vitesse) au sein du matériau.

Cette approche est donc largement incompatible, pour l'instant, avec des applications interactives. Il existe d'autres méthodes pour résoudre les problèmes de type $\Delta \mathbf{u} = \mathbf{0}$, assortis de conditions aux limites. Par exemple, des méthodes utilisant des transformées de Fourier. Malheureusement, elles font généralement d'importantes hypothèses quant à la forme du domaine, ou sur les conditions aux limites de ce dernier. Jos Stam, par exemple, résout de façon directe dans [Sta99] une équation en tous points similaire, mais fait l'hypothèse pour cela d'un espace de travail cubique, aux conditions aux limites nulles ou périodiques. Bien évidemment, ces hypothèses ne s'appliquent pas au cas qui nous occupe.

3.6.3 Quelques exemples

Il va donc nous falloir essayer de trouver un moyen pour trouver un champ de vitesse similaire à celui solution de l'équation $\Delta \mathbf{u} = \mathbf{0}$ et satisfaisant aux conditions aux limites. Comme nous l'avons dit, il n'existe pas de solution générale à ce problème, mais nous pouvons essayer de trouver des méthodes fournissant de bonnes solutions dans certains cas particuliers.

Pour ce faire, considérons quelques cas simples de mécanique des fluides pouvant être résolus de façon analytique (voir par exemple [LL86]), comme ceux présentés sur la figure 3.17. Dans le premier cas, un volume de fluide est enfermé entre deux plaques verticales infinies (correspondant aux plans $x = 0$ et $x = d$), se déplaçant à des vitesses différentes \mathbf{u}_1 et \mathbf{u}_2 . Le champ de vitesse solution à ce problème sera alors :

$$\mathbf{u} = \mathbf{u}_1 + (\mathbf{u}_2 - \mathbf{u}_1) \frac{x}{d}. \quad (3.18)$$

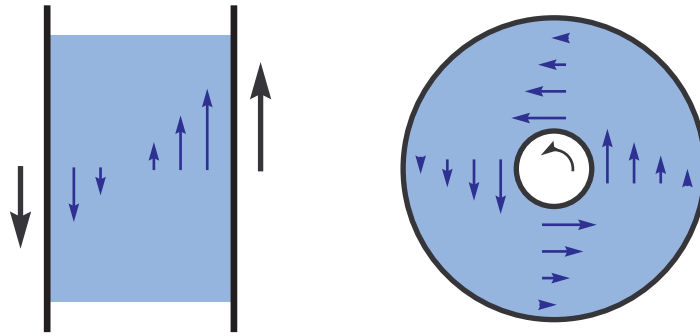


FIG. 3.17 – Exemples de la cinématique d'un fluide, respectivement entre deux plaques se déplaçant latéralement, et entre deux cylindres dont l'un est en rotation. Le champ de vitesse varie linéairement au sein du fluide.

On pourra vérifier que ce champ satisfait bien aux conditions aux limites, ainsi qu'aux équations $\Delta \mathbf{u} = \mathbf{0}$ et $div(\mathbf{u}) = 0$. Dans le second cas, nous avons un problème similaire, avec un volume de fluide compris entre deux cylindres infinis de rayons R_1 et R_2 , tournant autour d'un même axe z à des vitesses de rotation respectives ω_1 et ω_2 . La solution s'écrit cette fois :

$$\mathbf{u} = (\omega_1 + (\omega_2 - \omega_1) \frac{\sqrt{x^2 + y^2} - R_1}{R_2 - R_1}) \sqrt{x^2 + y^2} \mathbf{u}_\theta \quad (3.19)$$

où \mathbf{u}_θ est un vecteur unitaire orthoradial.

Dans chacun de ces deux cas, si l'on examine le champ de vitesse, il donne l'impression d'"interpoler" le champ de vitesse imposé par les conditions aux limites (dans le cas qui nous occupe, les plaques ou les cylindres). Cela peut se comprendre par le fait que des expressions affines pour la vitesse \mathbf{u} sont bien compatibles avec l'équation $\Delta \mathbf{u} = \mathbf{0}$. Ces variations affines dépendent de deux paramètres, qui sont simplement fixées par les conditions aux limites.

3.6.4 Idée générale

Pour estimer le champ de vitesse que nous cherchons, nous allons donc nous appuyer sur cette constatation. Pour connaître le mouvement du fluide en un point donné du matériau, nous allons estimer l'influence qu'exerce chacune des conditions aux limites en ce point. Plus on sera proche d'une de ces conditions, plus son influence sera grande, nous chercherons donc à déterminer la distance qui sépare ce point des différentes zones où s'exercent des contraintes. Nous combinerons ensuite ces différentes contraintes, en se basant sur leur influence respective au point considéré. En pratique, cela reviendra à utiliser une combinaison linéaire des contraintes, les coefficients de pondération des différentes contributions étant directement déduits des différentes distances que nous aurons calculées, selon une formule que nous préciserons ultérieurement.

Pour pouvoir utiliser cette approche, il faut cependant que l'on ait un nombre raisonnable de conditions aux limites distinctes, puisque pour chacune d'entre elles, il faudra calculer la distance qui la sépare de chacune des cellules à l'intérieur du matériau. Idéalement, chaque cellule imposant une condition aux limites devrait être prise en compte séparément. Cela n'est généralement pas possible. Nous ferons donc un certain nombre d'hypothèses. D'abord, nous supposons que les conditions aux limites correspondent à un nombre limité de groupes de cellules de petite taille par rapport à la taille de l'objet. Ensuite, nous considérerons que le mouvement imposé aux cellules dans chacun de ces groupes peut être décrit de façon simple : dans un premier temps, nous supposons que chacun de ces groupes correspond à un simple mouvement de translation, puis nous verrons que l'on peut également considérer un mouvement de rotation rigide.

Ces hypothèses sont acceptables dans le cas de notre étude, puisque les conditions aux limites sont celles imposées par le contact avec des outils, qui sont en nombre limité, et dont le mouvement est rigide. La seule réelle contrainte concerne la taille des outils (ou plutôt de la surface de contact avec le matériau). En pratique, nous verrons que l'on peut quand même prendre des outils de grande taille, à condition d'admettre que les comportements seront moins bien simulés, ou bien de disposer de suffisamment de puissance de calcul pour diviser les outils de grande taille en plusieurs outils de taille moindre.

Cette approche se place dans la même perspective que les "*As-rigid-as-possible shapes*" de T. Igarashi, qui évaluent des champs de déformation (en deux dimensions) les plus simples et rigides possible, mais qui respectent néanmoins un certain nombre de conditions aux limites [IMH05].

3.7 Seconde couche : un modèle pour les déformations globales

3.7.1 Calcul de la distance à un outil

L'opération au cœur de notre algorithme consistera donc, pour chaque cellule, à évaluer la distance qui la sépare des différents outils, afin de pondérer l'influence qu'aura chacun de ceux-ci sur le champ de vitesse dans son voisinage, et donc sur son mouvement. Ce n'est hélas pas la simple distance euclidienne, aisée à calculer, qui nous intéresse. En effet, la diffusion des

vitesses se fait de proche en proche au sein du matériau seulement. C'est donc une pseudo-distance qui suit les courbes du matériau qu'il nous faut trouver, ainsi que le montre la figure 3.18.

En fait, ce problème, nous l'avons déjà abordé dans le chapitre précédent, lorsque nous parlions de la possibilité d'un déversement direct des excédents de matières dans les cellules vides les plus proches. Nous avons vu que le calcul de ce genre de pseudo-distance à l'intérieur de l'outil pouvait être réduit à un simple algorithme de propagation. Ici, nous partirons des cellules à l'intérieur de l'outil, puis calculerons la distance à chacune des cellules du matériau en sautant de proche en proche. On s'arrêtera lorsque l'on atteint des cellules vides. Ainsi, l'influence d'un outil sera limitée aux seules composantes connexe du matériau avec lesquelles celui-ci est en contact.



FIG. 3.18 – La pseudo-distance d'un point du matériau à un outil est calculée en suivant la forme du matériau.

Dans les zones contenant peu de matière, la quantité de mouvement est moins bien transmise de cellule en cellule. La pseudo-distance à un outil augmentera donc plus rapidement au niveau des cellules dont la densité est inférieure au seuil limite. On s'attend même à ce que ces zones deviennent rapidement des points de rupture pour le matériau. Nous en tiendrons donc compte dans le calcul de notre pseudo-distance.

Si une cellule (i, j, k) se trouve à une distance $d_{i,j,k}$ d'un outil, alors on peut écrire la distance de sa voisine et suivante dans le processus de propagation par :

$$d_{i',j',k'} = d_{i,j,k} + \frac{1}{n_{i,j,k}}. \quad (3.20)$$

On constate ainsi que la distance augmentera plus vite dans les zones où la densité n est faible. Hormis cette modification pour le calcul de la pseudo-distance, l'algorithme reste semblable à celui présenté dans le chapitre précédent : on utilise une pile pour stocker la progression et un drapeau dans chaque cellule pour indiquer celles dont la pseudo-distance à l'outil a déjà été évaluée. Ces mêmes drapeaux pourront, pour chaque cellule, déterminer quels outils influenceront le mouvement du fluide contenu dans la cellule considérée.

L'algorithme 1 présente l'ensemble des étapes à effectuer pour calculer la pseudo-distance à un outil pour chaque cellule concernée. Le seuil de densité n_{min} permet d'arrêter la propagation lorsque l'on arrive aux bords de l'outil. On peut choisir simplement $n_{min} = 0$, ou placer un seuil un peu plus haut, ce qui favorise alors les phénomènes de rupture et de déchirement de l'objet dans les endroits les plus vulnérables. Cela peut parfois créer quelques miettes lors

Algorithme 1 Calcul de la pseudo-distance à un outil

```

fixer un seuil de densité  $n_{min}$ 
baissier l'ensemble des drapeaux  $z_{i,j,k}$  indiquant que la cellule  $(i, j, k)$  est sous l'influence de
l'outil

pour chaque cellule  $(i, j, k)$  contenant de la matière incluse dans le volume de l'outil :
  initialiser la distance de la cellule à l'outil  $d_{i,j,k} \leftarrow 0$ 
  lever le drapeau  $z_{i,j,k}$ 
  placer la cellule  $(i, j, k)$  dans la file de traitement
fin pour

pour chaque cellule  $(i, j, k)$  dans la file de traitement :
  calculer la distance  $d \leftarrow d_{i,j,k} + 1/(n_{i,j,k})$ 
  pour chaque cellule voisine  $(i', j', k')$  vérifiant  $n_{i',j',k'} > n_{min}$  :
    si le drapeau  $z_{i',j',k'}$  est baissé, ou s'il est levé et  $d_{i',j',k'} > d$  alors
      fixer  $d_{i',j',k'} \leftarrow d$ 
      lever le drapeau  $z_{i',j',k'}$ 
      placer la cellule  $(i', j', k')$  dans la file de traitement
    fin si
  fin pour
fin pour

```

de la sculpture, sur lesquelles nous reviendrons ultérieurement. Un seuil de l'ordre de 0.1 à 0.2 semble être un bon compromis.

La démarche précédente est à répéter pour chaque outil. Chaque cellule disposera alors, pour chaque outil l , d'un drapeau z_l indiquant que le déplacement de l'outil en question va avoir une influence sur le mouvement de la matière dans la cellule (c'est-à-dire si l'outil est en contact avec l'objet auquel correspond la cellule considérée), et le cas échéant de la pseudo-distance à cet outil, d_l . A partir de ces informations, nous allons pouvoir estimer le mouvement du fluide contenu dans la cellule.

3.7.2 Influence d'un seul outil

Lorsqu'une cellule n'est sous l'influence d'aucun outil, il ne se pose pas de problème particulier : la matière ne bouge pas. Nous nous sommes en effet affranchis de tout effet inertiel. Avec un seul outil, le problème est plus délicat. On peut soit attribuer directement la vitesse de l'outil au fluide présent dans la cellule, soit seulement une fraction de cette vitesse, en se basant sur la pseudo-distance que nous venons de calculer. La vitesse du fluide pourra s'écrire :

$$\mathbf{u}_{i,j,k} = k\mathbf{u}_l \quad (3.21)$$

où k vaudra simplement 1 ou correspondra à une expression de la forme $e^{-d_l(i,j,k)}$.

Dans la pratique, ce problème n'en est pas vraiment un. Lorsque l'on veut déformer un objet, on ne travaille jamais avec un seul outil. Dans le plus simple des cas, le support lui-même joue le rôle de second outil. Ce n'est qu'un cas particulier d'outil immobile, qui impose

le même genre de contraintes que les outils proprement dits. Sans support, par exemple dans l'espace, on peut penser que pousser l'objet avec un seul outil aura pour seule conséquence notable de le déplacer. Nous avons donc choisi la première solution, en gardant à l'esprit que ce choix n'aura aucune conséquence notable dans des conditions réelles de sculpture.

3.7.3 Champ de déplacement dans le cas de deux outils

Le problème devient plus intéressant lorsque des cellules sont sous l'influence de plusieurs outils, dont nous allons essayer d'interpoler le mouvement, selon notre idée initiale. Nous allons d'abord examiner le cas d'une cellule sous l'influence de deux outils seulement, l et l' , que nous numérotions 1 et 2 dans la suite pour plus de simplicité. On s'intéressera au cas d'une cellule (i, j, k) dont les pseudo-distances aux deux outils seront respectivement notées d_1 et d_2 . Les deux outils peuvent être indifféremment des outils en mouvement, ou bien des points d'appui immobiles pour permettre les déformations, comme par exemple un support permettant de tenir l'objet pendant qu'un outil travaille dessus.

Nous allons écrire le mouvement du fluide dans la cellule considéré comme une simple combinaison linéaire du mouvement des deux outils, soit $\mathbf{u} = k_1 \mathbf{u}_1 + k_2 \mathbf{u}_2$. Reste donc seulement à évaluer les poids respectifs des deux coefficients k_1 et k_2 à partir des pseudo-distances aux outils d_1 et d_2 .

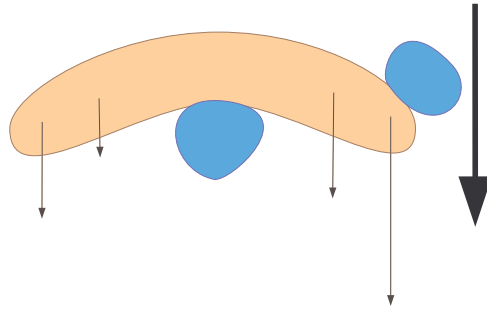


FIG. 3.19 – Erreur commise avec la première formulation : Le mouvement de l'outil de droite est également transmis au matériau de l'autre côté du point d'appui central.

On souhaiterait que la vitesse évolue linéairement lorsque l'on se déplace à l'intérieur du matériau d'un outil vers l'autre. L'idée qui vient immédiatement à l'esprit est de choisir $k_1 = d_2 / (d_1 + d_2)$, de même pour k_2 . Cela répond à cette condition d'évolution linéaire des vitesses entre les deux outils. Seulement ce choix donne un certain nombre de résultats assez étranges, comme celui présenté sur la figure 3.19. À grande distance des outils, le mouvement du matériau a un mouvement qui correspond à la moyenne des déplacements des deux outils.

Cet effet indésirable provient du fait que l'on a étendu la fonction précédente loin de la zone sise entre les deux outils, là où l'interpolation est effectivement correcte. Nous proposons donc l'usage de deux fonctions un peu différentes pour le calcul des poids :

$$\begin{cases} k_1 = \frac{1}{2} + \frac{d_2 - d_1}{2d_{12}} \\ k_2 = \frac{1}{2} + \frac{d_1 - d_2}{2d_{12}} \end{cases} \quad (3.22)$$

où le terme d_{12} représente la pseudo-distance entre les deux outils. Elle se calcule simplement

en considérant la plus petite distance à l'outil 1 des cellules à l'intérieur du volume de l'outil 2. Le principe de l'inégalité triangulaire nous assure que les coefficients k_1 et k_2 voient bien leurs valeurs comprises entre 0 et 1, et on peut vérifier aisément que leur somme fait 1. Ce sont ces poids que nous allons utiliser dans le calcul du mouvement car ils donnent de meilleurs résultats. En effet, le comportement anormal présenté ci-dessus n'est plus d'actualité. Chaque outil fait en quelque sorte "écran", et les cellules derrière l'outil en question subissent essentiellement la seule influence de ce dernier.

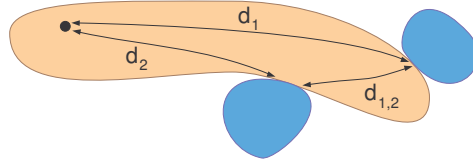


FIG. 3.20 – Distances intervenant dans le calcul des coefficients k_i : d_1 et d_2 entre l'objets et chacun des outils, et $d_{i,2}$ séparant les deux outils.

3.7.4 Résultats et limitations

Sur la figure 3.21, on peut voir le résultat de la déformation d'une barre d'argile sous l'effet de deux outils, l'un (celui de gauche), immobile, servant de point d'appui, l'autre se déplaçant verticalement.

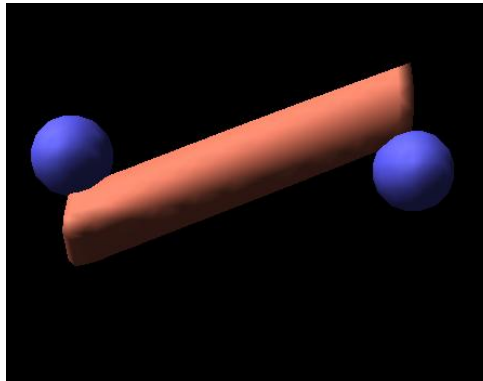


FIG. 3.21 – Le déplacement vertical de l'outil de droite a permis la déformation de la barre d'argile.

Les déformations obtenues de cette façon sont dans l'ensemble acceptables, même si elles ne correspondent pas toujours à la réalité. La principale différence avec un matériau réel, lorsque l'on veut plier une partie de la sculpture, tient au fait que l'on ne peut pas compter sur la rigidité du matériau et exercer l'effort au milieu de la partie à déformer. On est obligé d'en prendre l'extrémité si l'on souhaite que la déformation corresponde effectivement à ce que l'on souhaitait obtenir.

3.7. SECONDE COUCHE : UN MODÈLE POUR LES DÉFORMATIONS GLOBALES 97

De façon générale, les déformations sont de meilleure qualité dans la zone située entre les outils. Ceci tient à notre méthode, basée sur l'interpolation des contraintes au niveau du contact avec les outils, qui a des difficultés à proposer une solution valable hors de cette zone, même si nous avons pu limiter les défauts les plus évidents par l'utilisation de coefficients appropriés.

À noter qu'il est relativement aisé de limiter les déformations à une partie du matériau, simplement en disposant des outils qui serviront de "point d'appui" et maintiendront la matière en place là où l'on souhaite que les déformations s'arrêtent. Des outils servant de simple "fixation" peuvent ainsi être utilisés, par exemple pour contraindre à l'immobilité certaines parties du matériau.

Les déformations obtenues ne sont toutefois pas toujours exactement celles auxquelles on peut s'attendre. C'est en partie le cas lorsque le matériau présente des faiblesses, comme un étranglement, comme le montre la figure 3.22. Comme un fluide, les cellules "glissent" un peu trop aisément les unes par rapport aux autres, et le matériau n'a globalement qu'assez peu de tenue.

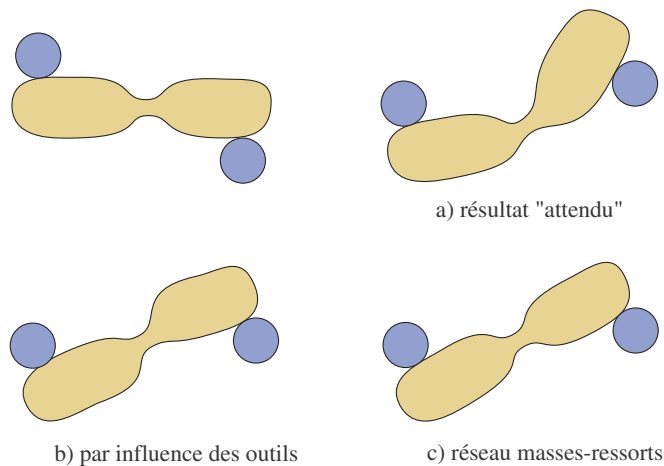


FIG. 3.22 – Exemples de déformations d'une barre présentant un étranglement. En haut à droite, le résultat que l'on s'attend à obtenir avec un matériau réel. En bas à gauche, le comportement que tend à avoir notre modèle : le matériau se "déchire" au niveau de la faiblesse, mais manque de tenue. En bas à droite, un réseau masses-ressorts tend au contraire vers un matériau trop rigide.

On peut trouver deux raisons à cette différence de comportement entre notre matériau et le matériau réel.

- D'abord, nous avons fait l'hypothèse implicite que le mouvement permanent était rapidement atteint, et que ce qui pouvait se passer durant l'établissement du régime permanent n'avait que peu d'importance pour la situation finale. En fait, ce régime permanent n'est pas atteint au même instant en tout point de l'objet, et les retards en certains endroits peuvent occasionner des déformations. Les résultats que nous obtenons sont donc valables, dans le cas d'un fluide visqueux, pour des déplacements très lents des outils.
- Ensuite et surtout, les matériaux comme l'argile (et encore davantage la plasticine)

ne sont pas correctement représentés par les équations de *Navier-Stokes*. Leur modèle suppose en effet que les particules sont essentiellement indépendantes (les effets macroscopiques de viscosité sont dus à des échanges de particules, au niveau microscopique, à l'intérieur de l'écoulement, et non à des efforts internes entre les particules). Ce qui n'est pas le cas ici : il existe des forces de cohésion supplémentaires, de type *Van der Waals*, à l'intérieur du matériau. Ces forces sont négligées dans ce modèle, d'où un comportement trop "fluide" du matériau virtuel.

L'utilisation d'un réseau masses-ressorts, que nous avons écarté précédemment, peut, dans certaines situations, donner de meilleurs résultats. Mais c'est loin d'être toujours le cas. Dans la situation de la figure 3.22, un réseau masse-ressorts donnerait au matériau un comportement bien plus rigide. S'il ne peut pas se déformer, et qu'on laisse le réseau évoluer jusqu'à un état d'équilibre, il gardera généralement sa forme, et se contentera de se déplacer et de tourner. La solution pour obtenir la bonne forme consisterait, là encore, à ne pas le laisser évoluer jusqu'à l'équilibre. Mais ainsi que nous l'avons souligné plus haut, il est très difficile de savoir pendant combien de temps nous devrions le laisser évoluer, d'autant que ce temps dépend énormément de l'échelle de l'objet, de la façon dont les ressorts sont placés, ainsi que du choix de leur raideur.

Nous n'avons pas de solution définitive à ce problème. Il n'est pas absolument gênant dans le cadre de la sculpture, puisqu'il est possible de rajouter un outil au niveau de l'étranglement, par exemple, afin de mieux définir la déformation souhaitée. Mais il faut reconnaître que le comportement n'est pas toujours aussi naturel que nous le souhaiterions.

3.7.5 Collaboration des deux couches

Simulation

Dans notre simulation, les deux couches (la couche fluide présentée au chapitre précédent utilisant de nombreuses itérations de notre automate, et la couche de déformation que nous venons de proposer) sont appliquées successivement. Les déformations sont calculées les premières, et s'appuient sur le mouvement des outils. La couche fluide vient ensuite chasser la matière restant éventuellement à l'intérieur des outils, et corriger les accumulations locales produites lors de la déformation du matériau.

La combinaison de ces deux couches permet d'obtenir des résultats assez convaincants, comme par exemple celui présenté sur la figure 3.23. La boule d'argile virtuelle est maintenue en son milieu, et un outil vient simplement pousser la matière. La couche de déformation permet d'obtenir correctement la déformation générale, et la couche fluide vient s'assurer que l'outil laisse bien son empreinte dans le matériau. Le résultat est comme on peut le constater assez proche de ce que l'on obtient avec un matériau réel.

Outils "vides"

On peut définir des outils servant uniquement aux déformations, c'est-à-dire des outils vides à l'intérieur, qui ne chasseront pas la matière. Cela permet d'obtenir des déformations

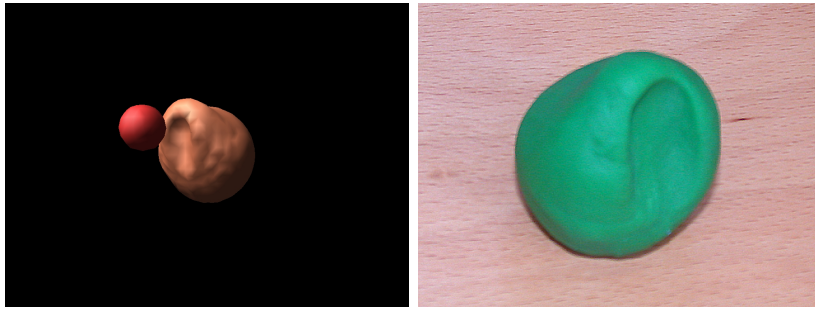


FIG. 3.23 – Déformation virtuelle, à comparer à l’empreinte obtenue en effectuant un mouvement similaire sur une sphère d’argile réelle.

sans avoir à laisser l’empreinte de l’outil dans le matériau. Dans la réalité, pour obtenir le même effet, le sculpteur s’efforce généralement d’augmenter autant que possible la surface de contact pour mieux transmettre les efforts, mais il est difficile de procéder de même dans un simulateur car on dispose d’un nombre limité d’outils indéformables dont la forme n’est généralement pas adaptée à celle de l’objet.

C’est également très utile lorsque l’on souhaite définir un support pour l’objet : c’est un outil qui empêche simplement le mouvement du fluide à l’intérieur, sans réellement occuper une partie de l’espace de travail. On peut ainsi fixer librement divers points d’appui pour le solide, qui faciliteront le travail de l’argile.

L’inconvénient de ces outils tient justement au fait qu’ils ne chassent pas la matière. Il peut parfois être difficile de les décoller du matériau, dans la mesure où il est difficile de les en écarter suffisamment de sorte qu’il ne reste plus de matière à l’intérieur qui puisse être entraînée. Le matériau persiste à adhérer à l’outil lorsque l’utilisateur veut le retirer, une fois la déformation terminée. Il faut alors signifier à l’interface que l’on souhaite déplacer l’outil sans conséquence sur la forme de l’objet, ce qui revient en pratique à ne plus tenir compte de l’outil en question dans la simulation.

Nous avons essayé de détecter automatiquement cette tentative de retrait des outils, en se basant sur l’évolution de la quantité de matière dans l’outil. Si elle diminue, alors on peut penser que l’utilisateur s’efforce de retirer l’outil, et on peut désactiver la seconde couche. Malheureusement, ce critère n’est pas toujours suffisamment discriminant, en particulier lorsque l’on fait glisser l’outil le long de la surface, ou que l’on essaie de déformer un morceau filiforme de la sculpture en manipulant son extrémité. Nous n’avons donc pas pu exprimer, pour l’instant, de critère réellement fiable permettant de détecter les moments opportuns pour la désactivation de la seconde couche du modèle.

Pondération des contributions des deux premières couches

Si l’on se penche à nouveau sur l’expression de Stokes, les deux termes que nous avons simulés jusqu’à présent agissent en fait de façon simultanée. Une partie de la matière est déplacée sous l’effet des contraintes de pression avant que la quantité de mouvement ait pu être correctement diffusée dans le matériau. Il n’est donc pas correct d’utiliser dans un premier

temps la seconde couche de notre modèle, et dans un second temps seulement la première couche, la couche fluide.

L'importance des deux termes, celui de la couche fluide faisant intervenir le gradient de la pression, $-\mathbf{grad}(p)$ et celui de la seconde couche $\nu\Delta(\mathbf{u})$, est modulée par la valeur de la viscosité cinématique ν . Plus elle est importante, plus vite le second terme prendra le dessus. En pratique, cela se traduit par une transmission incomplète du mouvement de l'outil au fluide qui l'environne. Plus le matériau est fluide, plus l'outil va glisser sur le matériau avant que la diffusion de la quantité de mouvement dans le fluide permette de l'entraîner.

Nous pouvons tenir compte de cet effet en remplaçant les vitesses \mathbf{u}_1 des outils, lorsqu'elles apparaissent dans les expressions de la seconde couche de notre modèle, par une fraction de celles-ci :

$$(1 - \eta)\mathbf{u}_1 \tag{3.23}$$

où η est un paramètre compris entre 0 et 1 décrivant la fluidité du matériau. Lorsque par exemple le matériau est très fluide, *i.e.* lorsque η est proche de 1, le mouvement est très mal transmis de l'outil vers l'objet⁴.

L'usage de vitesses modulées permet d'avoir une meilleure approximation des interactions entre les outils et la matière. En faisant varier la fluidité du matériau, on peut insister sur ses effets élastiques, ou bien favoriser les effets fluides qui permettront de percer par exemple plus facilement un trou dans le matériau. La figure 3.24 compare les résultats obtenus avec deux matériaux de fluidité différente. Dans les deux cas, un outil tente de traverser un bloc d'argile. Les quatre coins du bloc sont maintenus en place par quatre outils "vides". Dans le premier cas, on a choisi un matériau très fluide, et malgré son épaisseur, l'outil le traverse sans difficulté, le mouvement de l'outil étant transmis de façon trop inefficace pour occasionner des déformations importantes (tout au plus constate-t-on un léger creusement de la surface). Dans le second cas, on a choisi un matériau nettement plus élastique (sa fluidité est proche de 0) et moins épais. Les déformations du matériau sont nettement plus importantes avant que l'outil ne parvienne à le traverser.

Dans le second cas, les déformations créées laissent clairement apparaître une forme particulière, essentiellement carrée. Cette forme trouve son origine dans la disposition des points d'ancrage, aux quatre coins du bloc d'argile. Elle n'est pas due, comme on pourrait le penser, à notre manière de calculer notre pseudo-distance à l'intérieur du matériau, qui s'apparente à une distance de type *Manhattan*. Une pseudo-distance plus évoluée donnerait essentiellement les mêmes résultats. Cette forme trouve son origine directement dans le calcul des déplacements, qui ne tient compte que de la distance à l'outil et au coin le plus proche. Le lieu géométrique des points équidistants de l'outil et d'un coin se trouve effectivement être un carré de ce genre.

Dans le cas d'une argile réelle, on retrouve une forme similaire à celles des images de la figure 3.24, mais un peu adoucie, de sorte que la forme ne ressort pas de la même façon. C'est l'un des principaux défauts de notre approche, qui donne des résultats parfois un peu plus heurtés que la réalité par son interpolation un peu brutale. Il devrait être possible d'améliorer

⁴Nous reviendrons plus en détail, dans le chapitre suivant, sur la fluidité du matériau. Le paramètre η est ici considéré comme un paramètre global décrivant le comportement plus ou moins fluide du matériau. Nous verrons en fait que cette fluidité peut en fait être définie indépendamment en tout point du matériau.

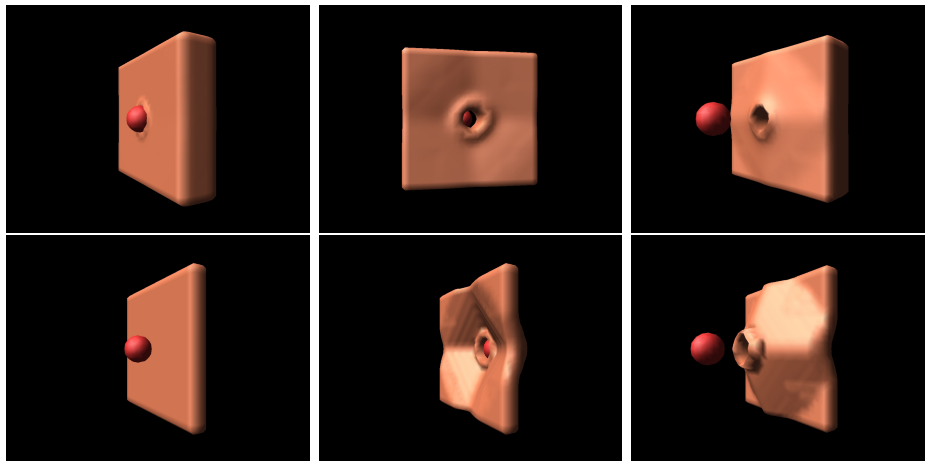


FIG. 3.24 – Tentative de percée d’un bloc d’argile épais et fluide (en haut) et d’une feuille fine et élastique (en dessous). Le matériau est maintenu au niveau de ses quatre coins. L’image de droite est prise depuis l’arrière du matériau.

notablement les résultats en filtrant quelque peu le champ de déformation, par exemple en appliquant quelques pas de l’équation de relaxation déjà rencontrée. C’est une ouverture qui nous semble prometteuse, même si pour des raisons de temps de calcul, il n’était pas possible de l’intégrer à notre modèle.

3.8 Troisième couche : une “tension de surface” limitant la dispersion

3.8.1 Nécessité d’une troisième couche

Comme nous l’avons déjà signalé au chapitre précédent, après un certain temps, une quantité notable de cellules au voisinage de l’outil contiennent une petite quantité de fluide, reliquat de l’interaction avec les outils. Tout se passe comme si, au cours du temps, la matière se diffusait quelque peu dans l’espace de travail. Ce fluide, en quantité trop faible par rapport à l’isovaleur utilisée, n’apparaît pas à l’utilisateur, mais il n’en continue pas moins à être poussé, déplacé par les outils.

Plusieurs conséquences gênantes peuvent découler de cette dispersion invisible de la matière. La première est un phénomène d’apparition parasite de matériau. Un outil peut en effet, en se déplaçant, amasser suffisamment de ces reliquats de matière devant lui pour que la densité atteigne l’isovaleur, causant l’apparition d’un peu de matière en aval de l’outil. Sans être véritablement gênant, cet effet n’en est pas moins désagréable.

Une seconde conséquence de la présence de cette matière dispersée dans l’espace de travail est une imprécision croissante des interactions entre les outils et le matériau. Même si une surface apparaît propre, il peut y avoir suffisamment de matière invisible pour que l’on obtienne des résultats imprévisibles lorsqu’un outil s’approche d’elle. En particulier, les reliquats de

matière interviennent dans le calcul des normales à l'isosurface (normales calculées à partir du gradient du champ au voisinage de ladite isosurface), normales qui peuvent alors devenir très bruitées.

Un troisième effet indésirable consiste en une diminution apparente du volume de l'objet, étant donné qu'il peut y avoir de plus en plus de matière invisible dans l'espace de travail. Là encore, ce n'est pas un gros problème, la décroissance étant assez lente, mais elle peut néanmoins se révéler gênante.

3.8.2 Mise en œuvre de la troisième couche

Principe

Il est possible de combattre ces différents effets indésirables en luttant contre la dispersion de la matière dans l'espace de travail, lors de l'interaction avec les outils. Pour ce faire, nous allons essayer de rassembler cette matière, dans le but de compacter autant que possible le matériau. Cela peut se concevoir comme un effet de surface : la matière reste au contact du matériau, et ne s'éparpille pas, comme le ferait un liquide, dans le reste de l'espace de travail.

Dans notre modèle, cet effet sera implémenté sous la forme d'une couche supplémentaire de simulation. Partout au voisinage de la surface, à l'extérieur du matériau, nous allons nous efforcer de ramener la matière vers l'intérieur. Dans les cellules où la densité est inférieure à l'isovaleur, nous allons examiner la norme du gradient du champ scalaire f . Si celui-ci se trouve être en deçà d'un certain seuil, c'est que la matière commence à s'éparpiller dans l'espace de travail. La troisième couche va alors déplacer une partie du fluide vers la cellule qui, parmi ses six voisines, est la plus proche de la direction indiquée par le gradient.

Précisons enfin qu'il n'est point besoin d'utiliser cette couche à chaque pas de temps. Celle-ci ne fait que corriger des effets qui ne seront visibles qu'à plus ou moins long terme. On peut donc l'oublier un temps pour laisser davantage de temps de calcul aux autres couches du modèle, et ne l'appliquer qu'occasionnellement, par exemple lorsque l'objet est au repos.

Mise en œuvre pratique

La quantité de matière à déplacer et le seuil sur la norme du gradient sont des paramètres difficiles à choisir. Il est préférable de ne pas opter pour un seuil trop élevé pour le gradient, car l'échelonnement des densités sur deux ou trois cellules, au voisinage du bord de l'outil, est ce qui garantit une bonne qualité de l'isosurface obtenue, en particulier au niveau des normales. Si l'on suppose que les cellules sont de taille unitaire, un seuil sur le gradient compris entre 0.3 et 0.5 semble être un bon compromis entre compacité et qualité de la surface.

Quant à la quantité de matière à déplacer, on pourrait envisager de la choisir de façon à s'assurer que le gradient soit suffisant après le déplacement de la matière. Dans la pratique, précipiter les mouvements de matière a tendance à occasionner quelques instabilités, et une quantité correspondant à la moitié de ce qu'il faudrait pour rétablir le gradient donne des résultats plus acceptables.

3.8.3 Un ramasse-miettes

L'un des défauts de cette troisième couche est qu'elle est beaucoup plus sensible aux problèmes de commutativité que la couche fluide. Tout mouvement de matière d'une cellule à une autre influence notablement le gradient dans les cellules voisines, ce qui peut poser des problèmes. C'est l'une des raisons qui motivent à limiter les déplacements de matière. On pourra revenir, au besoin, à une application simultanée sur l'ensemble des cases du modèle, en particulier si on applique la troisième couche lorsque l'objet est au repos.

On peut toutefois s'attendre à quelques effets indésirables. Par exemple, un maximum local de la densité aura tendance à attirer vers lui la matière environnante. On verra donc parfois apparaître des “miettes” de matière dans l'espace de travail. Pour se débarrasser de ces restes de matière indésirable, nous avons décidé d'implémenter un “ramasse-miettes”. Ce dispositif parcourt occasionnellement l'espace de travail, et vide toute cellule située à plus de deux cases d'une cellule dont la densité est supérieure à l'isovaleur. Cette démarche assure que l'espace situé entre les objets est vide de tout fluide, sans que cela influe en rien sur les objets eux-mêmes.

Ce ramasse-miettes est lancé périodiquement, lorsque du temps de calcul est disponible. La matière ainsi récoltée conduit elle aussi à une diminution globale du volume du matériau. Pour éviter cela, on peut replacer la matière retirée dans la cellule pleine la plus proche, la couche fluide s'occupant alors de faire gonfler le matériau à cet endroit. Tout se passe comme si le matériau attirait à lui ces miettes. Cependant, comme la perte d'une petite partie du volume n'est guère importante, nous avons préféré simplement nous en débarrasser.

3.8.4 Résultats et limitations

Visuellement, les effets de l'ajout de cette troisième couche sont essentiellement ceux que l'on attendait : il n'y a plus d'apparitions parasites de miettes, moins d'effets imprévisibles lors de l'action des outils, et une meilleure conservation du volume (même si la préservation des normales implique que cet effet ne saurait complètement disparaître). Elle permet aussi, dans certaines conditions, de faciliter la rupture du matériau lorsqu'il est étiré (chacun des deux morceaux après la rupture étant “recompacté”). Elle ne résout cependant pas complètement les problèmes liés des champs de déformation fortement divergents issus de la seconde couche du modèle.

Cette démarche a par ailleurs ses limites. En particulier, elle va à l'encontre de l'entropie, et ce genre de problème souffre généralement d'un manque de stabilité. Si l'on prend par exemple le cas du défloutage d'une image (un cas relativement proche de celui qui nous occupe), il n'est pas rare que les filtres fassent apparaître des artefacts indésirables, ou conduisent à une image notablement différente de l'originale avant qu'elle ne soit floutée. L'approche proposée ici est particulièrement basique, et elle peut assurément être notablement améliorée.

Chapitre 4

Enrichissement du modèle

4.1 Introduction

Dans le précédent chapitre, nous avons jeté les bases d'un modèle d'argile virtuelle, basé sur trois couches de simulation (une première couche modélisant les effets locaux et assurant la conservation du volume, une seconde permettant des déformations globales du matériau, une troisième, enfin, imitant une tension de surface pour limiter la dispersion de la matière dans l'espace de travail).

Ce modèle présentait encore, toutefois, plusieurs limitations notables, pour lesquelles nous n'avons pas, pour l'instant, apporté de solution. Par exemple, la seconde couche limite à deux le nombre d'outils, et ne permet que des translations de ces outils. Nous avons également tantôt parlé de la possibilité, pour l'utilisateur, d'agir sur certains paramètres physiques du matériau, comme la teneur en eau, mais n'avons que brièvement abordé les effets possible d'une modification globale de la fluidité du matériau.

Dans ce présent chapitre, nous allons nous efforcer à répondre à ces questions, que nous avons laissé en suspens afin de mieux faire ressortir les idées principales qui gouvernaient notre modèle. Nous verrons d'abord une expression plus générale de la seconde couche permettant l'usage d'un nombre quelconque d'outils, qui pourront également tourner dans l'espace de travail. Nous expliquerons ensuite comment introduire des paramètres supplémentaires dans notre modèle (fluidité, mais aussi couleur), et comment les précédentes équations sont modifiées en conséquence. Nous explorerons enfin quelques pistes pour améliorer les performances de notre modèle, tant pour la simulation que pour le rendu.

⁰Ces améliorations de notre modèle d'argile ont fait l'objet d'une publication [DC04b].

4.2 Améliorations de la seconde couche

4.2.1 Extension à un nombre quelconque d'outils

La seconde couche ne permet donc, pour l'instant, qu'une interaction limitée à deux outils. En fait, on peut plus exactement considérer plus de deux outils, la seule condition étant qu'il n'y ait que deux "mouvements" distincts : on peut par exemple utiliser un nombre quelconque de points d'appuis, il suffit de les considérer comme un seul outil (non connexe) lors du calcul de la pseudo-distance.

Il est assez difficile de modéliser un objet avec un nombre aussi limité d'outils (un seul si l'on considère le besoin d'un support pour tenir l'objet). Ce serait comme si un sculpteur essayait de travailler avec seulement l'extrémité d'un de ses doigts, plutôt que d'utiliser l'ensemble de ses deux mains. La démarche peut heureusement être aisément étendue à un nombre quelconque d'outils, seulement limité par des impératifs de temps de calcul. Pour chaque outil, on calculera les pseudo-distances à chacune des cellules à l'intérieur du matériau, et on calculera le champ de vitesse en interpolant les vitesses de déplacement des outils. Le mouvement de la cellule (i, j, k) , soumis à l'action d'un ensemble d'outils, s'exprimera par une combinaison linéaire de celui de chacun des outils :

$$\mathbf{u}_{i,j,k} = \frac{\sum_l k_l \mathbf{u}_l}{\sum_l k_l} \quad (4.1)$$

où la somme est étendue à tous les outils qui exercent une influence sur la cellule.

Les différents coefficients k_l de la combinaison, représentant l'influence de chacun des outils sur la cellule étudiée, s'exprimeront quant à eux par :

$$k_l = \frac{1}{2} - \frac{d_l - d_{l'}}{2d_{l'}} \quad (4.2)$$

où l' correspond à l'outil le plus proche de la cellule, l excepté (c'est-à-dire l'outil l' correspondant à la plus petite pseudo-distance $d_{l'}$). À travers cette formule, le matériau est en quelque sorte "découpé" et partagé entre les outils, et les mouvements varient continuellement (au premier ordre) à l'intérieur de celui-ci.

Le résultat de la déformation d'une barre d'argile sous l'action conjointe de quatre outils aux mouvements indépendants est présenté sur la figure 4.1. La déformation obtenue est à nouveau un peu plus heurtée et anguleuse que ce que l'on obtiendrait avec un matériau réel. C'est un défaut que l'on retrouve régulièrement. Dans la réalité, la réponse du matériau n'est pas linéaire, et il apparaît des déplacements dans des directions différentes des outils, que notre modèle (basé sur une combinaison linéaire des déplacements des outils) n'est pas en mesure de reproduire. La matière réelle, un peu moins fluide que notre matériau, fait parfois montre de réponses un peu plus riches à certaines contraintes, conduisant à de tels déplacements latéraux. Les courbures obtenues au niveau d'un pli avec un matériau réel sont ainsi un peu plus adoucies.

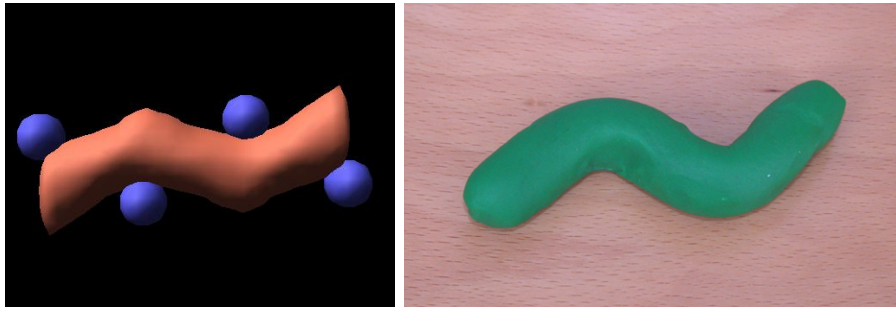


FIG. 4.1 – Déformation d'une barre d'argile sous l'action de quatre outils.

4.2.2 Rotation des outils pour une interaction plus riche

Pour l'instant, nous avons examiné uniquement le cas d'outils en translation dans l'espace de travail. Cela permet d'obtenir une assez large gamme d'effets, mais nous aimerions également pouvoir tourner les outils. La rotation des outils devrait permettre d'obtenir plus facilement des plis dans l'objet sculpté, ou bien des effets plus spécifiques, de torsion par exemple.

Le mouvement d'un solide, quel qu'il soit, peut être défini en un point \mathbf{O} quelconque par la donnée de la vitesse de ce point, $\mathbf{u}_\mathbf{O}$, et du mouvement de rotation du solide ω . À partir de ces informations, il est possible d'en déduire le mouvement $\delta_\mathbf{A}$ d'un point \mathbf{A} quelconque rigidement lié au solide :

$$\mathbf{u}_\mathbf{A} = \mathbf{u}_\mathbf{O} + \mathbf{AO} \times \omega. \quad (4.3)$$

Ce type de relation définit un champ de vissage, c'est-à-dire la combinaison d'une rotation suivant un axe et d'une translation le long de ce même axe. Une illustration en est donnée sur la figure 4.2. Pour tenir compte de la rotation des outils, nous allons simplement utiliser ces champs de déplacement en lieu et place de la seule translation des différents outils dans l'équation 4.1. Nous sommes alors en mesure de déformer l'objet sculpté simplement par rotation des différents outils. Une barre prise entre deux outils en rotation pourra donc aisément être pliée, sous l'action conjointe des deux outils.

Comme évoqué précédemment, les déformations que l'on peut obtenir grâce à cet ajout sont nombreuses et variées. En particulier, il est alors possible d'obtenir des effets de torsion qu'il aurait été difficile de simuler avec de simples outils en translation. Si l'on prend le cas d'une barre, par exemple, et que l'on tourne ses extrémités dans des sens opposés, on obtient un tel effet de torsion du matériau. Si la section de la barre est carrée, la torsion est alors bien visible. La figure 4.3 montre que les résultats obtenus avec notre simulation sont raisonnablement proches de ce que l'on peut effectivement obtenir avec de la pâte à modeler.

Dans la réalité, on s'attend parfois à une rupture de la barre en son milieu après une torsion importante. En fait, pour peu que le matériau soit suffisamment fluide, cette rupture n'aura pas lieu. Elle est en fait provoquée par un éloignement inconscient des deux mains de l'utilisateur qui étire la barre en même temps qu'il la tord (la conservation du volume empêcherait tout amincissement de la barre en son centre, n'eût été l'éloignement des deux extrémités).

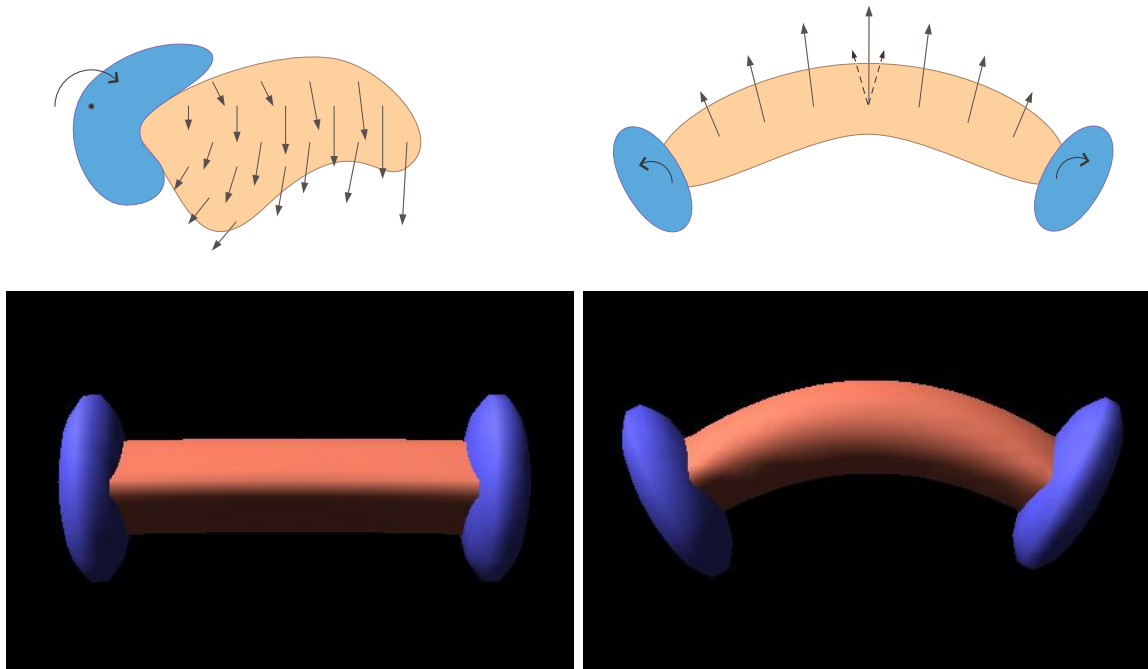


FIG. 4.2 – En haut à gauche, le champ de déplacement associé à un outil en rotation. En haut à droite, la combinaison de deux outils permet la flexion d’une barre d’argile. En dessous, les résultats obtenus par notre simulation.

4.3 Des paramètres intrinsèques supplémentaires

4.3.1 Introduction

La seule répartition des masses ne décrit bien évidemment pas complètement la matière simulée. Elle dépend d’une grande quantité d’autres paramètres parmi lesquels on peut citer la température, la viscosité, la composition, et de nombreuses autres grandeurs thermodynamiques. Pour notre application, nous n’avons pas besoin d’introduire toutes ces grandeurs, mais l’adjonction de certaines de ces dernières nous permettrait d’enrichir notre modèle de glaise.

Les grandeurs physiques peuvent généralement être classées parmi deux ensembles : les paramètres qualifiés d’extrinsèques et ceux que l’on appelle intrinsèques. Dans le premier cas, les grandeurs sont directement proportionnelles à la quantité de matière considérée. C’est le cas par exemple de la masse ou du volume. Dans le second cas, la grandeur ne dépend pas de la quantité de matière. C’est par exemple le cas de la température, de la teneur en eau du matériau, ou bien encore de la composition du matériau.

On s’intéressera essentiellement dans la suite aux seules grandeurs intrinsèques. Il est de toute façon aisé de passer de grandeurs extrinsèques à des grandeurs intrinsèques en les rapportant à la quantité de matière considérée. On supposera que les grandeurs étudiées sont homogènes à l’échelle d’une cellule de notre grille (en fonction de la précision que l’on souhaite vis-à-vis de ces paramètres, il faudra donc éventuellement affiner ladite grille pour que cette

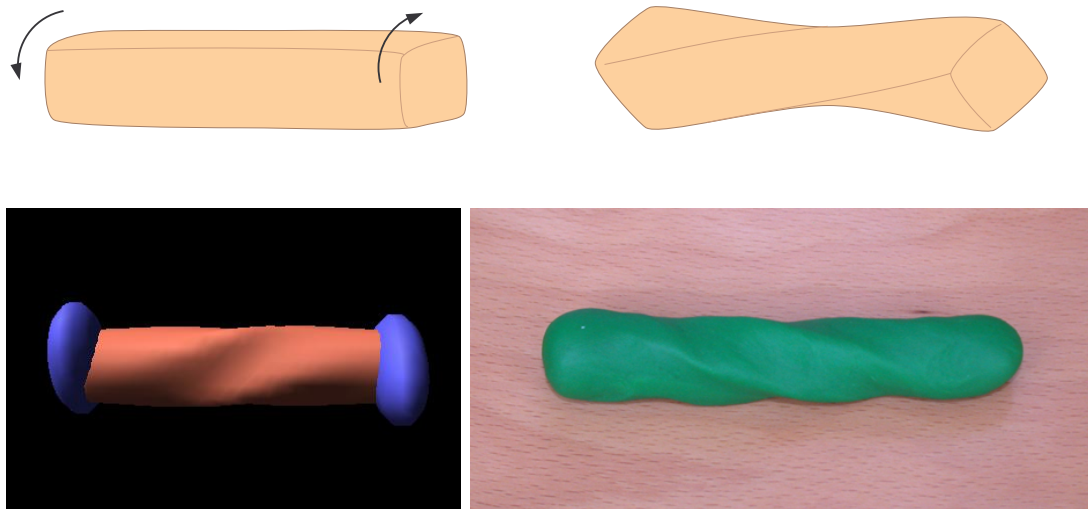


FIG. 4.3 – L’effet de torsion illustré sur une barre d’argile de section carrée. Le comportement de la simulation, en bas à gauche, est similaire à celui d’un matériau réel.

hypothèse puisse être vérifiée). Pour chaque grandeur intrinsèque introduite, les cellules de la grille s’enrichiront donc d’un nouveau paramètre, qui décrira la grandeur en question pour le fluide contenu dans chacune de ces cellules.

4.3.2 Fluidité du matériau

De l’équation 3.2, nous avons retenu deux termes pour notre modèle. Chacun de ces deux termes est à l’origine d’une des couches de simulation. Nous avons vu que si l’on s’intéresse à leurs importances respectives, on constate qu’elle est gouvernée par la présence d’un terme de viscosité cinématique, ν : lorsque cette viscosité est grande, le matériau aura un comportement plus élastique, moins fluide. Cette viscosité nous a déjà permis tantôt de pondérer les contributions des deux premières couches de notre modèle.

Dans le cas de l’argile, la viscosité est essentiellement une question de teneur en eau. Plus le matériau est riche en eau, plus sa viscosité cinématique tend à être faible, et plus le matériau est fluide. Le sculpteur exploite naturellement cette possibilité en ajoutant de l’eau, ou en faisant sécher son argile pour l’amener au degré de fluidité qu’il souhaite. Notre but, encore une fois, n’est pas ici de reproduire fidèlement et de façon quantitative le comportement d’une argile. Nous ne chercherons donc pas à lier précisément la teneur en eau et la viscosité du matériau. Ce qui nous semble utile, c’est que l’artiste puisse jouer sur ces paramètres, et que le comportement du matériau, d’un point de vue qualitatif, soit suffisamment convaincant.

Nous allons donc introduire un premier paramètre supplémentaire pour le matériau, noté η , qui représentera la fluidité du matériau (cette fluidité trouvant son origine dans la teneur en eau du matériau, sa température, etc). Pour plus de simplicité, on imposera que ce paramètre varie entre 0 (le matériau est essentiellement dépourvu de tout comportement fluide, et se présente comme très élastique) et 1 (le matériau étant alors pratiquement un fluide dépourvu

de viscosité). Cette fois, ce n'est plus un paramètre global, mais bien une grandeur définie de façon indépendante en tout point du matériau.

La principale difficulté à résoudre lorsque l'on souhaite simuler une grandeur supplémentaire résidera dans la façon dont nous traiterons le mélange de deux quantités de matériaux dont les caractéristiques sont différentes. Par exemple, lorsque l'on a une cellule comportant une quantité n_1 de matière, dont la teneur en eau vaut η_1 et que l'on souhaite y ajouter une quantité n_2 de matière de teneur en eau η_2 . La quantité totale de matière obtenue vaut naturellement $n_1 + n_2$, et la quantité d'eau présente dans le mélange correspond alors à $n_1\eta_1 + n_2\eta_2$. La teneur en eau de l'ensemble s'exprime donc comme le barycentre des teneurs en eau des deux échantillons initiaux, dont les poids sont les quantités de matière respectivement apportées par ces deux échantillons.

La plupart des grandeurs physiques se comporteront, vis-à-vis des mélanges, de façon identique, et il est généralement possible d'utiliser un calcul barycentrique similaire pour obtenir les caractéristiques du mélange. Si l'on souhaitait s'intéresser à la température, il faudrait estimer les quantités de chaleur apportées par chacun des deux échantillons, qui font intervenir leurs températures et leurs capacités thermiques respectives. Généralement, les capacités thermiques des deux échantillons sont identiques (dans la mesure où le matériau est homogène et où les températures varient suffisamment peu pour que l'on puisse négliger toute variation de la capacité thermique), et on peut considérer $T_{finale} = n_1T_1 + n_2T_2$. Le mélange de quantités égales d'argile respectivement à vingt et quarante degrés permettra d'obtenir un mélange à une température avoisinant les trente degrés.

4.3.3 Utilisation de la fluidité dans la simulation

Nous avons déjà vu, dans le chapitre précédent, que la fluidité η est utile pour déterminer l'efficacité de la transmission de la quantité de mouvement de l'outil au matériau. Si un outil tente d'exercer un effort sur une zone très fluide du matériau, il s'y enfoncera quelque peu avant de parvenir à le déplacer.

Le paramètre η qui intervient dans l'expression 3.23 est donc celui caractérisant le matériau au niveau du contact avec l'outil. A condition toutefois de pouvoir supposer que l'outil en question soit d'une taille suffisamment petite pour que ladite fluidité ne varie pas dans des proportions importantes sur l'ensemble de la surface de contact. On pourra en prendre la valeur médiane, ou plus simplement la valeur maximale

Ce n'est cependant pas le seul avantage que l'on peut tirer de l'introduction d'un terme local de fluidité à notre matériau. Nous pouvons en effet envisager de tenir compte des effets d'une viscosité non-homogène à l'intérieur du matériau dans la transmission du mouvement à l'intérieur de celui-ci. Dans les zones peu visqueuses, par exemple là où la glaise contient une importante quantité d'eau, le mouvement est en effet transmis moins efficacement de proche en proche. L'influence d'un outil se fera donc moins sentir si elle est vue à travers une zone de l'objet plus fluide.

Cet effet peut être traduit au travers du calcul de la pseudo-distance à un outil, qui correspond directement à son influence. La pseudo distance à un outil augmente plus rapidement

dans les zones où le matériau est très fluide (traduisant un affaiblissement plus rapide de son influence). L'expression 3.20 est, pour ce faire, remplacée par :

$$d_{i',j',k'} = d_{i,j,k} + \frac{1}{n_{i,j,k}(1 - \eta_{i,j,k})}. \quad (4.4)$$

La figure 4.4 illustre les effets induits par une fluidité non-homogène dans le matériau. La barre de droite, plus fluide au niveau de son extrémité droite, ne s'est pas déformée de la même façon que la barre homogène, sur la gauche. Les parties les plus fluides du matériau se déforment plus facilement que les autres, ce qui permet d'obtenir divers effets intéressants. Précisons que, dans cet exemple, afin de mieux illustrer les effets de non-homogénéité du matériaux, la transmission du mouvement de l'outil au matériau a été supposée parfaite (comme si le matériau avait une fluidité nulle au niveau du contact). C'est par ailleurs un autre avantage du travail avec une argile virtuelle : même si le matériau est fluide, il est possible d'éviter à l'outil, si petit soit-il, de s'enfoncer dans le matériau si on ne le souhaite pas.

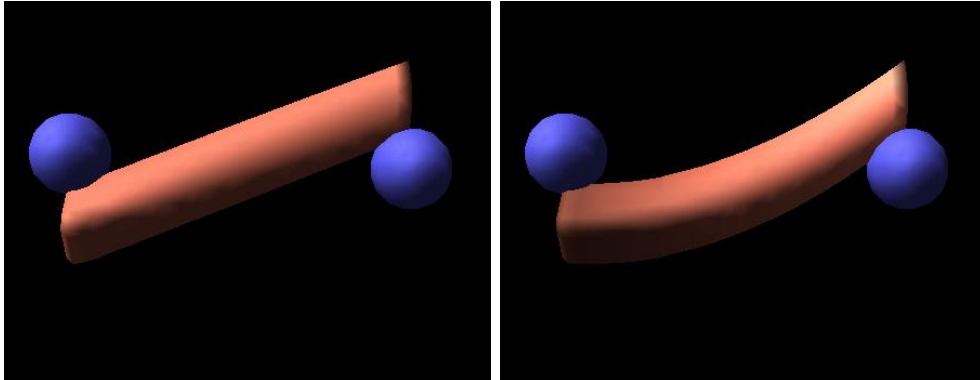


FIG. 4.4 – À gauche, la barre déformée par les outils est homogène. À droite, sa fluidité est plus grande au niveau des zones plus claires, ce qui permet de la rendre localement plus flexibles (les zones plus foncées étant quant à elles plus rigides).

4.3.4 Évolution des paramètres

Il peut arriver que dans le cas d'un objet auquel on ne touche pas, on observe néanmoins une évolution locale des grandeurs physiques du matériau. L'exemple le plus simple que l'on puisse citer est bien évidemment la température : par simple conduction, celle-ci va s'écouler depuis les parties les plus chaudes de l'objet en direction des parties les plus froides. Il est possible d'ajouter au modèle n'importe quel type de loi d'évolution qui peut sembler utile. La description en grille de la matière rend généralement cette opération relativement simple.

Par exemple, dans le cas de la température, on dispose de nombreux modèles permettant de décrire les échanges thermiques dus à la conduction à l'intérieur d'un objet. Généralement, deux cellules voisines peuvent s'échanger une certaine quantité de chaleur, proportionnelle à la différence de température, et on calcule ces différents échanges sur l'ensemble du volume de l'objet. Les équations de diffusion mises en jeu sont similaires à celles présentées

précédemment, à la différence près que la matière cette fois-ci ne bouge pas. Seuls les paramètres supplémentaires du modèle évoluent.

En général, l'échelle des temps sur laquelle interviennent ces phénomènes diffère notablement de celle associée à la sculpture de l'objet. Les échanges de température, par exemple, se font beaucoup plus lentement que les déformations en réponse à l'action d'un outil sur le matériau. On peut donc appliquer ces modèles de façon asynchrone avec la simulation des déformations. Par exemple, à chaque pas de temps, on calculera complètement les déformations issues du mouvement des outils (jusqu'à la convergence, ces effets apparaissant instantanés pour l'utilisateur), et ensuite une ou plusieurs itérations du modèle décrivant la diffusion de la chaleur dans le matériau (cette fois-ci, on ne cherche pas à atteindre une quelconque convergence, les temps pour la diffusion de la chaleur se comptant en minutes. Cette diffusion se fera progressivement, à chaque pas de temps de la simulation).

Dans le cadre de la sculpture virtuelle, la plupart de ces évolutions ne sont pas utiles, voire peuvent être gênantes. Ce sont généralement des diffusions qui conduisent à une uniformisation des propriétés du matériau. Si ces propriétés ne sont pas homogènes, c'est *a priori* un souhait de la part de l'utilisateur qui préfère qu'une zone donnée de l'objet ait un comportement différent. On pourra envisager une uniformisation de ces paramètres sur une demande explicite de l'artiste, mais nous n'avons pas vu d'usage utile à l'adjonction d'un modèle de ce genre à notre modèle, même si nous avons pu constater qu'ils donnaient les résultats attendus.

4.4 Un matériau en couleurs

4.4.1 De l'origine des couleurs

La couleur ne fait *a priori* pas partie des principaux attraits de l'argile, mais il n'en est pas de même avec des matériaux comme la pâte à modeler. La possibilité de jouer avec les couleurs, de les mêler, est un aspect important de cette dernière. Nous allons voir que la couleur n'est qu'un paramètre physique un peu particulier, et qu'une démarche similaire à celle menée précédemment nous permet d'introduire ce concept de couleur à notre modèle. La couleur d'un matériau n'est en fait qu'une capacité de ce dernier à absorber certaines des longueurs d'onde composant le spectre visible, tandis qu'il en renvoie d'autres.

La vision humaine des couleurs n'est pas encore pleinement comprise. Elle réside dans la présence, au niveau de la rétine de l'œil, de cellules appelées cônes sensibles à certaines longueurs d'onde spécifique du rayonnement. En simplifiant les choses, il y aurait essentiellement deux types de cônes : le premier mesurerait la différence de rayonnement dans les longueurs d'onde verte et rouge, le second travaillant de la même façon avec le vert et le bleu. En fonction de la stimulation de ces cellules par la lumière entrant dans l'œil, on verra un objet plutôt dans les tons verts, ou bien dans les tons rouges.

Pratiquement, cela signifie que nous ne sommes pas sensibles à la forme complète du spectre émis par les différents objets dans le visible, mais que nous comparons essentiellement les intensités relatives du rayonnement dans trois longueurs d'onde précises : le rouge, le vert, et le bleu. Avec ces trois seules longueurs d'onde, il n'est bien évidemment pas possible de



FIG. 4.5 – La couleur fait partie des caractéristiques importantes de la pâte à modeler.

reproduire n'importe quel spectre d'émission, mais cela suffit pour générer l'ensemble des nuances que l'œil est capable de discerner ou presque.

Cette constatation a été largement exploitée dans tous les domaines. Les écrans combinent ces trois couleurs de base (rouge, vert et bleu) pour générer l'ensemble des nuances, tandis que les caméras et appareils photo mesurent l'intensité du rayonnement dans ces trois mêmes longueurs d'onde. C'est le principe de la synthèse des couleurs qualifiée d'additive.

Les teintures et les pigments fonctionnent de façon un peu différente : ils reçoivent de la lumière, absorbent certaines longueurs d'onde, tandis qu'ils en diffusent d'autres. Un pigment absorbant la longueur d'onde bleue va réémettre les longueurs d'onde verte et rouge, et aura donc une couleur jaune, résultant du mélange de vert et de rouge. Les encres et peintures utilisent donc trois couleurs de base dites primaires, jaune, magenta et cyan (qui correspondent à l'absorption respective des longueurs d'onde bleue, rouge et verte) pour donner toute la gamme des couleurs. On parle alors de synthèse soustractive¹.

4.4.2 Adaptation de notre modèle

Pour avoir un modèle de pâte à modeler disposant de toute la gamme des couleurs visible par l'œil, il nous suffira donc d'y adjoindre trois pigments de base, absorbant chacune des trois longueurs d'onde auxquelles l'œil est sensible (un pigment jaune, un pigment cyan et un pigment magenta dont nous noterons les quantités respectives par unité de masse κ^J , κ^C et κ^M). Plus le pigment sera présent dans une cellule, plus la longueur d'onde correspondante sera absorbée. Une zone complètement dépourvue de pigment apparaîtra donc blanche, tandis qu'une zone riche en chacun des trois pigments sera noire.

La densité de pigments à l'intérieur de la matière peut être gérée de la même façon que nous avons traité le cas de la teneur en eau. Un mélange de deux échantillons dont la pigmentation diffère verra sa composition en pigments déterminée par un simple calcul barycentrique. Par

¹Pour des raisons sur lesquelles nous ne nous attarderons pas, on ne couvre pas parfaitement nécessairement rigoureusement l'ensemble des nuances avec ces synthèses, mais les zones faisant défaut sont négligeables.

exemple, pour le pigment jaune :

$$\kappa_f^J = n_1\kappa_1^J + n_2\kappa_2^J.$$

Il en est de même pour les deux autres pigments. Le mélange conduit donc à une modification des absorptions relatives dans les différentes longueurs d'onde du matériau. En mélangeant de la matière riche en pigment jaune (absorbant le bleu) et de la matière riche en pigment cyan (absorbant le rouge), on obtient un mélange qui absorbe à la fois le bleu et le rouge, et apparaîtra donc vert à l'œil.

Au niveau du rendu, lorsque l'on définit la couleur d'un matériau, on spécifie généralement simplement sa capacité à renvoyer chacune des trois longueurs d'onde rouge, verte et bleue, exprimée par exemple par un réel compris entre 0 (absorption complète) et 1. Plus un pigment est présent, plus la composante de la couleur qui lui correspond doit donc être faible. Un matériau riche en pigment jaune voit ainsi sa composante bleue tendre vers zéro. On pourra par exemple écrire :

$$c^B = 1 - \kappa^J \text{ si } \kappa^J < 1 \text{ et } c^B = 0 \text{ sinon}$$

où c^B est la composante bleue pour le matériau en question, et n^J la densité du pigment jaune. À partir du moment où la densité d'un pigment dépasse la valeur 1, la couleur correspondante est donc complètement absorbée. Etant donné que les transformations affines se comportent correctement vis-à-vis des barycentres, on pourra, pour des raisons de simplicité, travailler directement et de façon similaire avec les paramètres c^R , c^V et c^B en lieu et place des κ^J , κ^C et κ^M .

4.4.3 Résultats obtenus

Les résultats obtenus de cette façon sont assez satisfaisants, et il est possible de combiner les couleurs de la même façon qu'avec un matériau réel, ainsi que le montrent les images de la figure 4.6. Un mélange d'un ensemble quelconque de couleurs tend quant à lui vers le résultat grisâtre que l'on attend. Les résultats obtenus par notre modèle semblent de façon générale légèrement plus ternes, en particulier lorsque l'on utilise la couleur jaune. Cela vient probablement du fait de l'usage, dans la pâte à modeler réelle, de composés réagissant à l'ultraviolet donnant aux couleurs un peu plus d'éclat et un caractère quelque peu fluorescent délicat à reproduire à l'écran.

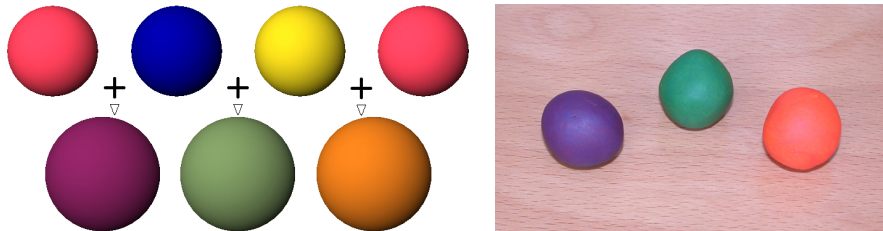


FIG. 4.6 – À gauche, les résultats obtenus en combinant deux à deux chacune des couleurs de base. À droite, les mêmes mélanges obtenus avec de la pâte à modeler réelle.

Précisons par ailleurs que les échelles ont été conservées lors de la simulation. La différence de taille entre les sphères vient simplement de la conservation du volume. Le mélange pouvant

être (dans la simulation comme dans la réalité!) un processus fastidieux, nous avons simplement placé chacune des deux boules de pâte à mélanger dans les mêmes cellules, et laissé la couche fluide traiter l'excédent de matière ainsi occasionné.

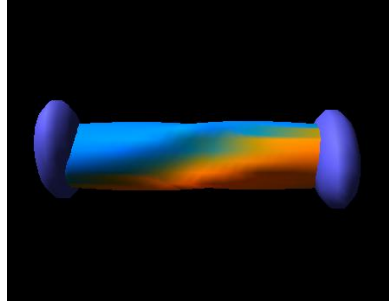


FIG. 4.7 – Un essai de déformation d'une pièce colorée d'argile avec notre modèle.

Évidemment, la nécessaire mise à jour de ces trois paramètres supplémentaires ralentit quelque peu la simulation. En outre, les variations de la couleur ne peuvent être observées qu'à des échelles plus grandes que celles de la cellule, et il peut se révéler nécessaire d'affiner la résolution de la grille. Si l'on souhaite aller plus loin dans cette direction, il est possible d'envisager d'intégrer d'autres paramètres régissant la couleur du matériau dans des conditions particulières (spéculaire, diffusion, etc).

4.5 Vers une approche multirésolution

4.5.1 Introduction

L'un des inconvénients du modèle présenté est qu'il devient rapidement gourmand en terme de ressources. En effet, bien que les opérations de base soient des plus simples, leur nombre est rapidement très élevé (cela se compte en dizaines, en centaines de millions de mouvements de matière entre les cellules). Et ceci, d'autant plus que l'on souhaite travailler avec des grilles relativement fines afin d'obtenir un maximum de précision. La vitesse actuelle des processeurs, combinée avec les quelques optimisations présentées précédemment, permettent de travailler interactivement (à raison de plusieurs dizaines de pas de simulation par seconde) sur des objets dont la taille est de l'ordre de quinze, vingt cellules dans chacune des dimensions, voire davantage, sur des machines tout à fait courantes.

En outre, la méthode que nous présentons se prête très bien à une parallélisation des tâches entre plusieurs machines. De par le caractère local des opérations de déplacement de matière, tant que les cellules concernées sont éloignées de plus de deux cases, les calculs peuvent être faits simultanément sans précaution particulière. Nous n'avons pas particulièrement creusé cette voie, mais on peut aisément imaginer un arbitre, gérant la pile des cellules à traiter, répartir ces dernières entre un groupe de processeurs. La tendance actuellement suivie pour les processeurs tend par ailleurs à privilégier les algorithmes utilisant une grande quantité d'opérations simples et systématiques. En attendant ces progrès techniques, on peut cependant réfléchir à d'autres approches visant à diminuer la quantité de calculs nécessaire à la simulation

du modèle.

4.5.2 Maillage multirésolution

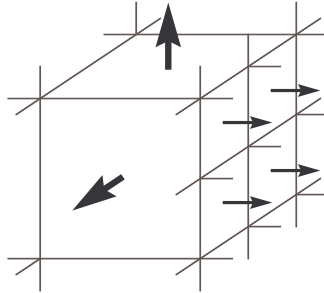


FIG. 4.8 – Lorsque les cellules ont des voisines de taille différente, le volume de chaque déversement (ici figuré par la taille des différentes flèches) est directement proportionnel à la surface d'échange entre les cellules.

Un objet présente fréquemment des endroits nettement plus détaillés que d'autres. Utiliser une grille régulière pour représenter cet objet est souvent inefficace : la grille doit en effet être suffisamment détaillée pour pouvoir contenir les détails les plus fins, quand bien même cette résolution serait superflue dans l'essentiel de l'espace de travail. Par exemple, dans le cas d'un objet sculpté, la surface contient plus d'informations que l'intérieur de l'objet, généralement plein et homogène. En outre, la précision requise peut être plus grande à certains endroits de la surface (le visage et les mains dans le cas d'un personnage par exemple). Avec un maillage régulier, ce sont les endroits où les détails sont les plus fins qui devront fixer la résolution de la grille, conduisant à de la précision inutile dans d'autres parties de l'objet.

On pourrait envisager un maillage de notre objet plus grossier à l'intérieur de ce dernier, et un maillage plus fin au niveau de la surface. Bien que rien ne l'interdise, ce n'est cependant pas simple à mettre en œuvre. Nous examinerons ici les difficultés posées par l'introduction d'une grille non-régulière en ce qui concerne la première couche de notre modèle, qui utilise de façon importante la notion de voisinage, et les modifications qu'il convient d'apporter.

La première difficulté rencontrée se situe au niveau des voisinages des cellules. Si le maillage n'est plus régulier, une cellule donnée n'a plus nécessairement six voisins, mais peut en avoir davantage, et ils peuvent ne pas être tous équivalents. Il faut alors modifier les opérations de déversement en conséquence. La fraction de matière échangée entre une cellule et sa voisine sera directement proportionnelle avec la surface qu'elles partagent (voir figure 4.8).

4.5.3 Correction des dissymétries dues au maillage non régulier

D'autres difficultés plus subtiles vont cependant émerger : cette grille non régulière va introduire des dissymétries et des erreurs dans la diffusion (dissymétries que l'on retrouvera dans le comportement du matériau), ainsi que le montre la figure 4.9. Ces dissymétries peuvent être gênantes, dans la mesure où les bourrelets pourraient par exemple apparaître d'un seul

côté de l'outil. Pour pallier cette limitation, il faut pouvoir se rappeler, lorsque l'on déverse la matière, l'endroit où elle a été introduite. Pour des raisons de simplicité, nous allons raisonner dans le cas d'une grille à deux dimensions, mais les idées seront les mêmes dans le cas d'une grille à trois dimensions. De même, nous traiterons uniquement le cas de cellules voisines dont les tailles vont du simple au double.

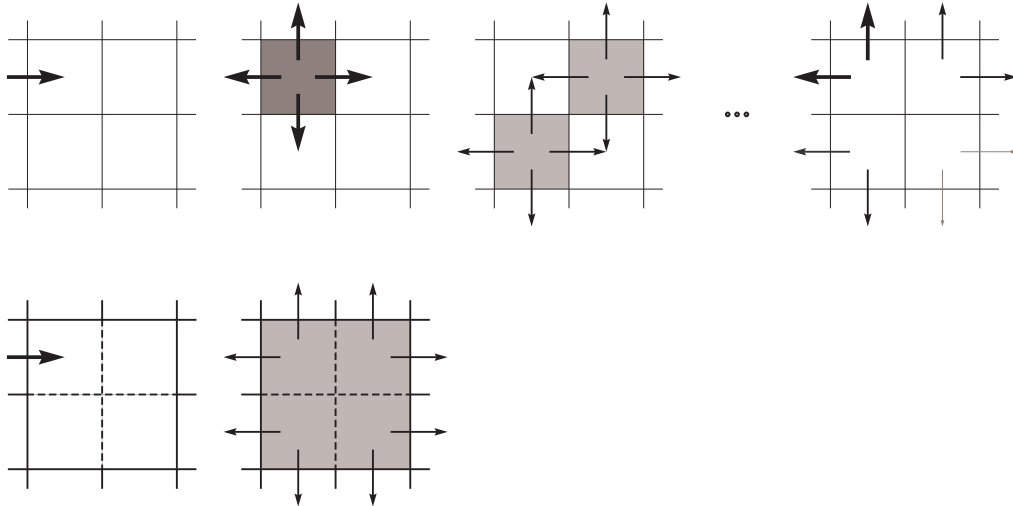


FIG. 4.9 – En haut, les échanges sur une grille régulière, faisant suite à l'introduction de matière en haut à gauche, conduisent à une répartition non uniforme de la matière en excès. La taille des flèches correspond aux quantités de matière transférées. En bas, cette dissymétrie est perdue lorsque l'on utilise des cellules de plus grande taille, toutes les voisines étant traitées de façon identique.

Si une cellule déverse de la matière dans une cellule de taille plus grande, l'excès éventuel de matière n'est pas au centre de la cellule la plus grande, mais plutôt au voisinage du centre d'une cellule de même taille que la cellule qui a apporté de la matière, sise de l'autre côté de la surface de contact entre les deux cellules. Le déversement devrait donc en tenir compte. La figure 4.9 montre ce que l'on attend du déversement pour différents positionnement de l'excès initial de matière (les calculs ont été faits en utilisant une grille de même taille que la cellule de petite taille).

Lorsque l'excès se trouve au milieu, les huit cellules entourant la grande cellule doivent recevoir une même quantité de matière. Si au contraire il se déplace vers l'un des bords, une moitié des cellules recevra une quantité de matière plus importante que l'autre. Si enfin l'on s'approche d'un des quatre coins de la cellule, la répartition entre voisines est encore plus complexe. Tout dépend donc de l'endroit où se situe l'apport de matière. Nous noterons cette position par rapport à un repère situé à l'intérieur de la cellule, choisi tel que son centre corresponde au centre de la cellule, et que les bords de la cellule vérifient $x = \pm 1$ et $y = \pm 1$. Si l'on exprime les coordonnées du barycentre dans ce repère, les quantités $\frac{(1 \pm x_G)(1 \pm y_G)}{8}$ représentent une approximation acceptable des proportions à attribuer aux voisines dans chacun des quatre quadrants entourant la cellule (le résultat est "exact" au niveau des axes dudit repère, et donne des résultats satisfaisants ailleurs).

On va donc calculer le barycentre des apports de matière à chaque cellule, et s'en servir

au moment de reverser la matière en excès. Chaque cellule dont la taille est plus grande que la taille élémentaire du maillage dispose donc d'informations supplémentaires : les coordonnées du barycentre de l'apport (x_G, y_G, z_G) , et la quantité de matière apportée n_{apport} . Lorsqu'une cellule voisine déverse une quantité de matière n_{supp} , le barycentre de l'apport est mis à jour :

$$x_G = \frac{n_{apport}x_G + n_{supp}x}{n_{apport} + n_{supp}}.$$

De même pour y_G et z_G . On met également n_{apport} à jour au moyen de la relation $n_{apport} = n_{apport} + n_{supp}$.

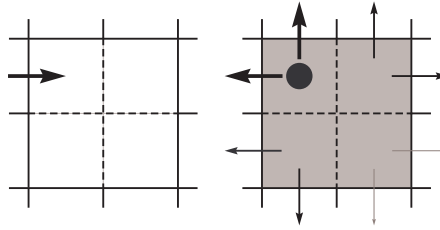


FIG. 4.10 – Avec la connaissance du centre de masse, on peut retrouver une répartition non uniforme entre les voisines. Le centre de masse y est figuré par un point noir.

Lorsque le moment vient pour la cellule de reverser dans ses voisines un éventuel excès, on peut exploiter la connaissance du barycentre de l'apport (x_G, y_G, z_G) . Les déversements sont alors répartis selon la loi proposée précédemment. Le résultat est illustré sur la figure 4.10. On constate que la connaissance du centre de masse permet de favoriser, lors du déversement, les cellules voisines les plus proches de l'endroit par lequel la matière a été introduite, avec des résultats très proches de ceux obtenus en considérant chacune des sous-cellules.

4.5.4 Bilan

Ces résultats montrent qu'il est possible d'envisager l'usage de techniques multi-résolution dans le cadre de la première couche de notre modèle. De même, on peut l'envisager pour chacune des deux autres couches de simulation.

La seconde couche est vraisemblablement celle qui posera le moins de difficultés théoriques. Il faudra simplement s'assurer que les déplacements de matière se font d'une cellule à une cellule de taille égale ou inférieure, afin de préserver les détails de la sculpture. Si ce n'est pas le cas, il faudra effectuer un raffinement local de la grille.

Si les cellules d'arrivée sont de taille inférieure aux cellules dont est originaire la matière, il faudra cependant répartir la matière déplacée dans un grand nombre de cellules. Ces problèmes de répartition peuvent s'avérer très délicats, et la solution la plus simple consiste probablement à découper la cellule de départ en un ensemble de cellules de plus petite taille, afin de déterminer précisément où doit parvenir la matière déplacée.

Au fil du temps, la grille va donc devenir de plus en plus précise, en particulier chaque fois qu'une déformation à grande échelle déplace des détails. Il va donc également falloir envisager une méthode capable de déterminer les endroits où la précision est devenue superflue, et

où il est possible de regrouper plusieurs cellules en une seule cellule de plus grande taille. Même si ce genre de méthode existe, c'est un problème supplémentaire qui vient s'ajouter aux nombreux autres.

Pour ces nombreuses raisons, nous n'avons pas exploité, en pratique, la possibilité d'utiliser une grille non régulière. Les problèmes supplémentaires sont nombreux, les opérations à effectuer notablement plus complexes, et le gain en temps n'est pas forcément toujours net. Nous nous contenterons donc ici de signaler que ces améliorations sont envisageables, indispensables d'ailleurs si l'on veut pouvoir travailler sur des détails bien plus petits que l'objet lui-même, mais nécessiteront un travail approfondi.

4.6 Quelques considérations sur le rendu

Nous terminerons la présentation de notre modèle en abordant un problème soulevé par le choix d'une description implicite pour notre argile. Nous avons signalé tantôt que l'usage des surfaces implicites présentait un inconvénient notable : il faut généralement explorer l'espace de travail afin de localiser le lieu géométrique des points constituant l'isosurface. Cette opération peut être coûteuse, lorsqu'il s'agit de la répéter plusieurs dizaines de fois par seconde, en prenant soin de laisser un maximum de temps de calcul à disposition de l'animation proprement dite.

Dans la suite, nous examinerons les avantages et inconvénients pour notre usage des principales méthodes disponibles, et proposerons une méthode originale de tracé rapide pour les instants où il serait intéressant de laisser un maximum de temps de calcul à disposition des différentes couches simulant le comportement du matériau.

4.6.1 Rendu d'une surface implicite

Visualisation par raytracing

La première approche possible pour rendre une surface implicite consiste à employer des méthodes de lancer de rayon. Cela consiste à rechercher la première intersection entre un ensemble de rayons issu de la caméra et la surface implicite. La méthode est probablement celle qui fournira les meilleurs résultats, mais se trouve être généralement lente. En posant quelques hypothèses concernant le caractère lipschitzien du champ scalaire, D. Kalra et A. H. Barr proposent dans [KB89] une méthode permettant de déterminer rapidement et de façon sûre l'ensemble des intersections entre chacun des rayons et la surface. Un tel rendu est cependant trop coûteux en terme de temps de calcul pour que nous puissions l'utiliser.

Polygonisation de la surface

Une autre approche consiste à déterminer une polygonisation de la surface implicite. L'algorithme le plus populaire est celui dit des *marching cubes*, présenté par W. E. Lorensen et H. E. Clien dans [LC87]. Celui-ci propose de découper l'espace en un ensemble de cubes

élémentaires. Dans chacun de ces cubes, il détermine un ensemble décrivant l'isosurface, en se basant sur la situation (à l'intérieur ou hors du volume) de chacun des sommets. J. Bloomenthal suggère une démarche de subdivision, basée principalement sur la qualité du maillage (courbure estimée à partir des normales aux faces et aux sommets, taille et forme des polygones), afin d'affiner le résultat obtenu [Blo88]. Une variante de cette approche a également été proposée par P. Ning et J. Bloomenthal dans [NB93], où des tétraèdres sont utilisés en lieu et place des cubes élémentaires. De par le nombre réduit de sommets (quatre au lieu de huit), le nombre de cas à examiner est nettement plus restreint. En contrepartie, la qualité du maillage obtenu est un peu moindre.

Lorsque l'on connaît précisément le nombre de composantes connexes que comprend la surface étudiée, et que l'on dispose d'une graine sur chacune de ces composantes (un cube intersectant chaque composante), il est aisé de reconstruire la surface simplement en parcourant le maillage de proche en proche, sans avoir à explorer l'intégralité de l'espace. Quelques difficultés peuvent apparaître lorsque l'on se penche sur le cas de surfaces quelque peu exotiques (en particulier, les surfaces ne correspondant pas à un volume, et qui présentent des contours, des intersections ou des branchements), mais J. Bloomenthal et K. Fergusson ont montré, à partir de considérations sur la théorie des bifurcations et des catastrophes, que ces difficultés pouvaient être circonscrites [BF95].

C'est cette approche que nous avons retenue pour le rendu de notre surface. Son seul inconvénient réside dans le fait que, ne connaissant pas la topologie exacte de notre surface (qui peut changer au cours de la modélisation), il nous est nécessaire de parcourir l'ensemble de l'espace de travail pour s'assurer que l'intégralité de l'isosurface a bien été polygonisée. Heureusement, ainsi que le soulignent T. A. Galyean et J. F. Hughes dans [GH91], il est possible d'adopter une démarche incrémentale, et de ne mettre à jour que les cubes où le champ scalaire définissant l'isosurface a changé. Dans notre cas, il nous suffit de mémoriser dans une pile l'ensemble des cellules dans lesquelles on retire ou on ajoute de la matière, ainsi que le propose Eric Ferlay dans [Fer02], et lorsque vient le rendu, de ne mettre à jour que les triangles dans les cubes jouxtant ces cellules.

Mise à jour par particules flottantes

Lorsque la forme de l'isosurface évolue doucement et continûment dans le temps, on peut envisager d'autres approches que celle proposée par T. A. Galyean et J. F. Hughes. Plutôt que d'essayer de reconstruire la surface dans tous les cubes élémentaires où les valeurs prises par le champ ont pu évoluer, on peut essayer de maintenir la polygonisation initiale, et la faire évoluer au cours du temps en déplaçant les nœuds du maillage.

Par exemple, si le champ scalaire augmente au niveau d'un des nœuds du maillage, c'est que ce nœud est à présent à l'intérieur du volume, suite à un gonflement ou un déplacement de celui-ci. Pour mettre à jour le maillage, il suffira de déplacer le point en question vers l'extérieur, jusqu'en un point où le champ scalaire passe par notre isovaleur. Cette approche a par exemple été utilisée par Matthieu Desbrun dans [DC95]. Le déplacement peut se faire principalement le long de la normale de la surface au point considéré, ou selon le gradient du champ scalaire, qui peut parfois également fournir une estimation de l'amplitude du déplacement nécessaire. Inversement, si le champ scalaire diminue au niveau d'un point du maillage, ce

dernier devra reculer pour retrouver l'isosurface. Tout se passe comme si les nœuds du maillage "flottaient" sur l'isosurface.

L'une des difficultés de cette approche consiste à traiter les problèmes soulevés par les différents changements de topologie de la surface considérée. Pour éviter cet écueil, A. Witkin et P. Heckbert proposent initialement d'appliquer cette approche en utilisant simplement un rendu par particules de la surface, particules qui ne s'efforcent pas de maintenir de relations de voisinage particulières [WH94]. En revanche, pour assurer un bon échantillonnage de la surface, ils proposent que les différentes particules se repoussent deux à deux. Ainsi, les particules vont chercher à occuper tout l'espace disponible sur l'isosurface. Si, à la suite d'une modification de la surface, cet échantillonnage est insuffisant à un endroit, les particules étant trop éloignées les unes des autres, il est aisé de réensemencer la surface. À l'inverse, on pourra retirer des particules des endroits où elles sont en surnombre.

Cette approche a été notablement améliorée par B.T. Stander and J.C. Hart dans [SH97] qui ont montré qu'il était possible de maintenir une triangulation malgré les changements de topologie, en détectant de façon astucieuse ces derniers. Leur démarche se base sur une recherche des points critiques du champ scalaire. En fonction des signes des valeurs propres de sa hessienne, qui déterminent la forme de la singularité, ils examinent les différents cas pouvant se produire.

Si par exemple un point critique a ses trois valeurs propres négatives, il dénote un maximum local pour le champ. Lorsque ce maximum passe en dessous de l'isovaleur, c'est qu'une composante connexe de l'objet située autour de ce point a disparu (il suffit de lancer un simple rayon dans une direction quelconque pour la déterminer et la retirer). À l'inverse, s'il passe au-dessus de l'isovaleur, il faut créer une nouvelle composante à la surface. Un point "selle" (dont les valeurs propres sont de signes distincts) traitera des problèmes de cassures et de recollement (une seule valeur propre positive) ou de l'apparition et de la disparition de trous (deux valeurs propres positives). Les minima locaux, avec trois valeurs propres positives, gèrent la création et la destruction de cavités dans le volume.

L'approche nous a paru très intéressante, mais semble ne pas réellement offrir, dans notre cas, de meilleures performances que la démarche des *incremental marching cubes*.

Autres approches

De nombreuses autres méthodes ont été proposées. D'aucunes suggèrent l'utilisation d'un rendu volumique, des approches présentées par D. Mahoney dans [Mah97]. Ces méthodes ont généralement beaucoup profité, ces dernières années, d'améliorations et de nouvelles possibilités proposées par les cartes graphiques. On citera par exemple les textures-3D, que R. Westermann et T. Ertl suggèrent d'employer pour accélérer le rendu volumique d'une isosurface [WE98].

Dans le cas particulier de surfaces implicites à squelette, M. Desbrun proposait quant à lui dans [DTC95] d'essayer de localiser la surface en lançant des rayons à partir des éléments du squelette. L'isosurface se situe en effet dans un voisinage immédiat et quantifiable des primitives constituant le squelette. Lorsque nous travaillions avec un réseau masses-ressorts pour calculer les déformations du modèle, nous avons envisagé la possibilité d'utiliser ce même

réseau comme un squelette à partir duquel nous pourrions identifier la surface. L'idée étant de lancer des rayons à partir des masses du réseau, dans un ensemble de directions.

La tâche s'est cependant révélée bien plus ardue, principalement à cause du fait que, contrairement au cas des squelettes, il peut arriver que certains des nœuds se situent à l'*extérieur* du volume, auquel cas il faut alors rechercher la surface dans la direction opposée. En outre, les réseaux de déformation présentaient parfois des topologies exotiques (tantôt un volume, tantôt une surface, parfois encore une simple chaîne), qui démultipliaient de façon considérable le nombre de cas à envisager. Bien que cette solution soit *a priori* utilisable, quoique très délicate, nous l'avons délaissée en même temps que les réseaux de déformation.

4.6.2 Une méthode de tracé rapide

Motivation

La méthode de mise à jour que nous utilisons est à n'en pas douter rapide. Cependant, lors d'une déformation de l'objet, plusieurs milliers de cubes élémentaires peuvent nécessiter une mise à jour, à chaque pas de simulation. Si l'on souhaite assurer à l'utilisateur une manipulation interactive (qui requiert plus d'une dizaine de pas de simulation par seconde, voire bien davantage), en laissant un maximum de puissance de calcul à disposition des différents modèles de simulation, on ne dispose guère que de deux ou trois millisecondes pour chacune des mises à jour. Il pourrait arriver que cela devienne insuffisant, et pour ne pas pénaliser la simulation, nous avons également mis au point un rendu rapide (quoique incomplet) de l'objet, dans les moments où les différentes déformations imposent de trop nombreuses mises à jour.

Idée générale

Notre approche se base sur un lancer de rayons, dirigés le long d'un des axes principaux du maillage. Pour que l'illusion soit suffisante, l'axe de la caméra doit être aussi proche que possible de la direction choisie. Les différents rayons sont lancés avec le même pas que celui de la grille contenant la matière, et alignés sur le centre des cellules. Seule la première intersection du rayon avec l'isosurface sera considérée. Les volumes apparaissent alors comme s'ils étaient éclairés, par un côté, et l'arrière de ces derniers, dans l'ombre, n'est pas visible. Si une partie du volume se trouve derrière une autre, dans son "ombre", là encore, elle ne sera pas affichée.

Cette approche peut être réellement rapide, car elle se prête à de très nombreuses optimisations. L'intersection entre le rayon et l'isosurface consiste simplement à chercher la première cellule de la rangée qui contient plus de matière que l'isovaleur choisie. Elle se trouve nécessairement entre la cellule en cours de test et la cellule qui la précède dans la rangée (et donc plus proche de la caméra). Son positionnement exact est alors une simple interpolation des valeurs contenues dans chacune des deux cellules. En outre, cette recherche de cellule dans la rangée n'a en général pas à être conduite sur toute la profondeur de l'espace de travail. En effet, elle peut être dans une large mesure préparée durant la simulation même des déformations.

Mise en œuvre pratique

Pour chaque rangée, on conserve donc les cellules extrêmes c_{near} et c_{far} entre lesquelles la recherche devra s'effectuer. A l'origine, ces cellules sont simplement respectivement la première et la dernière de la rangée. Lorsque le contenu d'une cellule c_i de la rangée dépasse l'isovaleur, suite à un déplacement de matière, on impose $c_{far} = c_i$. Si la cellule c_{far} se trouve être vidée lors de la simulation, on choisit à nouveau la dernière cellule de la rangée pour c_{far} . Bien évidemment, à tout moment, on maintient la condition $c_{near} \leq c_{far}$.

Lorsque vient le moment du rendu, seules les cellules situées entre c_{near} et c_{far} seront examinées. Lorsque l'intersection est trouvée dans la cellule c_i , les deux bornes sont mises à jour par $c_{near} = c_{far} = c_i$. Ces bornes sont conservées d'un pas de simulation sur l'autre, de sorte que les recherches ultérieures seront améliorées.

Cette approche est efficace principalement grâce au fait que la recherche aura, la plupart du temps, à être conduite sur un nombre très restreint de cellules. Si la surface est immobile, ou s'approche de la caméra, l'intersection est déjà connue sans qu'il soit besoin de la chercher. Si elle s'en éloigne, la zone de recherche sera restreinte à la zone de recul. Le seul moment où la zone de recherche pourra être étendue sera celui où le déplacement latéral d'une partie de la sculpture laisse découvrir ce qui se trouve derrière. Cela ne concerne cependant qu'un nombre limité de rayons.

Résultats et améliorations possibles

La figure 4.11 présente le résultat obtenu par cette méthode de rendu rapide sur un objet constitué de deux sphères. Il est relativement convaincant lorsque la caméra est convenablement placée, mais bien évidemment, l'illusion est perdue si l'on s'éloigne trop de l'axe choisi pour le lancer de rayons. L'utilisateur ne tourne pas la caméra lorsqu'il est en train de déformer l'objet : il alterne généralement des étapes de positionnement de la caméra suivant un angle lui permettant d'avoir une bonne vision de la zone qu'il souhaite modifier, avec des étapes d'édition proprement dite de la forme de l'objet. Nous pouvons donc choisir le meilleur point de vue (l'axe le plus proche de celui de la caméra) avant l'édition. Les points de vue proches des axes donnent bien évidemment les meilleurs résultats.

Pour améliorer le résultat, on peut lancer à faible coût une série de rayons supplémentaires là où les polygones sont trop étirés, ce qui dénote probablement une fusion parasite de deux objets distincts, ou de deux parties distinctes d'un même objet. Il suffit alors d'éliminer les polygones restant étirés pour obtenir un rendu acceptable.

Ce rendu a bien évidemment pour vocation d'être utilisé seulement lorsque la quantité de calculs nécessaire à la simulation met en danger l'interactivité du système. Dès que l'on dispose à nouveau de suffisamment de temps de calcul, nous reviendrons à la méthode des *marching cubes* itératifs pour obtenir une visualisation de meilleure qualité du volume.

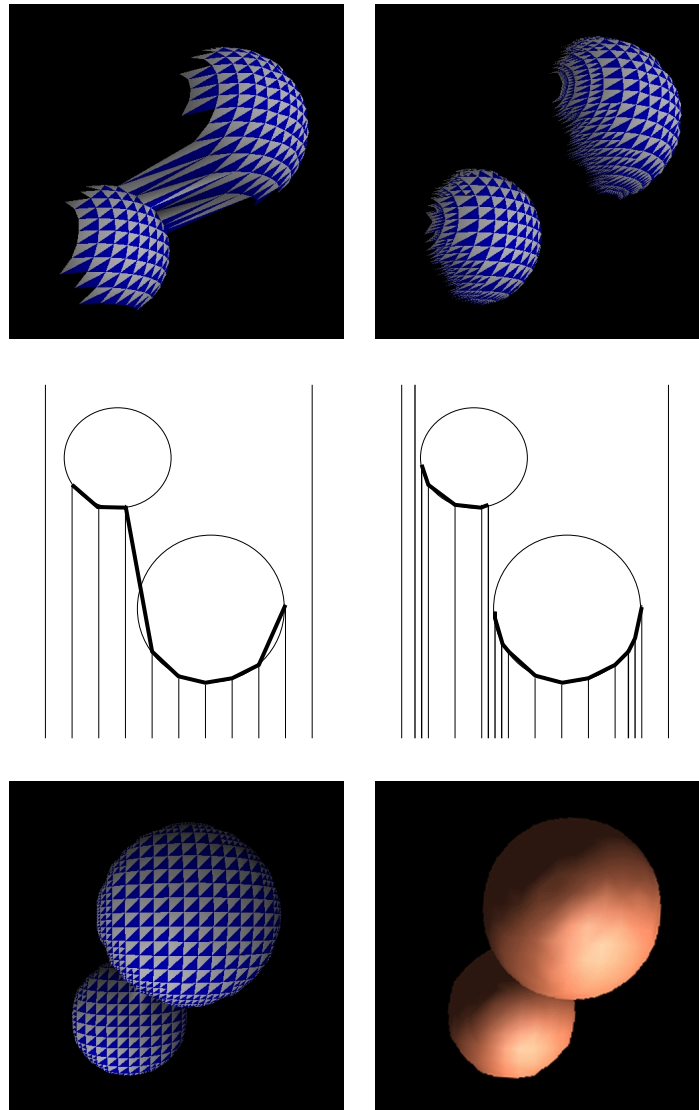


FIG. 4.11 – Exemple de résultat obtenu par la méthode de rendu rapide. Les deux images du haut montrent l'objet selon une vue latérale afin d'illustrer le principe. Les défauts visibles en haut à gauche sont éliminés par le lancer de rayons supplémentaires. Les images du bas montrent le résultat final selon la perspective correcte.

Deuxième partie

Capture de gestes par vidéo

Chapitre 5

La capture de gestes

5.1 Introduction

Nous allons à présent nous pencher sur le problème de la réalisation d'un capteur de mouvements de la main, qui permettrait entre autres choses d'interagir avec l'argile virtuelle que nous venons de présenter. Ce premier chapitre a pour but de présenter les travaux existant à l'heure actuelle. Afin de mieux cerner les avantages que présenterait une interface fondée sur la vision pour interagir avec un ordinateur, il est utile d'examiner les autres interfaces disponibles à ce jour.

La souris, créée dans les laboratoires de Xerox il y a un demi-siècle, est assurément l'interface la plus couramment rencontrée aujourd'hui. C'est cependant, avec seulement deux degrés de liberté, un moyen très inefficace pour interagir avec un monde virtuel à trois dimensions. C'est aussi le cas des trackballs et des tablettes graphiques, lesquelles sont éventuellement en mesure de mesurer la position 2D, mais aussi l'inclinaison d'un stylet. Pour interagir avec des mondes tridimensionnels, des *souris 3D* telles que la *space mouse* existent. Elles permettent de contrôler librement, par exemple, le déplacement d'un objet rigide dans l'espace, en mesurant à la fois un déplacement sur chacun des trois axes et trois rotations autour de ces mêmes axes.



FIG. 5.1 – À gauche, les Space Mouse et Space Ball proposés par 3D Immersion. À droite, des dispositifs capable de détecter leur orientation dans l'espace.

Des *bras articulés* proposent également de mesurer la position et l'orientation spatiale d'un objet, tel qu'un stylet, que tient l'utilisateur. Comme les *space mouses*, cela permet

donc d'obtenir six degrés de liberté, mais de façon un peu plus naturelle que ne le proposent les souris, puisque l'on contrôle directement la position du stylet dans l'espace. Par ailleurs, beaucoup de ces bras articulés sont motorisés, permettant ainsi un retour d'effort. L'utilisateur peut ainsi avoir une notion du toucher à travers l'interface¹. Le plus populaire des ces bras articulés est le *PHANTOM* de la société *Sensible*, qui par ailleurs propose un modèle pourvu d'une sorte de dé à coudre, en lieu et place du stylet, pour y placer l'extrémité du doigt. On peut ainsi, en combinant plusieurs de ces bras, suivre le mouvement spatial de l'extrémité de plusieurs doigts, avec toutefois des limitations sur les mouvements possibles dûs justement à la présence des bras.

Il existe également divers capteurs de position, qui permettent d'avoir des informations sur la position et/ou l'orientation d'un objet dans l'espace. Certaines sont disponibles pour le grand public, comme la souris gyroscopique de la société *Gyration*, qui est en mesure d'estimer son orientation spatiale au moyen de gyroscopes. De nombreuses technologies existent, basées sur des capteurs magnétiques, optiques, ou encore sonores, avec diverses contraintes quant à l'environnement et la calibration. La *motion capture* utilise généralement de tels capteurs. Ce sont parfois de simples capteurs passifs, comme des marqueurs, dont la position est déterminée de façon externe.

L'inconvénient commun à toutes ces approches est d'apporter une information relativement pauvre, en mesurant au plus 6 degrés de liberté, alors que l'on est en mesure d'en utiliser bien davantage avec la seule main. Une solution consisterait à multiplier les capteurs de position dont nous venons de parler (et c'est effectivement ce qui est fait dans le cadre de la *motion capture*), mais si c'est aisé dans le cas des bras et des jambes, c'est plus difficile (quoique possible) dans le cas de la main. Dans le cadre d'une interface, on préférerait sans doute avoir à bouger une main en face de son ordinateur plutôt que d'avoir à faire des mouvements des bras. La main propose par ailleurs une information bien plus riche que ne le pourrait le bras, particulièrement utile dans bien des applications comme la sculpture virtuelle.



FIG. 5.2 – À gauche, deux versions du bras articulé *PHANTOM*. À droite, des gants de données, dont le *Data Glove*.

Les gants de données, dont le *Data Glove* est le premier représentant, sont un moyen intéressant de récupérer cette information de configuration de la main. Ils se présentent comme de simples gants à enfiler, munis d'un appareillage permettant de mesurer une partie des informations articulaires des doigts. Par ailleurs, ils sont parfois motorisés afin de fournir un retour haptique à leur utilisateur. Le principal reproche que l'on peut faire à ces systèmes, qui présentent l'avantage d'être fiables, est d'imposer à l'utilisateur de porter quelque chose qui pourrait le gêner dans ses mouvements.

Un capteur basé sur la vision permettrait, à condition d'être suffisamment robuste, une

¹Précisons qu'il existe également quelques souris pourvues de retour d'effort.

interaction plus libre que ne le permettent ces solutions. C'est une solution intéressante au problème de l'interaction avec un monde virtuel. Avec un tel capteur, un artiste pourrait ainsi utiliser ses gestes habituels pour modeler un matériau virtuel, se servant de toute la surface de sa main. Dans la suite, nous allons nous pencher plus précisément sur ce problème.

5.2 Suivi de mouvements par vidéo

5.2.1 Introduction

La capture de mouvement par vidéo est un problème qui a été largement étudié, et qui continue à l'être aujourd'hui. En effet, compte tenu de sa difficulté, il n'a pas encore été proposé de solution définitive à ce problème. Dans bien des cas, des hypothèses supplémentaires, concernant l'éclairage ou encore le fond de l'image, ont dû être faites pour permettre un suivi dans de bonnes conditions.

Divers artifices peuvent venir faciliter la détection des mouvements. On peut porter des vêtements particuliers, des gants par exemple dans le cas de la main. E. Holden suggère l'usage de gants où les articulations sont matérialisées par des couleurs pour les situer plus aisément dans l'image [HOR99]. Il est aussi très fréquent, dans le domaine de la *motion capture*, d'utiliser des objets aisément identifiables, généralement des sphères, liées à l'objet à suivre. Cette approche a également été utilisée pour suivre le mouvement d'un doigt [DUS01]. Le suivi, assisté par de tels éléments, est aujourd'hui suffisamment fiable pour être largement utilisé dans des applications professionnelles.

Pour notre part, nous nous intéresserons plutôt au cas du suivi de mouvement n'utilisant pas de marqueurs. De nombreuses approches ont été présentées pour résoudre ce problème. Chacune d'entre elles présente des avantages et des inconvénients distincts, sur lesquels nous allons tour à tour revenir dans la suite de ce chapitre.

Il est d'usage de distinguer deux façon d'approcher le problème, ainsi que le propose V. Pavlović dans [PSH97]. Une première série de méthodes utilise un modèle de l'objet à suivre, fonction d'un certain nombre de paramètres, et s'efforce de déterminer les différents paramètres qui permettront de placer le modèle dans une configuration aussi proche que possible de celle observée. Une seconde approche repose sur un ensemble de vues de l'objet à suivre, chacune préalablement étiquetée, et essaie de déterminer la vue parmi cet ensemble qui s'apparente le plus à celle actuellement observée. La détermination de la position et de la configuration est alors essentiellement une question de classification.

Bien que cette distinction soit de moins en moins nette à l'heure actuelle, ces deux approches présentent toutefois encore des forces et des faiblesses bien différentes, et nous avons donc décidé de conserver cette distinction dans notre tour d'horizon des méthodes de suivi. Certaines approches se contentent d'estimer la position, et éventuellement l'orientation, de la main dans une image, sans se préoccuper de la troisième dimension. Nous en dirons également un mot.

Le suivi de mouvement, on l'a dit, est un problème vaste. Nous nous intéresserons dans la suite au cas particulier du suivi de la main, à travers une série d'articles qui nous ont

semblé intéressants et représentatifs. Une étude plus exhaustive des différentes méthodes de détection, de modélisation et de suivi des mouvements de la main peut être trouvée dans [WH01, PSH97].

Les méthodes employées pour le suivi de la main sont souvent similaires, pour l'essentiel, à celles que l'on trouve dans d'autres problèmes de suivi de mouvement. Celui du corps humain dans son intégralité, par exemple, avec toutefois quelques spécificités. Nous aborderons brièvement ensuite le cas du suivi du corps humain, car les similarités sont suffisamment importantes pour que l'on puisse trouver de l'inspiration dans ces modèles.

5.3 Suivi de la main

5.3.1 Approches bidimensionnelles

La plupart des méthodes sur lesquelles nous allons nous pencher tentent de déterminer la position de la main dans l'espace. Dans bon nombre d'application, ces informations ne sont cependant pas utiles, et une simple estimation bidimensionnelle de la position de la main est suffisante. Les méthodes présentées ici se contentent d'estimer la position de la main dans l'image, éventuellement son orientation et sa distance à la caméra à travers sa taille apparente, et ne tiennent pas compte de la position précise des doigts ou des autres rotations. C'est évidemment un problème beaucoup plus simple, avec généralement seulement quatre degrés de liberté.

Le moyen le plus simple pour déterminer la position d'une main dans une image consiste probablement à rechercher les zones dans l'image dont la couleur correspond effectivement à la couleur de la peau. Cette couleur est très caractéristique, et de façon remarquable, elle dépend assez peu de la pigmentation de la peau. La taille et la forme des zones ainsi identifiées peut ainsi servir de méthode basique pour le suivi de la main. Cette approche peut également servir d'initialisation à des méthodes de suivi plus élaborées [IB98].

Le suivi du mouvement de la main dans une séquence d'images peut ensuite être réalisé par le suivi de ses contours. J. Mac Cormick propose par exemple dans [MI00] de modéliser ces contours au moyen de courbes de Bezier, et d'utiliser un filtre à particules pour mettre à jour la position de la main au cours du temps. Ces méthodes donnent généralement des résultats satisfaisants, tant que les mouvements de la main ne sont pas trop rapides.

L'idée du filtre à particules pour le suivi de la main est reprise par Bretzner *et al* dans [BLL02]. Ils proposent de rechercher dans l'image des "blobs" de couleur correspondant à la paume et aux différents doigts. L'algorithme permet par ailleurs de distinguer cinq configurations différentes de la main, dans lesquelles les différents doigts sont pliés ou non.

Ces méthodes sont un moyen simple, intuitif et robuste de contrôler quelques actions de contrôle ou d'édition. Pour contrôler un outil de sculpture virtuelle tel que celui présenté précédemment, cependant, ces modèles sont insuffisants, les dispositifs de pointage classiques fournissant généralement autant d'informations. Il nous est nécessaire d'avoir la pleine connaissance de la position de la main dans l'espace, ce qui est difficile à obtenir sans un modèle ou une description plus précise de la main. Dans la section suivante, nous allons exa-

miner les diverses méthodes qui s'attaquent à ce problème, plus riche mais également plus difficile.

5.3.2 Approches basées sur un modèle

Principe

Dans ce genre d'approche, l'objet à suivre (dans notre cas, la main), est représenté par un modèle. Ce modèle comprend généralement une description cinématique de l'objet, qui permet d'exprimer sa configuration et sa forme au moyen d'un nombre limité de paramètres. D'autres approches sont cependant parfois employées de préférence à une description cinématique : T. Heap a par exemple préféré utiliser des PDM (*Point Distribution Models*) pour décrire la forme de la main dans [HH96]. Le mouvement des doigts y est exprimé sous la forme des cinq principaux modes de déformations du modèle, élaborés à partir d'un ensemble de données obtenu par résonance magnétique. Le mouvement réel des doigts n'étant pas linéaire, ce type de description donne cependant des résultats moins précis. Toujours est-il que le principe reste le même : la forme de l'objet est décrite au moyen d'un nombre restreint de valeurs.

Le modèle, dans une configuration, une position et une orientation donnés, est ensuite confronté aux mesures, par exemple à des images prises par une ou plusieurs caméras. La comparaison se fait généralement au moyen d'une fonctionnelle qui exprime les différences observées entre le modèle et les mesures sous la forme d'une erreur. De nombreuses expressions pour cette erreur ont été proposées, basées sur les informations qui peuvent être extraites de l'image : contours, silhouettes, formes caractéristiques ou bien encore cartes de profondeur sont quelques-unes des informations qui peuvent être utilisées. Parfois, la combinaison de plusieurs éléments permet d'augmenter la robustesse du suivi.

Le calcul de l'erreur permet de mettre à jour les différents paramètres du modèle, ce qui permet d'obtenir une nouvelle configuration du modèle que l'on peut confronter aux mesures. Le processus est ainsi répété jusqu'à converger une position qui minimisera les différences entre les observations et le modèle.

Silhouettes et contours

Ce sont probablement les contours qui sont les éléments les plus utilisés pour comparer le modèle aux données. Les contours présente l'avantage d'être relativement faciles à extraire d'une image, même si les cartes de contour présentent l'inconvénient notable d'être relativement sensibles au bruit. *Digiteyes*, un des tous premiers systèmes capable de suivre le mouvement la main proposé par J. Rehg dans [RK94], utilise ainsi les contours pour estimer l'erreur en position du modèle. Celle-ci s'exprime sous la forme d'une somme des carrés des distances entre les contours du modèle projetés dans l'image et les contours effectivement extraits de l'image. Cette erreur est alors minimisée grâce à un algorithme de type Gauss-Newton.

La détection des extrémités des doigts vient augmenter la robustesse de ce modèle. Il ne permet cependant pas de traiter le cas des occlusions éventuelles, et suppose que le fond est neutre éviter de détecter des contours qui ne correspondent pas à la main. Le problème des

occlusions a trouvé une ébauche de solution dans [RK95]. Des masques liés aux différents éléments du modèle permettent l'élimination des contours cachés du modèle lors de la projection.

De nombreux autres systèmes utilisent les contours pour estimer l'erreur en position du modèle [HH96, WH99, WLH01]. Les contours sont généralement considérés comme les lieux dans l'image où l'intensité varie brusquement spatialement. On peut aussi parfois considérer les endroits où la couleur varie brusquement, comme le propose J. Mac Cormick dans [MI00].

Les silhouettes sont également fréquemment utilisées pour comparer la position du modèle à celle de l'objet dans les images. Les silhouettes sont généralement extraites dans les images grâce à des critères de couleur. L'erreur s'exprime alors en fonction du recouvrement entre la silhouette extraite des images et celle calculée à partir de la position du modèle. L'inconvénient des silhouettes par rapport aux contours réside dans le fait qu'elles contiennent moins d'information, même si elle est plus robuste. Silhouettes et contours sont souvent utilisés de façon conjointe, comme le propose Y. Wu dans [WLH01].

Identification de zones d'intérêt

Dans le cas du suivi du corps humain, on utilise parfois des détecteurs pour localiser dans les images quelques éléments intéressants : le visage, les mains et les pieds, etc. Il n'est, en revanche, généralement pas aisé d'identifier des formes caractéristiques de certaines parties de la main. N. Shimada *et al* propose cependant d'essayer de localiser les extrémités des différents doigts afin de guider le suivi [SS96].

Utilisation de plusieurs points de vue

Le suivi des mouvements de la main au moyen d'une seule vue de celle-ci est un problème impossible. De nombreuses ambiguïtés, quant à la position des doigts notamment, ne peuvent être levées en ne se basant que sur une seule image. Les nombreuses occlusions qui ne manquent pas de survenir dans le cadre du suivi des mouvements de la main viennent s'ajouter à ces ambiguïtés. La connaissance de la position à l'instant précédent peut parfois aider à lever certaines de ces ambiguïtés, mais bien souvent, cela ne suffit pas.

Pour cette raison, de nombreuses méthodes utilisent plusieurs points de vue afin d'être en mesure de lever les inévitables ambiguïtés. On observe deux types d'approches pour gérer ces points de vue supplémentaires. La minimisation entre prédiction et observations est conduite simultanément dans l'ensemble des images. Ou bien, à l'inverse, on peut choisir l'image, parmi celles disponibles, qui présente le moins d'occlusions et semble la plus appropriée à l'estimation de la configuration de la main.

De façon surprenante, il est relativement rare que ces vues supplémentaires soient utilisées comme un moyen d'obtenir des informations tridimensionnelles sur la scène. Ce sont les projections du modèle dans les différentes images qui sont utilisées pour juger de la justesse de la position. Q. Delamarre, au contraire, utilise une paire d'images (obtenue en pratique au moyen d'une seule caméra et d'un jeu de miroirs) pour reconstruire effectivement, par

stéréoscopie, une carte de profondeur de la scène [DF98]. L'ajustement s'efforce alors de faire directement coïncider la position du modèle dans l'espace avec la carte de profondeur.

Réduction du nombre de paramètres du modèle

L'une des difficultés posées par le suivi de la main réside dans le grand nombre de degrés de liberté présents, problème accru par ailleurs de par la présence de nombreuses et régulières occlusions, ces libertés étant rassemblées dans un très petit volume. Les modèles articulaires généralement adoptés pour le suivi de la main présentent ainsi de vingt-six ou vingt-sept degrés de liberté : chacun des doigts dispose de quatre libertés en rotation, le pouce permet quatre ou cinq rotations supplémentaires selon les modèles, auxquelles viennent s'ajouter les six paramètres positionnant la paume dans l'espace. Nous reviendrons plus en détail sur la cinématique de la main et sur ces modèles dans le chapitre suivant.

Il a donc naturellement été suggéré de restreindre le nombre de ces degrés de liberté, en s'appuyant sur quelques considérations anatomiques de la main. Il n'est en effet pas possible d'avoir le plein contrôle de ces différentes libertés : les muscles, par l'intermédiaire des tendons, ne peuvent contrôler indépendamment chacune des articulations, et beaucoup de mouvements articulaires sont généralement corrélés. Il est par exemple impossible de plier la troisième phalange de l'index sans que la seconde ne bouge, et si l'inverse est possible, c'est un mouvement conscient qui n'apparaîtra guère naturellement.

Ces corrélations peuvent être directement explicitées sous la forme d'un couplage des paramètres, par exemple en liant les angles des articulations entre phalanges par une relation linéaire. Une approche plus systématique visant à diminuer le nombre de paramètres à estimer consiste à acquérir une grande quantité de mouvements de la main (par exemple au moyen d'un gant) et à réduire le nombre de degrés de liberté par exemple au moyen d'une analyse en composantes principales de ces mouvements [WLH01].

Deux choses sont à considérer à propos de ces simplifications articulaires. D'une part, il est toujours possible de les utiliser avec n'importe quelle méthode de suivi de la main basée sur un modèle comprenant un grand nombre de degrés de liberté. Soit pour diminuer le nombre de paramètres à estimer, soit, dans un premier temps, pour obtenir une première bonne approximation du mouvement, après quoi il est possible d'estimer plus précisément chacun des paramètres du modèle complet. D'autre part, ces simplifications sont généralement valables uniquement pour des mouvements naturels, lorsque les doigts sont peu repliés, et surtout ne sont vérifiées que lorsque la main n'est en contact avec aucun objet. Lorsque des efforts extérieurs interviennent, il est en effet courant que ces relations ne soient plus vérifiées.

Pour ces raisons, nous nous attacherons dans la suite à décrire un modèle complet, qui conviendra tout autant pour suivre un mouvement simple de la main qu'un geste d'un sculpteur qui exerce des efforts sur un matériau, en gardant à l'esprit que dans bon nombre d'applications, ces simplifications pourront éventuellement venir améliorer la robustesse et la vitesse du suivi.

Utilisation de la dynamique pour le suivi

La première estimation est, nous l'avons dit, souvent cruciale pour la robustesse de tels modèles. Divers modèles dynamiques de mouvement de la main ont donc été suggérés pour permettre une première estimation de la position. Le filtrage de Kalman est une solution régulièrement choisie pour ce faire. Les modèles les plus simples font l'hypothèse de vitesses constantes [SS96]. H. Zhu et T. S. Huang ont proposé un modèle dynamique plus élaboré, dans lequel chaque doigt est modélisé par un système dynamique travaillant avec un nombre restreint de paramètres [ZH03].

Cette fois encore, le modèle dynamique vient en support d'une méthode de suivi, fournissant généralement la première approximation des transformations survenues entre un instant et le suivant. Notons également que c'est à nouveau essentiellement valable pour une main isolée : les éventuels chocs de la main sur un objet quelconque rendront en effet ces estimations caduques. C. Tomasi note également dans [TPS03] que les mouvements de la main sont généralement trop vifs pour que les modèles dynamiques puissent être utilisés de façon efficace. Un mouvement vif de la main peut en effet conduire d'une main fermée à une main complètement ouverte en l'espace de quelques images seulement, rendant tout suivi très délicat.

Estimation du mouvement pour l'aide au suivi

L'ensemble des modèles présentés tantôt s'efforcent d'estimer directement la position de l'objet, à travers l'ajustement d'un modèle. Ces approches sont généralement la proie de nombreuses ambiguïtés. En effet, la plupart des éléments extraits des images (textures, contours...) sont très similaires d'un doigt à l'autre. En outre, la forme cylindrique des doigts rend le processus sensible à un éventuel "glissement" du modèle. Pour contourner ces difficultés, on peut s'intéresser au mouvement de l'objet entre les deux images, de façon à guider le modèle dans la bonne direction. Cette approche mixte, combinant une estimation du mouvement et un ajustement du modèle, a été proposée par K. Nirei dans [NSMO96], et plus récemment reprise par S. Lu dans [LMSO03]. Le flot optique mesuré dans chacune des images oriente le modèle dans la bonne direction, de façon à faciliter l'estimation de la position de la main.

Difficultés et limitations

Le principal écueil que doivent affronter les modèles basés sur un modèle qui minimisent une fonctionnelle réside dans la présence de nombreux minima locaux. Pour cette raison, il est généralement supposé que l'on connaisse une bonne estimation de la position de la main à l'instant précédent, et que l'on cherche la nouvelle position dans un voisinage immédiat. Le problème de l'initialisation se pose donc, d'abord au début de la séquence, puis à chaque fois que la méthode perd la trace de la main. C'est un problème à part entière, dont une ébauche de solution se trouve probablement dans les méthodes de positionnement par classification que nous allons examiner à présent.

5.3.3 Approches par classification

Introduction

Plutôt que d'ajuster les paramètres d'un modèle de façon à faire coïncider son apparence avec celle de l'objet, il est possible de rechercher dans une base d'images préalablement étiquetées celle qui se rapproche le plus de la vue actuelle de l'objet. Dès que l'on a identifié la position, l'orientation et éventuellement la taille de la main dans l'image, on est alors ramené à un problème de classification. Lorsque plusieurs angles de vue distincts de la main sont disponibles, K. Nirei propose par ailleurs de choisir celui dans lequel la classification sera la plus aisée, réduisant de ce fait les difficultés dues aux éventuelles occlusions [UO99].

L'étiquetage peut représenter un obstacle important, si l'on veut disposer d'une grande quantité d'images dans la base de référence. Y. Wu propose d'utiliser l'algorithme EM pour restreindre ce travail d'étiquetage, et de travailler avec la combinaison d'un petit groupes d'images étiquetées, représentant quatorze configurations distinctes, et d'un groupe plus large d'images qui ne le sont pas [WH00].

Aujourd'hui, la frontière entre les approches basées sur un modèle et celles basées sur une collection de vues tend à disparaître. Plutôt que d'étiqueter une série d'images prises par exemple par des caméra, certains auteurs préfèrent générer une grande base d'images obtenues au moyen d'un modèle de main semblable à ceux utilisés par les approches dites "*basées modèle*". L'étiquetage peut être fait de façon entièrement automatique avec les paramètres du modèle.

Principe

L'estimation de la pose de la main est généralement obtenue en plusieurs étapes. Dans un premier temps, l'image est segmentée de façon à pouvoir retrouver la main (voire éventuellement plusieurs mains) dans les images. Cette segmentation de l'image constitue un problème complexe, et qui n'est pas entièrement résolu. Une approche souvent choisie consiste à utiliser la couleur pour segmenter la main dans les images prises par les caméras, la couleur de la peau étant nettement caractéristique. L'utilisation de lumière contrôlée, ou infrarouge comme le propose Shimada dans [SKS01], vient aider à la détection, de même que parfois la présence d'un vêtement au niveau du poignet pour bien séparer la main du reste du bras [LF02]. Une autre approche populaire tente d'éliminer ce qui constitue le fond de l'image, généralement à travers des méthodes d'apprentissage du fond (qui doit alors rester figé). En pratique, cependant, il n'existe guère de méthode à la fois fiable et générale de segmentation.

Une fois la main extraite des images, il faut ensuite la comparer aux différents exemples présents dans la base de donnée. Un certain nombre de caractéristiques à même de décrire la forme de la main dans l'image sont déterminées, de façon à pouvoir conduire la recherche dans la base. Divers classifieurs sont utilisées pour ce faire, par exemple des techniques de recherche du plus proche voisin.

Parmi les méthodes utilisant une base d'image étiquetée, celle proposée par Y. Cui détecte dans l'image les zones en mouvement, afin d'éliminer le fond (dont on suppose qu'il est immo-

bile), qu'il convertit en fenêtres qui serviront de segmentation. La classification proprement dite, parmi vingt-huit configurations possibles, est effectuée grâce à un arbre de décision qui vient comparer le contenu de ces fenêtres aux exemples présents dans la base de données [CW96, CW00]. J. Triesch utilise quant à lui des comparaisons élastiques de graphes pour reconnaître douze signes différents, sur un fond qui peut être encombré [TdM01]. R. Lockton utilise dans [LF02] un algorithme de boosting pour augmenter la rapidité de reconnaissance d'une pose parmi quarante-six, obtenant un taux de reconnaissance remarquable de 99.87% d'après les auteurs..

Lorsque la base est construite grâce à un modèle, un plus grand nombre de configurations peuvent être distinguées, et il est plus difficile de trouver des descripteurs simples de la forme de la main permettant la classification. Les approches de Shimada *et al* [SKS01], et de Rosales *et al* [RASS01], comparent directement la silhouette extraite de l'image avec celles établies grâce au modèle.

Cependant, les silhouettes ne sont, comme nous l'avons dit plus tôt, guère fiables dans ce genre de situation, d'autant que de nombreuses occlusions peuvent survenir. Athistos propose dans [AAS01, AASK04] diverses autres méthodes pour comparer les images aux exemples de la base, dont beaucoup sont semblables à celles utilisées dans le domaine du suivi basé sur un modèle : moments principaux de la forme, histogrammes d'orientation des contours, distance de chanfrein (exprimant la distance entre les contours de l'image et du modèle), etc. Un critère global, regroupant ces différents éléments, est utilisé pour juger de la corrélation.

Une approche inadaptée au suivi

Cette fois encore, c'est le nombre de degrés de liberté qui va imposer des limitations à ces approches. Celles-ci se sont montrées efficaces pour classer une pose parmi quelques milliers de vues au plus dans la base de référence. Les travaux d'Athistos *et al* [AS03] utilisent 26 configurations de la main, observées selon 4128 angles de vue distincts, ceux de Shimada *et al* [SKS01] utilisent 125 poses différentes, réparties selon 128 orientations. Rosales *et al* vont jusqu'à considérer 2400 configurations distinctes dans [RASS01].

Cependant, à raison simplement de deux positions pour chacun des degrés de liberté de la main (qui en comporte, rappelons-le, une vingtaine), on obtient plus d'un million de configurations possibles. Ces diverses configurations doivent en plus être aperçues de divers points de vue tout autour de la main elle-même. Même en utilisant largement les différents couplages entre les articulations, il n'est pas possible d'espérer, pour l'instant, disposer une base assez vaste pour avoir une quelconque précision dans l'estimations des paramètres articulaires, d'autant que le temps de recherche du meilleur candidat dépend fortement de la taille de la base.

Ces méthodes sont donc adaptées à la reconnaissance d'un nombre limité de gestes et de poses, et peuvent par exemple être utilisées avec succès dans la reconnaissance du langage des signes. Elles ne résolvent en revanche pas correctement le problème du suivi. Une proposition a été faite par C. Tomasi pour tenter de contourner cette difficulté dans [TPS03]. Il propose d'identifier dans une séquence d'images quelques positions clés de la main, puis d'interpoler le mouvement entre ces positions de référence pour estimer le mouvement de l'image entre

ces dernières. Quoique cette approche ne constitue pas un réel suivi de la main, elle permet de partiellement résoudre la question de la non-exhaustivité de la base de référence.

Par ailleurs, il est intéressant de noter que, outre la génération de la base, il est un autre point sur lesquelles ces méthodes se rapprochent des méthodes basées sur un modèle. De plus en plus fréquemment, le caractère temporel est pris en compte, dans le but de fiabiliser la reconnaissance. Ainsi, T. Starner préfère utiliser une reconnaissance grossière de la forme, de la position et de l'orientation de la main et utilise des modèles de type chaînes de Markov pour reconnaître un vocabulaire d'une quarantaine de mots [SWP98].

La configuration de la main à l'instant précédent est aussi régulièrement utilisée pour guider la reconnaissance. Plutôt que de parcourir l'ensemble de la base de données, la recherche est effectuée dans le voisinage de la dernière position connue. Lorsque les données concernent plusieurs milliers d'images, cela permet d'augmenter tout à la fois la vitesse de traitement et la robustesse du suivi, puisqu'il n'est plus possible de "sauter" d'une configuration de poing fermé à une main complètement ouverte. Shimada garde ainsi plusieurs hypothèses quant à la position courante de la main, et restreint ses recherches dans la base au voisinage de ces mêmes hypothèses [SKS01].

5.3.4 Bilan sur le suivi de la main

Le grand nombre de méthodes et d'approches existantes pour le problème du suivi de la main démontre bien, s'il le fallait, l'intérêt qu'on lui accorde. La quasi-totalité des méthodes travaillent avec une seule image. Cela ne permet pas de résoudre un nombre important d'ambiguïtés parmi les configurations possibles. Il nous a donc paru intéressant de s'intéresser au cas du suivi de la main au moyen de plusieurs caméras, afin de disposer de davantage d'information tridimensionnelle.

Certaines méthodes proposent par ailleurs de s'aider d'informations de mouvement (par exemple des flots optiques) pour "guider" le suivi dans la bonne direction. Là encore, nous pensons que cet aspect mérite que l'on s'y attarde. Le mouvement entre deux images peut être important, en particulier dans le cas de la main qui montre des déplacements rapides des doigts. Les méthodes usuelles de minimisation ne sont pas nécessairement adaptées à la recherche d'une solution qui peut être loin de l'estimation courante. L'usage de la dynamique est souvent suggérée pour contourner ce problème, mais les expériences ont montré que cela ne marchait pas très bien dans le cas de la main.

Ces deux directions (utilisation de la troisième dimension, et guidage pour le suivi à partir d'informations de mouvement) sont celles que nous explorerons dans les chapitres à venir, afin de proposer un système de suivi robuste de la main. Mais avant cela, nous allons faire un rapide tour d'horizon des méthodes utilisées dans le suivi de mouvements du corps humain, relativement proche dans ses objectifs du suivi de la main, afin de voir s'il n'est pas possible d'en tirer des enseignements utiles.

5.4 Suivi du corps humain

5.4.1 Introduction

Le suivi de mouvements du corps humain est un problème relativement proche du suivi de la main, même si nous verrons qu'il existe quelques différences qui méritent d'être soulignées. Il nous a donc semblé naturel d'examiner les différentes approches employées dans ce domaine, et de jauger de leur efficacité. Certaines d'entre elles pourraient vraisemblablement être appliquées au cas qui nous occupe, celui du suivi de la main.

Parmi les différences les plus évidentes, qu'il convient de garder à l'esprit, le suivi du corps humain a généralement à tenir compte de la présence d'habits. Ces derniers sont généralement supposés ajustés afin de ne pas compliquer le problème. Toutefois, il n'est plus possible, comme dans le cas de la main, de se baser sur la reconnaissance de la couleur de la peau pour localiser le corps humain dans l'image. Cela complique les méthodes qui supposent une extraction efficace du fond de l'image.

À l'inverse, on dispose de détecteurs fiables de visages, et, au moyen de la couleur, on peut espérer situer et suivre les mains et les pieds des personnes dont on espère mesurer les mouvements. Le modèle développé par C. Théobald repose sur ce principe, et estime une première estimation de la position du modèle à partir des positions dans l'espace du visage, des mains et des pieds [TMSS02]. Cette première estimation sera corrigée par la suite, grâce à d'autres informations, mais elle permet de rendre le suivi nettement plus robuste, puisqu'il est possible de rapidement réinitialiser le modèle.

D'autres différences seront soulignées dans la suite. Au chapitre des points communs, de même que dans le cas de la main, les occultations sont souvent un problème. La solution d'utiliser plusieurs caméras est notablement plus répandue dans le domaine du suivi du corps humain. On trouve ainsi des systèmes utilisant des paires stéréoscopiques de caméras (R. Plänkner et P. Fua dans [PF01b], D. Demirdjian et T. Darell dans [DD02]) ou bien encore de nombreuses caméras observant la scène selon des points de vue différents (C. R. Wren *et al* dans [WADP97] ou bien encore R. Fablet and M. J. Black dans [FB02] entre autres exemples).

Dans la suite, nous passerons en revue quelques-unes des méthodes de suivi du corps humain parmi les plus marquantes, en nous attachant surtout aux modèles basés sur l'utilisation d'un modèle, qui nous intéressent tout particulièrement pour la suite. Un tour d'horizon plus vaste des méthodes de suivi de mouvement a été proposé par T.B. Moeslund dans [MG01], ou peut être trouvé dans l'ouvrage de A. Menache [Men00].

Les modèles estimant directement la pose d'un être humain, par exemple au moyen d'une base de données, sont relativement rares. Dans le cas de la main, le langage des signes par exemple permettait de définir un nombre restreint de poses à reconnaître. Il semble ne pas y avoir d'équivalent dans le cadre du corps humain dans son intégralité, et il existe une trop grande variété de poses pour qu'il soit aisé de procéder de cette façon. On pourra toutefois citer les travaux de G. Mori and J. Malik qui tentent d'estimer la pose à partir d'une seule vue, en la confrontant à une base d'images de référence au moyen d'un descripteur de forme [MM02]. L'estimation est répétée pour chaque image de la séquence. G. Shakhnarovich propose dans [SVD03] un moyen de recherche efficace dans une vaste base d'images de références générée

au moyen d'un logiciel de synthèse d'images.

Quelques rares travaux proposent un suivi qui ne nécessite pas un modèle 3D du corps humain. On pourra citer par exemple *Pfinder* qui suit le mouvement d'une personne en s'efforçant de déterminer le mouvement, au moyen de flots optiques, de blobs dans l'image. Ces blobs portent une information de forme et de couleur qui est mise à jour lors du suivi [WADP97]. L'essentiel des méthodes de suivi du corps humain reposent toutefois sur un modèle tridimensionnel, cinématiquement paramétré, que l'on confronte aux données mesurées. Ce principe, en tout point similaire à ce que nous avons discuté dans le cas de la main, a été introduit par les travaux de J. O'Rourke [OB80] et de D. Hogg [Hog83]. Ce sont sur ces modèles, les plus courants et les plus proches de ce qui nous intéresse, que nous allons à présent revenir.

5.4.2 Modèles articulaires

Les modèles cinématiques et géométriques décrivant le corps humain sont très semblables aux modèles utilisés pour la main. De nombreux modèles cinématiques sont utilisés, sur lesquels nous ne nous étendrons pas ici, avec généralement plusieurs dizaines de degrés de liberté. Les principales différences entre les modèles tiennent généralement à la modélisation du tronc, qui peut être rigide dans les modèles les plus simples, ou présenter de nombreux degrés de liberté au niveau du bassin ou encore des épaules et de la clavicule. Ces modèles incluent très fréquemment des contraintes sur les mouvements possibles, et parfois tiennent compte des auto-collisions.

Par dessus ces modèles cinématiques sont construits des modèles volumiques qui serviront au suivi de mouvement proprement dit. La plupart d'entre eux utilisent des formes géométriques simples, telles que des cylindres [MN78, Hog83], des superquadriques [MT93, GD96], ou bien encore des quadriques tronquées [DF99, DC01].

I. A. Kakadiaris a suggéré dans [KM96] d'utiliser des modèles déformables afin de disposer d'un modèle plus précis. Dans un but similaire, R. Plänkers propose dans [PF01b, PF03] d'utiliser des surfaces implicites (ou "soft-objects"). Encore une fois, ces modèles sont particulièrement intéressants, car ils permettent de fusionner des volumes simples comme des quadriques afin d'obtenir aisément un modèle volumique continu et déformable. Ce genre de modèle a été repris dans d'autres travaux, par exemple ceux de M. Niskanen [NBH05].

5.4.3 Confrontation du modèle aux mesures

La plupart du temps, on estime la qualité d'une estimation en projetant le modèle tridimensionnel dans les différentes images. Un critère est alors utilisé pour comparer la position supposée du modèle avec les informations que l'on peut extraire de l'image. Ces informations sont généralement les mêmes que dans le cas de la main. Les contours sont probablement les plus utilisés [Hog83, GD96, DC01], la distance de Hausdorff étant par exemple employée pour estimer la distance entre les contours du modèle projeté et ceux extraits de l'image.

Q. Delamarre se sert quant à lui des silhouettes dans [DF99]. Dans le cas du corps humain dans son ensemble, on ne peut toutefois plus bénéficier d'information de couleur pour distin-

guer l'objet à suivre du fond de l'image, et l'usage de silhouettes requièrent un bon algorithme d'extraction de fond, ce qui suppose généralement de travailler dans des conditions contrôlées, en particulier pour l'éclairage, et un fond fixe. C. Bregler propose également l'utilisation de flots optiques afin de mieux estimer le mouvement entre deux images [BM98].

Plus rarement, les images sont mises à contribution pour fournir des informations tridimensionnelles que l'on pourra confronter directement au modèle. I. Haritaoglu et R. Plänkers reconstruisent par exemple des cartes de profondeur à partir de données stéréoscopiques auxquelles ils confrontent leur modèle [HHD00, PF03]. Ces techniques, assez peu utilisées à ce jour dans le cas de la main, nous apparaissent particulièrement intéressantes.

Plutôt que des données stéréoscopiques, K. Cheung suggère l'usage d'une représentation voxelique de la scène [CKBH00]. On retrouve également cette approche dans les travaux de I. Mikic [MTHC01], qui propose également d'employer des enveloppes convexes construites à partir de silhouettes extraites de différents points de vue [MTHC03]. Cette idée est également reprise par M. Niskanen dans [NBH05], et permet à C. Theobald d'opérer des ajustements après une estimation grossière de la position du corps grâce à la détection des mains, des pieds et du visage au moyen de critères de couleur [TMSS02].

Compte tenu des difficultés du suivi de mouvement, plusieurs sources d'information sont souvent utilisées conjointement afin de rendre le suivi plus robuste. Ainsi, C. Sminchisescu associe flots optiques et détection de contours dans [ST03]. De la même façon, l'usage de silhouettes œuvre de concert avec des cartes de profondeur dans les travaux de R. Plänkers [PF03].

5.4.4 Prédiction du mouvement

Comme dans le cas de la main, le suivi de mouvement est un exercice difficile, et il est souvent utile de disposer de modèles dynamiques essayant de prédire le futur mouvement du corps. Des filtres de Kalman sont souvent utilisés dans ce but [KM96, MTHC03], de même que des filtres à particules [DBR00]. Des modèles simples, avec une vitesse ou une accélération constante, sont généralement envisagés [GD96, KM96], même si l'on trouve quelques modèles dynamiques plus évolués, issus généralement d'un apprentissage de mouvements réels [PRC00, SBS02].

Les modèles dynamiques semblent donner, dans le cas du suivi du corps humain, de meilleurs résultats que dans le cadre du suivi de la main. L'une des raisons tient probablement au fait que les mouvements du corps humain sont généralement mieux décomposés que ceux de la main, lesquels sont souvent vifs et brusques.

5.4.5 Bilan

Que l'on s'efforce de suivre les mouvements de la main ou bien du corps dans son entier, les problèmes sont souvent similaires, de même que les solutions proposées. Quelques différences notables sont cependant à souligner. Dans le cas du corps humain, les vêtements peuvent poser problème, l'extraction des silhouettes peut être plus difficile dans la mesure où l'on ne

peut plus se baser sur un critère de couleur comme cela est possible avec la main. En revanche, il est généralement plus aisé d'obtenir des points d'intérêt, et surtout d'identifier certaines parties du corps, comme le visage pour lequel on dispose de détecteurs efficaces, ou encore les mains et les pieds au moyen par exemple de la couleur.

Certaines approches qui semblent prometteuses dans le cadre du suivi de mouvement du corps humain ont cependant été peu creusées dans le cas de la main. En particulier, l'utilisation de caméras multiples, sous forme de paires stéréoscopique ou non, est bien plus courante que dans le cadre du suivi de la main, probablement parce que la première idée qui vient, lorsque l'on se propose de s'intéresser à ce sujet, consiste à utiliser la webcam posée au-dessus de l'écran. Pourtant, il ne semble pas y avoir d'inconvénients à la multiplication des caméras, qui permettraient de lever bien des ambiguïtés. Cela nous conforte dans notre souhait d'essayer de tirer parti de cette information tridimensionnelle.

Chapitre 6

Modèle articulé et déformable de la main

6.1 Introduction

Afin de pouvoir étudier le mouvement d'un objet, il va nous falloir en établir un modèle. Dans ce chapitre, nous verrons comment concevoir un tel modèle, et présenterons en détail le modèle de main que nous utiliserons par la suite pour le suivi de mouvement.

Un solide articulé correspond à un ensemble d'éléments rigides, dont les mouvements respectifs dans l'espace ne sont pas libres et indépendants, mais au contraire liés par des contraintes articulaires. On dénombre une grande variété d'objets de ce genre autour de nous. Un bras robotisé, par exemple, donne un parfait exemple d'un ensemble de pièces rigides formant une chaîne articulaire.

Toutefois, beaucoup d'objets n'entrent pas parfaitement dans un cadre si strict. Les squelettes des êtres humains et des animaux présentent, par exemple, de grandes similarités avec des solides articulés, mais la présence de chair, de peau, de vêtements ne permet pas de considérer de tels systèmes comme un assemblage d'éléments "rigides". Par-dessus la structure articulée, on observe pour ces objets une certaine capacité à se déformer, qui rend le mouvement plus complexe à caractériser. Ce sont ces objets, particulièrement répandus dans le domaine du vivant, qui vont nous intéresser plus particulièrement dans la suite. Nous les qualifierons d'objets "articulés et déformables", de par leur nature duale, déformable, mais disposant quand même d'une structure articulée, à la différence d'objets amorphes, librement déformables, comme un simple morceau de glaise.

Nous nous pencherons sur des objets dont il est possible de modéliser les déformations à partir des mouvements du squelette sous-jacent, en supposant notamment que la seule donnée de la position des éléments du squelette articulé suffit à déterminer la forme de l'objet. Un modèle décrivant un objet de ce genre comprendra naturellement plusieurs éléments, résumés sur la figure 6.1 :

- un squelette, définissant la façon dont les différentes parties peuvent se déplacer les unes par rapport aux autres.

- une représentation du volume de l’objet, construite à partir de la position du squelette sous-jacent. Elle correspond, dans le cas de la main par exemple, à la répartition des chairs autour du squelette.
- éventuellement une modélisation plus fine de la surface, permettant de déterminer précisément le mouvement des points situés à la surface de l’objet. C’est ce qui servira de peau à notre modèle.

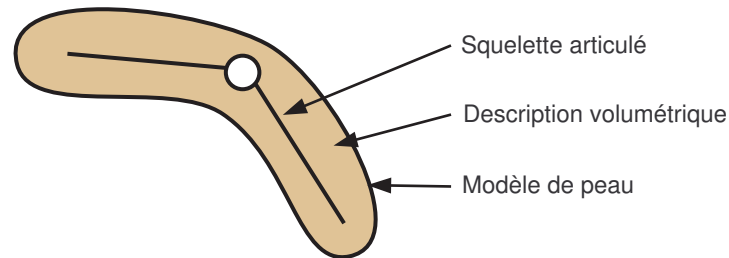


FIG. 6.1 – Les trois “couches” du modèle utilisé pour le suivi : un squelette articulé est habillé d’une description volumétrique de l’objet, sur laquelle on a greffé un modèle de peau.

Dans la suite, nous discuterons successivement chacun de ces points, afin d’établir un modèle de main humaine apte à servir de base aux travaux d’estimation de pose et de suivi.

6.2 Construction d’un squelette

6.2.1 Objectif poursuivi

La toute première étape, dans la construction d’un modèle articulé et déformable, consistera à décrire la cinématique de l’objet étudié. Lorsque l’on s’intéresse au mouvement d’un solide indéformable, six informations suffisent à définir, de façon complète, sa position : trois coordonnées permettent de le situer dans l’espace, et trois angles caractériseront son orientation.

Un “solide articulé” est, comme nous l’avons dit, un assemblage de plusieurs éléments rigides dont les mouvements respectifs ne sont pas indépendants, mais liés par une série de contraintes. Il existe une grande variété de contraintes possibles. Nous en étudierons ici quelques-unes, parmi les plus utiles pour le suivi d’objets articulés, et leurs conséquences sur le mouvement des pièces qu’ils lient. Dans tout ce qui suivra, sauf mention contraire, les différentes transformations et coordonnées seront exprimées dans le repère du monde.

6.2.2 Contraintes articulaires

Liaison rotule

La contrainte la plus simple que l'on peut établir entre deux éléments consiste à imposer qu'un point du repère lié à l'un des objets reste à tout instant coïncident avec un point du repère lié au second élément, le point en question étant usuellement appelé centre de la liaison. Ce type de liaison - appelée rotule - dans laquelle le second solide peut s'articuler librement autour d'un point du premier, et prendre une orientation quelconque, correspond par exemple à l'articulation d'une hanche ou bien encore d'une cheville. Il est possible en effet de faire pivoter la jambe autour de trois directions distinctes et de lui donner l'orientation souhaitée.

Cette contrainte peut être traduite en terme de relations liant les mouvements respectifs des deux éléments i et j . Aucune contrainte n'existe concernant l'orientation respective de ces deux éléments, les matrices \mathcal{R}_i et \mathcal{R}_j sont donc parfaitement indépendantes. En revanche, une relation lie les translations \mathbf{t}_i et \mathbf{t}_j , de façon à conserver la coïncidence des deux points :

$$\mathcal{R}_i \mathbf{X} + \mathbf{t}_i = \mathcal{R}_j \mathbf{X} + \mathbf{t}_j \quad (6.1)$$

où \mathbf{X} représente la position du centre de la liaison. Cette relation exprime simplement le fait que le point du solide i correspondant au centre de la liaison reste au cours du mouvement en coïncidence avec son homologue lié au solide j . On remarquera qu'une liaison rotule permet trois degrés de liberté en rotation, par conséquent un système composé de deux solides liés par une liaison rotule doit présenter $6 + 3$ degrés de liberté. L'équation (vectorielle) de contrainte fournit les trois contraintes scalaires qui permettent de passer des douze degrés de liberté de deux solides indépendants aux neuf degrés de liberté en question.

Liaison pivot

Il existe une multitude de façons de lier deux objets ensemble. Les liaisons les plus simples sont toutefois généralement basées sur des libertés de type rotation. Ce sont ces liaisons que l'on retrouve en particulier dans les articulations humaines et animales. Les squelettes sont constitués d'un assemblage d'articulations permettant une ou plusieurs rotations (avec toutefois l'existence occasionnelle d'un jeu plus ou moins important). Lorsqu'un mouvement de translation est souhaitable, il est généralement obtenu par la combinaison de plusieurs rotations.

Plutôt que de permettre une libre rotation, certaines liaisons permettent la rotation d'un solide par rapport à un autre uniquement autour d'un axe donné. C'est par exemple le cas de l'articulation du genou. On parle usuellement de liaison pivot. A la contrainte précédemment exprimée s'ajoute une seconde contrainte imposant qu'un axe lié au premier solide reste à tout moment coïncident avec un axe lié au second solide. Il représente l'axe de la liaison pivot : par rapport au solide i , le solide j peut uniquement tourner autour de ce dernier. On retrouve alors deux contraintes, l'une sur les translations, l'autre sur les rotations respectives des deux éléments i et j :

$$\begin{cases} \mathcal{R}_i \mathbf{X} + \mathbf{t}_i = \mathcal{R}_j \mathbf{X} + \mathbf{t}_j \\ \mathcal{R}_i \mathbf{u} = \mathcal{R}_j \mathbf{u} \end{cases} \quad (6.2)$$

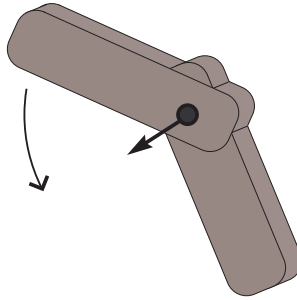


FIG. 6.2 – Dans une liaison pivot, les deux pièces sont articulées autour d’un axe commun.

où \mathbf{u} représente l’axe de la liaison pivot. La seconde contrainte impose deux équations de contraintes scalaires de plus (la norme du vecteur \mathbf{u} est conservée lors de la transformation, puisque l’on a affaire à des matrices de rotations, et l’une des trois équations scalaires correspondant au couplage des deux rotations \mathcal{R}_i et \mathcal{R}_j n’apporte donc pas de contraintes supplémentaires). Ces cinq équations de contrainte nous indiquent bien qu’il ne reste qu’un seul degré de liberté, de rotation en l’occurrence.

Joint de Cardan

À mi-chemin de la liaison rotule et de la liaison pivot, on peut parfois rencontrer des liaisons permettant deux rotations suivant des axes distincts. C’est par exemple le cas de l’articulation à la base des doigts, qui peuvent s’orienter suivant deux directions, mais ne peuvent pivoter sur eux-mêmes. La façon la plus simple de construire une liaison de ce type est de combiner en fait deux liaisons pivot entre les deux éléments, en intercalant une “pièce supplémentaire” au milieu, par exemple selon un montage dit “à la Cardan”. Dans ce cas, on retrouve deux liaisons pivot dont les axes sont généralement orthogonaux, et le centre est confondu.

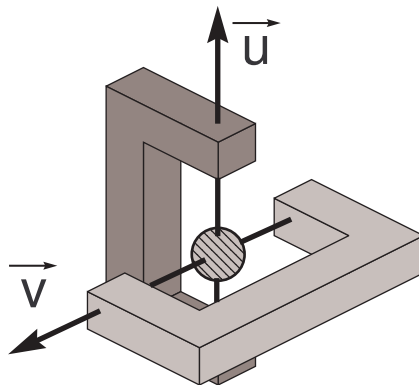


FIG. 6.3 – Un exemple typique de liaison de type “Cardan”, où les deux éléments sont articulés selon deux axes, autour d’une pièce centrale.

Il n'est pas nécessaire de considérer la pièce supplémentaire en question, dans la mesure où elle n'est pas (ou peu) souvent visible à l'extérieur, et aborder la liaison comme un ensemble, comprenant deux libertés de rotation. Il est possible d'exprimer les contraintes liant le mouvement des deux solides en remarquant que les deux axes des deux liaisons pivot doivent rester immobiles dans le repère liée à la pièce intermédiaire. En conséquence, le produit scalaire entre les vecteurs directeurs de ces deux axes doit rester constant (nul dans le cas d'axes orthogonaux).

L'un de ces vecteurs directeurs est lié rigidement à la première pièce, l'autre l'étant à la seconde pièce. Pour maintenir une liaison de ce type entre deux pièces i et j , il suffit donc d'imposer la conservation du produit scalaire entre un vecteur \mathbf{u} rigidement lié à i et un vecteur \mathbf{v} rigidement lié à j .

$$\begin{cases} \mathcal{R}_i \mathbf{X} + \mathbf{t}_i = \mathcal{R}_j \mathbf{X} + \mathbf{t}_j \\ \mathcal{R}_i \mathbf{u} \cdot \mathcal{R}_j \mathbf{v} = \mathbf{u} \cdot \mathbf{v} = cste \end{cases} \quad (6.3)$$

Cela fournit la contrainte scalaire supplémentaire permettant de faire disparaître quatre des douze degrés de liberté. Deux libertés de rotation apparaissent autour de ces deux vecteurs directeurs : en effet, la rotation de la pièce j autour de \mathbf{v} , par exemple, laisse le produit scalaire inchangé.

6.2.3 Paramétrisation des liaisons

Cas de la liaison pivot

La notion de contraintes géométriques au niveau des pièces est certes proche de l'idée que l'on se fait du mouvement articulé, mais il est relativement malaisé de travailler directement dans ce cadre. On préfère en général travailler avec une paramétrisation de la liaison. Revenons au cas de la liaison la plus simple, la liaison pivot, qui n'a qu'un seul degré de liberté. Rappelons que l'on a les contraintes suivantes :

$$\begin{cases} \mathcal{R}_i \mathbf{X} + \mathbf{t}_i = \mathcal{R}_j \mathbf{X} + \mathbf{t}_j \\ \mathcal{R}_i \mathbf{u} = \mathcal{R}_j \mathbf{u} \end{cases} \quad (6.4)$$

Nous allons remanier ces contraintes de façon à exprimer le mouvement de j en fonction du seul mouvement de i et d'un paramètre α correspondant au seul degré de liberté de rotation permis par la liaison pivot. La première chose que l'on peut remarquer, c'est que puisque \mathcal{R}_i et \mathcal{R}_j sont deux matrices de rotation, alors elles sont reliées par une expression de ce type :

$$\mathcal{R}_j = \mathcal{R}_i \mathcal{R}_{ij} \quad (6.5)$$

où \mathcal{R}_{ij} est nécessairement une matrice de rotation, de par la nature de groupe de l'ensemble des rotations de l'espace. \mathcal{R}_{ij} peut en effet être réécrite de la sorte :

$$\mathcal{R}_{ij} = \mathcal{R}_i^{-1} \mathcal{R}_j \quad (6.6)$$

Regardons à présent l'action de \mathcal{R}_{ij} sur le vecteur directeur de la liaison, \mathbf{u} . On a :

$$\begin{aligned}\mathcal{R}_{ij} \mathbf{u} &= \mathcal{R}_i^{-1} \mathcal{R}_j \mathbf{u} \\ &= \mathcal{R}_i^{-1} \mathcal{R}_i \mathbf{u} \\ &= \mathbf{u}\end{aligned}$$

où l'on a utilisé la contrainte exprimée précédemment sur les deux matrices de rotation \mathcal{R}_i et \mathcal{R}_j . On constate que la matrice de rotation \mathcal{R}_{ij} conserve le vecteur \mathbf{u} . C'est donc simplement une rotation autour de l'axe \mathbf{u} , et elle peut être exprimée au moyen de la formule de Rodrigues.

En définitive, nous pouvons réécrire le mouvement de j en fonction du mouvement de i , du vecteur \mathbf{u} , du point \mathbf{X} et du paramètre α :

$$\begin{cases} \mathcal{R}_j = \mathcal{R}_i \mathcal{R}_{ij} = \mathcal{R}_i \mathcal{R}_{\mathbf{u},\alpha} \\ \mathbf{t}_j = \mathcal{R}_i \mathbf{X} - \mathcal{R}_j \mathbf{X} + \mathbf{t}_i \end{cases} \quad (6.7)$$

où $\mathcal{R}_{\mathbf{u},\alpha}$ correspond à la matrice de Rodrigues d'axe \mathbf{u} et de paramètre α , ce que l'on peut simplifier en :

$$\begin{cases} \mathcal{R}_j = \mathcal{R}_i \mathcal{R}_{\mathbf{u},\alpha} \\ \mathbf{t}_j = \mathcal{R}_i (1 - \mathcal{R}_{\mathbf{u},\alpha}) \mathbf{X} + \mathbf{t}_i \end{cases} \quad (6.8)$$

Autres types de liaison

On peut procéder de la même façon pour les autres types de liaisons. Sans entrer dans les détails, la liaison rotule à doigt, nous l'avons vu, consiste essentiellement en deux liaisons pivot successives d'axe \mathbf{u} et \mathbf{v} , placées au même endroit \mathbf{X} . On va donc voir apparaître deux matrices de Rodrigues, de paramètres respectifs α et β . On aura donc :

$$\begin{cases} \mathcal{R}_j = \mathcal{R}_i \mathcal{R}_{\mathbf{u},\alpha} \mathcal{R}_{\mathbf{v},\beta} \\ \mathbf{t}_j = \mathcal{R}_i (1 - \mathcal{R}_{\mathbf{u},\alpha} \mathcal{R}_{\mathbf{v},\beta}) \mathbf{X} + \mathbf{t}_i \end{cases} \quad (6.9)$$

Le cas de la liaison rotule est un peu plus complexe. Le paramétrage le plus connu est celui dit des angles d'Euler. Ce n'est toutefois qu'une méthode parmi d'autres pour représenter les rotations d'un objet. Parmi les autres possibilités souvent retenues, on pourra citer les quaternions, particulièrement compacts et dépourvus de problème de "lock", et qui peuvent être étendus pour inclure non seulement les rotations mais également les translations [KM96]. Les "twists", enfin bien connus des roboticiens, ont été introduits dans le domaine de la capture de mouvement par C. Bregler dans [BMP03]. Bien souvent, c'est l'usage que l'on souhaite en faire qui guidera le choix d'une paramétrisation donnée. Comme nous n'aurons pas, dans le cas qui nous intéresse, à nous préoccuper de liaisons de ce type, nous n'irons pas plus avant.

6.2.4 Construction de la chaîne articulaire

Détermination itérative des rotations et translations

Ces relations vont nous permettre, dans le cas de chaînes ouvertes, de déterminer de proche en proche le mouvement de chacune des pièces constituant le modèle articulé, à partir de la

donnée du mouvement d'une pièce de référence, de la position de chacune des liaisons, et de l'ensemble des paramètres articulaires. Prenons l'exemple d'un système composé de trois éléments, numérotés 1, 2 et 3. Le premier élément peut se mouvoir librement dans l'espace. Le second s'articule autour du premier selon une liaison rotule à doigt, située en \mathbf{X}_2 , d'axes \mathbf{u}_2 et \mathbf{v}_2 . Quant au troisième, c'est une liaison pivot qui le lie au second élément, liaison sise en \mathbf{X}_3 et d'axe \mathbf{u}_3 .

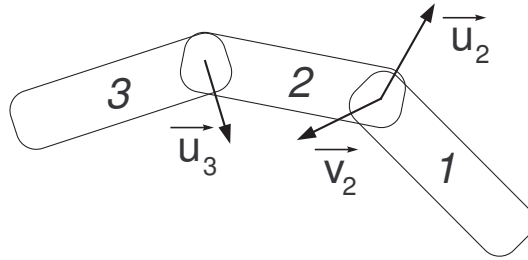


FIG. 6.4 – Un exemple de chaîne articulaire comprenant trois éléments, séparés respectivement par une liaison cardan (à droite) et une liaison pivot (à gauche).

Le mouvement de l'élément 1 est libre dans l'espace, et correspond donc à une matrice de rotation quelconque \mathcal{R}_1 et à une translation quelconque \mathbf{t}_1 . On construit le mouvement du second élément à partir du premier

$$\begin{cases} \mathcal{R}_2 = \mathcal{R}_1 \mathcal{R}_{\mathbf{u}_2, \alpha_2} \mathcal{R}_{\mathbf{v}_2, \beta_2} \\ \mathbf{t}_2 = \mathcal{R}_1 (1 - \mathcal{R}_{\mathbf{u}_2, \alpha_2} \mathcal{R}_{\mathbf{v}_2, \beta_2}) \mathbf{X}_2 + \mathbf{t}_1 \end{cases} \quad (6.10)$$

puis le mouvement du troisième élément à partir de celui du second

$$\begin{cases} \mathcal{R}_3 = \mathcal{R}_2 \mathcal{R}_{\mathbf{u}_3, \alpha_3} \\ \mathbf{t}_3 = \mathcal{R}_2 (1 - \mathcal{R}_{\mathbf{u}_3, \alpha_3}) \mathbf{X}_3 + \mathbf{t}_2 \end{cases} \quad (6.11)$$

Par substitution, si besoin est, il est possible d'exprimer le mouvement de la troisième pièce simplement en fonction du mouvement libre de 1, des positions des liaisons, et des différents paramètres :

$$\begin{cases} \mathcal{R}_3 = \mathcal{R}_1 \mathcal{R}_{\mathbf{u}_2, \alpha_2} \mathcal{R}_{\mathbf{v}_2, \beta_2} \mathcal{R}_{\mathbf{u}_3, \alpha_3} \\ \mathbf{t}_3 = \mathcal{R}_1 \mathcal{R}_{\mathbf{u}_2, \alpha_2} \mathcal{R}_{\mathbf{v}_2, \beta_2} (1 - \mathcal{R}_{\mathbf{u}_3, \alpha_3}) \mathbf{X}_2 + \mathcal{R}_1 (1 - \mathcal{R}_{\mathbf{u}_2, \alpha_2} \mathcal{R}_{\mathbf{v}_2, \beta_2}) \mathbf{X}_3 + \mathbf{t}_1 \end{cases} \quad (6.12)$$

On obtient de cette façon le mouvement de l'ensemble des éléments en combinant simplement la structure de la chaîne (à travers le type et la position des différentes liaisons) et les différents paramètres du mouvement (ici neuf degrés de liberté, six pour le mouvement de la pièce 1, les trois autres étant représentés par les trois paramètres α_2 , β_2 et α_3).

Ce type de représentation ne convient pas à des chaînes articulaires fermées, qui ne peuvent pas être représentées par des arbres, parce qu'elles comportent des cycles cinématiques entre les différentes pièces. Ce cas de figure est fréquent dans le domaine industriel car il permet d'augmenter les efforts pouvant être transmis, mais fait figure d'exception dans le domaine

du vivant (du moins d'un point de vue extérieur). Le corps humain dans son ensemble, par exemple, peut être représenté par une chaîne ouverte, depuis le tronc jusqu'à l'extrémité des doigts. La main humaine, par voie de conséquence, l'est donc tout autant.

Les chaînes fermées posent des problèmes spécifiques, car la fermeture de chaînes cinématiques entraîne des contraintes supplémentaires, de sorte que les paramètres précédemment présentés ne sont plus tous indépendants. La fermeture retire certains degrés de liberté, et il faut donc reconsidérer le paramétrage. Les méthodes pour ce faire abondent dans la littérature robotique, mais dépasse le cas de notre étude.

Dérivation des transformations

La représentation que nous proposons se prête bien à la dérivée par rapport à un paramètre quelconque. Nous pouvons aisément construire les matrices dérivées pour tous les éléments avec la même démarche itérative.

$$\begin{cases} \frac{\partial \mathcal{R}_j}{\partial \alpha} = \frac{\partial \mathcal{R}_i}{\partial \alpha} \mathcal{R}_{ij} + \mathcal{R}_i \frac{\partial \mathcal{R}_{ij}}{\partial \alpha} \\ \frac{\partial \mathbf{t}_j}{\partial \alpha} = \frac{\partial \mathcal{R}_i}{\partial \alpha} (1 - \mathcal{R}_{ij}) \mathbf{X} - \mathcal{R}_i \frac{\partial \mathcal{R}_{ij}}{\partial \alpha} \mathbf{X} + \frac{\partial \mathbf{t}_i}{\partial \alpha} \end{cases} \quad (6.13)$$

En général, les paramètres n'apparaissent que dans une seule matrice (matrice de Rodrigues, dont il est aisé d'avoir une dérivée) ou un seul vecteur des formules précédentes, de sorte qu'un terme au minimum est nul. Par exemple, supposons que l'on cherche à dériver les matrices précédentes par rapport au paramètre α_2 . Seule la matrice $\mathcal{R}_{\mathbf{u}_2, \alpha_2}$ en dépend. On a donc :

$$\begin{cases} \frac{\partial \mathcal{R}_1}{\partial \alpha_2} = 0 \\ \frac{\partial \mathbf{t}_1}{\partial \alpha_2} = 0 \end{cases} \quad (6.14)$$

$$\begin{cases} \frac{\partial \mathcal{R}_2}{\partial \alpha_2} = \mathcal{R}_1 \frac{\partial \mathcal{R}_{\mathbf{u}_2, \alpha_2}}{\partial \alpha_2} \mathcal{R}_{\mathbf{v}_2, \beta_2} \\ \frac{\partial \mathbf{t}_2}{\partial \alpha_2} = -\mathcal{R}_1 \frac{\partial \mathcal{R}_{\mathbf{u}_2, \alpha_2}}{\partial \alpha_2} \mathcal{R}_{\mathbf{v}_2, \beta_2} \mathbf{X}_2 \end{cases} \quad (6.15)$$

$$\begin{cases} \frac{\partial \mathcal{R}_3}{\partial \alpha_2} = \frac{\partial \mathcal{R}_2}{\partial \alpha_2} \mathcal{R}_{\mathbf{u}_3, \alpha_3} \\ \frac{\partial \mathbf{t}_3}{\partial \alpha_2} = \frac{\partial \mathcal{R}_2}{\partial \alpha_2} (1 - \mathcal{R}_{\mathbf{u}_3, \alpha_3}) \mathbf{X}_3 + \frac{\partial \mathbf{t}_2}{\partial \alpha_2} \end{cases} \quad (6.16)$$

6.2.5 Limitation des mouvements

Dans un système articulaire, l'amplitude des mouvements est le plus souvent limitée à une zone bien précise. Ainsi, il n'est pas possible de lever l'index de plus d'une trentaine de degrés au-dessus du plan de la paume. Ces limitations peuvent servir de guide au suivi, puisqu'elles permettent d'éviter des configurations impossibles de la chaîne articulaire. Usuellement, ces limitations sont traduites par des contraintes sur les valeurs que peuvent prendre les paramètres des diverses articulations. Plus précisément, on définit des angles limites de rotation de chaque pièce par rapport à celle à laquelle elle est liée.

Ces angles sont en général mesurés par rapport à une position de référence du squelette. Il n'est pas toujours aisé, cependant, de déterminer cette position de référence, surtout lorsque l'on a affaire à un squelette articulaire initialement dans une position quelconque. Nous allons montrer qu'il est possible de retrouver aisément ces différents angles en assignant des axes de références à chacun des éléments de la chaîne articulaire.

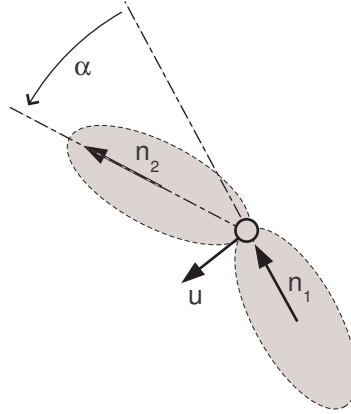


FIG. 6.5 – L'angle à mesurer entre deux pièces articulées autour d'une liaison pivot.

Dans le cas d'une liaison pivot entre deux pièces i et j , l'angle qui nous intéresse est celui que l'on trouve entre les deux pièces (cf schéma 6.5). Considérons \mathbf{u} , le vecteur normé correspondant à l'axe de la liaison pivot, et deux vecteurs normés \mathbf{n}_i et \mathbf{n}_j associés à chacune des deux pièces, chacun de ces deux vecteurs étant orthogonal à \mathbf{u} . Les équations suivantes permettent alors de retrouver l'angle entre les deux pièces :

$$\begin{aligned} \cos(\alpha) &= \mathbf{n}_i \cdot \mathbf{n}_j \\ \sin(\alpha) &= [\mathbf{n}_i, \mathbf{n}_j, \mathbf{u}] = (\mathbf{n}_i \times \mathbf{n}_j) \cdot \mathbf{u} \end{aligned} \quad (6.17)$$

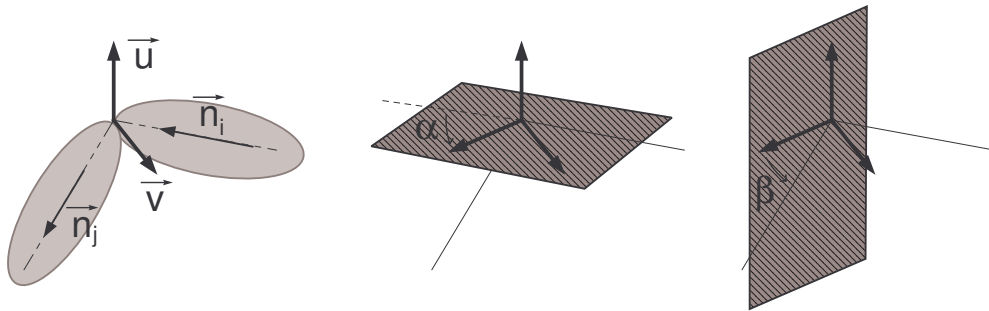


FIG. 6.6 – Dans une liaison “Cardan”, les deux angles sont mesurés par rapport à l'axe $\mathbf{v} \times \mathbf{u}$.

Les liaisons dites *rotules à doigt* (généralement dénommées liaisons “Cardan”) posent un problème un peu plus épineux, mais néanmoins soluble. Rappelons que ce type de liaison peut être vue comme la combinaison de deux liaisons pivot en cascade, entre lesquelles on trouve

une pièce “virtuelle”. Pour mesurer chacun des deux angles, il faut disposer d’un axe lié à cette pièce virtuelle. Afin d’obtenir un axe orthogonal à l’axe de chacune des deux liaisons pivot, il est naturel de choisir comme axe de référence le produit vectoriel de ces deux axes de rotation.

$$\begin{aligned}\cos(\alpha) &= [\mathbf{v}, \mathbf{u}, \mathbf{n}_i] = (\mathbf{v} \times \mathbf{u}) \cdot \mathbf{n}_i \\ \sin(\alpha) &= [\mathbf{n}_i, \mathbf{v} \times \mathbf{u}, \mathbf{u}] = (\mathbf{n}_i \times (\mathbf{v} \times \mathbf{u})) \cdot \mathbf{u}\end{aligned}\quad (6.18)$$

$$\begin{aligned}\cos(\beta) &= [\mathbf{v}, \mathbf{u}, \mathbf{n}_j] = (\mathbf{v} \times \mathbf{u}) \cdot \mathbf{n}_j \\ \sin(\beta) &= [\mathbf{v} \times \mathbf{u}, \mathbf{n}_j, \mathbf{v}] = ((\mathbf{v} \times \mathbf{u}) \times \mathbf{n}_j) \cdot \mathbf{v}\end{aligned}\quad (6.19)$$

Ces formules sont valables dans le cas où les deux axes \mathbf{u} et \mathbf{v} sont orthogonaux, mais peuvent être étendues sans difficulté à tout autre cas en prenant soin de renormaliser le vecteur $\mathbf{v} \times \mathbf{u}$. Ce vecteur est toujours bien défini, puisque dans le cas où \mathbf{u} et \mathbf{v} sont colinéaires, la liaison se trouve être dégénérée et correspond simplement à une liaison pivot.

De cette façon, il est aisé de mesurer les angles correspondant à chacune des liaisons, et donc d’imposer des limites aux valeurs qu’ils sont susceptibles de prendre. Reste toutefois à choisir, pour chaque liaison, un axe de référence pour chacune des deux pièces qu’elle associe. Il est en général possible de choisir un seul axe par élément.

Dans le cas de la main, ces choix seront relativement simples. Pour chacune des phalanges des doigts, l’axe médian est un choix naturel et aisé. Le cas de la paume est à peine plus délicat, et on peut aisément lui associer un repère dont un axe est perpendiculaire au plan auquel elle correspond, et un second est orienté dans la direction générale de la main, direction dans laquelle pointeraient les doigts si la main était tendue.

6.2.6 La cinématique de la main

Présentation

Ainsi que l’ont montré de nombreuses études en robotique, la préhension est un problème des plus épineux. La main, qui joue ce rôle chez l’être humain, se trouve donc naturellement être une partie particulièrement complexe du corps humain du point de vue mécanique : un grand nombre de degrés de liberté, permettant une vaste variété de prises, sont rassemblés dans un espace restreint.

Lorsque l’on regarde l’ensemble des os composant la main, on se retrouve avec un assemblage complexe d’os tous liés et articulés entre eux. Le poignet, souvent dénommé carpe en anatomie, se compose de huit osselets regroupés en deux rangées, et donne accès au métacarpe qui forme le squelette proprement dit de la paume. Il consiste en un ensemble de cinq os allongés, qui serviront de base aux cinq doigts de la main. Chacun des doigts proprement dits se compose d’un ensemble de trois phalanges (deux seulement dans le cas du pouce).

Les cas de l’index, du majeur, de l’annulaire et de l’auriculaire sont essentiellement semblables. Les quatre os du métacarpe qui leur correspondent sont essentiellement liés dans la paume de la main, et leurs mouvements relatifs sont des plus restreints. Pour cette raison, les degrés de liberté situés à ce niveau sont généralement négligés.



FIG. 6.7 – Le squelette de la main, où l'on identifie carpe, métacarpe et phalanges.

Entre le métacarpe et les premières phalanges, on trouve des liaisons de type rotule à doigt, autrement dit des liaisons dite “à la Cardan”, qui fournissent à chacun des doigts deux degrés de liberté. Le premier suivant un axe perpendiculaire au plan de la paume permettant l'orientation du doigt. Le second se situe quant à lui dans le plan de la paume, perpendiculaire au doigt, et permet sa flexion. À la base des secondes et troisièmes phalanges, on retrouve des liaisons pivot selon des axes parallèles aux précédents.

Chaque doigt dispose donc essentiellement de quatre degrés de liberté par rapport à la paume de la main. D'un point de vue anatomique, beaucoup de ces degrés de liberté ne sont pas indépendants, et le contrôle de chacun des mouvements qui leur correspondent n'est pas toujours possible. Certains modèles utilisent ces limitations anatomiques pour restreindre plus avant le nombre de degrés de liberté du modèle. L'exemple le plus simple consiste à considérer que le mouvement des liaisons pivot de part et d'autre de la seconde phalange sont liés, ce qui se trouve être généralement le cas en pratique. Toutefois, ces simplifications restent des approximations, et lorsque la main tient un objet, par exemple, ou bien encore prend appui sur quelque chose, ces contraintes sont très souvent violées.

Le cas du pouce est quelque peu différent. Le mouvement du métacarpe ne peut pas, pour ce dernier, être négligé. Son articulation par rapport au carpe consiste essentiellement en une liaison à deux degrés de liberté de rotation, tandis que les deux articulations suivantes (entre le métacarpe et la première phalange, et entre la première et la seconde phalange) correspondent à des liaisons de type pivot. Toutefois, l'articulation à la base du métacarpe est assez complexe, et son positionnement comme son orientation se révèlent assez délicats. C'est la raison pour laquelle on trouve régulièrement dans la littérature un degré de liberté supplémentaire entre le métacarpe et la première phalange.

En ajoutant les six degrés de liberté correspondant au positionnement et à l'orientation de la paume dans l'espace, nous arrivons donc à un modèle comprenant un total de vingt-

six (ou vingt-sept) degrés de liberté pour décrire complètement la position d'une main. On retrouvera la chaîne cinématique complète résumée dans la figure 6.8. Ce sont ces vingt-six paramètres qu'il va nous falloir estimer à tout instant lors du suivi. C'est le modèle que l'on retrouve le plus généralement dans les applications de suivi de la main (citons par exemple [SMC01, DF98, WH99, HOR99, LH00]...), et que nous avons décidé d'utiliser.

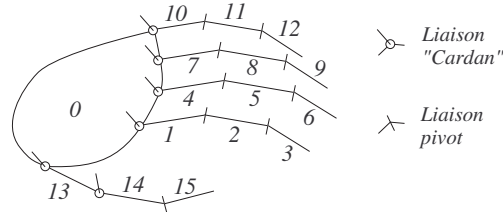


FIG. 6.8 – Le modèle à 27 degrés de liberté que nous utiliserons, comprenant seize éléments, articulés au moyen de six liaisons cardan et neuf liaisons pivot. La paume, qui peut prendre une position quelconque dans l'espace, ajoute six degrés de liberté supplémentaires.

Il est rare que des modèles plus complexes soient utilisés, même si quelques rares travaux ont mis à profit des modèles cinématiques où les mouvements entre le carpe et le métacarpe n'étaient pas négligés. On citera le modèle utilisé par K. Nirei *et al* [NSMO96] qui comporte trente-trois degrés de liberté. Compte tenu de la relative imprécision qui plane sur le modèle, il ne nous est pas apparu utile de prendre en compte ces degrés de liberté supplémentaires. En revanche, nombreux sont les travaux qui, comme nous l'avons dit, ont essayé de réduire le nombre de ces libertés. Les travaux de Y. Wu *et al* [WLH01] utilisent par exemple une réduction en composantes principales, à partir de données issues de motion capture, pour n'avoir à considérer que sept degrés de liberté, la pose n'étant pas prise en compte. Ce choix limite évidemment beaucoup la variété des mouvements pouvant être suivis.

Expression du squelette au format BVH

Le format BVH est probablement l'un des plus usités dans le domaine de l'animation pour la description des squelettes. Notre modèle de main peut très simplement être exprimé au moyen de cet outil. Le format BVH décrit chaque squelette comme une hiérarchie arborescente d'éléments appelés *bones*. La position d'un élément est exprimée dans le repère de son parent direct par la donnée d'un point, défini par trois translations le long de chacun des axes du repère lié au parent, et de trois rotations autour de ces mêmes axes. Il est possible de choisir librement l'ordre dans lequel ces rotations seront effectuées. Dans notre cas, nous utiliserons d'abord une rotation suivant l'axe \mathbf{z} , puis suivant l'axe \mathbf{y} , et enfin autour de l'axe \mathbf{x} .

Nous utiliserons comme position de référence une main posée à plat dans le plan $(O, \mathbf{x}, \mathbf{z})$, les quatre principaux doigts étant parallèles, le long de l'axe \mathbf{z} (dans ce qui suit, nous décrirons une main droite). La paume de la main sert, comme nous l'avons dit précédemment, de base sur laquelle viendront se greffer les autres éléments. Les trois translations et les trois rotations qui correspondent à cet élément sont donc libres. On choisira librement l'origine du repère lié à la paume, par exemple à proximité de son centre. Comme le mouvement est libre, ce choix

n'a que peu d'importance.

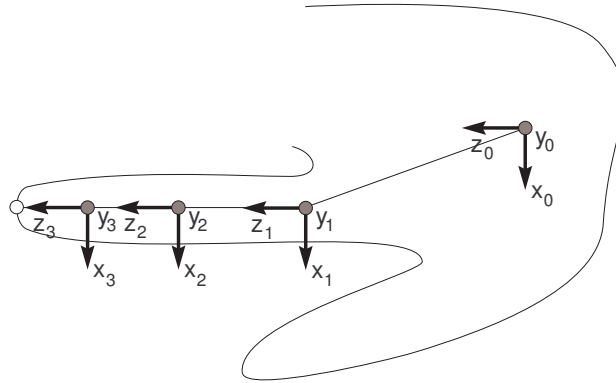


FIG. 6.9 – Les différents repères associés à la main et à chacune des phalanges de l'index dans un fichier BVH. Le cas des trois autres doigts est semblable, avec parfois une légère torsion à la base de la première phalange autour de l'axe z .

Le cas de chacun des quatre doigts (pouce excepté) est similaire, et consistera en une arborescence semblable de trois éléments. Le premier correspond à la première phalange. Les trois translations fixent la position de l'articulation à la base du doigt par rapport à la paume, et ne changeront pas au cours du temps. La première rotation, suivant l'axe z , est fixe au cours du temps elle aussi. Elle traduit l'inclinaison du premier axe de rotation de la liaison cardan à la base de chacun des doigts par rapport à l'axe y . On la choisira généralement nulle pour le majeur, légèrement négative pour l'index et positive pour les deux doigts restants.

Les deux rotations suivantes, suivant les axes y et x , peuvent librement évoluer au cours du temps, et représentent chacun des deux degrés de liberté de la liaison cardan. La première correspond au mouvement d'écartement des doigts (l'angle correspondant pouvant prendre des angles allant approximativement de -20 degrés à 20 degrés), la seconde au repliement du doigt (de -30 à 70 degrés environ). Dans la position de référence, chacune de ces rotations est nulle.

Vient ensuite le cas des seconde et troisième phalanges, qui sont similaires. Les translations suivant x et y sont nulles, celle suivant z fixe la longueur de la phalange précédente (distance entre les deux articulations). Les rotations autour des axes z et y permettent de modéliser d'éventuels défauts dans la cinématique de la main, mais sont en principe nulles (les axes correspondant aux trois rotations permettant de plier un même doigt sont alors parallèles). La rotation autour de l'axe x est la seule qui évoluera au cours du temps (un seul degré de liberté), avec des valeurs s'échelonnant entre -10 et 110 degrés pour la seconde phalange, et -5 et 85 degrés pour la troisième et dernière phalange.

Le cas du pouce est un peu plus délicat à établir. Si les positions et orientations des articulations des quatre doigts précédents sont relativement aisées à situer, il en va différemment pour le pouce, d'autant que les liaisons réelles ne sont pas aussi ponctuelles que le modèle

que nous utilisons. En première approximation, le premier élément (métacarpe du pouce) peut être décrit de façon similaire aux premières phalanges des quatre autres doigts. Pour des raisons d'encombrement, cependant, le pouce ne peut être parallèle aux autres doigts.

Entre le premier et le second élément, la rotation autour de l'axe \mathbf{z} n'est pas négligeable, ni la conséquence d'un éventuel défaut : elle permet en fait au pouce d'être opposable. Sa valeur est fixée, voisine d'une soixantaine de degrés. En théorie, une seule rotation (autour de l'axe \mathbf{x}) est présente entre le métacarpe du pouce et sa première phalange. Cependant, compte tenu des nombreuses imprécisions précitées du modèle, il est généralement utile de laisser une liberté en rotation supplémentaire autour de l'axe \mathbf{y} . Le cas de la seconde phalange du pouce, enfin, est similaire à ceux des phalanges terminales des quatre autres doigts.

La structure de la main correspondra donc à un fichier du genre de celui présenté ci-dessous. À noter que l'absence d'offset pour les rotations dans la norme BVH force à préciser chacune des rotations, qu'elle soit libre ou fixée. Le fichier BVH a été résumé au seul index ainsi qu'à la paume de la main, pour des raisons de longueur. On retrouve la structure correspondante sur la figure 6.9.

```

HIERARCHY
ROOT PAUME
{
  OFFSET 0 0 0
  CHANNELS 6 Xposition Yposition Zposition Zrotation Yrotation Xrotation
  JOINT INDEX_1
  {
    OFFSET 28.0 5.0 56.0
    CHANNELS 3 Zrotation Yrotation Xrotation
    JOINT INDEX_2
    {
      OFFSET 0.0 0.0 48.0
      CHANNELS 3 Zrotation Yrotation Xrotation
      JOINT INDEX_3
      {
        OFFSET 0.0 0.0 26.0
        CHANNELS 3 Zrotation Yrotation Xrotation
        END SITE
        {
          OFFSET 0.0 0.0 24.0
        }
      }
    }
  }
}
JOINT MAJEUR_1
...
}

```

6.3 Habillage du squelette

6.3.1 Modélisation du volume

La seule donnée de la cinématique du problème ne saurait, dans la très grande majorité des cas, suffire pour permettre le suivi d'un objet articulé. Il est tout aussi nécessaire de pouvoir décrire la forme de l'objet lui-même. En effet, la plupart des informations que l'on peut extraire des images ou des séquences vidéo sont liées à des éléments situés à la surface de l'objet : points d'intérêt, contours, silhouette, etc.

Cette représentation du volume ou de la surface d'un objet doit répondre au mieux à un certain nombre de critères :

- La représentation choisie doit permettre une description correcte, et dans la mesure du possible précise, de l'objet suivi. Il est nécessaire de pouvoir calculer, pour une position quelconque des articulations dans l'espace, la surface de notre modèle.
- La représentation doit également permettre, si possible, une modélisation aisée de l'objet étudié. Le modèle obtenu devrait pouvoir être adapté assez facilement si l'on remplace ce dernier par un autre de forme sensiblement différente. C'est particulièrement important dans le cas du suivi d'objets "naturels" comme le corps humain ou la main, dont les caractéristiques changent d'un individu à l'autre. Dans la mesure du possible, on préfère disposer d'un modèle ajustable, soit au moyen d'un apprentissage préalable, soit par affinage au cours du suivi.
- Enfin, on attend de la représentation qu'elle soit bien adaptée au suivi que l'on cherche à mettre en place. Dans notre cas, nous chercherons à associer des points 3D reconstruits par stéréoscopie à notre modèle. Il nous faudra donc disposer d'une représentation permettant aisément le calcul de la distance entre un de ces points et la surface du modèle. D'autres méthodes pourront préférer des représentations différentes, présentant d'autres avantages (facilité de projection, d'extraction de contours, etc).

L'informatique graphique met à notre disposition une grande variété de moyens pour représenter surfaces et volumes : volumes élémentaires, CSG, surfaces triangulées, surfaces de Bézier, nuages de points, volumes implicites, etc. Chacun d'entre eux présente des avantages et inconvénients distincts en ce qui concerne le suivi, il n'existe très probablement pas de représentation universellement adaptée à tous les usages.

6.3.2 Solutions usuellement retenues

La solution la plus fréquemment choisie dans le domaine du suivi consiste à utiliser un ensemble de formes de base pour décrire l'objet : un assemblage de cônes, cylindres, sphères et autres prismes droits permet d'obtenir une forme plus ou moins réaliste. Dans le cas de la main humaine, des cônes modélisant les différentes phalanges, combinés à des sphères au niveau des articulations pour permettre d'obtenir une surface relativement continue pour toute position de la main, donnent un résultat satisfaisant pour la modélisation des doigts. La paume peut toutefois poser un problème sensiblement plus délicat.

Bien évidemment, il n'est pas question avec ce genre de modélisation de simuler, par exemple, les plis de la peau et le gonflement des phalanges lorsqu'un doigt se plie. Toutefois,

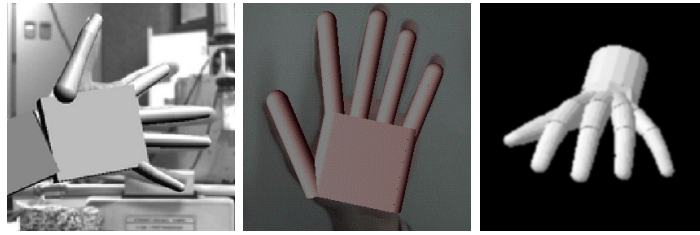


FIG. 6.10 – Quelques exemples de modélisations de mains au moyen de formes simples. De gauche à droite : Delamarre *et al* [DF98], Ouahdi *et al* [OH98], Nirei *et al* [NSMO96].

ces effets sont suffisamment négligeables pour que l'on puisse les négliger dans le suivi, au moins en première approximation. Ces modèles permettent une extraction aisée des contours, et un calcul analytique simple de la distance d'un point de l'espace à la surface du modèle. Ces avantages font de ce type de représentation l'un des plus utilisés dans le domaine du suivi.

Beaucoup de descriptions surfaciques et volumétriques entrent dans le cadre de ce que l'on appelle les "surfaces implicites". Toutes sont fondées sur un même principe : on dispose d'un champ continu de valeurs, sur tout ou partie de l'espace. La surface considérée est le lieu des points où ce champ prend une valeur précise, parfois appelée isovaleur. On peut, très simplement, décrire également un volume avec le même outil : on définira par exemple l'intérieur du volume comme le lieu des points correspondant à une valeur supérieure au seuil choisi. On peut alors aisément déterminer, pour tout point, s'il se trouve à l'intérieur de l'objet, à l'extérieur, ou bien encore à proximité de sa surface, simplement en examinant la valeur que prend le champ à cet emplacement.

Il est également possible de se dispenser complètement d'une description de la surface, et prendre pour référence un simple nuage de points, répartis à la surface du modèle. A la condition que l'on dispose d'une grande densité de points sur toute la surface du modèle, il est possible d'estimer la distance d'un point quelconque de l'espace au modèle, simplement en prenant la distance au point du nuage le plus proche [Gra03]. La grande quantité de données nécessaires en rend toutefois l'usage malaisé pour le suivi. D'autres descriptions plus spécifiques de la forme étudiée ont été utilisées. Heap et Hogg [HH96] utilisent par exemple des PDM (Point Distribution Model) pour décrire la forme d'une main.

On a vu se développer ces dernières années, principalement dans le domaine de la synthèse d'image, de nouveaux outils pouvant se révéler utiles. Plus particulièrement, les surfaces implicites ont rapidement vu leur importance croître. Elles présentent de nombreux avantages (réalisation aisée de surfaces courbes complexes, obtention d'objets de topologie quelconques, nombreux outils mathématiques, etc.) qui ont poussé à leur adoption dans de nombreux domaines. Plus récemment, on a pu envisager leur utilisation en vue de suivi d'objets. Les travaux de R. Plänkers et P. Fua [PF01b] ont par exemple montré leur intérêt pour le suivi du corps humain.

6.3.3 Les surfaces implicites

Motivation

Décrire des surfaces gauches comme peuvent l'être celles issues du monde organique, ou bien encore des volumes dont la topologie est complexe, peut se révéler être un problème délicat. Il est en général difficile d'obtenir une paramétrisation simple de ces surfaces et volumes, et lorsque cela est envisageable, leur manipulation (au moyen de points de contrôle, par exemple) est malaisée. Une solution a été proposée avec l'introduction des surfaces implicites.

Une surface lisse, de bonne qualité, peut être d'un secours appréciable lorsque l'on souhaite suivre le mouvement d'un objet déformable. Les surfaces implicites, que nous avons présentées dans le chapitre 1.5, permettent d'obtenir ce genre de résultat.

Les surfaces implicites les plus adaptées pour la description d'un solide indéformable sont celles qui fusionnent des volumes élémentaires. À chaque élément du squelette seront attachés un ou plusieurs de tels volumes, comme c'est le cas avec les descriptions habituellement utilisées. Toutefois, dans le cas présent, plutôt que d'ajouter des éléments comme des sphères au niveau des articulations pour assurer la continuité de la surface, ces volumes seront naturellement fusionnés par la description implicite.

Rappelons que les surfaces implicites sont décrites, au travers d'une fonction f , comme le lieu géométrique des points \mathbf{X} vérifiant $f(\mathbf{X}) = k$ où k est une constante appelée isovaleur (dans la suite, nous utiliserons $k = 1$). Nous utiliserons les expressions proposées par John Blinn dans [Bli82], qui décrivent la surface implicite comme la fusion d'un ensemble de sphères élémentaires. Le potentiel en un point \mathbf{X} permettant d'obtenir une sphère de rayon R_i centrée en \mathbf{C}_i s'exprime — pour une isovaleur de 1 — sous la forme :

$$f_i(\mathbf{X}) = e^{-\frac{|\mathbf{X}-\mathbf{C}_i|^2 - R_i^2}{\nu}}. \quad (6.20)$$

La fusion de ces sphères élémentaires est obtenue par la somme de leurs potentiels f_i . Le paramètre ν permet d'ajuster la façon dont les volumes élémentaires seront fusionnés, en particulier la distance à partir de laquelle cette fusion s'opère. L'isosurface construite par la fusion de telles sphères correspond donc alors au lieu géométrique des points qui vérifient :

$$\sum_{i=1}^n f_i(\mathbf{X}) = 1. \quad (6.21)$$

Plutôt que de travailler uniquement avec des sphères, il serait utile de pouvoir utiliser des ellipsoïdes. Cela permettra un peu plus de souplesse dans la modélisation. En particulier, dans le cas de la main, il est raisonnable de penser que l'on peut considérer chacun des doigts comme la fusion de quelques ellipsoïdes, plus précisément un par phalange. Pour ce faire, nous allons réécrire les fonctions f_i de façon à étirer ces sphères de façon différente selon trois directions différentes de l'espace.

Nous utiliserons toujours la distance à un point, mais nous allons remplacer la distance euclidienne par une distance qui n'est pas isotrope. Une telle distance peut être exprimée par

une simple fonction quadratique. Dans le cas euclidien, la distance entre deux points \mathbf{X} et \mathbf{Y} s'exprime comme

$$d(\mathbf{X}, \mathbf{Y}) = \sqrt{{}^t(\mathbf{X} - \mathbf{Y}) \cdot (\mathbf{X} - \mathbf{Y})}. \quad (6.22)$$

Cette distance devient alors

$$d'(\mathbf{X}, \mathbf{Y}) = \sqrt{{}^t(\mathbf{X} - \mathbf{Y})\mathcal{Q}(\mathbf{X} - \mathbf{Y})} \quad (6.23)$$

où \mathcal{Q} est une matrice 3×3 symétrique définie positive.

Cette matrice \mathcal{Q} peut être décomposée de la façon suivante :

$$\mathcal{Q} = {}^t\mathcal{R} \begin{pmatrix} 1/a^2 & 0 & 0 \\ 0 & 1/b^2 & 0 \\ 0 & 0 & 1/c^2 \end{pmatrix} \mathcal{R}. \quad (6.24)$$

Cette distance aura pour effet d'étirer les distances d'un facteur a , b , ou c suivant les trois directions de l'espace définies par la matrice de rotation \mathcal{R} . Ainsi, les volumes élémentaires ne sont plus des sphères, mais des ellipsoïdes dont a , b et c représentent les dimensions (plus précisément les longueurs des trois demi-axes de l'ellipsoïde). Nous allons utiliser cette expression de la distance en lieu et place de la distance euclidienne dans l'expression de f_i . Pour tout point \mathbf{X} de l'espace, cette distance s'écrit donc

$$d'(\mathbf{X}, \mathbf{C}_i) = \sqrt{q_i(\mathbf{X})} = \sqrt{{}^t(\mathbf{X} - \mathbf{C}_i)\mathcal{Q}(\mathbf{X} - \mathbf{C}_i)}. \quad (6.25)$$

On peut ensuite créer notre fonction $f_i(\mathbf{X})$ à partir de cette distance $q_i(\mathbf{X})$ au point \mathbf{C}_i :

$$f_i(\mathbf{X}) = e^{-\frac{(q_i(\mathbf{X}) - R_i^2)}{\nu}}. \quad (6.26)$$

Cette expression de f_i nous permettra d'obtenir non plus des sphères, mais bien des ellipsoïdes, dont l'orientation est définie par la matrice \mathcal{R} . Si l'on choisit 1 comme isovaleur pour la surface, alors les dimensions de cet ellipsoïde correspondront aux termes aR_i , bR_i , et cR_i .

6.3.4 Distance à une surface implicite

Motivation

S'il est relativement aisé de déterminer l'endroit où passe une surface implicite, ou de déterminer si un point est à l'intérieur ou bien à l'extérieur du volume correspondant simplement en examinant le signe pris par la fonctionnelle f au point considéré, il est nettement plus délicat d'estimer la distance entre un point donné et la surface. Pourtant, cette information de distance peut être très utile lorsque l'on souhaite, par exemple, ajuster la position d'une isosurface sur un nuage de points.

Il est d'usage d'utiliser la quantité $|f(\mathbf{X}) - k|$ pour obtenir une estimation de la distance d'un point \mathbf{X} à la surface. Cependant, il est bien évident que cela ne constitue généralement

pas une véritable distance. Il suffit pour s'en convaincre de constater que lorsque l'on s'éloigne infiniment de l'objet, cette distance tend vers l'isovaleur k , sans diverger. Elle n'a pas, loin s'en faut, toutes les propriétés d'une distance de type euclidienne.

Parmi les défauts que l'on peut noter, cette expression pour la distance à l'isosurface se comporte différemment pour de gros et petits éléments, ou selon les axes dans le cas d'éléments ellipsoïdaux, devient rapidement non-linéaire lorsque l'on s'éloigne de la surface, etc. Les problèmes sont acceptables dans le cas où la surface est utilisée principalement à des fins d'affichage (encore que ces limitations soient néanmoins gênantes, les éléments de grande taille ayant tendance à avaler les plus petits, par exemple), mais dans le cas où l'on a réellement besoin d'une estimation de la distance entre un point et la surface, il faut trouver d'autres solutions.

Pseudo-distance à un ellipsoïde

Trouver très exactement la distance euclidienne entre un point et une isosurface n'est pas envisageable dans un temps relativement court : dans le simple cas de la distance d'un point à un ellipsoïde, il faut résoudre une équation du sixième degré, quatrième si l'on a une symétrie de révolution. On peut toutefois s'en approcher quelque peu. Plutôt que de considérer la distance euclidienne \mathbf{XH} , nous nous intéresserons à la distance \mathbf{XR} qui est la longueur du segment porté par \mathbf{CX} qui sépare le point de l'ellipsoïde. Nous l'appellerons "pseudo-distance" dans la suite, et nous la noterons d' .

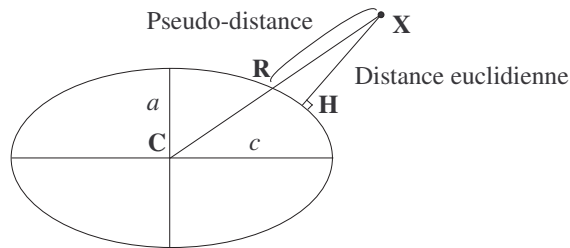


FIG. 6.11 – Comparaison des distances euclidienne et pseudo-euclidienne.

Cette pseudo-distance d' a des propriétés intéressantes. On peut tout de suite constater qu'elle coïncide avec la distance euclidienne d dans le cas d'une sphère. Si l'on se place dans le cas d'un ellipsoïde de révolution, tant que l'excentricité c/a de l'ellipsoïde est bornée, elle se révèle équivalente à la distance euclidienne. On peut en effet montrer que $d \leq d' \leq d \frac{c/a + a/c}{2}$ (la démonstration pourra être trouvée dans l'annexe B). Pour des ellipsoïdes peu allongés, notre pseudo-distance d' se révèle donc être une approximation acceptable de la distance euclidienne.

6.3.5 Calcul de la pseudo-distance à un ellipsoïde

La pseudo-distance précédemment introduite se révèle beaucoup plus aisée à calculer que la distance euclidienne. On a très simplement

$$d'(\mathbf{X}) = |\mathbf{CX}| - |\mathbf{CR}| = |\mathbf{CX}| \left(1 - \frac{1}{\sqrt{q(\mathbf{X})}} \right). \quad (6.27)$$

Dans le cas particulier d'un ellipsoïde de révolution, l'expression de $q(\mathbf{X})$ peut même être explicitée, et l'on obtient :

$$d'(\mathbf{X}) = |\mathbf{CX}| - \frac{a}{\sqrt{1 + \frac{(\mathbf{u} \cdot \mathbf{CX})^2}{|\mathbf{CX}|^2} \left(\frac{a^2}{c^2} - 1 \right)}}. \quad (6.28)$$

Retour sur les surfaces implicites

Afin de profiter pleinement des avantages de cette distance, nous allons nous en servir dans l'expression même des différentes composantes f_i de l'isosurface. Plutôt que de considérer les points à une distance donnée R d'un point (que la distance utilisée soit euclidienne ou non), nous nous intéresserons aux points dont la distance d' à l'ellipsoïde lui-même est nulle. Pour chacun des ellipsoïdes i composant notre objet (définis par la donnée d'un centre \mathbf{C}_i , d'une orientation \mathbf{u}_i et des dimensions a_i et c_i), nous utiliserons le terme :

$$f_i = e^{-\frac{d'_i(\mathbf{X})}{\nu}} \quad (6.29)$$

où d'_i est l'expression 6.28 présentée plus haut.

La fusion s'effectue de la façon usuelle, l'isosurface considérée correspond donc à l'équation :

$$f(\mathbf{X}) = \sum_i f_i(\mathbf{X}) = \sum_i e^{-\frac{d'_i(\mathbf{X})}{\nu}} = 1. \quad (6.30)$$

Le résultat obtenu avec cette expression particulière des f_i est similaire à celui obtenu avec les autres expressions précédemment citées (blobs, metaballs, soft objects). En fait, comme on peut le voir sur la figure 6.12, elle donne même des résultats de meilleure qualité lors de la fusion. L'utilisation d'un terme plus proche d'une véritable distance a en effet éliminé bon nombre des effets indésirables précédemment cités.

D'une part, les expressions de f_i usuellement employées montrent une nette anisotropie le long du grand axe des ellipsoïdes. Cette anisotropie provient de l'étirement d'une fonction sphérique selon une direction particulière. La fonction f_i décroît moins rapidement le long de cet axe privilégié, de sorte que la fusion se fait préférentiellement, et à plus grande distance, suivant cet axe. Notre expression de f_i ne fait pas montre de cette anisotropie.

D'autre part, les ellipsoïdes de grande taille et ceux de petite taille sont d'ordinaires considérés comme des éléments semblables, mais à une échelle différente. La fonction f_k

décroit donc plus rapidement au voisinage des petits ellipsoïdes qu’au voisinage des grands, qui indiquent une propension à “avaler” ceux de taille moindre. Cela dû au fait que les distances à l’ellipsoïde sont en quelques sortes comptées en “unités de rayon”. Cet effet disparaît également avec l’usage de notre distance pseudo-euclidienne, et les metaballs de plus grande taille n’étouffent plus ceux de taille moindre.

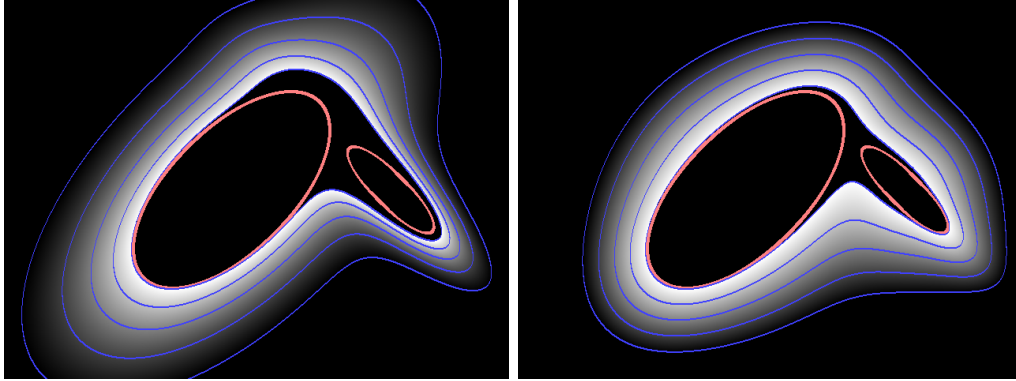


FIG. 6.12 – Isosurface et courbes de distance (en coupe) obtenues par fusion de deux ellipsoïdes. À gauche, les expressions classiques montrent des défauts (anisotropie, gonflement, non-linéarités) qui sont corrigées à droite par l’usage de la distance pseudo-euclidienne.

D’autres inconvénients des expressions classiques des isosurfaces tiennent dans le fait que l’on utilise des termes quadratiques et non linéaires pour le calcul de la distance. La raison en est essentiellement le coût souvent élevé du calcul d’une racine carrée, en particulier sur les machines datant d’il y a quelques années. Dans le domaine de l’infographie, les inconvénients de l’usage d’une distance fortement non-linéaire sont souvent suffisamment modérés pour que ce ne soit pas un réel souci. Rien, toutefois, n’empêche l’usage d’une racine carrée dans les expressions classiques pour régulariser les distances, si le besoin s’en fait sentir.

Pseudo-distance à la surface implicite

On peut simplement obtenir la pseudo-distance d' d’un point \mathbf{X} au metaball i en calculant la quantité $-\nu \ln(f_i(\mathbf{X}))$. Pour estimer la distance de ce même point \mathbf{X} à l’isosurface, on supposera par analogie que l’on peut simplement évaluer la quantité suivante :

$$d'(\mathbf{X}) = -\nu \ln(f(\mathbf{X})). \quad (6.31)$$

Cette expression n’est bien évidemment au mieux qu’une approximation de la distance à la surface. Au voisinage d’un ellipsoïde isolé, on retrouve l’expression de la pseudo-distance. Comme on peut le constater sur la figure 6.12, lorsque plusieurs éléments sont fusionnés, elle reste une bonne approximation de la distance à la surface, en témoignent les différentes lignes de niveaux rassemblant les points considérés comme à distance égale à la surface. Cette approximation sera suffisante pour l’usage que l’on en fera par la suite.

6.3.6 Modèle géométrique de main

Une main peut être obtenue au moyen d’une surface implicite décrite comme précédemment avec simplement seize ellipsoïdes, un pour chacune des parties articulées de la main. Les seuls défauts notables se situent au niveau de la paume, un peu trop arrondie. Il serait utile, éventuellement, d’y adjoindre quelques ellipsoïdes supplémentaires pour la décrire de façon plus fine. Le modèle obtenu avec seize ellipsoïdes seulement est toutefois d’ores et déjà satisfaisant.

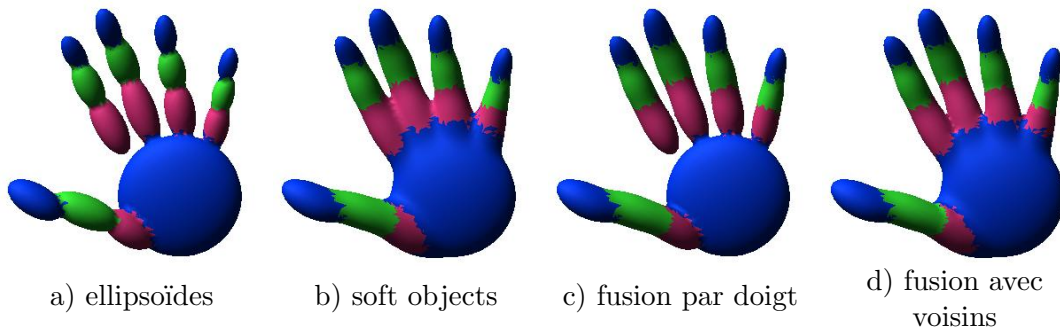


FIG. 6.13 – Différentes façons de fusionner les ellipsoïdes présentés à gauche : simple fusion selon le modèle des soft objects, fusion des ellipsoïdes appartenant à un même doigt, et fusion des ellipsoïdes avec leurs voisins dans la chaîne articulaire.

On constate toutefois un léger défaut si l’on se contente d’utiliser les surfaces implicites définies comme précédemment. En effet, la description par metaballs permet bien de fusionner les différentes phalanges d’un même doigt comme souhaité, mais si deux doigts se rapprochent, ils auront tendance à fusionner également, ce qui pourrait se trouver quelque peu gênant, la main se révélant très souvent “palmée”. Nous souhaiterions évidemment éviter d’ajouter les termes venant de phalanges appartenant à des doigts différents.

Habituellement, le problème est résolu en séparant le solide en plusieurs parties distinctes. Ici, nous aurions pu considérer chacun des cinq doigts et la paume comme six volumes déformables distincts, puis assemblés. Le résultat obtenu est montré dans la figure 6.13-c. Cela n’est hélas pas non plus pleinement satisfaisant. En effet, les doigts ne sont plus solidaires de la paume, et donc la forme prise par la surface au niveau de la base des doigts n’est pas convaincante, et présente de sérieux problèmes de régularité.

C’est pourquoi nous avons préféré une approche différente, hiérarchique, en limitant la sommation au terme de plus forte contribution et aux éventuelles phalanges voisines du même doigt, $f(\mathbf{X}) = \sum_{p \text{ vois.}} f_p(\mathbf{X})$, ou, pour être plus précis, aux nœuds voisins dans l’arbre que constitue la géométrie de l’objet : on commence par chercher en un point \mathbf{X} de l’espace quelle est la contribution f_p la plus grande, puis on ajoute à cette contribution tous les termes f'_p correspondant à des parties de l’objet liées à celle décrite par f_p . La description finale obtenue pour la main est présentée sur la figure 6.13-d. Bien que simple, elle est suffisamment proche de la réalité pour que l’on puisse espérer l’ajuster aux mesures.

Le volume ainsi obtenu n’est pas parfaitement lisse en raison du fait que la sommation est réservée uniquement aux plus proches voisins. Cependant, les discontinuités observées dans



FIG. 6.14 – Notre modèle de main, obtenu par la fusion des seize éléments.

le cas de la main sont inférieures au millimètre, ce qui est de toute façon bien en-deçà des imprécisions inévitables du modèle si l'on considère les nombreuses déformations et plis des tissus.

6.4 Habillage de l'isosurface

6.4.1 Nécessité de l'habillage

Certains types de suivi de mouvement, dont ceux que nous espérons mettre en œuvre, supposent que l'on puisse prédire le mouvement de points d'intérêt lorsque l'on agit sur les articulations de l'objet à suivre. Ces méthodes cherchent en effet, par exemple, à déterminer les mouvements articulaires permettant d'obtenir un mouvement prédit des points d'intérêt conforme à celui observé dans les images obtenues par les caméras.

Il nous faut pouvoir déduire le mouvement d'un point quelconque à la surface de l'objet, en fonction des variations dans les paramètres décrivant notre modèle. Il n'est pas correct, en général, de simplement supposer que les points se déplacent de façon rigide par rapport à un des éléments (par exemple, l'élément le plus proche). En effet, dans le cas d'objets recouverts de peau, par exemple, on observe des déformations et des plis au niveau des jointures : la peau s'étend, se rétracte et se plie lorsque les paramètres articulaires changent. Il nous faut donc pouvoir décrire, au moins de façon grossière, ces déformations afin d'être en mesure de prédire correctement le mouvement de points situés à ces endroits.

6.4.2 Principe du "skinning"

Pour trouver une solution à ce problème, il est instructif de se tourner vers des techniques dites de "skinning", couramment utilisées en animation. Afin de donner des résultats visuellement acceptables, il est apparu indispensable de poser des textures sur les différents objets à animer, par exemple des images imitant le grain de la peau, qui donneront une impression visuelle plus acceptable qu'une surface unie.

Toutefois, il est absolument nécessaire que lorsque l'objet se déplace, ces textures bougent en accord. Si ce n'est pas le cas, elles donneront au mieux l'impression de "glisser" sur l'objet, ce qui ruinera l'effet recherché. Lorsque l'on a affaire à des objets parfaitement rigides, il est facile de lier les différentes textures aux objets. Dans le cas d'objets déformables décrits par exemple par des surfaces de Bezier, là encore, le problème n'en est pas vraiment un, la texture, liée aux points de contrôle, se déformera aisément en même temps que la surface elle-même sans intervention de l'utilisateur.

En revanche, la question se trouve être plus délicate lorsque la surface est obtenue au moyen de fonctions implicites comme celles que nous utilisons. Pour permettre le placage d'une texture, il faut pouvoir associer à chaque point à la surface de l'objet une paire de coordonnées. En quelque sorte, il faut trouver une paramétrisation de la surface implicite.

Dans le cas de surfaces de type "Metaballs" comme celles que nous utilisons, des solutions ont pu être trouvées dans le cadre de l'animation [Blo02]. L'idée consiste à utiliser, pour un point donné de la surface, le potentiel correspondant à chacun des différents éléments en ce point pour en déduire une "position". En effet, lorsque l'on suit la surface en s'éloignant progressivement d'un élément pour s'approcher d'un autre, ces potentiels varient continûment, et il est donc envisageable de s'en servir comme paramétrisation.

6.4.3 Application à la prédiction de mouvement d'un point

Dans le cas de l'animation, cette remarque ne suffit pas à régler complètement le problème. Notamment, on n'obtient généralement, de cette façon, de paramétrage que selon une seule direction. Toutefois, notre propre but n'est pas de texturer l'objet que l'on souhaite étudier, mais de pouvoir prédire le mouvement d'un point donné à la surface de l'objet déformable entre deux positions données du squelette sous-jacent. On pourrait évidemment chercher dans tout l'espace un point à la surface de l'objet ayant les mêmes coordonnées de textures que le point considéré dans la position précédente, mais cette recherche serait trop coûteuse en temps de calcul pour être exploitable.

Nous avons préféré repartir de l'idée que les potentiels des différents éléments du squelette nous donnent des informations utiles sur la position d'un point sur la surface, qui nous permettront d'estimer le mouvement de ce point. Dans notre cas, nous avons supposé qu'il était possible d'approcher le mouvement d'un point quelconque sis sur la surface de l'objet par une combinaison linéaire des mouvements de chacun des éléments de la chaîne articulaire. Ces informations sur la position du point à la surface relativement aux différents éléments devraient nous permettre de choisir des poids appropriés pour cette combinaison linéaire.

6.4.4 Déformations de la peau

Cette hypothèse ne saurait se révéler justifiée dans bien des cas, mais lorsque l'on a affaire essentiellement à des extensions relativement modérées au niveau des joints des articulations, elle se révèle suffisamment correcte, comme nous allons le voir, pour l'usage que l'on souhaite en faire.

Dans le cas de la main, les déformations au voisinage d'une articulation trouvent quasiment exclusivement leur origine dans le mouvement relatif des deux éléments liés par cette articulation. C'est pourquoi, pour un point donné à la surface du solide, nous considérerons que son mouvement peut être décrit par une combinaison linéaire du mouvement de l'élément le plus proche p , et du *second* élément le plus proche p' du point choisi parmi les voisins de p :

$$\mathcal{T}(\mathbf{X}) = \frac{f_p(\mathbf{X})^\alpha \mathcal{T}^p \mathbf{X} + f_{p'}(\mathbf{X})^\alpha \mathcal{T}^{p'} \mathbf{X}}{f_p(\mathbf{X})^\alpha + f_{p'}(\mathbf{X})^\alpha}. \quad (6.32)$$

Les poids utilisés dans la combinaison linéaire dépendent directement de la distance du point à chacun des éléments p et p' . Plus le point est loin de l'élément p' , plus il sera rigidement lié à p .

Le paramètre α permet d'ajuster le comportement des points proches des joints du modèle (voir figure 6.15), pour lesquels les distances aux éléments p et p' tendent à être comparables. Un paramètre α faible donne des transitions très douces et interpolées, tandis que l'on retrouve un comportement où tous les points sont pratiquement rigidement liés à l'un des solides lorsque α tend vers l'infini (même si la continuité est néanmoins préservée).

Dans le cas de la main, un paramètre α de l'ordre de 2 semble être un bon compromis, ainsi qu'on peut le voir sur le schéma 6.15. On obtient ainsi un comportement réaliste, à la fois sur l'extérieur où la peau se tend, comme à l'intérieur de l'articulation où l'on observe la formation d'un pli.

Il convient de noter que l'on fait ici une simple estimation du mouvement d'un point situé à la surface de l'objet. Le point ainsi déplacé peut ne pas se trouver précisément à la surface de l'objet dans sa nouvelle position, en particulier si les déplacements relatifs des deux éléments est important. Ce n'est cependant pas en soi réellement problématique. Pour des variations modérées des paramètres articulaires, les points restent approximativement situés à la surface du modèle. En outre, rappelons encore une fois que cette "déformation" de la peau reste une approximation, dans la mesure où les déformations et plis réellement observés peuvent être plus complexes.

Toutefois, il convient de noter que cette difficulté à reproduire très exactement les déformations de la peau (aucun modèle, si riche soit-il, ne peut espérer modéliser pleinement ces déformations, le simple contact avec un objet pouvant induire des mouvements de la peau sans même que le squelette bouge) rend tout suivi de points d'intérêt entaché d'erreurs contre lesquelles il est impossible de se prémunir complètement. Il conviendra donc, lors du suivi, de prendre en compte la possibilité d'erreurs dans une estimation du mouvement basée sur le mouvement de points d'intérêt, erreurs qui peuvent rapidement devenir importantes, par accumulation.

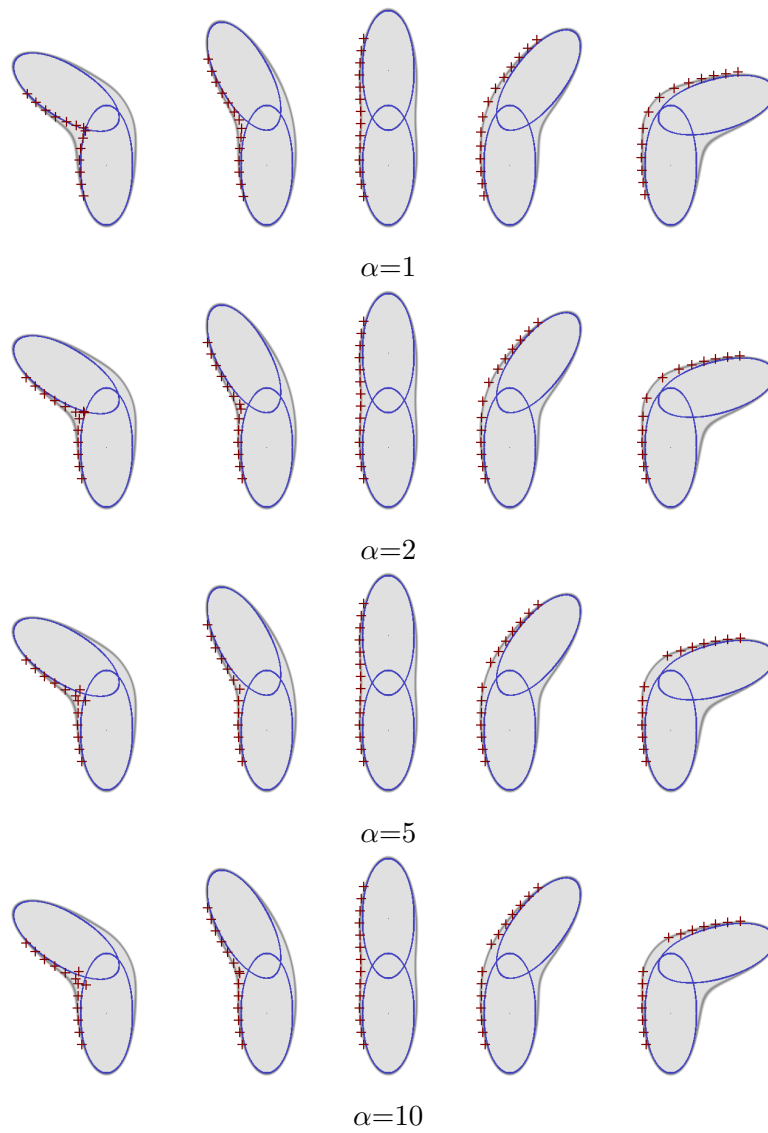


FIG. 6.15 – Le déplacement des points à la surface du modèle dépend du choix du paramètre α . La surface du modèle est représentée ici en gris.

Chapitre 7

Suivi robuste d'objets rigides

7.1 Introduction au problème du suivi de mouvement

7.1.1 Deux approches différentes

Ainsi que nous l'avons laissé entendre dans le chapitre 5, il existe deux approches permettant de déterminer la position d'un objet à un instant $t + 1$ connaissant celle qu'il occupait à l'instant t . On peut soit chercher une position dans le voisinage de la dernière position connue qui corresponde aux observations que l'on peut faire à l'instant t , ou bien déterminer le mouvement de l'objet entre les deux instants, et utiliser ce mouvement pour en déduire la nouvelle position de l'objet. Dans le domaine du suivi de la main, c'est la première approche qui est généralement utilisée, même si récemment on a vu apparaître des méthodes qui prenaient appui sur une estimation du déplacement pour guider l'estimation de la nouvelle position [LMSO03].

Nous pensons également, et nous nous efforcerons de le montrer dans le présent chapitre, que ces approches ne sont pas antagonistes, mais que bien au contraire elles peuvent se compléter à merveille, de façon à ce que chacune d'entre elles compense les défauts de l'autre. Il nous est en effet apparu que chacune des deux approches, utilisées de façon isolée, ne pouvait convenir à l'usage que nous souhaitions en faire, mais que combinées, elles donnent des résultats des plus encourageants.

7.1.2 Organisation du chapitre

Dans un premier temps, nous présenterons brièvement le dispositif que nous avons mis en place pour permettre le suivi de mouvements, ainsi que les différents moyens mis en œuvre pour obtenir les données qui serviront de base au suivi proprement dit. Nous aborderons les difficultés pouvant survenir dans cette phase d'extraction des données, afin de savoir quels écueils il nous faudra éviter dans la suite.

Puis nous allons successivement étudier différentes méthodes permettant le suivi d'un ob-

jet. Nous commencerons par nous intéresser aux techniques utilisant le mouvement de points d'intérêt extraits des images pour déduire le déplacement, puis la position de l'objet étudié. Nous reviendrons en détail sur les méthodes les plus couramment utilisées pour l'estimation conjointe de la transformation et des mises en correspondance des points dans les sections 7.3.3, 7.3.4 et 7.3.5.

Puis nous présenterons notre propre méthode dans la section 7.3.6, fondée sur l'algorithme EM, qui permet un suivi robuste de l'objet grâce à ces points d'intérêt, grâce à une élimination automatique des points aberrants que nous essaierons de justifier d'un point de vue statistique. Ensuite, nous verrons comment des méthodes d'ajustement à un modèle permettent d'obtenir de façon différente cette même position, et comment les idées de EM peuvent inspirer une élimination similaire des points aberrants.

Nous discuterons des défauts respectifs de ces deux approches, qui comme nous le verrons dans la section 7.5, peuvent être contournés. Il est en effet possible de les combiner afin d'augmenter tout à la fois rapidité et robustesse du suivi.

Afin de simplifier les descriptions qui vont suivre, en particulier les expressions des différents critères, nous présenterons, dans ce chapitre, ces différentes méthodes dans le cas particulier du suivi d'objets indéformables. Nous verrons dans le chapitre suivant qu'elles peuvent être naturellement étendues au cas d'un solide articulé et déformable, pourvu que le modèle le décrivant nous fournisse un certain nombre d'informations quant à sa cinématique, sa géométrie et ses déformations éventuelles.

7.2 Présentation du dispositif

7.2.1 Un système multi-caméras

Dans notre cas, nous nous intéresserons à des données provenant de sources vidéo, plus précisément des séquences d'images obtenues à l'aide de caméras. La scène, principalement les mains de l'utilisateur dans le cas qui nous intéresse, est filmée par plusieurs caméras, sous différents angles. Nous souhaitons pouvoir utiliser des données tridimensionnelles obtenues avec ce dispositif, donc deux caméras au moins seront nécessaires (encore qu'il soit tout à fait possible de n'utiliser qu'une seule caméra équipée d'un jeu de miroirs permettant d'obtenir deux points de vue distincts dans la même image, ainsi que le proposent les travaux de Quentin Delamarre et Olivier Faugeras [DF99, Del03]).

Des images de ces deux caméras, nous extrairons par exemple des points d'intérêt, obtenus avec une méthode telle que le détecteur de Harris. Ces points d'intérêt, une fois appariés, permettent d'obtenir, par reconstruction, un nuage de points 3D. Pour ce faire, il est nécessaire d'avoir au préalable calibré la scène, de façon à pouvoir déterminer la géométrie épipolaire liant les deux caméras.

7.2.2 Efficacité de la reconstruction 3D sur la main

Ces différents points 3D reconstruits correspondent, en principe, à des éléments visibles des objets filmés. Dans notre cas, par exemple, une bonne part de ces points correspondent à des détails sur la peau des mains de l'utilisateur : plis, taches et autres imperfections. Même en l'absence de marqueurs, et malgré son apparence assez uniforme, la peau peut fournir quantité de détails à condition d'avoir pris soin d'utiliser un éclairage adéquat et de s'être assuré une netteté suffisante de l'image.

La plupart de ces détails se trouvent au niveau des plis de la peau, à l'articulation des phalanges et au voisinage de la paume. Ce ne sont malheureusement pas les endroits idéaux pour le suivi d'un objet articulé, puisqu'ils se situent dans des zones où les déformations sont importantes. En outre, d'une image à l'autre, ces éléments peuvent apparaître ou disparaître tandis que la peau se tend et se détend, et il paraît des plus difficile d'établir une sorte de "carte de texture" qui serait valable pour toutes les positions de la main.

Les extrémités des ongles sont également potentiellement de très bons candidats à fournir des points d'intérêt de qualité, fiables et précis, lorsqu'ils apparaissent dans les images des différentes caméras. Ce qui se révèle une aubaine, dans la mesure où un bon positionnement de l'extrémité des doigts est particulièrement utile lorsque l'on souhaite retrouver une description complète de la position de la main et de ses articulations.

Pour parfaire ce positionnement, il faut également pouvoir situer la paume de façon précise. Si l'intérieur de la paume est en général richement pourvu de détails, il n'en est pas toujours de même pour le dessus de cette dernière. Cependant, il est en général possible d'y trouver suffisamment de défauts pour disposer d'un nombre satisfaisant de points d'intérêt. Dans de bonnes conditions, donc, on peut disposer *a priori* de suffisamment de points 3D pour permettre le suivi de mouvement sans avoir besoin d'ajouter de marqueurs supplémentaires. Les peaux les plus lisses, cependant, sont susceptibles de poser problème.

7.2.3 Points 3D parasites

En pratique, compte tenu des difficultés inhérentes à l'extraction du fond d'une image, et à la présence possible d'autres objets dans le champ des caméras, on reconstruira inévitablement des points 3D qui ne correspondent pas directement à l'objet de l'étude. Il est parfois possible de limiter le nombre de points 3D parasites reconstruits, simplement en utilisant un arrière-plan uniforme, un fond fixe ou bien encore identifiable, et propre à une extraction. Cependant, ce genre de traitement n'est pas toujours applicable, et nous verrons comment il est possible d'éliminer l'essentiel de ces points de façon automatique lors du suivi.

En outre, la mise en correspondance des différents points, malgré les contraintes épipolaires, n'est pas toujours exempte de défauts. En particulier, les éléments repérés à la surface de la peau sont très souvent redoutablement similaires les uns aux autres, et des appariements erronés ne sont pas rares. De sorte qu'il apparaît une quantité importante de points 3D dont la position ne correspond à rien de réel, dispersés dans tout l'espace, et ce même avec des images de qualité. Là encore, nous aurons à éliminer ces points et nous assurer qu'ils ne conduisent pas notre dispositif de suivi vers des solutions incorrectes.

7.2.4 Enrichissement des informations

Comme dit précédemment, il n'est besoin que de deux caméras pour permettre la reconstruction du nuage de points 3D que nous utiliserons par la suite. Cependant, plus on disposera de caméras, meilleures seront les informations à notre disposition. D'une part, avec plus de deux caméras, il est possible d'augmenter la fiabilité de la reconstruction 3D des différents points, et limiter l'apparition de points aberrants. Comme nous disposerons d'une méthode pour rejeter les points par trop erronés, ce n'est cependant pas le principal avantage que nous voyons à l'utilisation d'un nombre plus élevé de caméras.

En fait, la principale difficulté pour le suivi de la main est lié aux très nombreuses occultations que l'on ne manquera pas de constater. A moins d'imposer à l'utilisateur de restreindre ses mouvements à ceux parfaitement visibles par la caméra, une bonne partie de la main risque d'être souvent invisible pour une paire stéréo de caméras. Il apparaît donc des plus souhaitable (même si ce n'est pas rigoureusement indispensable), de disposer d'au moins une seconde paire, que l'on placera avantageusement selon un angle très différent de la première, par exemple de l'autre côté de la scène.

7.3 Suivi point à point

7.3.1 Introduction

Principe du suivi point à point

Nous allons donc, dans un premier temps, nous intéresser aux méthodes qui infèrent le déplacement de l'objet à partir du mouvement d'éléments qui lui sont rigidement liés. Dans notre cas, ces éléments seront des points 3D, visibles par les caméras, et situés à la surface de l'objet.

Nous supposerons, pour l'instant, que l'on connaît la position de notre objet à l'instant t , et que l'on souhaite connaître sa position à l'instant suivant $t + dt$. Pour ce faire, on dispose de deux nuages de points, correspondant respectivement à ces deux instants t et $t + dt$. Nous les noterons dans la suite \mathbf{X}_i et \mathbf{Y}_j . Nous supposerons également qu'aux occultations et erreurs de reconstruction près, les points issus des deux nuages correspondent, dans leur majorité, aux mêmes points physiques.

Objectif

Nous cherchons à déterminer le déplacement $\mathcal{T}_{t,t+1}$ (ici la combinaison d'une translation et d'une rotation) qu'a subi l'objet étudié entre les deux instants. Les points \mathbf{X}_i étant rigidement liés à l'objet lui-même (nous nous affranchissons, dans un premier temps, de toute considération sur des points ne correspondant pas à l'objet suivi), leur mouvement doit être décrit par cette même transformation $\mathcal{T}_{t,t+1}$. Ainsi, après un déplacement $\mathcal{T}_{t,t+1}$ de l'objet, le point \mathbf{X}_i se retrouvera à la position $\mathcal{T}_{t,t+1} \mathbf{X}_i$.

Si ce point \mathbf{X}_i est encore visible à l'instant $t + dt$, il correspond à un point \mathbf{Y}_j avec $j = c(i)$ où c est une fonction qui apparie les points du nuage \mathbf{X}_i avec ceux du nuage \mathbf{Y}_j . La transformation $\mathcal{T}_{t,t+1}$ doit donc amener les \mathbf{X}_i sur les $\mathbf{Y}_{j=c(i)}$ correspondants. Autrement dit, nous cherchons la transformation $\mathcal{T}_{t,t+1}$ qui minimisera les distances entre les positions prédites des points \mathbf{X}_i à l'instant $t + dt$, c'est à dire $\mathcal{T}_{t,t+1} \mathbf{X}_i$, et les positions effectivement observées \mathbf{Y}_j .

Il existe de nombreuses façons de construire un critère de distance entre deux nuages de points, mais le plus fréquent consiste à utiliser la somme des distances quadratiques séparant les points deux à deux, car ce sont les critères qui se comportent le mieux vis-à-vis des algorithmes d'optimisation. Le but est donc de déterminer la transformation $\mathcal{T}_{t,t+1}$ qui va minimiser l'expression suivante :

$$E_p(\mathcal{T}_{t,t+1}) = \sum_{i \in [1, N_t]} \|\mathcal{T}_{t,t+1} \mathbf{X}_i - \mathbf{Y}_{c(i)}\|^2 \quad (7.1)$$

Appariement des points

L'appariement des points entre les deux instants successifs n'est pas toujours chose aisée, et on ne trouvera généralement pas une fonction bi-univoque qui associe les points \mathbf{X}_i aux \mathbf{Y}_j . En effet, nombre de points visibles à l'instant t pourront ne plus l'être à l'instant suivant, parce qu'ils n'ont pu être reconstruits correctement, ou simplement parce qu'ils ont été masqués à l'occasion d'une quelconque occlusion. Inversement, il peut apparaître à l'instant $t + dt$ des points qui n'étaient pas visibles à l'instant précédent, pour des raisons similaires.

La notation $c(i)$ est donc généralement maladroite, et guère commode à utiliser. On préfère définir une matrice d'association $\alpha_{i,j}$, telle que $\alpha_{i,j}$ prenne la valeur 1 lorsque deux points \mathbf{X}_i et \mathbf{Y}_j sont appariés, et 0 dans le cas contraire. Cette notation procure davantage de souplesse, puisque chaque point \mathbf{X}_i peut n'être associé à aucun \mathbf{Y}_j , à un unique point \mathbf{Y}_j voire à plusieurs d'entre eux, comme nous le verrons ultérieurement. De même, chaque \mathbf{Y}_j peut avoir un nombre quelconque de correspondants à l'instant t .

L'expression que l'on cherche à minimiser se réécrit alors :

$$E_p(\mathcal{T}_{t,t+1}) = \sum_{(i,j) \in [1, N_t] \times [1, N_{t+1}]} \alpha_{i,j} \|\mathcal{T}_{t,t+1} \mathbf{X}_i - \mathbf{Y}_j\|^2 \quad (7.2)$$

7.3.2 Cas des points non appariés

Difficultés à résoudre

Pour permettre le calcul de la transformation $\mathcal{T}_{t,t+1}$, il nous faut donc connaître les différents appariements entre les \mathbf{X}_i et \mathbf{Y}_j , c'est-à-dire pouvoir écrire la matrice $\alpha_{i,j}$. Tout dépend alors de la façon dont on a obtenu les différents nuages de points.

Dans le cas le plus favorable, on suit effectivement un ensemble de points d'intérêt dans les différentes séquences d'images. Le problème est alors pleinement résolu, et la matrice $\alpha_{i,j}$ est connue. On sait très précisément à quel \mathbf{Y}_j correspond un point \mathbf{X}_i (si toutefois ce dernier a bien un correspondant visible à l'instant $t + dt$). Toutefois, il peut survenir des erreurs dans le suivi, de sorte que certains appariements seront erronés.

Les nuages de points ne seront cependant pas toujours obtenus de cette façon. Bien souvent, les différents nuages seront obtenus par une simple extraction de points d'intérêt à chaque instant, et une reconstruction basée sur la géométrie épipolaire. Dans ce cas, il nous faudra explicitement déterminer les associations, inconnues *a priori*.

Il est possible d'essayer d'apparier les points en comparant les voisinages des points étudiés dans les différentes images, et de les associer selon les ressemblances. Toutefois, dans le cas de textures comme celle de la main, beaucoup de ces voisinages se ressembleront, rendant l'association incertaine. En outre, la peau étant déformable, les voisinages peuvent notablement changer entre deux images. On pourra voir disparaître un pli, ou bien une zone s'étirer quelque peu, déformant la texture au voisinage du point d'intérêt. Bref, si l'on peut avoir une estimation des appariements possibles, beaucoup d'erreurs risquent de se glisser dans la matrice $\alpha_{i,j}$. Il nous faut donc prévoir la possibilité de corriger cette matrice au cours du suivi, voire de la déterminer complètement.

Une démarche de minimisation alternée

Si la correspondance entre les points \mathbf{X}_i et \mathbf{Y}_j n'est pas connue, ou peut être erronée, le problème devient double. Il nous faut en effet tout à la fois déterminer la transformation recherchée $\mathcal{T}_{t,t+1}$, mais également la matrice des assignements $\alpha_{i,j}$. Plusieurs solutions à ce problème difficile ont été suggérées. Nous allons voir, dans la suite, les avantages et inconvénients de chacune d'entre elles. Ces approches ont en commun une démarche alternée pour l'estimation des inconnues $\mathcal{T}_{t,t+1}$ et $\alpha_{i,j}$.

En effet, lorsque l'on connaît les différents assignements $\alpha_{i,j}$, on peut en déduire la transformation qui nous intéresse $\mathcal{T}_{t,t+1}$. Mais il se trouve que l'inverse est également exact : lorsque l'on connaît la transformation correcte, il est aisé de déduire les différentes mises en correspondances idéales entre les points des deux nuages, les différents points \mathbf{X}_i ayant été déplacés à proximité de leur correspondant \mathbf{Y}_j . D'où l'idée d'essayer de déterminer tour à tour les assignements $\alpha_{i,j}$ en supposant la transformation $\mathcal{T}_{t,t+1}$ connue, puis d'évaluer la transformation $\mathcal{T}_{t,t+1}$ correspondant à ces assignements.

Les différences qui existent entre les approches proposées résident principalement dans la manière de construire la matrice $\alpha_{i,j}$, c'est-à-dire dans l'étape d'estimation des correspondances entre points. Toutes ensuite essaient de déterminer la transformation $\mathcal{T}_{t,t+1}$ qui minimisera le critère E_p pour des $\alpha_{i,j}$ déterminés. Il existe cependant plusieurs façons de calculer cette transformation, qui peuvent être utilisées indifféremment avec chacune des approches.

Estimation du déplacement

Supposons dans un premier temps les correspondances $\alpha_{i,j}$ connues. Le critère ne dépend alors plus que de la transformation $\mathcal{T}_{t,t+1}$. Celle-ci peut être déterminée de nombreuses façons, tant directes qu'itératives. Pour $\alpha_{i,j}$ donnée, on cherche à déterminer la rotation \mathcal{R} et la translation \mathbf{t} qui minimisent

$$E_p(\mathcal{R}, \mathbf{t}) = \sum_{(i,j)} \alpha_{i,j} \|\mathcal{R} \mathbf{X}_i + \mathbf{t} - \mathbf{Y}_j\|^2 \quad (7.3)$$

Après paramétrage de la rotation (par exemple au moyen des angles d'Euler), il est possible d'effectuer cette minimisation au moyen de techniques itératives d'optimisation numérique telles qu'une descente de gradient ou un filtrage de Kalman. Toutefois, dans ce cas particulier, il est intéressant de noter qu'il existe de nombreux moyens d'évaluer directement \mathcal{R} et \mathbf{t} , par exemple au moyen de la SVD.

Cette méthode, proposée par Arun *et al* en 1987 [AHB87] et reprise par Umeyama en 1991 [Ume91] permet d'obtenir \mathcal{R} et \mathbf{t} en exploitant le fait que la rotation et la translation peuvent être séparées dans l'expression de E_p . Dans un premier temps, on cherche à déterminer la translation \mathbf{t} qui minimisera le critère E_p pour une rotation \mathcal{R} donnée :

$$\frac{\partial E_p}{\partial \mathbf{t}} = 0 = 2 \sum_{(i,j)} \alpha_{i,j} (\mathcal{R} \mathbf{X}_i + \mathbf{t} - \mathbf{Y}_j)$$

Ce qui conduit à

$$\mathbf{t} = \frac{\sum_{(i,j)} \alpha_{i,j} \mathbf{Y}_j}{\sum_{(i,j)} \alpha_{i,j}} - \mathcal{R} \left(\frac{\sum_{(i,j)} \alpha_{i,j} \mathbf{X}_i}{\sum_{(i,j)} \alpha_{i,j}} \right) \quad (7.4)$$

En pratique, cela consiste à aligner le premier moment des deux nuages de points, à savoir le barycentre des différents points auxquels on a affecté les poids $\alpha_{i,j}$. On introduit cette translation \mathbf{t} dans l'expression de E_p , ce qui revient en pratique à se placer dans le repère barycentrique de chaque nuage. Le critère devient alors :

$$E_p(\mathcal{R}) = \sum_{(i,j)} \alpha_{i,j} \|\mathcal{R} \mathbf{X}'_i - \mathbf{Y}'_j\|^2$$

$$\text{où } \mathbf{X}'_i = \mathbf{X}_i - \frac{\sum_{(i,j)} \alpha_{i,j} \mathbf{X}_i}{\sum_{(i,j)} \alpha_{i,j}} \text{ et } \mathbf{Y}'_j = \mathbf{Y}_j - \frac{\sum_{(i,j)} \alpha_{i,j} \mathbf{Y}_j}{\sum_{(i,j)} \alpha_{i,j}}$$

On a fait disparaître la translation de l'expression de E_p , ce qui va nous permettre d'estimer \mathcal{R} . Il suffit de décomposer le critère $E_p(\mathcal{R})$ en trois termes distincts :

$$E_p(\mathcal{R}) = \sum_{(i,j)} \alpha_{i,j} \|\mathcal{R} \mathbf{X}'_i\|^2 + \sum_{(i,j)} \alpha_{i,j} \|\mathbf{Y}'_j\|^2 - 2 \sum_{(i,j)} \alpha_{i,j} \langle \mathcal{R} \mathbf{X}'_i, \mathbf{Y}'_j \rangle$$

Parmi ces trois termes, on remarque que les deux premiers sont constants vis-à-vis de la rotation, et n'interviendront donc pas dans la minimisation du critère. Il reste donc à maximiser le terme

$$\begin{aligned} \sum_{(i,j)} \alpha_{i,j} \langle \mathcal{R} \mathbf{X}'_i, \mathbf{Y}'_j \rangle &= \text{trace} \left(\sum_{(i,j)} \mathcal{R} (\alpha_{i,j} \mathbf{X}_i \cdot \mathbf{Y}_j^t) \right) \\ &= \text{trace}(\mathcal{R} \cdot \mathcal{K}^t) \text{ où } \mathcal{K} = \sum_{(i,j)} \alpha_{i,j} \mathbf{Y}_j \cdot \mathbf{X}_i^t \end{aligned}$$

Le maximum est atteint pour $\mathcal{R} = \mathcal{U} \cdot \mathcal{S} \cdot \mathcal{V}^t$ où \mathcal{U} et \mathcal{V} sont issus de la décomposition en valeurs singulières de \mathcal{K} ($\mathcal{K} = \mathcal{U} \cdot \mathcal{D} \cdot \mathcal{V}^t$) et où \mathcal{S} est la matrice diagonale $\text{diag}(1, 1, \dots, 1, \det(\mathcal{U}), \det(\mathcal{V}))$. Cela permet de calculer la rotation \mathcal{R} de la transformation recherchée. La translation \mathbf{t} peut alors en être déduite au moyen de l'expression 7.4.

De nombreuses autres méthodes permettent l'évaluation directe de la transformation, basées sur l'utilisation de quaternions [FH86, WSV91] ou des matrices orthogonales [HW88] et donnent des résultats équivalents tant en terme de vitesse que de précision du résultat.

7.3.3 L'algorithme ICP (ou *Iterative Closest Point*)

Mise en correspondance

Nous allons à présent examiner plusieurs démarches permettant l'évaluation de la matrice d'assignation $\alpha_{i,j}$, afin d'en cerner les forces et les limitations. Cette fois-ci, nous supposons que le déplacement $\mathcal{T}_{t,t+1}$ est connu, et nous cherchons à déterminer la matrice d'appariements $\alpha_{i,j}$ qui va minimiser notre critère E_p . Rappelons la forme que prend le critère à minimiser E_p :

$$E_p(\mathcal{T}_{t,t+1}, \alpha_{i,j}) = \sum_{(i,j) \in [1, N_t] \times [1, N_{t+1}]} \alpha_{i,j} \|\mathcal{T}_{t,t+1} \mathbf{X}_i - \mathbf{Y}_j\|^2 \quad (7.5)$$

Ce critère peut être réécrit sous la forme :

$$E_p = \sum_{i \in [1, N_t]} E_i \text{ avec } E_i = \sum_{j \in [1, N_{t+1}]} \alpha_{i,j} \|\mathcal{T}_{t,t+1} \mathbf{X}_i - \mathbf{Y}_j\|^2 \quad (7.6)$$

Nous allons nous placer dans le cas où chacun des points \mathbf{X}_i possède un et un seul correspondant dans le nuage de points \mathbf{Y}_j . Pour tout i , un seul des coefficients $\alpha_{i,j}$ vaudra donc 1, tous les autres étant nuls. La somme des termes E_i est alors réduite à un seul terme.

Si l'on n'exige pas que chaque point \mathbf{Y}_j ne peut avoir qu'un seul antécédent à l'instant t , chaque point \mathbf{X}_i peut librement choisir son correspondant parmi les points \mathbf{Y}_j . Il est alors possible de minimiser séparément chacun des termes E_i . Il restera donc simplement à trouver, pour tout point \mathbf{X}_i , le point \mathbf{Y}_j du nuage à l'instant $t + dt$ le plus proche de la position estimée de \mathbf{X}_i après le déplacement, à savoir $\mathcal{T}_{t,t+1} \mathbf{X}_i$. Les seules difficultés pouvant intervenir dans

cette recherche proviennent de la possible grande quantité de points à examiner, ce qui peut être résolu en découpant l'espace pour faciliter les recherches, par exemple au moyen de "k-d trees".

D'autres approches, dans lesquelles les mises en correspondance ne sont pas indépendantes, sont détaillées par [Zha93]. L'idée générale reste cependant toujours la même : choisir pour correspondant du point \mathbf{X}_i le point \mathbf{Y}_j dont la position est la plus proche possible de la position actuellement prédite $\mathcal{T}_{t,t+1}\mathbf{X}_i$.

L'algorithme ICP

La démarche itérative alternant cette détermination *a priori* des mises en correspondance, et une estimation de la transformation qui minimisera E_p pour ces mêmes mises en correspondance, porte le nom d'ICP (Iterative Closest Point). Elle est résumée, dans le cas d'un objet rigide, par l'algorithme 2.

Introduit au début des années 1990, notamment dans les travaux de Besl *et al* [BM92] et de Zhang [Zha94], ce principe a connu un succès croissant, et demeure le plus utilisé actuellement dans de nombreux domaines comme celui du recalage de surface.

Algorithme 2 Suivi d'un objet rigide avec ICP

initialiser $\mathcal{R}_0 \leftarrow Id$ et $\mathbf{t}_0 \leftarrow \mathbf{0}$

initialiser $k \leftarrow 0$

répéter

pour tout point \mathbf{X}_i :

déterminer le point prédit $\mathbf{X}'_i \leftarrow \mathcal{R}_k\mathbf{X}_i + \mathbf{t}_k$

déterminer le point \mathbf{Z}_i le plus proche de \mathbf{X}'_i parmi les \mathbf{Y}_j

fin pour

$k \leftarrow k + 1$

déterminer \mathcal{R}_k et \mathbf{t}_k minimisant $E_p(\mathcal{R}_k, \mathbf{t}_k) = \sum_i \|\mathcal{R}_k\mathbf{X}_i + \mathbf{t}_k - \mathbf{Z}_i\|^2$

jusqu'à ce que la différence entre \mathcal{R}_{k-1} et \mathcal{R}_k et entre \mathbf{t}_{k-1} et \mathbf{t}_k tombe en deçà d'un seuil, ou bien que $k > k_{max}$

Élimination des outliers

L'inconvénient de cette méthode, telle qu'elle a été décrite pour l'instant, est qu'elle n'est pas en mesure de prendre en compte le cas de points privés de correspondants, ou bien le problème de points aberrants. Certains points \mathbf{X}_i , en effet, n'auront pas de correspondants à l'instant suivant, parce que la reconstruction en a été mauvaise, ou simplement pour des raisons d'occlusions.

Pour prendre en compte ce genre de problème, on peut définir une distance maximale acceptable d_0 lors de la recherche du point le plus proche, dans l'étape d'estimation des correspondances. Si un point \mathbf{X}_i ne peut trouver de correspondant dans un voisinage de

rayon d_0 autour de $\mathcal{T}_{t,t+1}\mathbf{X}_i$, alors tous les $\alpha_{i,j}$ correspondants seront placés à 0 quel que soit j , et le point \mathbf{X}_i n'interviendra plus dans la phase d'estimation de la transformation.

Il restera à choisir avec soin cette distance, qui devra être assez grande au début de l'optimisation pour ne pas rendre tous les points orphelins (et donc tendre vers la solution triviale où tous les $\alpha_{i,j}$ sont nuls), mais suffisamment petite à la fin (de l'ordre du bruit sur la reconstruction des points) pour éliminer efficacement les mauvaises associations. L'utilisation d'ICP pour le suivi d'un objet rigide avec élimination automatique des points aberrants est résumée dans l'algorithme 3.

Algorithme 3 Suivi d'un objet rigide avec ICP et élimination des outliers

définir d_{max} : ordre de grandeur du mouvement à suivre

définir d_{min} : bruit sur les points 3D

initialiser $d \leftarrow d_{max}$

initialiser $\mathcal{R}_0 \leftarrow Id$ et $\mathbf{t}_0 \leftarrow \mathbf{0}$

initialiser $k \leftarrow 0$

répéter

pour tout point \mathbf{X}_i :

déterminer le point prédit $\mathbf{X}'_i \leftarrow \mathcal{R}_k \mathbf{X}_i + \mathbf{t}_k$

déterminer le point \mathbf{Z}_i le plus proche de \mathbf{X}'_i parmi les \mathbf{Y}_j

si $\|\mathbf{Z}_i - \mathbf{X}'_i\| < d$ **alors**

$\lambda_i \leftarrow 1$

sinon

$\lambda_i \leftarrow 0$

fin si

fin pour

$k \leftarrow k + 1$

déterminer \mathcal{R}_k et \mathbf{t}_k minimisant $E_p(\mathcal{R}_k, \mathbf{t}_k) = \sum_i \lambda_i \|\mathcal{R}_k \mathbf{X}_i + \mathbf{t}_k - \mathbf{Z}_i\|^2$

si $d > d_{min}$ **alors**

faire décroître d : $d \leftarrow cste \times d$

fin si

jusqu'à ce que la différence entre \mathcal{R}_{k-1} et \mathcal{R}_k et entre \mathbf{t}_{k-1} et \mathbf{t}_k tombe en deçà d'un seuil, ou bien que $k > k_{max}$

Limitations de ICP

Bien que les méthodes de type ICP soient souvent efficaces, le fait d'utiliser des valeurs soit unitaires soit nulles pour les coefficients $\alpha_{i,j}$ pose des problèmes de robustesse et de répétabilité. Le critère, tel qu'il est actuellement défini, n'est en effet continu que par morceaux. Une variation infinitésimale de la transformation peut conduire à un changement dans les appariements des différents points. On bascule alors vers une solution radicalement différente.

En outre, un point va participer pleinement à la minimisation, ou sera complètement

mis de côté. Il n'est pas possible de donner davantage d'importance à des appariements qui semblent plus plausibles que d'autres. Cette sélection des appariements introduit une seconde discontinuité gênante dans cet algorithme. C'est parmi les algorithmes exploitant la notion de "maximum de vraisemblance" que l'on trouvera solution à ces questions.

7.3.4 L'algorithme EM (ou *Expectation Maximization*)

Origine

Si l'on se penche sur notre problème, on identifie différents types paramètres que l'on souhaite estimer. D'une part, la position de l'objet suivi, exprimée ici à travers une rotation \mathcal{R} et une translation \mathbf{t} , et d'autre part un ensemble de paramètres $\alpha_{i,j}$ qui expriment les assignements entre les points obtenus aux deux instants successifs. Ces paramètres $\alpha_{i,j}$ ne nous intéressent pas particulièrement en eux-mêmes, c'est la position de l'objet qui représente le but de l'estimation. Toutefois, il est indispensable d'estimer ces paramètres afin d'être en mesure de déterminer la position de l'objet. On qualifiera parfois les paramètres $\alpha_{i,j}$ de variables "cachées", tandis que la rotation et la translation font office d'"inconnues" à estimer.

Pour une position et un appariement donné, le critère que l'on a construit nous permet d'estimer l'erreur commise (basée sur la distance entre la position prédite des points \mathbf{X}_i et celle des points \mathbf{Y}_j qui leur sont appariés). Cette erreur a toutes les raisons d'être minimale lorsque la position et les appariements sont corrects. L'objectif est donc de trouver ces positions et appariements qui correspondront à l'erreur la plus faible. Dans ce genre de problèmes, il est d'usage d'alterner une estimation des variables cachées et un calcul des inconnues qui minimiseront le critère pour ces mêmes variables cachées.

Principe

Dans beaucoup de situations, et c'est le cas de l'algorithme ICP, lorsque l'on doit estimer les variables cachées, on choisit simplement le jeu de variables le plus vraisemblable étant donné les valeurs courantes des inconnues. Cette façon de procéder n'est pas toujours idéale, comme nous l'avons déjà laissé entendre. Une idée assez ancienne, que l'on retrouve dès 1958 [Har58], propose plutôt de déterminer une distribution de probabilité, parmi tous les jeux de variables cachées possibles, de préférence à une série de variables bien particulière, puis d'utiliser cette distribution pour le calcul des inconnues.

L'idée consisterait donc à exprimer l'erreur comme une somme pour tous les assignements $a_{i,j}$ possibles de l'expression précédente. À chacun de ces appariements possibles correspond un poids $c_{\{a_{i,j}\}}$ d'autant plus important que l'appariement correspondant est probable étant donné les inconnues (rotation \mathcal{R} et translation \mathbf{t}) actuelles. Soit plus précisément :

$$E_p(\mathcal{R}, \mathbf{t}) = \sum_{\{a_{i,j}\}} c_{\{a_{i,j}\}} \left(\sum_{(i,j)} \alpha_{i,j} \|\mathcal{R} \mathbf{X}_i + \mathbf{t} - \mathbf{Y}_j\|^2 \right). \quad (7.7)$$

Cette expression se réécrit aisément sous la forme suivante :

$$E_p(\mathcal{R}, \mathbf{t}) = \sum_{(i,j)} \left(\sum_{\{a_{i,j}\}} c_{\{a_{i,j}\}} \alpha_{i,j} \right) \|\mathcal{R} \mathbf{X}_i + \mathbf{t} - \mathbf{Y}_j\|^2. \quad (7.8)$$

Soit plus simplement :

$$E_p(\mathcal{R}, \mathbf{t}) = \sum_{(i,j)} \alpha'_{i,j} \|\mathcal{R} \mathbf{X}_i + \mathbf{t} - \mathbf{Y}_j\|^2. \quad (7.9)$$

Autrement dit, le principal changement viendra de l'expression des coefficients $\alpha_{i,j}$. Plutôt que de prendre uniquement les valeurs 0 (points non appariés) et 1 (points appariés), les $\alpha_{i,j}$ prendront dorénavant des valeurs quelconques, exprimant la probabilité qu'ils soient appariés dans la position courante. Les premières utilisations du principe d'*Expectation Maximization*, abrégé en EM dans la suite, remontent aux travaux de Dempster, Laird et Rubin [DLR77], qui en particulier ont pu en montrer la convergence. Elles vont naturellement conduire à exprimer les coefficients $\alpha_{i,j}$ comme des probabilités.

Approche statistique de la question

Notre objectif est d'évaluer la transformation $\mathcal{T}_{t,t+1}$, mais pour ce faire, il nous manque l'association entre les points \mathbf{Y}_j et les \mathbf{X}_i . D'un point de vue statistique, ce genre de problème est qualifié de *problème à données manquantes*. Pour le résoudre, nous allons utiliser l'approche dite EM, pour *expectation maximisation*. Nous allons à présent formaliser le cadre dans lequel nous allons travailler en termes statistiques.

Associer un point \mathbf{X}_i à l'un des points \mathbf{Y}_j peut être vu comme un problème de classification : il nous faut ranger chacun des points \mathbf{X}_i dans une des classes que représentent les \mathbf{Y}_j .

Dans ce problème, l'association d'un point \mathbf{X}_i est vue comme un problème de classification parmi les différentes classes que représentent les points \mathbf{Y}_j . Nous disposons d'un ensemble de m classes, $\{\mathbf{Y}_j\}_{1 \leq j \leq m}$ et d'une série de candidats, notés $\{\mathbf{X}_i\}_{1 \leq i \leq n}$. A chacun de ces candidats est associée une mesure de position \mathcal{X}_i , et le point \mathcal{Z}_i , initialement inconnu, qui correspond effectivement au point \mathbf{X}_i à l'instant $t+1$. \mathcal{X}_i et \mathcal{Z}_i sont deux variables aléatoires, la première à valeur dans R^3 , la seconde dans un ensemble fini de m valeurs, $\mathbf{Y}_1, \dots, \mathbf{Y}_m$.

Nous supposons pour l'instant les \mathcal{Z}_i indépendants, ce qui revient à dire que deux points \mathbf{X}_i et $\mathbf{X}_{i'}$ peuvent se voir associer au même point \mathbf{Y}_j . Cela laisse certes davantage de possibilités pour les affectations, mais il est bien plus aisé d'exprimer EM lorsque cette condition est vérifiée. Nous verrons un peu plus loin un moyen d'imposer aux affectations d'être bi-univoques.

La probabilité *a priori* d'une des classes \mathbf{Y}_j sera notée dans la suite $\Pi_j = \mathcal{P}(\mathcal{Z}_i = \mathbf{Y}_j)$. La normalisation impose $\sum_{j=1}^m \Pi_j = 1$. Comme les classes jouent toutes des rôles équivalents¹,

¹Ce qui n'est pas toujours vrai dans le domaine des statistiques, et la méthode EM permet également d'estimer ces probabilités *a priori* Π_j .

nous ferons l'hypothèse supplémentaire que tous les Π_j sont égaux, soit :

$$\forall i, \text{ on a } \Pi_j = \frac{1}{m}. \quad (7.10)$$

Nous avons également besoin de définir la distribution de probabilité de la variable aléatoire \mathcal{X}_i lorsque l'on connaît la classe à laquelle appartient le point \mathbf{X}_i . Si \mathcal{T} est la transformation qui amène le nuage des points $\{\mathbf{X}_i\}_{1 \leq i \leq n}$ sur celui des points $\{\mathbf{Y}_j\}_{1 \leq j \leq m}$, et que la classe du point \mathbf{X}_i est celle du point \mathbf{Y}_j , alors le point \mathbf{X}_i devrait se trouver proche de la position qu'occupait le point \mathbf{Y}_j juste avant le déplacement, soit $\mathcal{T}^{-1}\mathbf{Y}_j$. Nous choisirons une distribution gaussienne \mathcal{N} , centrée en $\mathcal{T}^{-1}\mathbf{Y}_j$ et de variance σ^2 :

$$f_i(\mathbf{X}_i | \mathcal{Z}_i = \mathbf{Y}_j) = \mathcal{N}(\mathbf{X}_i; \mathcal{T}^{-1}(\mathbf{Y}_j), \sigma^2). \quad (7.11)$$

Cette loi de densité exprime la probabilité que la variable aléatoire \mathcal{X}_i prenne la valeur \mathbf{X}_i lorsque le point est effectivement associé au point \mathbf{Y}_j , c'est-à-dire lorsque $\mathcal{Z}_i = \mathbf{Y}_j$.

La formule de Bayes nous permet d'exprimer, à partir des expressions 7.10 et 7.11, la distribution de probabilité pour la variable aléatoire \mathcal{X}_i :

$$f(\mathbf{X}_i) = \sum_{j=1}^m \Pi_j f_i(\mathbf{X}_i | \mathcal{Z}_i = \mathbf{Y}_j) = \sum_{j=1}^m \frac{1}{m} \mathcal{N}(\mathbf{X}_i; \mathcal{T}^{-1}(\mathbf{Y}_j), \sigma^2). \quad (7.12)$$

Utilisation de l'algorithme EM

Par simplicité, dans la suite, Ψ représentera l'ensemble des paramètres inconnus, à savoir la transformation $\mathcal{T}_{t,t+1}$ et la variance σ . Nous cherchons les paramètres Ψ qui vont maximiser la vraisemblance $\mathcal{P}(\mathbf{X} | \Psi)$ où \mathbf{X} désigne l'ensemble des positions mesurées des points $\{\mathcal{X}_i\}_{1 \leq i \leq n}$, soit $\{\mathbf{X}_i\}_{1 \leq i \leq n}$.

Les \mathcal{Z}_i étant inconnus, nous utiliserons donc l'algorithme EM qui nous fournira, en plus d'estimations pour Ψ , les valeurs pour ces \mathcal{Z}_i . Nous utiliserons par ailleurs la notation suivante :

$$\alpha_{i,j} = \mathcal{P}(\mathcal{Z}_i = \mathbf{Y}_j | \mathbf{X}, \Psi). \quad (7.13)$$

Le coefficient $\alpha_{i,j}$ exprime donc la probabilité que la classe \mathcal{Z}_i du point \mathbf{X}_i soit celle du point \mathbf{Y}_j . Le coefficient $\alpha_{i,j}$ exprime donc bien la notion que nous avons évoqué précédemment.

Partant d'une première estimation Ψ^0 , l'algorithme EM procède itérativement, l'itération q consistant à déterminer les paramètres Ψ qui maximiseront le terme :

$$Q(\Psi | \Psi^q) = E[\log(\mathcal{P}(\mathbf{X}, \mathcal{Z} | \Psi)) | \mathbf{X}, \Psi^q] \quad (7.14)$$

où Ψ^q est l'estimation des paramètres Ψ disponible à l'itération q . La moyenne est calculée pour toute les valeurs de \mathcal{Z} , prenant ainsi en compte toutes les valeurs possibles des \mathcal{Z}_i . Cette maximisation va être répétée jusqu'à convergence.

Nous allons développer le terme $Q(\Psi|\Psi^q)$ pour les distributions que l'on a précédemment établies. D'abord, nous avons :

$$\mathcal{P}(\mathbf{X}, \mathcal{Z}|\Psi) = \prod_{i=1}^n f_i(\mathbf{X}_i|\mathcal{Z}_i = \mathbf{Y}_j, \mathcal{T}, \sigma^2)\mathcal{P}(\mathcal{Z}_i)$$

soit

$$\mathcal{P}(\mathbf{X}, \mathcal{Z}|\Psi) = \prod_{i=1}^n \prod_{j=1}^m [f_i(\mathbf{X}_i|\mathcal{Z}_i = \mathbf{Y}_j, \mathcal{T}, \sigma^2)\Pi_j]^{\delta_{\{\mathcal{Z}_i = \mathbf{Y}_j\}}}$$

où δ est la notation usuelle de Dirac, c'est-à-dire que $\delta_{\{\mathcal{Z}_i = \mathbf{Y}_j\}}$ vaut 1 si la classe \mathcal{Z}_i du point \mathbf{X}_i est bien celle du point \mathbf{Y}_j , et 0 dans le cas contraire.

Le passage de l'expression précédente au logarithme conduit à :

$$\log(\mathcal{P}(\mathbf{X}, \mathcal{Z}|\Psi)) = \sum_{i=1}^n \sum_{j=1}^m (\log f_i(\mathbf{X}_i|\mathcal{Z}_i = \mathbf{Y}_j, \mathcal{T}, \sigma^2) + \log \Pi_j) \delta_{\{\mathcal{Z}_i = \mathbf{Y}_j\}}. \quad (7.15)$$

En réinjectant l'expression 7.15 dans l'équation 7.14, on obtient l'expression de Q :

$$Q(\Psi|\Psi^q) = \sum_{i=1}^n \sum_{j=1}^m (\log f_i(\mathbf{X}_i|\mathcal{Z}_i = \mathbf{Y}_j, \mathcal{T}, \sigma^2) + \log \Pi_j) E[\delta_{\{\mathcal{Z}_i = \mathbf{Y}_j\}}|\mathbf{X}, \Psi^q]$$

soit encore

$$Q(\Psi|\Psi^q) = \sum_{i=1}^n \sum_{j=1}^m (\log f_i(\mathbf{X}_i|\mathcal{Z}_i = \mathbf{Y}_j, \mathcal{T}, \sigma^2) + \log \Pi_j) \mathcal{P}(\mathcal{Z}_i = \mathbf{Y}_j|\mathbf{X}, \Psi^q). \quad (7.16)$$

Les probabilités $\alpha_{i,j}^q = \mathcal{P}(\mathcal{Z}_i = \mathbf{Y}_j|\mathbf{X}, \Psi^q)$ apparaissant dans la formule précédente peuvent être explicités grâce à la formule de Bayes :

$$\alpha_{i,j}(y, \Psi^q) = \frac{f_i(\mathbf{X}_i|\mathbf{Y}_j, \mathcal{T}_q, \sigma_q^2)\Pi_j}{\sum_{k=1}^m f_i(\mathbf{X}_i|\mathbf{Y}_k, \mathcal{T}_q, \sigma_q^2)\Pi_k} \quad (7.17)$$

où, selon l'expression de la loi normale,

$$f_i(\mathbf{X}_i|\mathbf{Y}_j, \mathcal{T}_q, \sigma_q^2) = \frac{1}{(2\pi\sigma_q^2)^{3/2}} e^{-\frac{\|\mathbf{X}_i - \mathcal{T}_q^{-1}\mathbf{Y}_j\|^2}{2\sigma_q^2}}. \quad (7.18)$$

Nous obtenons ainsi l'expression des probabilités $\alpha_{i,j}^q$:

$$\alpha_{i,j}^q = \frac{e^{-\frac{\|\mathbf{X}_i - \mathcal{T}_q^{-1}\mathbf{Y}_j\|^2}{2\sigma_q^2}}}{\sum_{k=1}^m e^{-\frac{\|\mathbf{X}_i - \mathcal{T}_q^{-1}\mathbf{Y}_k\|^2}{2\sigma_q^2}}}. \quad (7.19)$$

On peut contrôler aisément que ces différentes probabilités sont bien normées, c'est-à-dire vérifient $\sum_j \alpha_{i,j} = 1$.

Estimation des paramètres \mathcal{T} et σ

Pour évaluer les paramètres \mathcal{T} et σ , l'algorithme EM propose de procéder en alternance à une évaluation, selon la formule précédente, des $\alpha_{i,j}$ (étape "E"), et à une maximisation de Q pour les paramètres \mathcal{T} et σ (étape "M"). Pour ce faire, on va simplement déterminer les valeurs pour lesquelles la dérivée de Q par rapport à ces paramètres va s'annuler. Nous avons, d'après 7.16 et 7.18 :

$$Q(\mathcal{T}, \sigma^2 | \Psi^q) = \sum_{i=1}^n \sum_{j=1}^m \alpha_{i,j}^q \left(-\frac{3}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \|\mathbf{X}_i - \mathcal{T}^{-1} \mathbf{Y}_j\|^2 - \log(m) \right). \quad (7.20)$$

Par exemple, pour σ^2 , cela donne :

$$Q(\mathcal{T}, \sigma^2 | \Psi^q) = cste - \frac{3}{2} \log(\sigma^2) \sum_{i=1}^n \sum_{j=1}^m \alpha_{i,j}^q - \frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^q \|\mathbf{X}_i - \mathcal{T}^{-1} \mathbf{Y}_j\|^2)$$

$$Q(\mathcal{T}, \sigma^2 | \Psi^q) = cste - \frac{3n}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^q \|\mathbf{X}_i - \mathcal{T}^{-1} \mathbf{Y}_j\|^2)$$

$$\frac{\partial Q(\mathcal{T}, \sigma^2 | \Psi^q)}{\partial \sigma^2} = -\frac{3n}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^n \sum_{j=1}^m (\alpha_{i,j}^q \|\mathbf{X}_i - \mathcal{T}^{-1} \mathbf{Y}_j\|^2)$$

La condition $\frac{\partial Q(\mathcal{T}, \sigma^2 | \Psi^q)}{\partial \sigma^2} = 0$ nous permet donc d'obtenir une nouvelle estimation pour σ^2 pour l'étape $q+1$, plus précisément :

$$\sigma_{q+1}^2 = \frac{\sum_{i=1}^n \sum_{j=1}^m \alpha_{i,j}^q \|\mathbf{X}_i - \mathcal{T}^{-1} \mathbf{Y}_j\|^2}{3n} \quad (7.21)$$

On peut procéder de même pour déterminer une nouvelle estimation de \mathcal{T} , \mathcal{T}_{q+1} . Si l'on y regarde de près, l'expression 7.20 de Q est similaire à l'expression 7.3 de E_p présentée tantôt, à quelques détails près (un terme en $\log(m)$ et un facteur multiplicatif). Maximiser Q pour \mathcal{T} revient donc à minimiser l'expression E_p pour ce même paramètre, ce que nous avons déjà fait.

EM et entropie

Notons que cette démarche peut être interprétée dans un cadre entropique. Elle est équivalente, en effet, à la minimisation alternée de la quantité

$$D(\Psi, \{\alpha_{i,j}\}) = \sum_{(i,j)} \alpha_{i,j} \log(\Pi_j f_i(\mathbf{X}_i | \mathbf{Y}_j, \mathcal{T}, \sigma^2)) - \sum_{(i,j)} \alpha_{i,j} \log(\alpha_{i,j}) \quad (7.22)$$

soit à une constante près, à la minimisation alternée de

$$\sum_{(i,j)} \alpha_{i,j} \|\mathbf{X}_i - \mathcal{T}^{-1} \mathbf{Y}_j\|^2 - \sigma^2 \sum_{(i,j)} \alpha_{i,j} \log(\alpha_{i,j}) \quad (7.23)$$

Dans cette expression, on voit apparaître la quantité $\sum_{(i,j)} \alpha_{i,j} \log(\alpha_{i,j})$ qui correspond à l'entropie associée aux coefficients $\alpha_{i,j}$ ². Celle-ci permet de mesurer l'ordre dans un ensemble de valeurs. Si tous les $\alpha_{i,j}$ ont la même valeur, l'ensemble est peu ordonné, dans la mesure où l'on ne voit pas apparaître d'appariement bien défini entre les différents points. Ce désordre se traduit par une grande entropie. À l'inverse, lorsqu'une assignation émerge clairement, lorsque les $\alpha_{i,j}$ sont tous proches des valeurs 0 ou 1, l'entropie sera faible.

Initialement, le terme σ est grand. La minimisation de l'expression 7.23 tendra donc à maximiser l'entropie des coefficients. Cela peut s'expliquer par notre souhait de ne pas favoriser, au départ, des mises en correspondance une à une, comme c'est le cas avec l'algorithme ICP. Au fur et à mesure de la convergence, toutefois, le paramètre σ décroît, la composante exprimant l'entropie voit son importance diminuer, et nous revenons vers une mise en correspondance unitaire.

La méthode des multiplicateurs de Lagrange, via le paramètre σ^2 , permet de retrouver l'expression des différents coefficients. On obtient ainsi à nouveau l'expression :

$$\alpha_{i,j} = \frac{e^{-\|\mathcal{T}^{-1}\mathbf{Y}_j - \mathbf{X}_i\|^2/\sigma^2}}{\sum_k e^{-\|\mathcal{T}^{-1}\mathbf{Y}_k - \mathbf{X}_i\|^2/\sigma^2}}. \quad (7.24)$$

7.3.5 Utilisation de EM dans le cadre du suivi de mouvements

EM est, de façon générale, plus robuste que des méthodes comme ICP, et a donc connu un net succès dans le domaine de la mise en correspondance de nuages de points. Plusieurs travaux de Rangarajan *et al* ont montré que cette approche pouvait avec succès être utilisée afin d'améliorer les performances de l'algorithme ICP, en particulier en termes de robustesse et de stabilité [RCB97, CR03]. On parle généralement de la méthode EM-ICP.

S. Granger a par exemple utilisé cette démarche pour permettre le recalage d'une surface (représentée en fait par une grande quantité de points \mathbf{X}_i) sur un petit nombre de points de mesure \mathbf{Y}_i [Gra03]. Quelques difficultés ont conduit d'autres auteurs à introduire diverses variantes, que nous allons brièvement examiner. Elles ont cependant toutes en commun une expression pour les coefficients $\alpha_{i,j}$ inspirée du résultat précédent, de la forme :

$$\alpha_{i,j} = \frac{1}{z_{i,j}} e^{-\|\mathcal{T}^{-1}\mathbf{Y}_j - \mathbf{X}_i\|^2/\delta}. \quad (7.25)$$

Les hypothèses que nous avons faites précédemment (indépendance des différentes classes, chaque point \mathbf{X}_i correspond à l'un des points \mathbf{Y}_j , etc.) ne sont en effet pas toujours vérifiées. Le terme de normalisation $z_{i,j}$ a donc pris différentes formes dans la littérature, en fonction des besoins et des hypothèses que l'on pouvait faire. Nous en verrons deux exemples un peu plus loin. Mais avant cela, penchons-nous un moment sur le paramètre σ .

A noter que, dans le cas d'une transformation \mathcal{T} rigide, la distance entre les points \mathbf{X}_i et $\mathcal{T}^{-1}\mathbf{Y}_j$ est la même que celle entre $\mathcal{T}\mathbf{X}_i$ et \mathbf{Y}_j . On peut donc réécrire l'équation suivante sous

²Pour une description plus complète de l'expression de l'entropie, on pourra se reporter à des ouvrages spécialisés tels que [DGLR89].

une forme plus couramment rencontrée :

$$\alpha_{i,j} = \frac{1}{z_{i,j}} e^{-\|T\mathbf{X}_i - \mathbf{Y}_j\|^2 / \delta}. \quad (7.26)$$

Interprétation du paramètre σ

Nous pouvons essayer d'interpréter le sens que prend le paramètre σ dans le cadre de notre étude. Comparons deux coefficients $\alpha_{i,j}$ et $\alpha_{i,j'}$, qui correspondent à des mises en correspondance d'un point \mathbf{X}_i avec deux points \mathbf{Y}_j et $\mathbf{Y}_{j'}$.

$$\frac{\alpha_{i,j}}{\alpha_{i,j'}} = \frac{e^{-\|T\mathbf{X}_i - \mathbf{Y}_j\|^2 / \sigma^2}}{e^{-\|T\mathbf{X}_i - \mathbf{Y}_{j'}\|^2 / \sigma^2}} \quad (7.27)$$

Soit encore

$$\frac{\alpha_{i,j}}{\alpha_{i,j'}} = e^{\frac{\|T\mathbf{X}_i - \mathbf{Y}_{j'}\|^2 - \|T\mathbf{X}_i - \mathbf{Y}_j\|^2}{\sigma^2}} \quad (7.28)$$

On constate que le coefficient le plus grand correspond au point le plus proche de $T\mathbf{X}_i$ entre \mathbf{Y}_j et $\mathbf{Y}_{j'}$, et que le rapport est d'autant plus grand que la différence entre les distances est grande devant σ . Si d_i est la distance entre $T\mathbf{X}_i$ et le point le plus proche du nuage à l'instant $t + dt$, les points \mathbf{Y}_j qui interviendront effectivement sont ceux dont la distance à $T\mathbf{X}_i$ n'excède pas d_i de quelques σ . Plus simplement, σ représente ainsi la zone de recherche dans laquelle on envisagera les mises en correspondance. Il est donc logique de s'attendre à voir le coefficient σ décroître progressivement.

Plutôt que de laisser à l'algorithme EM le soin d'adapter la valeur du paramètre σ^2 , Rangarajan propose de s'inspirer des méthodes dites de *recuit simulé*, et d'imposer à σ^2 de suivre une loi décroissante [RCB97]. L'une des raisons pour imposer une décroissance de ce paramètre plutôt que de laisser l'algorithme EM s'occuper de son ajustement est citée par S. Granger dans [Gra03], et reportée par d'autres auteurs : la décroissance automatique du paramètre σ est souvent trop rapide, conduisant à des minima locaux qui ne correspondent pas à la solution recherchée. Nous avons nous-mêmes pu observer un tel phénomène, ainsi que le montreront les tests que nous mènerons à la fin de ce chapitre³. Il peut également arriver que le paramètre σ diverge un temps, avec des conséquences néfastes sur la solution obtenue.

On peut comprendre les raisons d'une telle décroissance. Initialement, on cherche en effet à dégager une tendance générale dans le placement de l'objet, puis progressivement, on souhaite que les mises en correspondance s'établissent. Diverses expériences ont montré la pertinence de cette suggestion, qui donne à la fois robustesse et précision à la méthode. La valeur initiale pour σ devra correspondre à la zone dans laquelle on recherchera les correspondants à l'origine, et donc à une estimation du déplacement possible de l'objet entre les deux images. Il est raisonnable de ne pas faire décroître σ en deçà du bruit de reconstruction sur les points. Dans la suite, nous noterons σ_p ce paramètre σ , afin d'éviter toute confusion avec d'autres grandeurs qui seront introduites ultérieurement.

³La stricte application de la méthode EM sera dénoté "EM avec σ automatique.

Autre expression pour $\alpha_{i,j}$: Soft Assign

Comme nous l'avons déjà dit, le problème tel que nous l'avons présenté jusqu'ici n'est pas symétrique. Par exemple, dans la variante suggérée à la fin du paragraphe 7.3.5, nous assurions à travers la condition $\sum_j \alpha_{i,j} = 1$, qu'à un point \mathbf{X}_i correspondait à un unique point \mathbf{Y}_j , mais l'inverse n'était pas imposé, et un même point \mathbf{Y}_j pouvait être associé à plusieurs \mathbf{X}_i .

A. Rangarajan juge cette dissymétrie malvenue. Pour remédier à ce problème, il faut de plus imposer que $\sum_i \alpha_{i,j} = 1$. L'expression statistique de ce problème est alors bien plus délicate. En effet, les classifications des différents points considérés ne sont plus indépendants, puisque l'on doit retrouver exactement un et un seul point \mathbf{Y}_j dans chacune des classes représentées par les \mathbf{X}_i .

Il n'existe alors plus de façon simple d'exprimer les coefficients $\alpha_{i,j}$. On supposera qu'ils sont toujours de la forme $\frac{1}{z_{i,j}} e^{-\|\mathcal{T}\mathbf{X}_i - \mathbf{Y}_j\|^2 / \sigma_p^2}$. Le calcul des $z_{i,j}$ n'est pas trivial. La méthode des multiplicateurs de Lagrange nous permet cependant de les mettre sous la forme d'un produit de termes : $z_{i,j} = \mu_i \nu_j$.

Dans l'algorithme Soft-Assign présenté dans [RCB97], Rangarajan propose d'évaluer ces μ_i et ν_j de façon itérative, en initialisant la matrice $\alpha_{i,j}$ avec les termes $e^{-\|\mathcal{T}\mathbf{X}_i - \mathbf{Y}_j\|^2 / \sigma_p^2}$, et en normalisant alternativement les lignes et les colonnes de la matrice, jusqu'à parvenir à une convergence, assurée par le théorème de Sinkhorn.

L'un des inconvénients majeurs de cette méthode est qu'elle est coûteuse en terme de calculs, puisque de nombreuses normalisations sont à effectuer sur la matrice. Elle est donc à réserver aux cas où une mise en correspondance bi-univoque est absolument nécessaire, par exemple dans la mise en correspondance de graphes, où l'on est assuré qu'aucune occlusion ou erreur de mesure ne viendra poser de problème au niveau de la mise en correspondance des points.

Principe de l'élimination des points aberrants

Encore une fois, il peut apparaître dans l'un des deux nuages à mettre en correspondance des points aberrants ou sans correspondants. Ces points doivent impérativement pouvoir être éliminés, en l'absence de quoi ils viendraient perturber le suivi. L'une des méthodes proposées par Rangarajan consiste à ajouter à la matrice des appariements $\alpha_{i,j}$ une ligne et une colonne supplémentaire.

Les coefficients supplémentaires permettront d'avoir des points \mathbf{X}_i ou \mathbf{Y}_j sans correspondants. En effet, il sera alors possible que, pour un i donné, tous les $\alpha_{i,j}$ correspondant effectivement à un point \mathbf{Y}_j soient nuls, le terme supplémentaire $\alpha_{i,k}$ valant quant à lui 1 pour conserver la condition de normalisation. C'est également le cas pour un point \mathbf{Y}_j .

La figure 7.1 montre un exemple de matrice d'appariement $\alpha_{i,j}$, dans laquelle toutes les mises en correspondance ont été résolues. Bien que la plupart des points aient été appariés, on note la présence de points \mathbf{X}_i ou \mathbf{Y}_j dépourvus de correspondants. Les ligne et colonne supplémentaires permettent d'avoir quand même des lignes et colonnes normalisées (à l'ex-

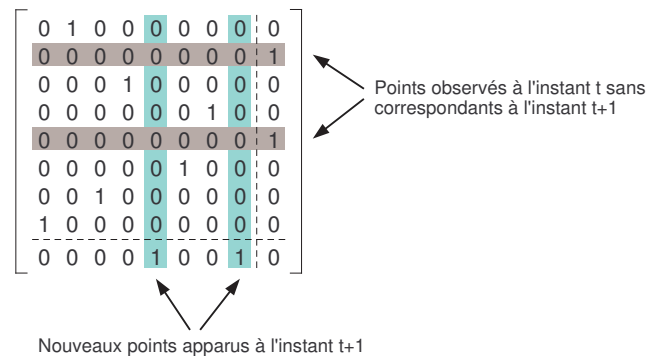


FIG. 7.1 – La matrice d’appariement $\alpha_{i,j}$ pourvue d’une colonne et d’une ligne supplémentaire permet de traiter le cas des points privés de correspondants.

ception des lignes et colonnes supplémentaires, puisque l’on peut avoir plusieurs points \mathbf{X}_i ou \mathbf{Y}_j sans correspondants).

L’essentiel de la méthode présentée auparavant peut être appliqué sans grand changement, et permettra ainsi de traiter le cas où des points disparaissent ou apparaissent entre les deux instants. Cependant, la méthode de normalisation proposée dans la section précédente ne peut plus être appliquée sans précaution. La difficulté vient d’abord du fait que les ligne et colonne nouvellement ajoutées ne doivent pas faire l’objet d’une normalisation, car plusieurs points peuvent être privés de correspondants. Il est donc naturel de trouver plusieurs coefficients à 1 dans les ligne et colonne supplémentaires.

Limitations et défauts de l’approche

Si l’on applique la méthode de normalisation itérative aux seules lignes et colonnes restantes, on s’aperçoit que la solution obtenue dépend grandement des valeurs initiales que l’on aura placé dans les ligne et colonnes supplémentaires. Si on les choisit trop grandes, on aboutit à la solution triviale où tous les points sont privés de correspondants (seules les dernières ligne et colonne contiennent des valeurs non nulles). Trop petites, l’élimination des points sans correspondants sera impossible. Le choix de la valeur initiale dans les ligne et colonne ajoutées se révèle donc délicat, et déterminant pour la bonne convergence de l’algorithme.

Nous avons essayé de comprendre dans quelles conditions on pouvait obtenir une convergence correcte de cet algorithme. En fait, d’un point de vue pratique, il faut que l’algorithme dispose d’une distance “seuil” d_0 à partir de laquelle deux points ne seront pas appariés. Les valeurs que l’on trouvera initialement dans les ligne et colonne supplémentaires vont, indirectement, définir ce seuil. C’est la raison pour laquelle des valeurs trop petites ou trop grandes conduisent respectivement à une détection d’aucun point aberrant, et à un rejet de tous les points observés.

De par la présence de cette normalisation itérative de la matrice, il est cependant assez délicat de savoir exactement ce qui passe, et la détermination des valeurs supplémentaires peut être problématique. Nous avons donc essayé de reprendre cette idée, mais en faisant

apparaître explicitement cette distance d_0 dans les expressions. C'est l'objet de la section suivante.

7.3.6 Solution proposée pour un suivi robuste avec EM

Expression retenue pour les coefficients $\alpha_{i,j}$

Dans le cas qui nous intéresse, nous ne considérons pas l'association bi-univoque comme rigoureusement indispensable, ce qui simplifie grandement les choses. C'est en effet cette condition, qui impose que les lignes et les colonnes de la matrices soient toutes normalisées, qui nous avait imposé l'utilisation de cette normalisation itérative. Dans notre cas, nous pourrions avoir une forme explicite pour les coefficients $\alpha_{i,j}$. Pour chaque point \mathbf{X}_i , on cherchera à savoir s'il dispose d'un correspondant parmi les points \mathbf{Y}_j ou non. Peu nous importe que les points \mathbf{Y}_j aient, quant à eux, un ou plusieurs correspondants, voire n'en aient pas. On ne cherchera donc pas à imposer de normalisation sur les colonnes $\sum_j \alpha_{i,j} = 1$.

On ne gardera donc de la méthode ci-dessus que la colonne supplémentaire afin d'assurer que les lignes, elles, soient bien normalisées. Le sens donné aux valeurs à placer dans cette colonne est alors un peu plus facile à exprimer. Pour permettre le décrochage des points \mathbf{X}_i sans correspondants, nous allons introduire un "point virtuel" en permanence à une distance d_0 de chacun des points \mathbf{X}_i . Ce sont les $\alpha_{i,k}$ correspondant à ce point virtuel qui apparaîtront dans la colonne supplémentaire.

Après normalisation sur les lignes, les différents coefficients $\alpha_{i,j}$ vaudront alors

$$\alpha_{i,j} = \frac{e^{-\frac{\|\mathcal{T}\mathbf{X}_i - \mathbf{Y}_j\|^2}{\sigma_p^2}}}{\sum_{k=1}^{N_{t+1}} e^{-\frac{\|\mathcal{T}\mathbf{X}_i - \mathbf{Y}_k\|^2}{\sigma_p^2}} + e^{-\frac{d_0^2}{\sigma_p^2}}}. \quad (7.29)$$

On peut ainsi vérifier que le comportement, illustré sur la figure 7.2 est bien celui recherché. Initialement, lorsque σ_p est grand devant d_0 (à gauche dans la figure), tous les correspondants possibles sont envisagés. Puis, σ_p va décroître et l'on verra émerger deux situations possibles. La première, en haut, dans laquelle on trouve effectivement un (ou plusieurs) point(s) \mathbf{Y}_j dans un voisinage d_0 autour de la position prédite de \mathbf{X}_i . Dans ce cas de figure, le terme qui deviendra prépondérant au dénominateur dans l'expression 7.29 sera l'un des termes de la somme. L'un des $\alpha_{i,j}$ tendra donc vers 1, tandis que les autres tendront vers 0.

A l'inverse, s'il n'est pas possible de trouver un point dans un voisinage d_0 entourant la position prédite, le terme prépondérant au dénominateur de 7.29 sera le terme $e^{-\frac{d_0^2}{\sigma_p^2}}$. Ce qui aura pour conséquence de faire tendre tous les coefficients $\alpha_{i,j}$ vers 0. Le point en question ne s'étant trouvé aucun correspondant, il n'interviendra plus dans l'expression du critère E_p . Cela permet donc effectivement d'éliminer les points qui disparaîtront entre les instants t et $t + dt$.

Pour l'utilisation que l'on souhaite en faire (suivi d'objets rigides, et ultérieurement d'ob-

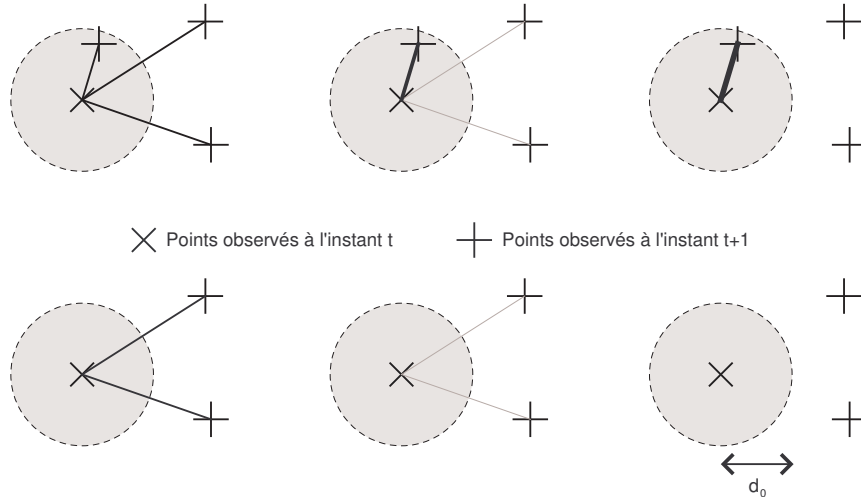


FIG. 7.2 – Évolution des probabilités d'appariement $\alpha_{i,j}$ pour un point avec correspondant (en haut) et un point sans correspondant (en bas) tandis que σ_p décroît.

jets déformables), il est intéressant de noter que les résultats obtenus par cette version dissymétrique et l'algorithme *Soft-Assign* proposé par Rangarajan *et al* sont largement similaires, ainsi que l'on peut en juger sur la figure 7.13 en fin de chapitre. Cependant, il n'est pas utile ici de normaliser de façon itérative une matrice, ce qui nous permet de gagner plus d'un ordre de grandeur en terme de temps de calcul. C'est donc à la présente expression que nous référerons dorénavant lorsque nous parlerons de EM-ICP.

Interprétation statistique

Cette expression s'est révélée satisfaisante, et on verra à la fin de ce chapitre qu'elle donne des résultats très similaires à ceux de *Soft-Assign*, du moins dans le cadre dans lequel nous l'avons appliquée⁴, sans qu'il soit besoin de recourir à une minimisation itérative d'une matrice, particulièrement coûteuse en terme de temps de calcul.

Nous allons à présent essayer de justifier d'un point de vue théorique l'expression que nous avons introduite. Pour ce faire, reprenons le cadre statistique défini dans le paragraphe 7.3.4. L'élimination de points aberrants peut y être vue comme l'adjonction d'une classe supplémentaire (notée \mathbf{Y}_{m+1}) qui viendra s'ajouter aux classes $\{\mathbf{Y}_1, \dots, \mathbf{Y}_m\}$. Cette classe jouera le rôle d'une classe "poubelle", dans laquelle devront se retrouver les points \mathbf{X}_i qui ne trouvent pas de correspondant parmi les $\{\mathbf{Y}_j\}_{1 \leq j \leq m}$.

La difficulté réside dans le fait qu'il nous faut déterminer la loi de probabilité $f_i(\mathbf{X}_i | \mathcal{Z}_j =$

⁴Il est évident que certaines applications, de mise en correspondance de graphes par exemple, ont un besoin plus grand du caractère bi-univoque des mises en correspondances, auquel cas ces résultats pourraient différer.

\mathbf{Y}_{m+1}) de la variable aléatoire \mathcal{X}_i , lorsque l'on sait que le point considéré est à ranger dans cette classe supplémentaire, ainsi que la probabilité à priori Π_{m+1} . La loi de probabilité ne doit pas faire intervenir la transformation $\mathcal{T}_{t,t+1}$ et devrait à priori être uniforme. La probabilité *a priori* Π_{m+1} de cette même classe est, elle aussi, plus difficile à déterminer.

Pour la loi de probabilité $f_i(\mathbf{X}_i | \mathcal{Z}_i = \mathbf{Y}_{m+1})$, nous allons choisir une simple loi uniforme. Afin d'éviter de travailler avec une loi dite "impropre" (c'est à dire dont l'intégrale sur l'ensemble de l'espace ne vaut pas 1), elle sera en fait uniforme (égale à $1/V$) dans un volume V , et nulle en dehors de ce volume.

Reste à établir les probabilités *a priori* Π_j . Supposons que l'on cherche les mises en correspondance dans des sphères de rayon r centrée sur les points $\mathcal{T}^{-1}\mathbf{Y}_j$. Pour un point pris au hasard dans le volume V , la probabilité de tomber dans une sphère donnée vaut $(4\pi r^3)/(3V)$ (nous supposons les probabilités indépendantes, ce qui correspond à négliger la possibilité que certaines de ces sphères se touchent). Celle de ne tomber dans aucune des sphères (ce qui correspond donc à un rejet du point) vaut $1 - (4m\pi r^3)/(3V)$ (en supposant toujours que les sphères sont disjointes).

Les coefficients $\alpha_{i,j}$ deviennent alors :

$$\alpha_{i,j}^q = \frac{e^{-\frac{\|\mathbf{x}_i - \mathcal{T}_q^{-1}\mathbf{Y}_j\|^2}{2\sigma_q^2}}}{\sum_{k=1}^m e^{-\frac{\|\mathcal{T}_q^{-1}\mathbf{Y}_k - \mathbf{x}_i\|^2}{2\sigma_q^2}} + \frac{(2\pi)^{3/2}\sigma^3 V}{4/3\pi r^3} \left(1 - \frac{4m\pi r^3}{3V}\right) \frac{1}{V}}. \quad (7.30)$$

Cette expression se simplifie, lorsque l'on choisit un volume V suffisamment grand pour que les sphères de rayon r soient de volume négligeable en comparaison, en :

$$\alpha_{i,j}^q = \frac{e^{-\frac{\|\mathbf{x}_i - \mathcal{T}_q^{-1}\mathbf{Y}_j\|^2}{2\sigma_q^2}}}{\sum_{k=1}^m e^{-\frac{\|\mathcal{T}_q^{-1}\mathbf{Y}_k - \mathbf{x}_i\|^2}{2\sigma_q^2}} + \frac{(2\pi)^{3/2}\sigma^3}{4/3\pi r^3}}. \quad (7.31)$$

Reste donc à choisir le rayon r des sphères dans lesquelles on cherchera des mises en correspondance. La solution la plus simple consiste à choisir $r = \sigma$, ce qui conduit à l'expression suivante pour les $\alpha_{i,j}^q$

$$\alpha_{i,j}^q = \frac{e^{-\frac{\|\mathbf{x}_i - \mathcal{T}_q^{-1}\mathbf{Y}_j\|^2}{2\sigma_q^2}}}{\sum_{k=1}^m e^{-\frac{\|\mathcal{T}_q^{-1}\mathbf{Y}_k - \mathbf{x}_i\|^2}{2\sigma_q^2}} + c} \quad \text{avec } c = 3\sqrt{\frac{\pi}{2}} \quad (7.32)$$

Notons que ces coefficients $\alpha_{i,j}$ sont au cœur de l'élimination des points aberrants. En effet, lorsque la transformation \mathcal{T}^{-1} n'amène aucun des points \mathbf{Y}_j dans un voisinage de rayon σ^q autour d'un point \mathbf{X}_i , tous les termes au dénominateur seront petits devant c , de sorte que tous les coefficients $\alpha_{i,j}^q$ correspondant au point \mathbf{Y}_j tendront vers 0.

On ne retrouve pas exactement la même expression que celle que nous avons suggéré plus haut, et qui fait intervenir une quantité d_0 . Dans la pratique, les deux formules se sont révélées

à peu près équivalentes, avec cependant une subtile différence : il est important, avec cette nouvelle formule, de ne pas faire décroître σ de façon trop importante, sous peine de voir tous les points \mathbf{X}_i associés à la classe “pouvelle”. La formule 7.29 permet éventuellement d’atteindre des valeurs plus petites σ , de façon à davantage forcer un “choix” si deux correspondants sont possibles, tout en ne rejetant pas un point à une distance de l’ordre de d_0 même si celle-ci est supérieure à σ .

Pour retrouver l’expression précédente, il faut donc imposer que le rayon r des sphères où s’effectuent les recherches ne doit pas descendre en-dessous d’une valeur d_0 , même si σ descend quant à lui plus bas. Considérons pour r l’expression :

$$r = \sigma e^{\frac{d_0^2}{6\sigma^2}}. \quad (7.33)$$

La courbe $r/d_0 = f(\sigma/d_0)$ est tracée sur la figure 7.3. On peut constater que le comportement est approximativement celui attendu. Certes, ce choix paraît un peu arbitraire, et n’est correct que pour des valeurs de σ supérieures à $d_0/3$ environ (la courbe remontant rapidement pour des valeurs plus faibles).

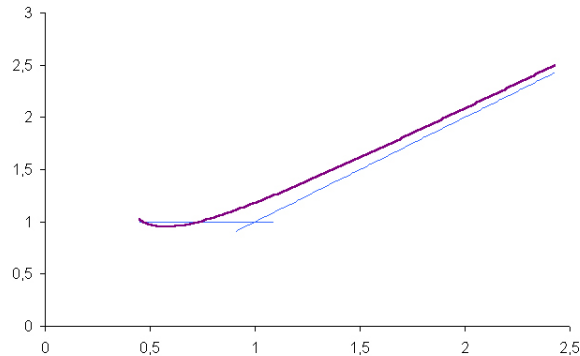


FIG. 7.3 – Tracé du rapport r/d_0 en fonction de la quantité σ/d_0 . La fonction proposée permet de retrouver grossièrement le comportement souhaité, représenté ici par les deux courbes rectilignes.

Avec cette expression pour r , on trouve, à un coefficient près⁵, pour les coefficients $\alpha_{i,j}$ l’expression suggérée précédemment :

$$\alpha_{i,j}(y, \Psi^q) = \frac{e^{-\frac{\|\mathbf{Y}_j - \mathcal{T}_{t,t+1}^q \mathbf{X}_i\|^2}{2\sigma_q^2}}}{\sum_{k=1}^m e^{-\frac{\|\mathbf{Y}_k - \mathcal{T}_{t,t+1}^q \mathbf{X}_i\|^2}{2\sigma_q^2}} + 3\sqrt{(\pi/2)} e^{-\frac{d_0^2}{2\sigma_q^2}}}. \quad (7.34)$$

⁵Coefficient que l’on pourrait éliminer en intégrant en fait une distribution de probabilité gaussienne plutôt que d’utiliser une simple sphère de rayon r .

7.3.7 Mise en œuvre pratique

Introduction

De façon générale, la démarche mise en place dans le cadre de l'algorithme ICP est semblable à celle des algorithmes dérivés de EM. Elle consiste en une minimisation alternée du critère précédemment défini :

$$E_p(\mathcal{T}_{t,t+1}, \alpha_{i,j}) = \sum_{(i,j) \in [1, N_t] \times [1, N_{t+1}]} \alpha_{i,j} \|\mathcal{T}_{t,t+1} \mathbf{X}_i - \mathbf{Y}_j\|^2. \quad (7.35)$$

Ainsi, comme dans le cadre de ICP, ou de l'algorithme EM, on alternera une étape d'évaluation des assignations $\alpha_{i,j}$ pour la valeur courante de la transformation $\mathcal{T}_{t,t+1}$, et une étape de minimisation du critère E_p pour ces mêmes valeurs $\alpha_{i,j}$. Dans la suite, nous allons décrire l'ensemble des étapes permettant le suivi d'un objet rigide, de l'initialisation à la convergence, en précisant quelques points importants de l'implémentation.

Filtrage des points

Parmi les points 3D reconstruits que l'on peut extraire d'une série d'images correspondant à un instant t , nombre d'entre eux ne correspondent pas au solide qui nous intéresse (points issus de mauvaises reconstructions, appartenant au fond ou à d'autres solides). Seuls les points à la surface de l'objet étudié nous intéressent pour évaluer le mouvement de celui-ci.

Dans notre cas, nous disposons de la position à l'instant t de la position de l'objet à suivre. Il est donc possible d'évaluer pour tout point \mathbf{X}_i la distance à sa surface pour ne garder que les points pour lesquels elle est, en valeur absolue, inférieure à un seuil ϵ . Comme illustré sur la figure 7.4, les points retenus sont donc ceux inclus dans un volume avoisinant la surface de l'objet. On supposera dans la suite que ces points sont bien liés au solide, et que leur mouvement dans l'espace entre les instants t et $t + 1$ est donc le même que celui de l'objet à suivre.

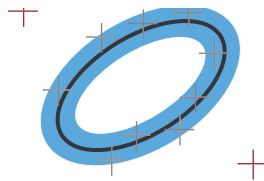


FIG. 7.4 – Les points dont la distance à la surface de l'objet (ici un ellipsoïde) n'excèdent pas ϵ , c'est à dire ceux apparaissant ici dans le domaine grisé, sont choisis pour estimer le mouvement du solide.

Si le modèle décrivant l'objet en reproduit très fidèlement la forme, ce seuil peut être choisi de façon à correspondre aux erreurs de reconstruction lors de la génération du nuage de points. Cette erreur est, dans notre cas, inférieure au millimètre pour des points d'intérêt correctement détectés et appariés. Si en revanche le modèle ne décrit qu'approximativement

la surface de l'objet étudié, le seuil retenu sera bien évidemment choisi sensiblement plus grand, afin de prendre en compte les erreurs de modélisation.

Estimation de la transformation $\mathcal{T}_{t,t+1}$

Supposons la matrice de correspondance $\alpha_{i,j}$ connue, et reprenons notre terme E_p apparaissant dans l'expression 7.5 :

$$E_p = \sum_{(i,j) \in [1, N_t] \times [1, N_{t+1}]} \alpha_{i,j} \|\mathcal{T}_{t,t+1} \mathbf{X}_i - \mathbf{Y}_j\|^2. \quad (7.36)$$

Lorsque l'on suppose les différents coefficients $\alpha_{i,j}$ connus, et que l'on s'intéresse à l'estimation de la transformation $\mathcal{T}_{t,t+1}$ qui minimisera E_p , l'expression précédente peut être notablement simplifiée, en postant

$$\lambda_i = \sum_{j=1}^{N_{t+1}} \alpha_{i,j} \quad (7.37)$$

$$\text{et } \mathbf{Z}_i = \frac{\sum_{j=1}^{N_{t+1}} \alpha_{i,j} \mathbf{Y}_j}{\sum_{j=1}^{N_{t+1}} \alpha_{i,j}} = \frac{1}{\lambda_i} \sum_{j=1}^{N_{t+1}} \alpha_{i,j} \mathbf{Y}_j. \quad (7.38)$$

Le critère se réécrit alors, en fonction de ces nouveaux termes, sous la forme

$$E_p = \sum_{i=1}^{N_t} \left(\lambda_i \|\mathcal{T}_{t,t+1} \mathbf{X}_i - \mathbf{Z}_i\|^2 + \langle \mathbf{X}_i \mathbf{Z}_i, \sum_{j=1}^{N_{t+1}} (\alpha_{i,j} \mathbf{Z}_i \mathbf{Y}_j) \rangle + cste \right), \quad (7.39)$$

le second terme se révélant en fait nul de par la définition de \mathbf{Z}_i . Puisque l'on cherche simplement à minimiser un critère, le troisième terme, constant, peut également être oublié, et l'on obtient alors l'expression définitive pour le critère E_p :

$$E_p = \sum_{i=1}^{N_t} \lambda_i \|\mathcal{T}_{t,t+1} \mathbf{X}_i - \mathbf{Z}_i\|^2. \quad (7.40)$$

Cette expression est plus simple que l'expression 7.5 apparaissant dans la section 7.3.2, dans la mesure où la somme a été restreinte aux seuls points \mathbf{X}_i . Du calcul des coefficients d'appariement $\alpha_{i,j}$, nous avons pu déduire les λ_i et les points \mathbf{Z}_i , avec lesquels nous pourrions évaluer directement et aisément la transformation $\mathcal{T}_{t,t+1}$ qui va minimiser le critère E_p .

Si l'on regarde cette expression d'un peu plus près, on voit naturellement ressortir le fait que les points \mathbf{X}_i non appariés vont disparaître progressivement de l'expression de E_p , puisque le coefficient λ_i qui leur est associé va tendre progressivement vers 0 lorsque σ_p tendra vers 0. Pour les points appariés, au contraire, λ_i va tendre vers 1. De la même façon, les différents \mathbf{Z}_i associés aux points effectivement appariés vont tendre vers le point \mathbf{Y}_j le plus proche de $\mathcal{T}_{t,t+1} \mathbf{X}_i$.

On retrouve donc, à la limite, le même comportement que l'algorithme ICP avec élimination des points aberrants. En revanche, lorsque σ_p sera plus élevé, les avantages de EM-ICP se feront sentir, avec un critère davantage souple et continu.

Mise en correspondance

Lorsque, au contraire, la transformation $\mathcal{T}_{t,t+1}$ est supposée connue, on peut en déduire les différents coefficients $\alpha_{i,j}$. Diverses expressions des coefficients $\alpha_{i,j}$ ont été examinées précédemment. Nous utiliserons ici la formule 7.29 suggérée au paragraphe 7.3.6 :

$$\alpha_{i,j} = \frac{e^{-\frac{\|\mathcal{T}\mathbf{x}_i - \mathbf{y}_j\|^2}{\sigma_p^2}}}{\sum_{k=1}^{N_{t+1}} e^{-\frac{\|\mathcal{T}\mathbf{x}_i - \mathbf{y}_k\|^2}{\sigma_p^2}} + e^{-\frac{d_0^2}{\sigma_p^2}}}. \quad (7.41)$$

Une minimisation alternée

Pour le reste, l'algorithme EM alterne deux étapes jusqu'à la convergence :

- une évaluation des coefficients $\alpha_{i,j}$, pour la transformation $\mathcal{T}_{t,t+1}$ courante, qui permettra de calculer les coefficients λ_i et les points \mathbf{Z}_i qui serviront dans la seconde étape,
- une estimation de la transformation $\mathcal{T}_{t,t+1}$ qui va minimiser, pour des λ_i et \mathbf{Z}_i fixés, le critère $E_p(\mathcal{T}_{t,t+1}) = \sum_{i=1}^{N_t} \lambda_i \|\mathcal{T}_{t,t+1}\mathbf{X}_i - \mathbf{Z}_i\|^2$.

Ces deux étapes seront répétées successivement jusqu'à parvenir à une convergence de l'algorithme. À chaque boucle, on fera décroître le paramètre σ_p apparaissant dans l'expression des $\alpha_{i,j}$, partant d'une valeur estimant le déplacement maximal des points entre les deux instants, jusqu'à atteindre quelque chose de l'ordre du bruit subsistant sur la reconstruction des points 3D. Une loi géométrique donne des résultats satisfaisants. L'ensemble des étapes à appliquer est résumé dans l'algorithme 4.

7.3.8 Avantages et inconvénients de la méthode

Cette méthode présente de nombreux avantages. Elle a le mérite d'être simple, et ne nécessite que l'extraction de nuages de points à chaque instant. Les algorithmes basés sur EM sont reconnus comme robustes, et si l'on peut profiter de la connaissance *a priori* d'une correspondance entre les points des différents nuages, la méthode peut elle-même corriger les mauvais appariements et rejeter les outliers. Enfin, pour une correspondance donnée, il existe des méthodes permettant d'évaluer efficacement la transformation correspondante, ce qui permet d'obtenir des performances aptes à un éventuel usage en temps réel.

Cependant, plutôt que d'évaluer directement la position de l'objet, cette méthode détermine le déplacement de ce dernier entre deux instants successifs, et ces déplacements sont intégrés pour permettre d'obtenir la position. Les erreurs commises sur l'estimation du déplacement risquent donc d'être accumulées, de sorte que la position estimée s'éloignera progressivement de la position réelle de l'objet.

Algorithme 4 Suivi robuste d'un objet rigide s'appuyant sur EM

définir σ_p^0 : ordre de grandeur du mouvement à suivre
 définir σ_p^∞ : bruit sur les points 3D

initialiser $\sigma_p \leftarrow \sigma_p^0$
 initialiser $\mathcal{R}_0 \leftarrow Id$ et $\mathbf{t}_0 \leftarrow \mathbf{0}$
 initialiser $k \leftarrow 0$

répéter

pour tout point \mathbf{X}_i :

déterminer le point prédit $\mathbf{X}'_i \leftarrow \mathcal{R}_k \mathbf{X}_i + \mathbf{t}_k$
 initialiser $\lambda_i \leftarrow 0$ et $\mathbf{Z}_i \leftarrow \mathbf{0}$

pour tout point \mathbf{Y}_j :

déterminer $d_{i,j}^2 \leftarrow \|\mathbf{X}'_i - \mathbf{Y}_j\|^2 / \sigma_p^2$

déterminer $\alpha'_{i,j} = e^{-d_{i,j}^2}$

$\lambda_i \leftarrow \lambda_i + \alpha'_{i,j}$

$\mathbf{Z}'_i \leftarrow \mathbf{Z}'_i + \alpha'_{i,j} \mathbf{Y}_j$

fin pour

définir $\mathbf{Z}_i \leftarrow \frac{1}{\lambda_i} \mathbf{Z}'_i$

fin pour

$k \leftarrow k + 1$

déterminer \mathcal{R}_k et \mathbf{t}_k minimisant $E_p(\mathcal{R}_k, \mathbf{t}_k) = \sum_i \|\mathcal{R}_k \mathbf{X}_i + \mathbf{t}_k - \mathbf{Z}_i\|^2$

si $\sigma_p > \sigma_p^\infty$ **alors**

faire décroître σ_p : $\sigma_p \leftarrow cste \times \sigma_p$

fin si

jusqu'à ce que la différence entre \mathcal{R}_{k-1} et \mathcal{R}_k et entre \mathbf{t}_{k-1} et \mathbf{t}_k tombe en deçà d'un seuil, ou bien que $k > k_{max}$

Certes, les erreurs successivement commises ne sont pas complètement indépendantes, et se compensent en partie. C'est en particulier le cas lorsque des points sont observés très longtemps dans la séquence d'image. Si occasionnellement la position reconstruite du point en question est légèrement décalée, l'erreur qui en découlera sera compensée à l'étape suivante par une erreur de signe contraire, lorsqu'il reprendra sa place correcte. En fait, si certains points sont visibles sur plusieurs dizaines d'images, on peut les considérer comme faisant partie du modèle décrivant l'objet suivi, et l'on revient à un suivi du premier ordre, directement en position.

Malheureusement, lors du suivi, il apparaît et disparaît des points en permanence, par exemple du fait des occlusions éventuelles. Les points extraits des images obtenues à l'instant t ne constituent plus un modèle satisfaisant de l'objet une centaine d'images plus tard. Les erreurs ne se compensent donc pas parfaitement, et la dérive est inévitable. En outre, les inévitables points aberrants, qu'ils soient issus d'objets du fond et apparaissant à proximité de l'objet à suivre, ou bien de mauvaises reconstructions 3D, et n'appartenant bien évidemment pas au modèle de l'objet, accentuent encore cet état de fait.

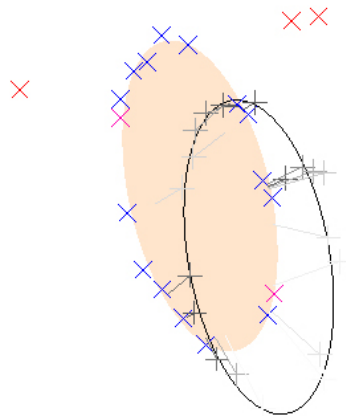


FIG. 7.5 – Notre plate-forme de test pour les différents algorithmes propose le suivi d'un ellipsoïde. Les signes + et l'ellipse noire indiquent la position initiale à l'instant t , les croix et la surface claire la position que l'on souhaite trouver à l'instant $t + 1$. On constate la présence de quelques points aberrants, la disparition et l'apparition de certains points entre les deux instants, et du bruit sur la mesure de la position des différents points.

Pour illustrer cette dérive, nous avons appliqué la méthode précédente au suivi d'un ellipsoïde d'une taille de l'ordre du centimètre, au moyen d'une vingtaine de points sis à la surface de l'ellipsoïde. Ces points sont perturbés par un bruit de mesure gaussien d'une amplitude de quelques dixièmes de millimètres. Les points considérés ont une espérance de vie d'une vingtaine d'images, et sont progressivement remplacés par de nouveaux points, autrement dit il disparaît à chaque image un point en moyenne, et il apparaît dans le même temps un nouveau point.

L'expérience étant simulée, la correspondance entre les différents points est parfaitement

connue, et nous nous en sommes servis plutôt que d'utiliser la détermination automatique des correspondances présentée plus haut. Les points sans correspondants (occlusions, nouveaux points) sont parfaitement éliminés, et nous n'avons pas introduit de points aberrants. Ceci afin de se placer dans le cas le plus favorable.



FIG. 7.6 – Dérive de l'erreur (en position et en angle) commise lors d'un suivi par points d'intérêt.

La figure 7.6 montre, en bleu, l'erreur commise sur l'estimation du centre de l'ellipsoïde. Bien que les conditions soient meilleures que celles que l'on rencontrera expérimentalement, on constate effectivement une dérive de l'estimation, due à l'accumulation des erreurs. En quelques secondes seulement (à raison d'une trentaine d'images par seconde), l'erreur atteint 3 millimètres, soit le tiers de la taille de l'ellipsoïde suivi. Bien avant cela, la plupart des points du nuage risquent d'être rejetés comme points aberrants parce que trop éloignés de la surface de l'ellipsoïde. Le suivi sera donc rapidement perdu.

Cette dérive trouve bien sa source dans le fait que les points disparaissent régulièrement. Dans le cas idéal que nous avons mis en place (correspondance parfaitement identifiée, rejet de la totalité des points aberrants), nous avons repris la même simulation avec des points qui restent visibles tout au long de la séquence, et sans apparition de nouveaux points. L'erreur commise, sur la figure 7.7, ne fait alors plus montre d'une dérive notable. Ceci peut s'expliquer par le fait que les points initialement identifiés décrivent un modèle de l'objet, et que l'on ajuste ce modèle aux différents nuages de points qui suivent, les erreurs commises se compensant au cours du temps.

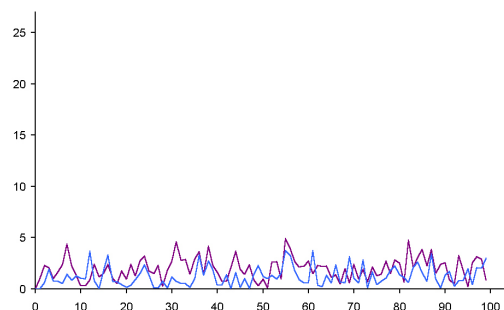


FIG. 7.7 – La dérive disparaît lorsqu'il n'y a aucune occlusion et que les mises en correspondances sont parfaites.

On pourrait penser qu'établir un modèle de l'objet constitué d'une sorte de carte de points d'intérêt, en y incorporant les nouveaux points lorsqu'ils apparaissent, serait une solution acceptable. En pratique, c'est sans compter sur le fait que cette dérive sera en outre accélérée par des mises en correspondance incorrectes et la présence de points aberrants. Cette seconde source d'erreur ne sera pas, quant à elle, totalement compensée au cours du temps.

En outre, lorsque l'on étudiera le cas d'objets déformables, les points d'intérêt ne permettront plus de décrire correctement le modèle de l'objet suivi, à moins de parfaitement modéliser les déformations, ce qui, nous l'avons dit, est en général impossible.

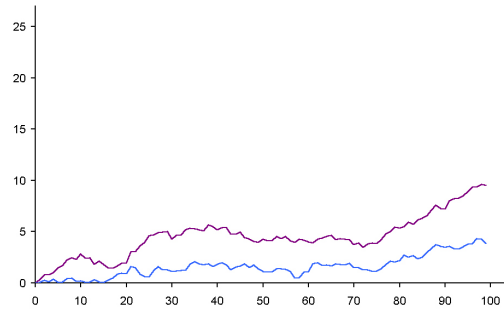


FIG. 7.8 – Bien qu'il n'y ait aucune occlusion, les mises en correspondance ne sont pas toujours parfaites du fait du bruit, entraînant une dérive certes lente, mais néanmoins présente.

Le suivi par points ne saurait donc convenir, du moins en l'état, pour suivre un objet correctement pendant plus de quelques instants, parce qu'il dérive dans le temps de par son principe même. Il est cependant utile de savoir que plus les points observés seront pérennes, plus la dérive sera lente. Il nous faut pourtant trouver des solutions pour limiter cette dérive, afin de pouvoir suivre l'objet étudié sur une séquence notablement longue.

7.4 Suivi par ajustement de modèle

7.4.1 Introduction

Le principal défaut de la méthode précédente, à savoir la possibilité de rencontrer des erreurs qui s'accumulent au cours du temps, provient essentiellement, comme nous l'avons dit, du fait que l'on tente d'évaluer les déplacements (ou de façon équivalente les vitesses) entre deux images, et que la position est obtenue par intégration. Pour contourner ces difficultés, il est nécessaire d'envisager des méthodes qui estiment directement la position de l'objet. C'est le cas des méthodes par ajustement, qui s'efforcent de placer au mieux un modèle décrivant l'objet en fonction des informations retirées de l'image (points, contours, silhouette, éclairage, couleur, texture, etc.).

Il nous faut choisir quel type d'information on souhaite utiliser pour comparer la position estimée du modèle et la position réelle de l'objet dans les différentes images. Comme nous l'avons souligné, les possibilités sont nombreuses. Compte tenu du fait que notre modèle nous fournit une surface non texturée, nous pouvons envisager d'utiliser la forme, les contours, la

silhouette de l'objet à suivre, ou bien encore des points observés à la surface de ce dernier.

L'extraction des contours d'un modèle (et ultérieurement la comparaison de ces contours à ceux effectivement extraits des différentes images) est généralement envisageable. Lorsque la forme de l'objet est décrite par une isosurface (comme c'est le cas de notre modèle de main, par exemple), il existe différentes solutions pour ce faire (extraction numérique des contours, extraction après une synthèse d'image du modèle, etc). qui cependant se révèlent particulièrement coûteuses en termes de temps de calcul, et donc inadaptées un suivi que l'on voudrait, à terme, possible en temps réel. L'extraction d'une silhouette ne serait pas notablement plus rapide, et notablement moins riche (dans le cas d'une main, par exemple, les doigts sont fréquemment les uns devant les autres, privant la silhouette d'une bonne partie de l'information).

En revanche, nous disposons d'un nuage de points 3D extraits par stéréoscopie des différentes images prises par les caméras. *A priori*, si l'objet est opaque (et l'on supposera que ce sera le cas), ces points doivent se situer au niveau de la surface de l'objet. Dans la mesure où l'on connaît la forme de l'objet, estimer sa position revient donc à trouver une position pour le modèle dans laquelle la distance entre sa surface et les points effectivement observés est minimale. C'est cette approche que nous allons à présent développer.

7.4.2 Objectif et critère de minimisation

Revenons au cas de notre objet indéformable. Nous disposons d'un nuage de points 3D, et nous souhaitons trouver le positionnement du modèle décrivant l'objet à l'instant $t + dt$ qui correspond le mieux à ce nuage. Autrement dit, le but consiste à placer le modèle de façon à ce que les points Y_j observés à l'instant $t + dt$ (du moins ceux correspondant effectivement à l'objet et non au fond) se situent au voisinage de sa surface.

On supposera que l'on dispose d'un moyen de mesurer la distance $d(\mathbf{Y}_j, \mathcal{R}, \mathbf{t})$ entre le point \mathbf{Y}_j et la surface du modèle, lorsque celui-ci est placé dans une position définie par la rotation \mathcal{R} et la translation \mathbf{t} . On souhaite trouver la position du modèle, définie par \mathcal{R} et \mathbf{t} , qui minimisera la quantité :

$$E_s(\mathcal{R}, \mathbf{t}) = \sum_i (d(\mathbf{Y}_i, \mathcal{R}, \mathbf{t}))^2. \quad (7.42)$$

On choisira généralement pour d la distance euclidienne à la surface du modèle, mais tout autre choix peut être envisagé. L'important est que cette distance soit nulle pour un point placé sur la surface du modèle, et croît lorsque l'on s'en éloigne. Dans le cas où l'objet rigide que l'on souhaite suivre est un ellipsoïde, par exemple, on pourra choisir la pseudo-distance d' introduite dans le chapitre précédent.

7.4.3 Élimination des outliers

Principe

Malheureusement, dans le nuage des points \mathbf{Y}_i ne contient pas, en général, uniquement des points correspondant effectivement à l'objet que l'on cherche à suivre. On y retrouvera également des points d'intérêt provenant de mauvaises reconstructions, ou bien encore des points issus du fond ou d'autres objets présents dans l'image. Ces points doivent être éliminés de l'expression de E_s , que l'on peut réécrire sous la forme :

$$E_s(\mathcal{R}, \mathbf{t}) = \sum_i \gamma_i (d(\mathbf{Y}_i), \mathcal{R}, \mathbf{t})^2 \quad (7.43)$$

où γ_i prendra idéalement la valeur 1 lorsque le point \mathbf{Y}_i appartient effectivement à la surface de l'objet étudié, et la valeur 0 lorsque ce n'est pas le cas. Il se pose, toutefois, le problème de la détermination des points du nuage appartenant effectivement à l'objet, et donc des différents facteurs γ_i . On ne dispose pas toujours d'éléments suffisamment discriminants sur ces points pour déterminer, *a priori*, si un point est à rejeter ou bien à conserver.

Utilisation de EM pour l'élimination

Une méthode inspirée des idées apportées par l'approche EM, similaire à celle proposée précédemment, peut néanmoins être mise en œuvre. Là encore, les coefficients γ_i font office de variables cachées qui ne nous intéressent pas pour elles-mêmes, mais qui sont indispensables à la bonne estimation des inconnues que sont \mathcal{R} et \mathbf{t} .

Nous attribuerons aux γ_i une valeur comprise entre 0 et 1 qui représentera une estimation de la probabilité que le point correspondant appartient effectivement à l'objet. Plus γ_i sera proche de 1, plus le point sera pris en compte dans le processus d'ajustement. En effet, si la position courante est proche de la solution correcte, les points 3D correspondant effectivement à des points situés à la surface de l'objet se situent à proximité de la surface définie par le modèle. On va donc essayer de s'appuyer davantage sur les points proches de cette surface plutôt que sur des points plus éloignés.

Pour le reste, nous procéderons comme précédemment, en alternant deux étapes durant la minimisation, jusqu'à parvenir à la solution :

- le calcul *a priori* des différents γ_i , basé sur l'estimation courante de la position de l'objet, en fonction de la distance des points \mathbf{Y}_i à la surface du modèle
- l'estimation de la position $\{\mathcal{R}, \mathbf{t}\}$ du modèle qui minimise l'expression E_s , pour les valeurs des γ_i précédemment calculées.

Expression des coefficients γ_i

À l'origine, il est possible de se trouver assez loin de la solution, et la distance des points \mathbf{Y}_i au modèle n'est guère instructive. On souhaiterait alors considérer de façon relativement équivalente les différents points. Mais plus on se rapproche de la solution, plus cette distance

va se révéler instructive, et on voudrait parvenir à rejeter plus efficacement les points les plus éloignés.

En s'inspirant de l'expression précédemment proposée exprimant la probabilité que deux points puissent être associés, nous définissons donc les γ_i de la façon suivante :

$$\gamma_i = e^{-\frac{d(\mathbf{Y}_i, \mathcal{R}, t)^2}{\sigma_s^2}}. \quad (7.44)$$

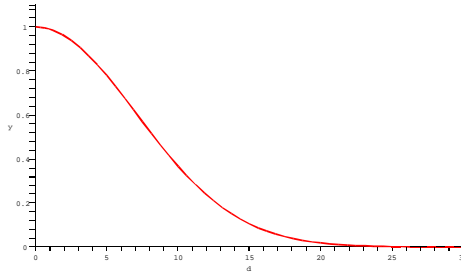


FIG. 7.9 – La probabilité qu'un point donné corresponde effectivement à la surface de l'objet étudié, exprimée à travers le coefficient γ_i , décroît avec la distance d à la position estimée de l'objet. L'équation 7.44 est ici tracée en fonction de d pour $\sigma_s = 10$.

Pour une valeur donnée de σ_s , plus un point \mathbf{Y}_i est situé loin de la position courante de l'objet, plus le coefficient γ_i est faible. Ces coefficients γ_i nous serviront donc à pondérer les contributions des différents points \mathbf{Y}_i lorsque l'on estimera la position de l'objet.

Le paramètre σ_s , quant à lui, nous renseigne sur le voisinage dans lequel il convient de chercher les candidats, comme c'était le cas précédemment avec le paramètre σ_p . À l'origine, on choisira une valeur élevée pour σ_s , de l'ordre de grandeur de la distance séparant initialement les points 3D de la surface du modèle. Les points trop éloignés de l'objet, qui appartiennent vraisemblablement au fond (ou sont issus d'une mauvaise reconstruction) seront d'ores et déjà éliminés. Parmi les points pas trop éloignés de l'objet, qui font de bons candidats en tant que points effectivement à la surface de l'objet, les coefficients γ_i seront comparables. Aucun d'entre eux ne sera rejeté trop rapidement.

Puis, au fur et à mesure que l'on se rapproche de la solution, nous allons faire décroître ce paramètre σ_s , exprimant la confiance croissante que l'on a dans la position du modèle. Tout point situé à une distance à la surface nettement supérieure à ce paramètre verra son importance dans la minimisation décroître, son paramètre γ_i tendant progressivement vers 0. De cette façon, les points trop éloignés seront progressivement éliminés, et n'influenceront plus l'estimation de la position de l'objet suivi. Si on le souhaite, on peut annuler complètement la contribution des points dont le γ_i devient trop faible, cette contribution n'influant de toute façon plus notablement sur l'estimation de la position de l'objet.

La décroissance du paramètre σ_s , encore une fois, doit s'achever lorsque l'on atteint une distance correspondant aux erreurs que l'on peut commettre par ailleurs. Dans le cas qui nous intéresse ici, à savoir l'ajustement à un modèle, c'est à la fois le bruit que l'on a sur les points du nuage, mais aussi éventuellement les défauts de la modélisation de l'objet. En effet, la surface du modèle ne reproduit pas nécessairement très précisément celle de l'objet étudié.

Algorithme 5 Suivi d'un objet rigide par ajustement, avec élimination des outliers

définir σ_s^0 : ordre de grandeur du mouvement à suivre
 définir σ_s^∞ : bruit sur les points reconstruits et imprécision sur la modélisation de la surface

initialiser $\sigma_s \leftarrow \sigma_s^0$

initialiser $\mathcal{R}_0 \leftarrow Id$ et $\mathbf{t}_0 \leftarrow \mathbf{0}$

initialiser $k \leftarrow 0$

répéter

pour tout point \mathbf{Y}_i :

estimer $\gamma_i = e^{-\frac{d(\mathbf{Y}_i, \mathcal{R}_k, \mathbf{t}_k)^2}{\sigma_s^2}}$

fin pour

$k \leftarrow k + 1$

déterminer \mathcal{R}_k et \mathbf{t}_k minimisant $E_s(\mathcal{R}_k, \mathbf{t}_k) = \sum_i \gamma_i (d'((\mathbf{Y}_i), \mathcal{R}_k, \mathbf{t}_k))^2$

jusqu'à ce que la différence entre \mathcal{R}_{k-1} et \mathcal{R}_k et entre \mathbf{t}_{k-1} et \mathbf{t}_k tombe en deçà d'un seuil, ou bien que $k > k_{max}$

7.4.4 Avantages et inconvénients de la méthode

L'avantage évident de cette méthode est celui que nous présentions précédemment. Puisque l'on estime directement la position de l'objet étudié, il n'est plus question d'erreurs qui pourraient être traînées d'une image sur l'autre. La qualité de l'ajustement ne dépend pas, en principe, de la première estimation, même si en pratique cette assertion doit être tempérée.

Il est en effet possible de tomber occasionnellement dans des minima locaux, si l'on part d'une estimation trop éloignée de la solution. La chose est d'autant plus courante lorsque les points observés ne correspondent pas à l'ensemble de la surface de l'objet, mais ne couvre que l'un de ses côtés. Partir d'une solution trop éloignée pourrait alors conduire à ajuster la surface du mauvais côté.

Le fait que les points ne couvrent pas de façon uniforme l'objet étudié peut conduire à d'autres difficultés. Même lorsqu'un point est situé à la surface de l'ellipsoïde, il reste de nombreux degrés de liberté possibles. La surface peut en effet "glisser" sur les points, comme on le voit sur la figure 7.10 ci-dessous. L'estimation de la position n'est alors pas très précise.

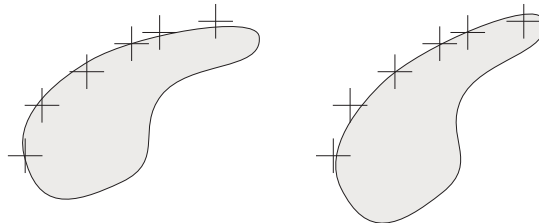


FIG. 7.10 – Il n'est pas rare, en particulier lorsque l'objet montre des éléments de symétrie ou lorsque les points ne couvrent qu'une partie de la surface, de trouver une incertitude importante dans le positionnement du modèle.

Même dans les cas les plus favorables où l'on dispose d'un grand nombre de points couvrant la surface, il est parfois possible qu'il reste des degrés de liberté non contraints. Dans le cas d'un ellipsoïde de révolution, par exemple, la rotation suivant l'axe de symétrie ne changera pas l'expression de E_s , et il n'est donc pas possible de complètement déterminer la pose (ces ambiguïtés pourront éventuellement être levées lorsque l'on aura des contraintes entre différents éléments, cependant).

Un autre inconvénient de cette méthode tient à l'expression de E_s . Le calcul d'une distance entre un point et une surface est bien souvent plus complexe, et par conséquent plus gourmand en terme de ressources, que celui de la distance entre deux points. Il faut donc s'attendre à ce qu'un suivi par ajustement de modèle demande davantage de temps de calcul que les méthodes de suivi de points précédemment présentées. Le seul suivi par ajustement n'est donc en général pas des plus adaptés au suivi en temps réel. On préférera, dans la mesure du possible, disposer d'une bonne première approximation de la position de l'objet avant d'utiliser ce genre de méthode.

7.5 Combinaison des deux approches

7.5.1 Motivation

En fait, les avantages et inconvénients des deux approches présentées se révèlent largement complémentaires. Le suivi point-point est rapide et précis, mais fait montre d'une tendance à accumuler des erreurs dans le temps. Le suivi par ajustement, quant à lui, se révèle plus robuste dans le temps, mais ses limitations et son coût en terme de temps de calcul en limite l'utilisation. Il semble donc tout naturel d'essayer de combiner ces deux méthodes pour obtenir un suivi aussi précis, robuste et rapide que possible.

Dans un premier temps, nous pouvons nous servir du suivi par points afin de calculer une bonne approximation du mouvement. Une fois celle-ci obtenue, l'adjonction de critères d'ajustement à un modèle va permettre de corriger les éventuelles erreurs du suivi par point, d'affiner la position de l'objet, et d'éviter toute dérive dans le temps.

7.5.2 Stratégie de minimisation

On commence donc simplement par utiliser la méthode de suivi de points illustrée au paragraphe 7.3.7 en ajustant les paramètres \mathcal{R} et \mathbf{t} de la transformation \mathcal{T} afin de minimiser l'énergie

$$E_p(\mathcal{R}, \mathbf{t}) = \sum_{i=1}^{N_t} \lambda_i \|\mathcal{T}_{t,t+1} \mathbf{X}_i - \mathbf{Z}_i\|^2 \quad (7.45)$$

$$\text{avec } \lambda_i = \sum_{j=1}^{N_{t+1}} \alpha_{i,j} \text{ et } \mathbf{Z}_i = \frac{1}{\lambda_i} \sum_{j=1}^{N_{t+1}} \alpha_{i,j} \mathbf{Y}_j$$

L'expression choisie pour les coefficients $\alpha_{i,j}$ est, rappelons-le,

$$\alpha_{i,j} = \frac{e^{-\frac{\|\mathcal{T}\mathbf{X}_i - \mathbf{Y}_j\|^2}{\sigma_p^2}}}{\sum_{k=1}^{N_{t+1}} e^{-\frac{\|\mathcal{T}\mathbf{X}_i - \mathbf{Y}_k\|^2}{\sigma_p^2}} + e^{-\frac{d_0^2}{\sigma_p^2}}}. \quad (7.46)$$

On applique donc la minimisation alternée du critère précédemment discutée, alternant évaluation des $\alpha_{i,j}$ et de la transformation $\mathcal{T}_{t,t+1}$. À chaque étape de l'alternance, le paramètre σ_p décroît progressivement ainsi que nous l'avons décrit dans la section 7.3.7. Expérimentalement, une décroissance régie par une loi géométrique, en quatre ou cinq étapes, nous a généralement donné de bons résultats.

Lorsque σ_p atteint sa valeur définitive, on dispose d'une bonne estimation de la position de l'objet. Toutefois, cette position peut être entachée d'erreurs, ainsi que nous l'avons dit, pour des raisons de mises en correspondance incorrecte, de position initiale erronée, ou bien encore à cause d'éventuels points aberrants. Afin d'affiner le résultat, et de corriger ces erreurs, nous allons ajuster la position du modèle sur les données, en faisant intervenir un terme supplémentaire. On cherchera donc dorénavant les paramètres de rotation et de translation minimisant le nouveau terme

$$E(\mathcal{R}, \mathbf{t}) = \omega_p E_p + \omega_s E_s \quad (7.47)$$

où E_p est le terme précédemment évoqué, et

$$E_s(\mathcal{R}, \mathbf{t}) = \sum_i \gamma_i (d(\mathbf{Y}_i), \mathcal{R}, \mathbf{t})^2 \quad (7.48)$$

avec

$$\gamma_i = e^{-\frac{d(\mathbf{Y}_i, \mathcal{R}, \mathbf{t})^2}{\sigma_s^2}} \quad (7.49)$$

La raison pour laquelle on conserve les deux critères en même temps, plutôt que de n'utiliser que l'ajustement à cette étape du suivi, est que l'on souhaite contraindre la solution à rester dans le voisinage de celle estimée par le suivi point à point. Cela permettra notamment d'empêcher tout degré de liberté interne, tel que la rotation de l'objet autour d'un éventuel axe de symétrie. En fait, les mises en correspondance entre points vont faire office de points d'ancrage, et seuls quelques ajustements en position seront possibles.

Il faut cependant, pour permettre la mise en commun des deux critères, leur attribuer un poids relatif dans l'optimisation qui suivra. Ce poids est modélisé par les deux termes ω_p et ω_s . Aucun des deux critères ne doit, *a priori*, jouer un rôle prépondérant, donc ces deux termes serviront à les équilibrer. De façon à ce que les deux contributions soient comparables, on fixera $\omega_p = E_s$ et $\omega_s = E_p$ au moment où l'on débute la phase "mixte", et l'on ne modifiera plus ces coefficients par la suite.

Lorsque l'on débute la phase "mixte", on se trouve en principe assez proche de la solution recherchée. Il n'est donc pas nécessaire de choisir une valeur initiale élevée pour le paramètre σ_s , que l'on fera décroître ensuite (on est d'ores et déjà près de la convergence). On peut, la plupart du temps, simplement choisir directement la valeur minimale de σ_s correspondant au bruit et à l'erreur quant à la forme de la surface et s'y tenir.

En principe, il est possible de continuer la minimisation en alternant des phases d'association (calcul des $\alpha_{i,j}$, λ_i , \mathbf{Z}_i et γ_i) et des phases d'estimation du mouvement (calcul de la transformation $\mathcal{T}_{t,t+1}$ qui minimise E , les éléments précédemment cités étant fixés), jusqu'à ce que les changements dans la position tombent en-dessous d'un seuil fixé à l'avance. Toutefois, dans la pratique, un nombre très réduit de passes (une ou deux) se révèle généralement amplement suffisant pour parvenir à une solution acceptable.

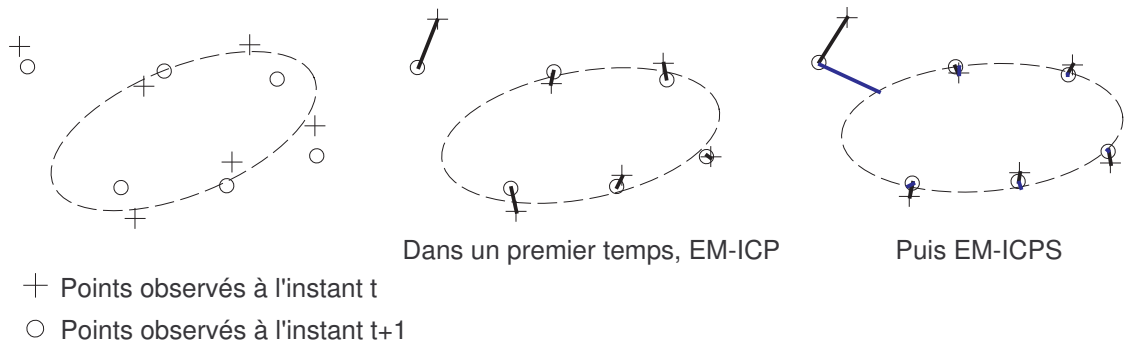


FIG. 7.11 – Le suivi se fait en deux étapes, d'abord un suivi des points 3D reconstruits (EM-ICP), auquel on ajoute un ajustement du modèle sur les observations.

7.5.3 Résultats sur des données simulées

Revenons un temps sur l'exemple décrit dans la section 7.3.8. Lorsque nous essayions de suivre l'objet au moyen des seuls points d'intérêt, on avait pu observer une dérive relativement rapide (après seulement quelques dizaines d'images) qui rendait le suivi impossible. Comme on peut le constater sur la figure 7.12, l'ajout de la phase d'ajustement, selon le schéma qui vient d'être proposé, permet de supprimer cette dérive dans l'estimation de la position. L'erreur reste limitée à quelques dixièmes de millimètres, ce qui correspond, rappelons-le, au bruit que l'on a introduit dans la position des points.

Pour mieux évaluer les qualités de chacune des six méthodes présentées dans ce chapitre, nous les avons confrontées à cinq cents situations identiques, afin de mesurer l'erreur moyenne commise, ainsi que la proportion d'échecs dans la mise en correspondance. Dans les conditions de l'expérience, 60% des points étaient effectivement vus dans chacune des deux images, 20% d'entre eux étaient des points situés à la surface de l'ellipsoïde, mais visibles uniquement dans l'une des deux images, les 20% restant étant des outliers (figurant les erreurs de reconstruction, le fond, etc.). La position de chacun de ces points est perturbée par un bruit gaussien dont l'écart-type est de l'ordre du millimètre. Les résultats obtenus sont résumés dans le tableau ci-dessous.

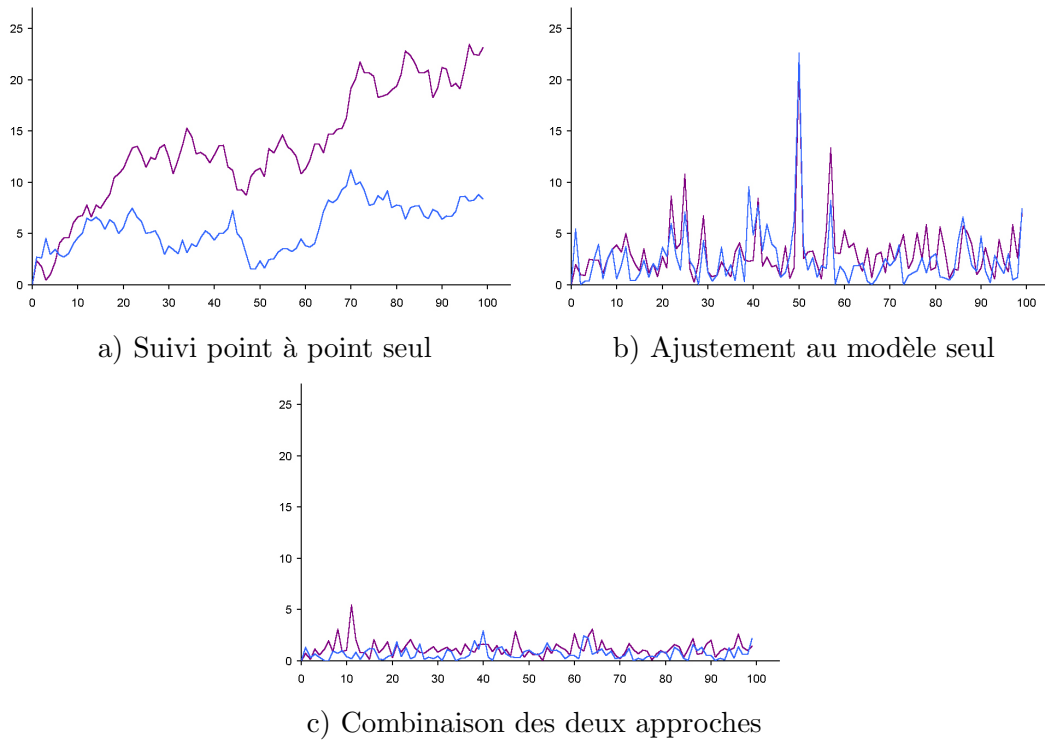


FIG. 7.12 – Erreur en position (courbe violette, en millimètres) et en angle (courbe bleue, en degrés) pour le suivi d'un ellipsoïde sur une même séquence. Le suivi utilisant seulement le mouvement de points d'intérêt dérive avec le temps, l'ajustement seul présente une certaine imprécision sur certaines images, et demande davantage de temps de calcul. La combinaison des deux approches permet d'obtenir un meilleur résultat.

Méthode utilisée	Erreur en position (mm)	Erreur en angle (degré)	Pourcentage de faux appariements
ICP	3,85	3,2	20,6%
EM (σ estimé)	3,75	2,97	15,6%
Soft-Assign	1,24	0,74	0%
Suivi P/P robuste ⁶	1,39	0,80	0,2%
Ajustement	2,16	1,38	2%
Méthode mixte	1,33	0,80	0%

Ainsi qu'on peut le voir, l'algorithme ICP est très sensible à des minima locaux, de même que les méthodes EM lorsqu'on laisse l'algorithme fixer lui-même le paramètre σ plutôt que de lui imposer une loi décroissante. La méthode Soft-Assign et notre propre système de suivi robuste point à point basé sur EM donnent des résultats très similaires. La correspondance bi-univoque entre les points, présente dans Soft-Assign, n'est pas suffisamment utile pour justifier les nombreux calculs supplémentaires nécessaires pour normaliser de façon itérative la matrice de mise en correspondance. La méthode par ajustement donne des résultats un peu moins bons, malgré une situation assez favorable (présence de points sur tout le pourtour de l'ellipsoïde, ce qui ne sera généralement pas le cas en pratique).

La méthode mixte, combinant une mise en correspondance point à point et un ajustement, fournit des résultats légèrement meilleurs que chacune de ses composantes. Mais surtout, elle est, avec les méthodes par ajustement, la seule méthode qui ne dérive pas dans le temps. Après cinq cents étapes successives de suivi, l'erreur moyenne en position reste de 1,3mm environ, alors que les quatre premières méthodes ont perdu la trace de l'ellipsoïde.

7.5.4 Utilisation conditionnelle

En fait, si l'on souhaite développer un système de suivi en temps réel, il n'est en général même pas nécessaire d'imposer une étape d'ajustement à chaque instant, pour chaque série d'images obtenue. En effet, les erreurs provenant du suivi par points sont en général relativement modérées et acceptables pour une suite de quelques images. C'est sur des périodes plus longues que ce type de suivi révèle ses limites et qu'il sera nécessaire de corriger les éventuelles dérives.

On peut donc n'appliquer qu'occasionnellement cette méthode mixte faisant intervenir un ajustement du modèle, et se contenter du suivi par points le reste du temps. Cela peut être fait périodiquement, à intervalles de temps constants, ou bien encore lorsque l'on constate effectivement une dérive dans l'évaluation de la position.

Cette seconde possibilité est plus intéressante, car elle exige un surcroît de ressources uniquement lorsque cela se révèle vraiment nécessaire, et éventuellement à chaque instant si les données se révèlent trop mauvaises pour que le suivi par points fonctionne correctement. La dérive peut être très simplement estimée au moyen du critère E_s . En effet, si la position estimée s'éloigne trop de la position réelle, les points d'intérêt seront relativement loin de la surface du modèle, et le paramètre E_s le mettra en évidence. On peut par exemple utiliser la méthode mixte lorsque, une fois le suivi par points achevé, la valeur de E_s reste supérieure au bruit de la reconstruction des nuages de points et à l'erreur commise sur la modélisation de la forme de l'objet.

7.5.5 Initialisation

Reste à régler le problème de l'initialisation de notre suivi. Dès le premier instant, nous avons besoin de la position estimée de l'objet, et l'on dispose en tout et pour tout d'un nuage de points 3D. Cependant, nous avons vu que la méthode d'ajustement d'une surface sur un nuage de points nous permettait, à condition de n'être pas trop loin de la solution, d'obtenir une estimation correcte de la position de l'objet. C'est donc cette seule seconde méthode, présentée à la section 7.4, qui servira pour initialiser le suivi.

À cause des problèmes liés aux nombreux minima locaux de cette méthode, on supposera toutefois avoir une grossière idée de la position de l'objet. Par exemple, on peut demander à l'utilisateur de le placer dans une position définie à l'avance, disons posé sur la table selon une orientation donnée. On commencera par donner à σ_s une valeur correspondant à la distance maximale probable dans laquelle il faut chercher l'objet, puis diminuer la valeur de σ_s jusqu'à atteindre celle correspondant au bruit et à l'erreur de modélisation précédemment évoquée.

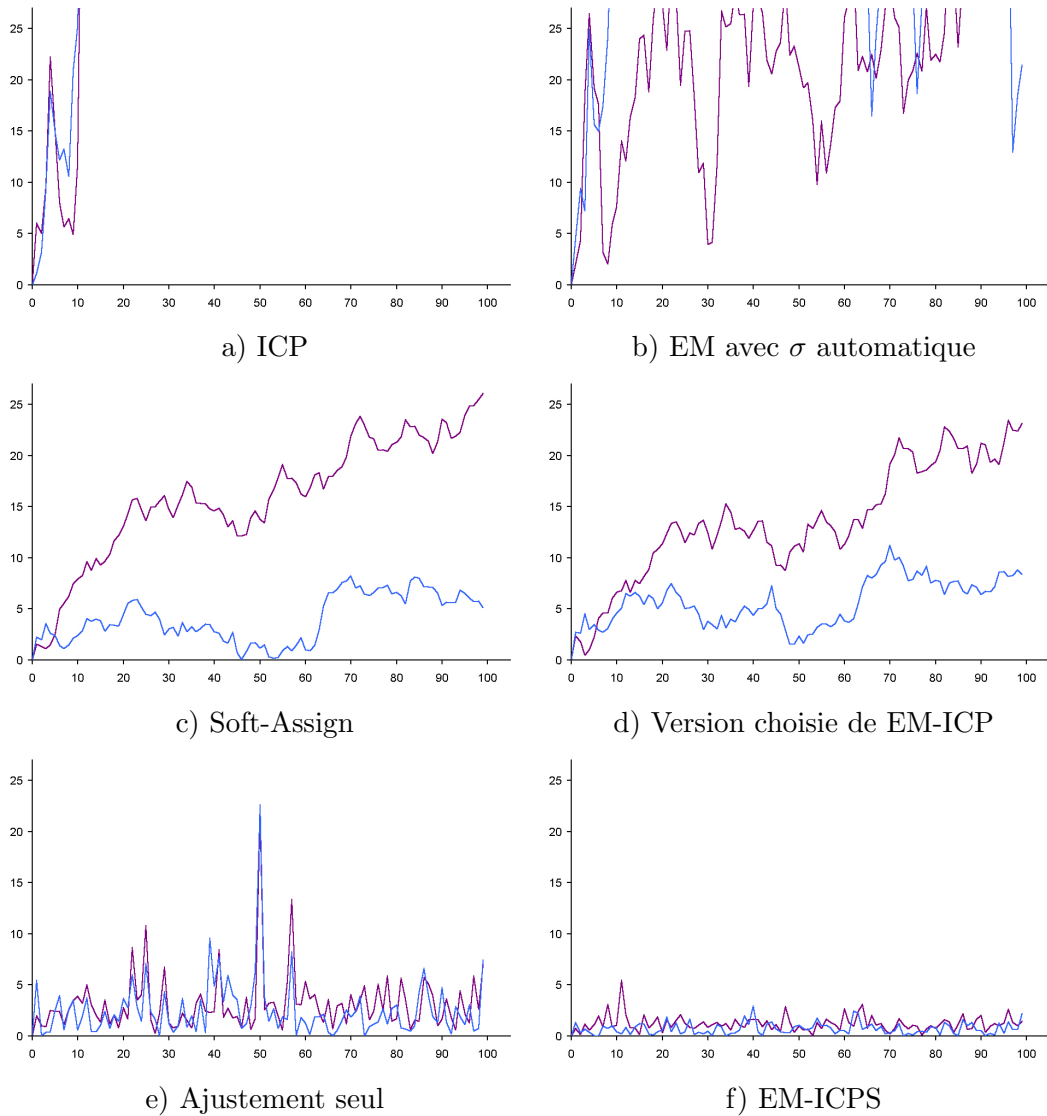


FIG. 7.13 – Erreur en position (courbe violette, en millimètres) et en angle (courbe bleue, en degrés) pour le suivi d'un ellipsoïde sur une même séquence, pour différents algorithmes. Conditions de l'expérience : 28 points par image, dont 8 outliers et un taux de renouvellement des points de 25%. Le bruit sur les mesures est de 2.1.

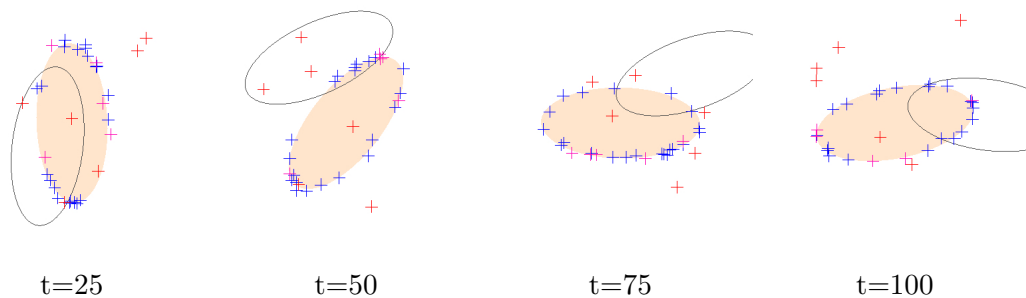


FIG. 7.14 – Suivi d'un objet rigide avec ICP.

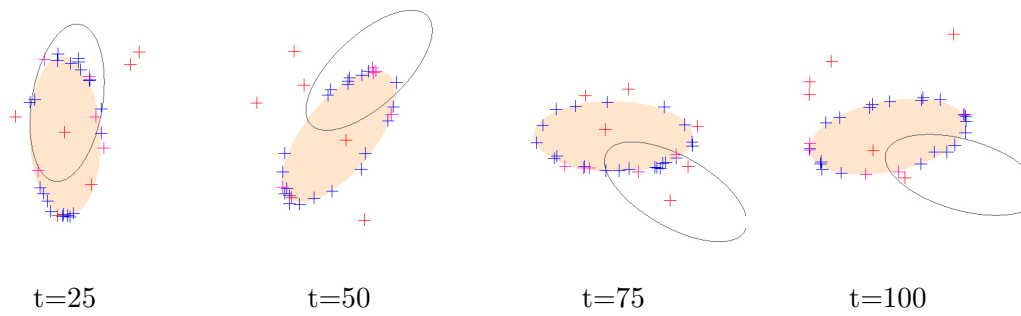


FIG. 7.15 – Suivi d'un objet rigide avec la méthode EM.

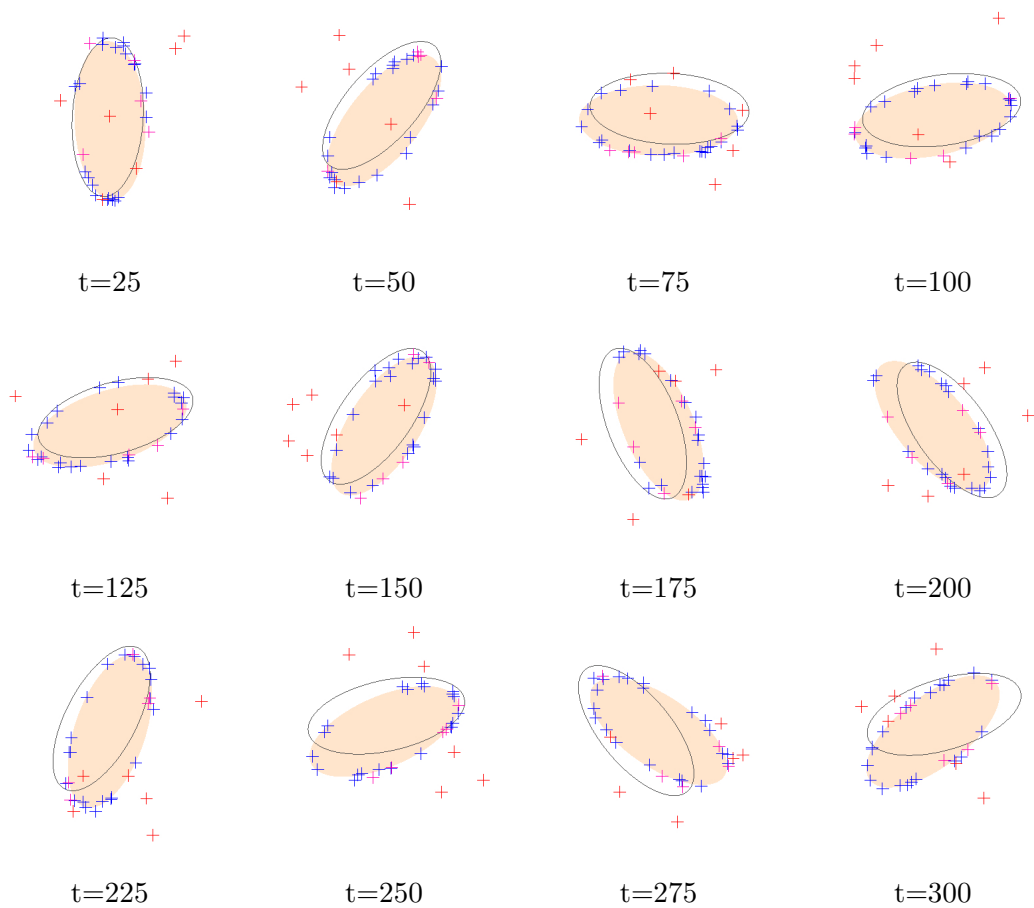


FIG. 7.16 – Suivi d'un objet rigide avec la méthode Soft-assign.

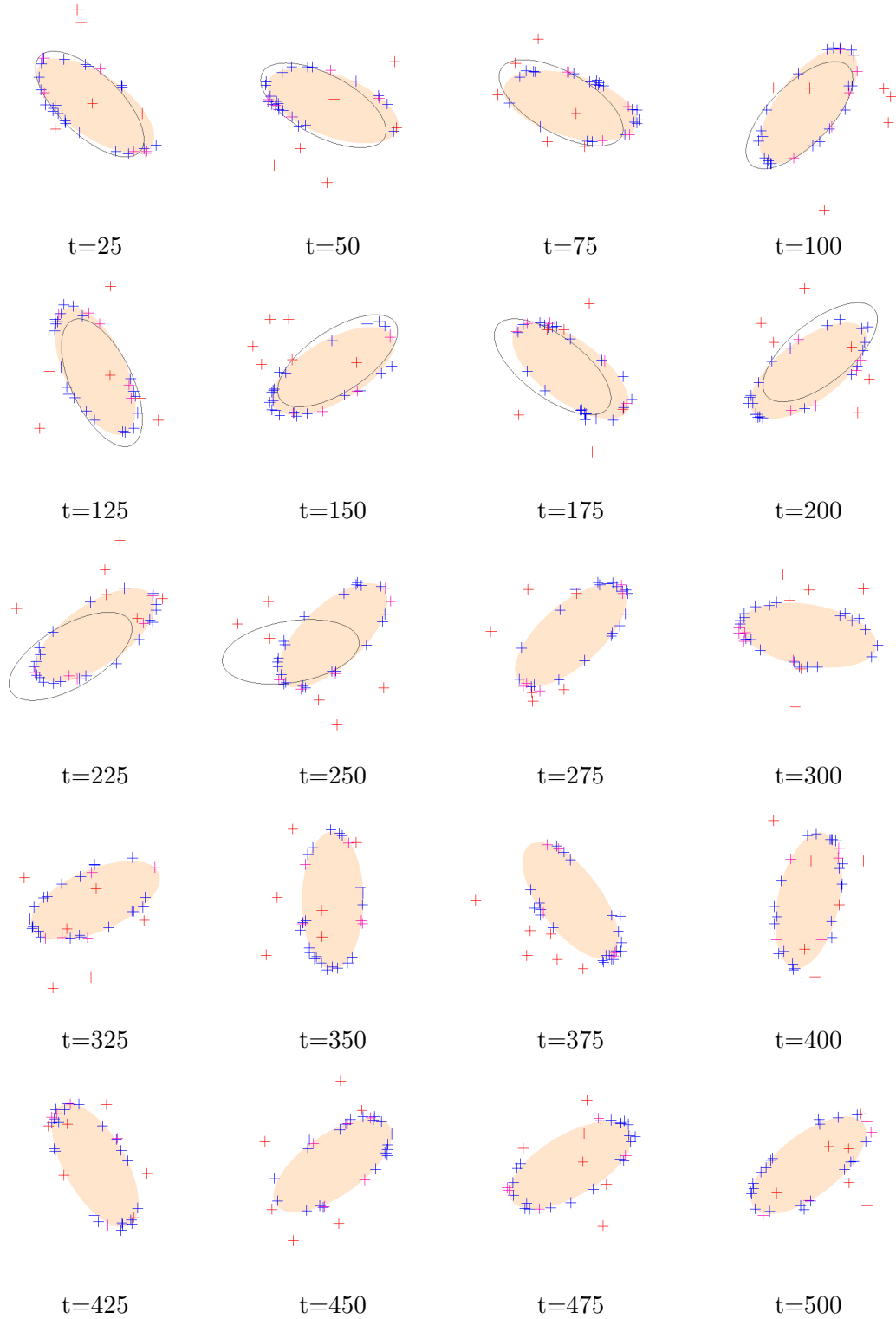


FIG. 7.17 – Suivi d'un objet rigide notre méthode de suivi point à point robuste basée sur EM.

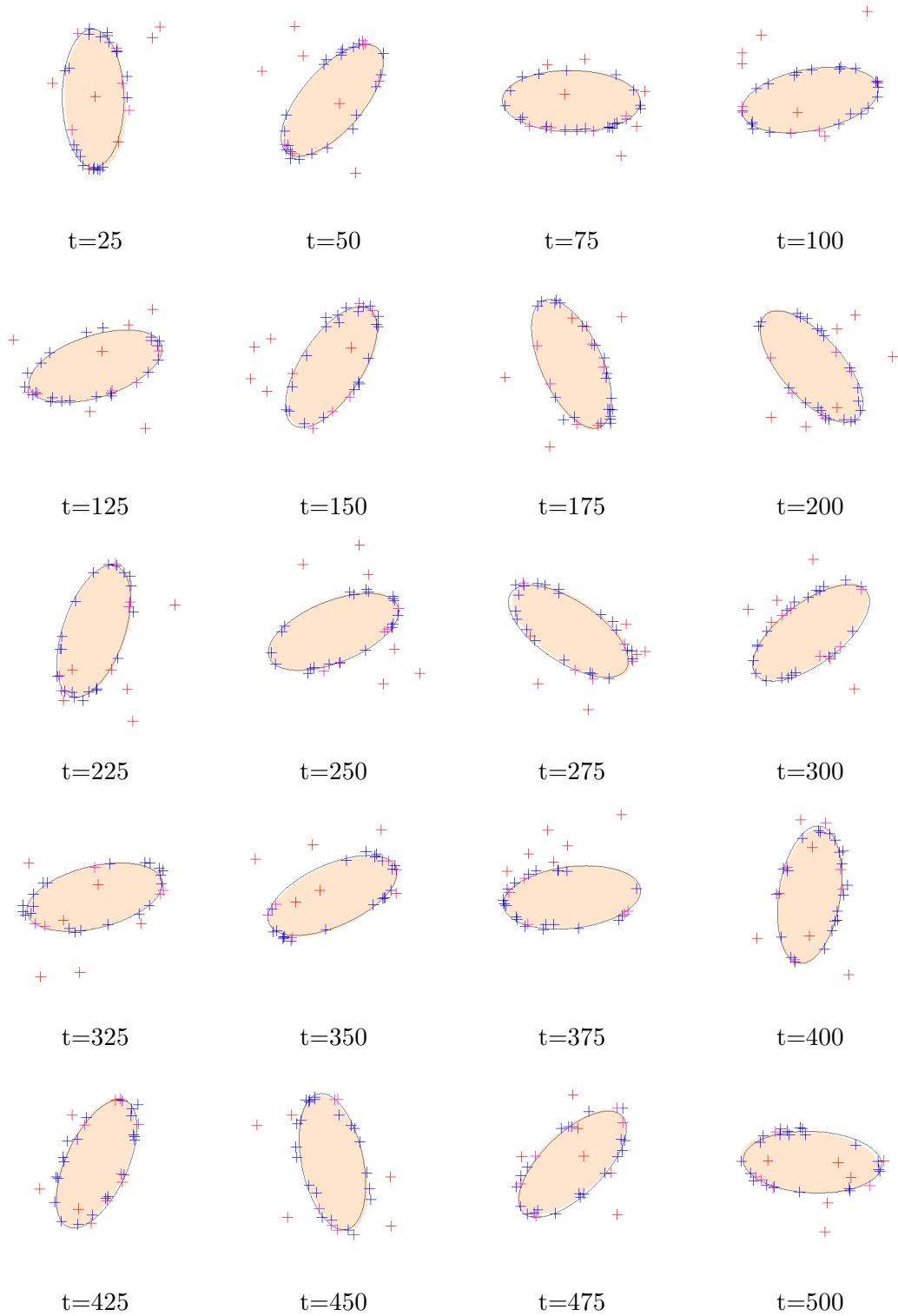


FIG. 7.18 – Suivi d'un objet rigide par simple ajustement d'un ellipsoïde aux données.

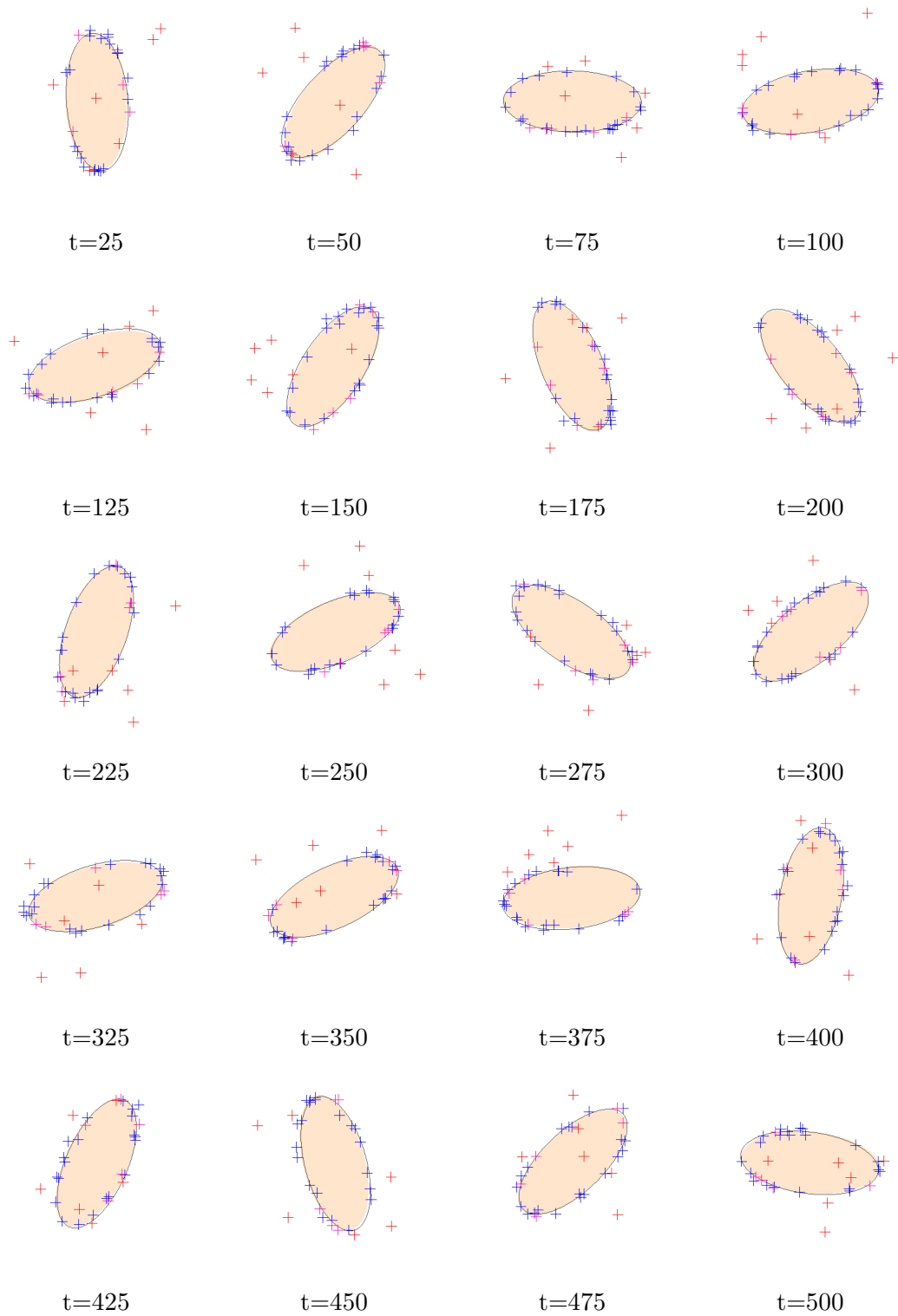


FIG. 7.19 – Suivi d'un objet rigide combinant suivi de points robuste et ajustement.

Chapitre 8

Suivi d'objets articulés et déformables

8.1 Introduction

Dans le chapitre précédent, nous avons proposé une démarche permettant de suivre de façon robuste un objet rigide, basée sur l'extraction de nuages de points 3D observés à sa surface. Nous allons à présent montrer comment ces principes peuvent s'appliquer au cas du suivi d'objets articulés, et même déformables, pour autant qu'il soit possible de décrire les déformations au moyen d'un nombre fini de paramètres. Notre but est de permettre le suivi des mouvements d'un être humain, plus particulièrement les mouvements de ses mains, mais la démarche que nous présenterons ici peut être appliquée au suivi d'un grand nombre d'objets déformables, et sera par conséquent présentée dans un cadre plus général.

Dans le chapitre qui va suivre, nous allons, dans un premier temps, voir comment la démarche exposée précédemment se réécrit dans le cas de solides articulés et déformables. Pour ce faire, nous précisons ce que nous attendons des modèles décrivant de tels objets, plus précisément les outils permettant de modéliser la géométrie de l'objet et les déformations que l'on en attend. Nous réécrirons ensuite chacun des critères dans le cadre de ces modèles articulés et déformables. Nous appliquerons enfin notre méthode au cas particulier du suivi de mouvement de la main, et présenterons les résultats que nous avons pu obtenir.

8.2 Modèles d'objets articulés et déformables

Par “objets articulés et déformables”, nous rappelons que nous entendons parler d'objets dont la forme peut changer, mais dont les déformations peuvent être exprimées au moyen d'un nombre fini de paramètres. Ainsi que nous l'avons dit, cela recouvre la plupart des objets qui nous entourent. Par exemple, les êtres humains et des animaux entrent dans cette définition : bien que susceptibles de subir une très large gamme de déformations, ceux-ci voient leur forme déterminée essentiellement par la configuration d'un squelette sous-jacent. On pourra donc

prendre en compte les mouvements d'un être humain, les déformations de son visage, etc.

A l'inverse, des objets totalement amorphes, dont on ne peut paramétrer les déformations, n'entrent pas dans le cadre de notre étude. Une masse d'argile, par exemple, dont la forme peut être absolument quelconque et n'obéit à aucun modèle particulier, ne pourra pas être étudiée au moyen des méthodes que nous allons à présent détailler. Cela reste toutefois des cas isolés, par ailleurs particulièrement délicats à étudier.

Nous allons formaliser un peu plus ces modèles. Dans la suite, nous noterons par $\{\theta_1, \dots, \theta_n\}$ les différents paramètres définissant la configuration du modèle. La première chose dont nous aurons besoin pour modéliser un objet de ce genre, c'est une description de la forme qu'il prendra, pour des valeurs quelconques desdits paramètres $\theta_1, \dots, \theta_n$. Plus précisément, on souhaite définir la surface de l'objet dans une configuration donnée. Ainsi que l'on a pu le constater dans le cas d'objets rigides, la manière de décrire la surface importe peu. Ce qu'il nous faut, c'est disposer d'un moyen d'évaluer la distance entre des points de l'espace et la surface en question. On pourra choisir la représentation la plus pratique pour la surface, tant que l'on disposera d'un moyen de calculer rapidement la distance $d'(\mathbf{Y}, \theta_1, \dots, \theta_n)$ d'un point \mathbf{Y} quelconque de l'espace à cette surface.

La simple donnée du mouvement de la surface de l'objet, entre deux configurations données, ne suffit toutefois pas à décrire pleinement les mouvements et déformations de l'objet étudié. En effet, même si cela nous permet de décrire les changements de la forme de l'objet, nous n'avons pour l'instant pas d'informations sur la façon dont se déplacerait un point précis de cette surface. La surface peut s'étirer ou se contracter localement, la texture peut glisser le long de la surface, des phénomènes qu'il n'est pas possible de décrire avec uniquement la forme de l'objet.

Or, ces mouvements de texture au niveau de la surface de l'objet seront indispensables à un suivi robuste et fiable de la surface. Ne serait-ce, par exemple, que pour lever les ambiguïtés dues à des symétries de l'objet. Il nous faudra donc également disposer d'un moyen de modéliser ces effets. Plus particulièrement, il nous faut pouvoir estimer le déplacement d'un point de la surface de l'objet lorsque celui-ci se déplace et se déforme, en quelque sorte d'une façon d'"habiller" la surface.

Ces déplacements peuvent être décrits par un champ de vecteurs f , qui dépend tout à la fois de la configuration courante de l'objet (exprimée à travers la donnée des paramètres $\theta_1, \dots, \theta_n$) et des variations de ces différents paramètres, $d\theta_i$, qui conduisent à un mouvement de l'objet. Ce champ de vecteurs permet de connaître le mouvement d'un point quelconque de la surface de l'objet, lorsque celui-ci se déforme. Il sera donc naturellement défini sur l'ensemble de la surface du modèle.

Les points observés à un instant donné ne seront toutefois pas toujours placés très exactement sur la surface de notre modèle. Les raisons en sont nombreuses : reconstruction stéréoscopique imprécise, modèle ne décrivant qu'imparfaitement la surface de l'objet, erreur dans le positionnement du modèle... Il serait donc également utile de savoir comment se déplacerait un tel point, sis à proximité immédiate de la surface. C'est pourquoi on étendra la définition de f à l'ensemble de l'espace, ou du moins à un voisinage immédiat de la surface.

Peu importera la méthode choisie pour étendre le mouvement des points de la surface

à son voisinage, du moment qu'elle reste raisonnablement continue au voisinage de l'objet. L'important étant qu'un point proche de l'objet se voie déplacé comme s'il appartenait à l'objet lui-même. On pourra choisir pour le mouvement d'un point quelconque de l'espace, par exemple, le mouvement du point de la surface le plus proche du point étudié. En définitive, tout dépendra des choix que l'on a faits concernant la représentation de la surface.



FIG. 8.1 – À gauche : mouvement des points de la surface lorsque l'objet se déplace et se déforme. À droite : un exemple de champ de déplacement pour les points au voisinage de l'objet.

Dans la suite, par simple commodité afin de ne pas alourdir les expressions, f représentera, non le déplacement d'un point \mathbf{X} , mais plutôt sa position dans l'espace après le déplacement en question. Pour un objet se trouvant dans une configuration définie par les paramètres $\theta_1, \dots, \theta_n$, un point \mathbf{X} se retrouvera donc à la position $f(\mathbf{X}, \theta_1, \dots, \theta_n, d\theta_1, \dots, d\theta_n)$ lorsque ces paramètres auront varié de $d\theta_i$.

Dans le cas d'un objet indéformable, f se résume simplement à l'application d'une transformation rigide à tout l'espace. Dans le cas d'un objet déformable, elle prendra des formes plus complexes. Ce genre d'expression permet d'envisager une très large gamme de mouvements et de déformations.

La possibilité d'estimer, pour une configuration quelconque de l'objet, la distance d' séparant un point de l'espace avec sa surface, et la façon dont se déplaceraient les points appartenant à l'objet étant donné les variations des paramètres $d\theta_i$, en pratique décrite par la fonctionnelle f , suffiront à permettre son suivi dans le temps, ainsi que nous allons le montrer.

8.3 Suivi point à point

8.3.1 Introduction

Comme nous l'avons expliqué dans le chapitre précédent, le suivi point à point est un suivi itératif dans lequel on s'efforce d'estimer le mouvement de l'objet étudié entre deux instants successifs t et $t + dt$, en supposant la position et la configuration dudit objet connues à l'instant t . On peut alors en déduire position et configuration à l'instant $t + dt$. C'est l'observation du déplacement d'un ensemble de points d'intérêt, situés à la surface de l'objet étudié, qui permettent d'estimer le mouvement de l'objet.

Ce type de suivi suppose donc que l'on puisse lier le mouvement des points observés aux déformations et déplacements de l'objet étudié. Dans le cas particulier d'un solide indéformable, cette relation était immédiate, les points ayant le même mouvement que l'objet auquel ils appartenaient. Les choses sont plus complexes lorsque l'on étudie des objets articulés et déformables, pour lesquels il nous faut également prendre en compte les déformations éventuelles de l'objet.

Le mouvement des différents points suivis n'est alors plus un simple déplacement rigide, mais peut, dans le cadre de notre étude, être décrit par un nombre restreint de paramètres, à travers la fonctionnelle $f(\mathbf{X}, \theta_1, \dots, \theta_n, d\theta_1, \dots, d\theta_n)$ précédemment introduite. L'objectif est alors de déterminer les $d\theta_i$ qui expliqueront le mieux le mouvement observé des différents points.

Les difficultés qu'il nous faudra résoudre sont les mêmes que dans le cas d'un objet rigide : les données extraites de l'image peuvent comprendre un nombre relativement grand de points qui ne correspondent pas à l'objet étudié (mauvaises reconstructions, points appartenant au fond, etc.), points qu'il conviendra de rejeter. En outre, la mise en correspondance entre les points observés à l'instant t et ceux obtenus à $t + dt$ n'est généralement pas triviale. Il faudra corriger, voire complètement déterminer, ces mises en correspondance en même temps que nous évaluerons le mouvement de l'objet. Pour contourner ces écueils, nous étendrons la méthode basée sur EM présentée au chapitre précédent, au cas d'un objet déformable.

8.3.2 Mise en œuvre

La connaissance de la position et de la configuration de l'objet étudié à l'instant t nous permet de filtrer le nuage de points obtenu à cet instant, de façon à éliminer les points aberrants qui ne correspondraient pas à des points situés à la surface de l'objet (points du fond ou points mal reconstruits). On ne conservera donc parmi les points \mathbf{X} du nuage uniquement ceux dont la distance $d'(\mathbf{X}, \theta_1, \dots, \theta_n)$ est inférieure à un seuil (cf. figure 8.2). Ce seuil doit correspondre aux erreurs à la fois sur la reconstruction des points, mais également sur l'imprécision du modèle quant à la forme de l'objet.

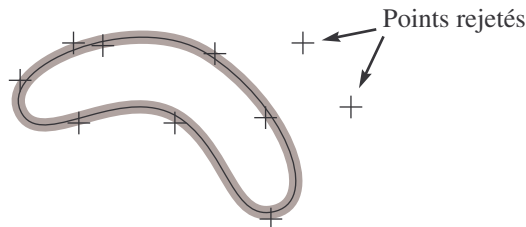


FIG. 8.2 – On ne garde des points reconstruits à l'instant t que ceux situés dans un voisinage de la surface du modèle.

Ensuite, on s'efforce de trouver la combinaison d'associations et de variation des paramètres qui va minimiser la distance entre les points prédits et ceux réellement observés à

l'instant $t + dt$. On s'intéresse donc à minimiser la fonctionnelle¹ :

$$E_p(\{a_{i,j}\}, d\theta_1, \dots, d\theta_n) = \sum_{i,j} \alpha_{i,j} d_{i,j}^2 \text{ avec } d_{i,j} = \|f(\mathbf{X}_i, \theta_1, \dots, \theta_n, d\theta_1, \dots, d\theta_n) - \mathbf{Y}_j\|. \quad (8.1)$$

La fonctionnelle f a endossé le rôle que jouait la transformation rigide dans le cas d'un solide indéformable (se reporter à l'équation 7.9 du chapitre précédent). En lieu et place des six paramètres (trois pour la rotation, trois pour la translation) qui décrivaient le mouvement d'un objet rigide, on trouve les variations $d\theta_1, \dots, d\theta_n$ des n paramètres déterminant position et configuration de l'objet étudié.

Les coefficients $a_{i,j}$, dont la valeur est comprise entre 0 et 1, expriment toujours la plausibilité de l'association du point \mathbf{X}_i du nuage à l'instant t avec celle du point \mathbf{Y}_j du nuage à l'instant $t + dt$. On retrouve enfin la démarche de type EM utilisée pour minimiser la fonctionnelle précédente : estimer alternativement les coefficients $a_{i,j}$ (pour une transformation donnée), puis les paramètres $d\theta_1, \dots, d\theta_n$ (les coefficients $a_{i,j}$ étant constants) qui minimiseront E_p .

8.3.3 Évaluation des coefficients $\alpha_{i,j}$

Les différents coefficients dits d'appariement $a_{i,j}$ restent estimés à partir de la distance entre la position prédite du point \mathbf{X}_i et celle observée du point \mathbf{Y}_j , selon le principe proposé au chapitre précédent reposant sur EM. La distance en question s'exprime cette fois-ci sous la forme

$$d_{i,j}(\theta_1, \dots, \theta_n, d\theta_1, \dots, d\theta_n) = \|f(\mathbf{X}_i, \theta_1, \dots, \theta_n, d\theta_1, \dots, d\theta_n) - \mathbf{Y}_j\|. \quad (8.2)$$

L'expression des différents coefficients $a_{i,j}$ ne change pas, et on a toujours :

$$\alpha_{i,j} = \frac{e^{-d_{i,j}^2/\sigma_p^2}}{\sum_k e^{-d_{i,k}^2/\sigma_p^2} + e^{-d_0^2/\sigma_p^2}}. \quad (8.3)$$

De même, on peut en déduire les coefficients λ_i et les points \mathbf{Z}_i qui serviront à l'estimation de la position avec des formules identiques à celles que l'on a définies dans le cas d'un objet rigide :

$$\lambda_i = \sum_{j=1}^{N_{t+1}} \alpha_{i,j} \quad (8.4)$$

$$\text{et } \mathbf{Z}_i = \frac{\sum_{j=1}^{N_{t+1}} \alpha_{i,j} \mathbf{Y}_j}{\sum_{j=1}^{N_{t+1}} \alpha_{i,j}} = \frac{1}{\lambda_i} \sum_{j=1}^{N_{t+1}} \alpha_{i,j} \mathbf{Y}_j. \quad (8.5)$$

¹En principe, si l'on reprend le cadre strict défini au chapitre précédent, on devrait plutôt considérer la distance entre le point \mathbf{X}_i et le point $f^{-1}(\mathbf{Y}_j, \theta_1, \dots, \theta_n, d\theta_1, \dots, d\theta_n)$, ce qui n'est pas exactement équivalent dans le cas des solides déformables. Toutefois, d'un point de vue pratique, les résultats obtenus sont similaires pour le type de transformation que nous utilisons. Cela revient en fait à échanger les rôles des nuages $\{\mathbf{X}_i\}_{1 \leq i \leq n}$ et $\{\mathbf{Y}_j\}_{1 \leq j \leq m}$ dans l'évaluation de la transformation \mathcal{T} .

Estimation de la position

L'étape suivante, dans l'algorithme EM, va consister à déterminer les déplacements $d\theta_i$, pour les λ_i et \mathbf{Z}_i précédemment calculés, le critère

$$E_p(d\theta_1, \dots, d\theta_n) = \sum_{i=1}^{N_t} \lambda_i \|f(\mathbf{X}_i, \theta_1, \dots, \theta_n, d\theta_1, \dots, d\theta_n) - \mathbf{Z}_i\|^2. \quad (8.6)$$

L'utilisation de méthodes telle que la SVD, couramment utilisées dans les cas les plus simples, n'est généralement plus possible lorsque l'expression de la fonctionnelle f est complexe. On utilisera donc simplement des méthodes d'optimisation numérique pour trouver la valeur des différents paramètres qui minimisera le critère E_p . On notera que ce critère se présente sous la forme d'une somme de termes au carré :

$$E_p(d\theta_1, \dots, d\theta_n) = \sum_{i=1}^{N_t} E_{p,i}^2 \quad (8.7)$$

$$\text{avec } E_{p,i}(d\theta_1, \dots, d\theta_n) = \sqrt{\lambda_i} \|f(\mathbf{X}_i, \theta_1, \dots, \theta_n, d\theta_1, \dots, d\theta_n) - \mathbf{Z}_i\|. \quad (8.8)$$

Cette optimisation s'adapte donc très bien à des techniques de minimisation numériques de type Levenberg-Marquart [GMW81], qui prendront en entrée les dérivées des différents termes $E_{p,i}$ par rapport à chacun des n paramètres du mouvement $d\theta_1, \dots, d\theta_n$.

Ce déplacement sera ensuite utilisé pour calculer de nouvelles mises en correspondances (plus précisément pour mettre à jour les coefficients $a_{i,j}$ puis les λ_i et \mathbf{Z}_i), avant de réitérer l'opération. L'ensemble des opérations est résumé dans l'algorithme 6.

Étant donné que le suivi alternera des mises en correspondance avec ces estimations du déplacement, il n'est pas nécessaire d'utiliser les méthodes d'optimisation sur le critère précédent jusqu'à en trouver le minimum. Il suffit en effet d'obtenir une meilleure estimation du mouvement, qui conduira à des mises en correspondance plus proches de la solution. La solution finale sera trouvée de façon progressive.

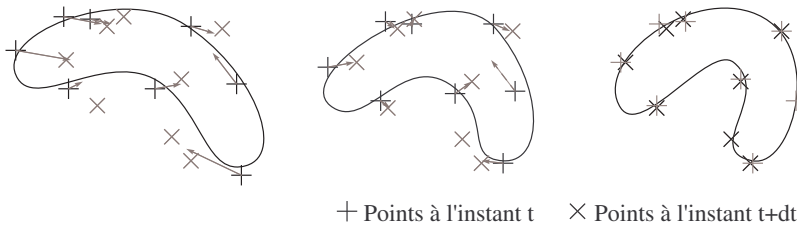


FIG. 8.3 – Suivi grâce à EM d'un objet déformable en éliminant les points aberrants.

Limitations de la méthode

Si un suivi de mouvements basé sur ce principe pouvait encore sembler envisageable dans le cadre d'un solide indéformable, les choses sont à présent beaucoup plus délicates.

Algorithme 6 Suivi point à point d'un objet déformable avec élimination des outliers ([‡] avec optimisation)

définir d_0 : bruit sur les points 3D

définir $\sigma_p^0 =$ ordre de grandeur du mouvement à suivre et $\sigma_p^\infty = \frac{d_0}{3}$

initialiser $\sigma_p \leftarrow \sigma_p^0$, $k \leftarrow 0$ et $d\theta_i \leftarrow 0$

répéter

déterminer $d_r^2 \leftarrow d_0^2 / \sigma_p^2$ et $\lambda'_r \leftarrow e^{-d_r^2}$

pour tout point \mathbf{X}_i :

déterminer le point prédit $\mathbf{X}'_i \leftarrow f(\mathbf{X}_i, \theta_1, \dots, \theta_n, d\theta_1, \dots, d\theta_n)$

initialiser $\lambda'_i \leftarrow 0$ et $\mathbf{Z}_i \leftarrow \mathbf{0}$

pour tout point \mathbf{Y}_j :

déterminer $d_{i,j}^2 \leftarrow \|\mathbf{X}'_i - \mathbf{Y}_j\|^2 / \sigma_p^2$

si $d_{i,j}^2 - d_r^2 < 5$ [‡] **alors**

déterminer $\alpha'_{i,j} = e^{-d_{i,j}^2}$

$\lambda'_i \leftarrow \lambda'_i + \alpha'_{i,j}$ et $\mathbf{Z}'_i \leftarrow \mathbf{Z}'_i + \alpha'_{i,j} \mathbf{Y}_j$

fin si

fin pour

définir $\lambda_i \leftarrow \frac{1}{1 + \lambda'_r / \lambda'_i}$ et $\mathbf{Z}_i \leftarrow \frac{1}{\lambda_i \lambda'_i} \mathbf{Z}'_i$

fin pour

$k \leftarrow k + 1$

déterminer les paramètres $d\theta_i^k$ minimisant $E_p(d\theta_1, \dots, d\theta_n) = \sum_i \lambda_i \|f(\mathbf{X}_i, \theta_1, \dots, \theta_n, d\theta_1, \dots, d\theta_n) - \mathbf{Z}_i\|^2$

si $\sigma_p > \sigma_p^\infty$ **alors**

faire décroître σ_p : $\sigma_p \leftarrow cste \times \sigma_p$

fin si

jusqu'à ce que $\sum_i |\theta_i^k - \theta_i^{k+1}|$ tombe en deçà d'un seuil, ou bien que $k > k_{max}$

Le modèle (plus particulièrement la fonctionnelle f décrivant les déformations) ne reproduit généralement pas fidèlement de ce qui se passe dans la réalité. Si l'on songe au cas du suivi de la main, la peau, qui se plisse et s'étire de façon complexe, ne fait l'objet que d'une représentation grossière.

De ce fait, les sources d'erreurs sont encore plus importantes, et dans la pratique, bien souvent, le suivi est perdu après seulement quelques dizaines d'images. Les points observés sont alors trop loin de la position estimée du modèle, et rejetés dès l'initialisation du suivi. Ce type de méthode n'est donc pas adapté, du moins de façon isolée, pour le suivi d'un objet déformable. Du moins le sera-t-il sans l'adjonction d'une méthode permettant un recalage régulier du modèle.

Stratégie de minimisation

Le nombre de degrés de liberté du modèle déformable peut être important. Dans le cas de la main, par exemple, pas moins de vingt-six à vingt-sept paramètres devront être évalués. C'est potentiellement beaucoup pour la minimisation, et l'on peut songer à procéder en plusieurs étapes successives. Dans un premier temps, la minimisation peut par exemple être restreinte à un sous-ensemble des paramètres $\{d\theta_1, \dots, d\theta_n\}$. Les autres seront gardés fixés, par exemple nuls.

Le choix de la stratégie dépend alors beaucoup du problème que l'on cherche à résoudre. On pourra parfois également exploiter le fait que les différents termes $E_{p,i}$ ne dépendent souvent que d'un nombre assez restreint de paramètres. Dans le cas de la main, par exemple, il est envisageable d'estimer d'abord la position de la paume, puis celle des doigts, l'un après l'autre. Le mouvement des points situés sur l'index, en effet, ne dépend que du mouvement de la paume et des différentes articulations de l'index, et non des dix-sept paramètres correspondant aux articulations des quatre autres doigts.

Dans le cas précis du suivi de la main, il se trouve que cette approche ne nous a pas fourni de résultats très convaincants. La raison en est que l'estimation de la position de la paume n'est pas particulièrement précise, car sa modélisation est la plus difficile. On voit en effet que des étirements complexes de la peau peuvent survenir lors du mouvement des doigts, et ces étirements sont assez mal retranscrits par notre modèle. Si la pose de la paume est mal estimée, la qualité du suivi s'en ressentira également sur chacun des doigts dont le placement par rapport à la paume est largement contraint. Dans le cas du suivi de la main, on gagne donc finalement à optimiser directement la totalité des paramètres, même si dans bien d'autres applications, une telle minimisation par étapes pourrait se révéler des plus utiles. En fait, l'estimation du mouvement des doigts va contraindre les positions possibles de la paume, qui de ce fait se verra suivie avec davantage de succès.

8.4 Suivi par ajustement au modèle

La méthode d'ajustement au modèle présentée dans le chapitre précédent peut elle aussi être étendue très naturellement au cas de solides articulés et déformables. Pour une posi-

tion et une configuration données de l'objet déformable dans l'espace (c'est-à-dire la donnée des différents paramètres $\theta_1, \dots, \theta_n$), il est possible de calculer la distance d'un quelconque point de l'espace \mathbf{Y}_i à la surface de l'objet. Nous noterons dans la suite cette distance $d_i = d(\mathbf{Y}_i, \theta_1, \dots, \theta_n)$.

L'ajustement déterminera la position et la configuration du modèle qui permettra de minimiser les distances d_i entre la surface du modèle et les points \mathbf{Y}_i du nuage correspondant effectivement à des points observés à la surface de l'objet. On cherchera donc à minimiser une énergie de la forme

$$E_s(\theta_1, \dots, \theta_n) = \sum_i \gamma_i d_i^2$$

où γ_i est un terme compris entre 0 et 1 représentant la confiance que l'on a dans le fait que le point \mathbf{Y}_i soit effectivement un point observé à la surface de l'objet. On se basera, là encore, uniquement sur leur distance au modèle pour le calcul des coefficients γ_i . On choisira donc pour expression de γ_i la quantité

$$\gamma_i = e^{-\frac{d_i^2}{\sigma_s^2}}.$$

Pour déterminer à la fois les paramètres de pose $\theta_1, \dots, \theta_n$ et les coefficients γ_i qui minimiseront la fonctionnelle E_s , on appliquera la stratégie de minimisation alternée propre à EM-ICP, que nous avons présentée précédemment dans le cas particulier d'un objet indéformable : on commence par estimer les γ_i à partir de la position initiale du modèle, puis, en maintenant ces paramètres fixés, on cherche les position et configuration du modèle minimisant E_s . Cette nouvelle position du modèle étant déterminée, on met à jour les coefficients γ_i , et l'on estime une nouvelle fois la position optimale, cette alternance étant répétée jusqu'à satisfaire un critère de convergence.

Le paramètre σ_s détermine l'espace entourant l'objet dans lequel on trouvera les points contribuant notablement à l'ajustement du modèle sur les données. Tout au long de l'ajustement, à mesure que l'on s'approche de la solution recherchée, on prendra donc soin de le faire décroître. Initialement, on choisira σ_s comme étant la distance possible entre l'estimation initiale de la position du modèle et la position réelle de l'objet.

La limite inférieure de σ_s en dessous de laquelle il convient de ne pas descendre doit prendre en compte deux choses : l'erreur de reconstruction sur les points du nuage utilisé pour l'ajustement et l'imprécision éventuelle du modèle quant à la description de la surface de l'objet. Lorsque l'on travaille avec des objets déformables, il n'est en effet pas toujours possible (voire souhaitable) de modéliser dans les détails l'ensemble des déformations observées. Il n'est pas rare que l'erreur commise sur le modèle de surface soit nettement plus grande que celle issue de la reconstruction, et il convient de ne pas rejeter des points appartenant effectivement à l'objet mais n'étant pas à proximité immédiate du modèle de surface.

L'ensemble de l'ajustement est résumé dans l'algorithme 7 ci-dessous.

Algorithme 7 Suivi par ajustement (inspiré par EM) d'un objet déformable, avec élimination des outliers

définir σ_s^0 : ordre de grandeur du mouvement à suivre

définir σ_s^∞ : maximum(erreur de reconstruction, imprécision du modèle)

initialiser $\sigma_s \leftarrow \sigma_s^0$

initialiser $\theta_i \leftarrow 0$

initialiser $k \leftarrow 0$

répéter

pour tout point \mathbf{Y}_i :

déterminer $\gamma_i \leftarrow e^{-\frac{d(\mathbf{Y}_i, \theta_1, \dots, \theta_n)^2}{\sigma_s^2}}$

fin pour

$k \leftarrow k + 1$

déterminer le jeu de paramètres $(\theta_1^k, \dots, \theta_n^k)$ minimisant $E_s(\theta_1, \dots, \theta_n) = \sum_i \gamma_i d(\mathbf{Y}_i, \theta_1, \dots, \theta_n)^2$

si $\sigma_s > \sigma_s^\infty$ **alors**

faire décroître σ_s : $\sigma_s \leftarrow cste \times \sigma_s$

fin si

jusqu'à ce que $\sum_i |\theta_i^k - \theta_i^{k+1}|$ tombe en deçà d'un seuil, ou bien que $k > k_{max}$

8.5 Combinaison des méthodes

C'est dans le cas des solides articulés et déformables que la combinaison des deux méthodes prend tout son sens. Suivre l'objet par ajustement d'un modèle conduirait à de nombreuses difficultés. Des problèmes de performances, tout d'abord, puisque les différents calculs de la distance à l'isosurface nécessitent beaucoup de puissance. Il semble peu raisonnable, sur les machines actuelles, d'espérer un suivi reposant entièrement sur des ajustements avec des performances permettant une utilisation en temps réel.

Mais aussi et surtout de robustesse. En effet, la surface du modèle est ici bien plus complexe qu'un simple ellipsoïde, et on s'expose à rencontrer un grand nombre de minima locaux lors de l'optimisation. Si deux doigts sont proches, par exemple, les points observés entre les deux doigts seront difficiles à attribuer à l'un ou l'autre des doigts. Il n'est alors pas rare de constater que les doigts restent "collés". Une des solutions serait d'examiner les normales au niveau des points d'intérêt pour différencier les différentes situations, mais cela ne saurait résoudre tous les problèmes (notamment de performances). D'un autre côté, compte tenu du bruit, des mises en correspondance incorrectes, des points aberrants et de l'imprécision du modèle concernant les déformations, un suivi des points d'intérêt du modèle ne pourrait fonctionner sur plus de quelques dizaines d'images, dans le meilleur des cas.

Nous combinerons donc les deux approches de la façon présentée au chapitre précédent. Dans un premier temps, un suivi de points d'intérêt nous fournira une première estimation de la nouvelle position de l'objet suivi. L'erreur alors commise peut être estimée par le calcul de E_s , image de la distance entre les points observés et le modèle. Si l'on estime que cette

erreur est trop importante, le mélange des deux approches permettra de recalculer le modèle.

Nous allons donc utiliser un suivi par points, que l'on va corriger par l'ajout d'un terme d'ajustement dès que le besoin s'en fera sentir. Le schéma général, pour le suivi d'un objet déformable, sera donc le suivant. Dans un premier temps, on applique le seul suivi par points, exposé dans l'algorithme 6. Ensuite, on évalue l'erreur sur la position en calculant le terme E_s . Si cette dernière est inférieure à un seuil donné, on s'arrêtera à cette étape. Dans le cas contraire, on applique une méthode mixte combinant EM-ICP et ajustement pour affiner la solution, résumée par l'algorithme 8.

Algorithme 8 Correction de l'estimation de la position d'un objet déformable combinant suivi robuste point à point et ajustement

définir σ_p : bruit sur les points reconstruits

définir σ_s : bruit sur les points reconstruits et imprécision sur la modélisation de la surface

définir $\omega_p = E_s(\theta_1 + d\theta_1, \dots, \theta_n + d\theta_n)$

définir $\omega_s = E_p(\theta_1, \dots, \theta_n, d\theta_1, \dots, d\theta_n)$

initialiser $k \leftarrow 0$

répéter

pour tout point \mathbf{X}_i :

déterminer le point prédit $\mathbf{X}'_i \leftarrow f(\mathbf{X}_i, \theta_1, \dots, \theta_n, d\theta_1, \dots, d\theta_n)$

initialiser $\lambda'_i \leftarrow 0$ et $\mathbf{Z}_i \leftarrow \mathbf{0}$

pour tout point \mathbf{Y}_j :

déterminer $d_{i,j}^2 \leftarrow \|\mathbf{X}'_i - \mathbf{Y}_j\|^2 / \sigma_p^2$

si $d_{i,j}^2 - d_r^2 < 5$ **alors**

déterminer $\alpha'_{i,j} = e^{-d_{i,j}^2}$

$\lambda'_i \leftarrow \lambda'_i + \alpha'_{i,j}$ et $\mathbf{Z}'_i \leftarrow \mathbf{Z}'_i + \alpha'_{i,j} \mathbf{Y}_j$

fin si

fin pour

définir $\lambda_i \leftarrow \frac{1}{1 + \lambda'_i / \lambda'_i}$ et $\mathbf{Z}_i \leftarrow \frac{1}{\lambda_i \lambda'_i} \mathbf{Z}'_i$

fin pour

pour tout point \mathbf{Y}_i :

estimer $\gamma_i = e^{-\frac{d(\mathbf{Y}_i, \mathcal{R}_k, \mathbf{t}_k)^2}{\sigma_s^2}}$

fin pour

$k \leftarrow k + 1$

déterminer le jeu de paramètres $(d\theta_1^k, \dots, d\theta_n^k)$ minimisant

$$E(d\theta_1, \dots, d\theta_n) = \omega_s E_s(\theta_1 + d\theta_1, \dots, \theta_n + d\theta_n) + \omega_p E_p(\theta_1, \dots, \theta_n, d\theta_1, \dots, d\theta_n)$$

jusqu'à $d\theta_{i_{k-1}}$ et $d\theta_{i_k}$ tombe en deçà d'un seuil, ou bien que $k > k_{max}$

8.6 Cas de la main

8.6.1 Application du schéma de suivi

La main entre parfaitement dans le cadre des objets articulés et déformables qui peuvent être suivis au moyen de la méthode que nous venons de présenter. Le modèle décrit dans le chapitre 6 nous fournit en effet tous les éléments dont nous avons besoin pour le mettre en œuvre. Les différents paramètres $\theta_1, \dots, \theta_n$ correspondent dans ce cas particulier à la position de la main dans l'espace, comprenant la position de sa paume (que l'on peut définir au moyen d'une rotation \mathcal{R} et d'une translation \mathbf{t}), et les positions relatives des différentes phalanges, correspondant à la donnée des vingt ou vingt et un paramètres angulaires que comprend le modèle articulaire.

Nous pouvons utiliser la description implicite de la main pour décrire sa surface, et plus précisément la pseudo-distance d' que nous avons introduite pour déterminer la distance entre cette surface et un point quelconque de l'espace, pour une position et une configuration donnée de la main. Cette pseudo-distance s'écrivait :

$$d'(\mathbf{X}) = -\nu \ln \left(\sum d'_k(\mathbf{X}) \right)$$

où $d'_k(\mathbf{X})$ représente la pseudo-distance au k -ième ellipsoïde composant le modèle de main, et où la somme est restreinte à l'ellipsoïde le plus proche du point \mathbf{X} et à ses voisins dans la chaîne articulaire.

Il nous faut également disposer d'une fonction f modélisant le mouvement des points au voisinage du modèle lorsque celui-ci se déplace. Nous avons vu que le mouvement d'un point à la surface du modèle lorsque varient les différents paramètres du mouvement ne peut pas être simplement exprimé, mais qu'il était possible d'en obtenir une approximation satisfaisante basée sur une combinaison linéaire des mouvements des différents éléments constituant le modèle articulaire.

8.6.2 Résultats obtenus

Séquence de synthèse

Dans un premier temps, nous avons testé notre méthode sur des séquences composées d'images de synthèse. Ces séquences ont été obtenues grâce au logiciel Poser, couramment utilisé dans le domaine de l'animation. L'avantage de telles séquences pour notre étude est double. D'une part, il nous est possible d'obtenir aisément les mouvements que l'on souhaite dans n'importe quelle situation, sans contrainte particulière sur le positionnement et la calibration des caméras par exemple. D'autre part, les images ainsi obtenues sont de très bonne qualité, et permettent une bonne extraction des points d'intérêt. C'est donc une première étape intéressante dans le cadre de la validation de notre approche.

Les séquences sont générées selon trois points de vue. Deux vues proches fournissent une vision stéréoscopique de la main, avec laquelle nous allons construire le nuage de points 3D

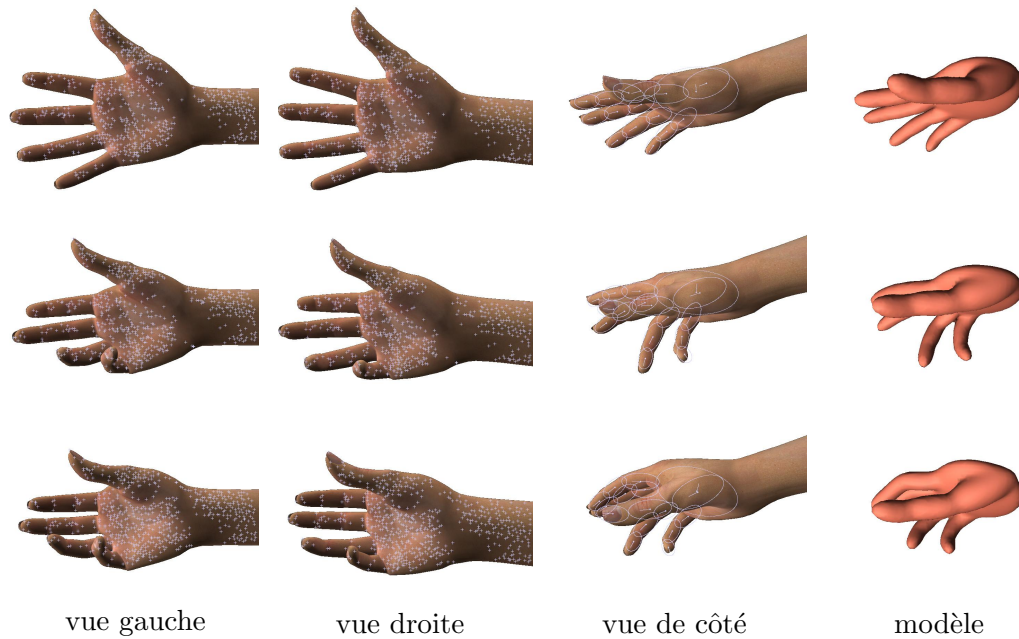


FIG. 8.4 – Résultat du suivi pour les images 10, 30 et 50. À gauche, les deux images stéréoscopiques utilisées pour la reconstruction, avec les points d'intérêt extraits des images. À droite, la vue latérale avec la position des ellipses déterminée par notre algorithme, et un rendu 3D montrant la position du modèle avec le même point de vue.

nécessaire à notre suivi. Pour ce faire, nous extrayons un ensemble de points dans l'une des deux images au moyen d'un détecteur de points d'intérêt utilisant la méthode de Harris. La géométrie épipolaire des caméras, qui a été préalablement établie, permet de déterminer la position tridimensionnelle de chacun de ces points. Pour ce faire, nous cherchons simplement un correspondant à chacun de ces points le long de la droite épipolaire dans la seconde image, et reconstruisons sa position grâce aux méthodes usuelles de stéréoscopie.

Une troisième séquence donne un point de vue latéral de la scène. Elle n'est aucunement utilisée pour le suivi proprement dit, mais nous permet de contrôler son bon fonctionnement. Certaines erreurs de positionnement ne sont en effet pas nécessairement évidentes à voir depuis le point de vue de la paire stéréoscopique, par exemple lorsqu'un des doigts est observé un peu trop près ou trop loin, mais dans l'axe de la caméra.

Si l'on regarde de près les points reconstruits, nous constatons que les points reconstruits entre deux instants successifs ne sont que très occasionnellement les mêmes. Lorsque certains doigts, en particulier l'auriculaire, se replient complètement, les points qui étaient visibles sur le dessous du doigt disparaissent complètement pour être remplacés par une série de nouveaux points situés sur le dessus de celui-ci. Malgré ces difficultés, on peut constater sur la figure 8.4 que le suivi du mouvement est raisonnablement correct.

Séquences réelles

Bien évidemment, les images réelles ne sauraient être aussi propres et nettes que celles utilisées précédemment, de sorte que les tests précédents ne sauraient à eux seuls valider notre démarche. Nous avons donc conduit des expériences similaires dans des conditions réelles. Les caméras utilisées ont une résolution de 640 pixels par 480, et sont réparties de la même façon que précédemment. En particulier, deux caméras monochromes fixées côte à côte nous fournissent une paire stéréoscopique permettant l'extraction des données tridimensionnelles de la scène.

Les points 3D ont cette fois-ci été reconstruits de façon un peu différente. La paire de caméras nous permet de construire une carte de profondeur, grâce à une reconstruction stéréoscopique dense, telle que présentée dans [Dev97]. Un ensemble de points d'intérêt, extraits de l'une des deux images de la même façon que précédemment, sont ensuite traduits en un nuage de points 3D grâce à cette carte de profondeur. La stéréoscopie dense permet d'avoir un peu plus de robustesse sur la reconstruction des points, même si on observe quand même du bruit dans la position des points, et des imprécisions au niveau des contours, où la profondeur varie brusquement.

Précisons qu'il n'a pas été fait d'effort particulier pour obtenir un fond aisément identifiable à la séquence. Les points reconstruits loin de la position du modèle représentant la main sont naturellement rejetés par notre algorithme, et la présence d'objets dans le fond de l'image ne pose pas de difficulté autre que la quantité accrue de données qu'il nous faut manipuler.

Dans la première séquence, on peut voir une main qui se ferme. Malgré quelques occlusions et le retournement de certains doigts, le suivi du mouvement est globalement satisfaisant, même si une géométrie sans doute un peu incorrecte (au niveau des axes des articulations) a pour conséquence un léger décalage dans l'estimation de la position de l'index. Le modèle a été, comme précédemment, initialisé à la main sur la première image de la séquence.

La deuxième séquence montre un mouvement similaire de prise, observé cette fois depuis l'autre côté de la main. Lorsque les doigts sont repliés, l'extraction des points sur les troisièmes phalanges devient délicate, ce qui occasionne une erreur de position lors des deux instants dans la séquence où la main est complètement repliée. Cette erreur est cependant corrigée rapidement lorsque les doigts s'étendent à nouveau, grâce à la partie "ajustement" de notre modèle.

La troisième séquence présente un cas un peu difficile de suivi dans lequel la main subit une rotation de 180 degrés. Cette rotation l'amène à se présenter par le côté, la plupart des doigts étant alors masqués. Notre méthode se sort plus ou moins correctement de ce genre de situation. On peut cependant remarquer une hésitation de l'algorithme sur l'annulaire et l'auriculaire au moment où ils disparaissent derrière les autres doigts. La partie ajustement en vient à positionner ces deux doigts du mauvais côté des quelques points restants avant l'occlusion.

Ce genre de difficulté peut être levée de plusieurs façons. On peut simplement ajouter une seconde caméra stéréoscopique pour lever ces ambiguïtés occasionnelles. Avoir à disposition plusieurs points de vue distincts s'est souvent révélé très utile pour le suivi de la main. Une autre solution possible consiste à tenir compte des normales lors de l'étape d'ajustement, afin

d'éviter que la surface puisse venir se coller du mauvais côté des données. Ce type de solution a été employé avec succès par M. Niskanen *et al*, dans le cadre du suivi du corps humain, dans [NBH05].

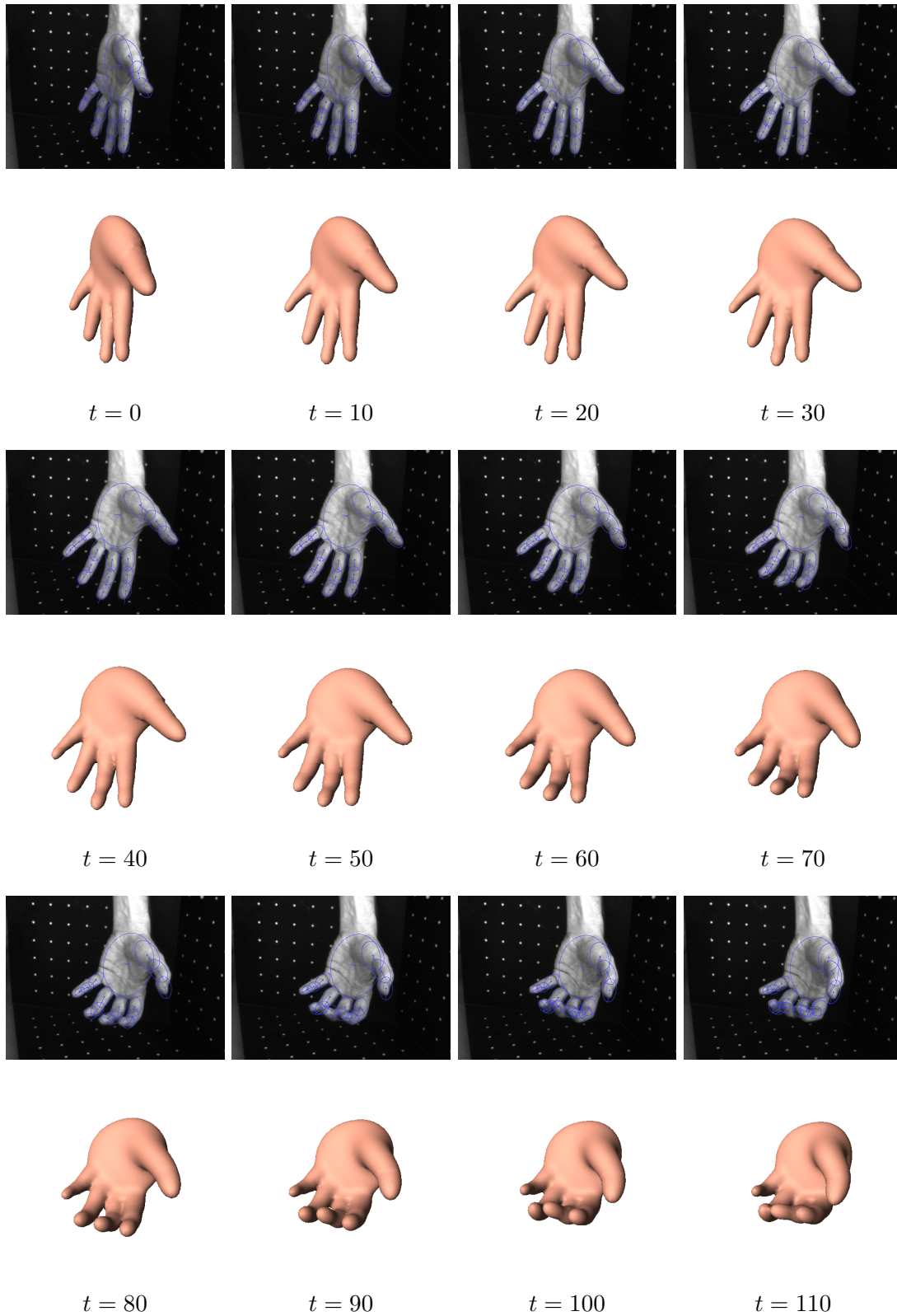


FIG. 8.5 – Un premier mouvement de prise, vu du côté de la paume de la main. Hormis un glissement de l'index dû à un défaut de la cinématique du modèle, le résultat obtenu est satisfaisant.

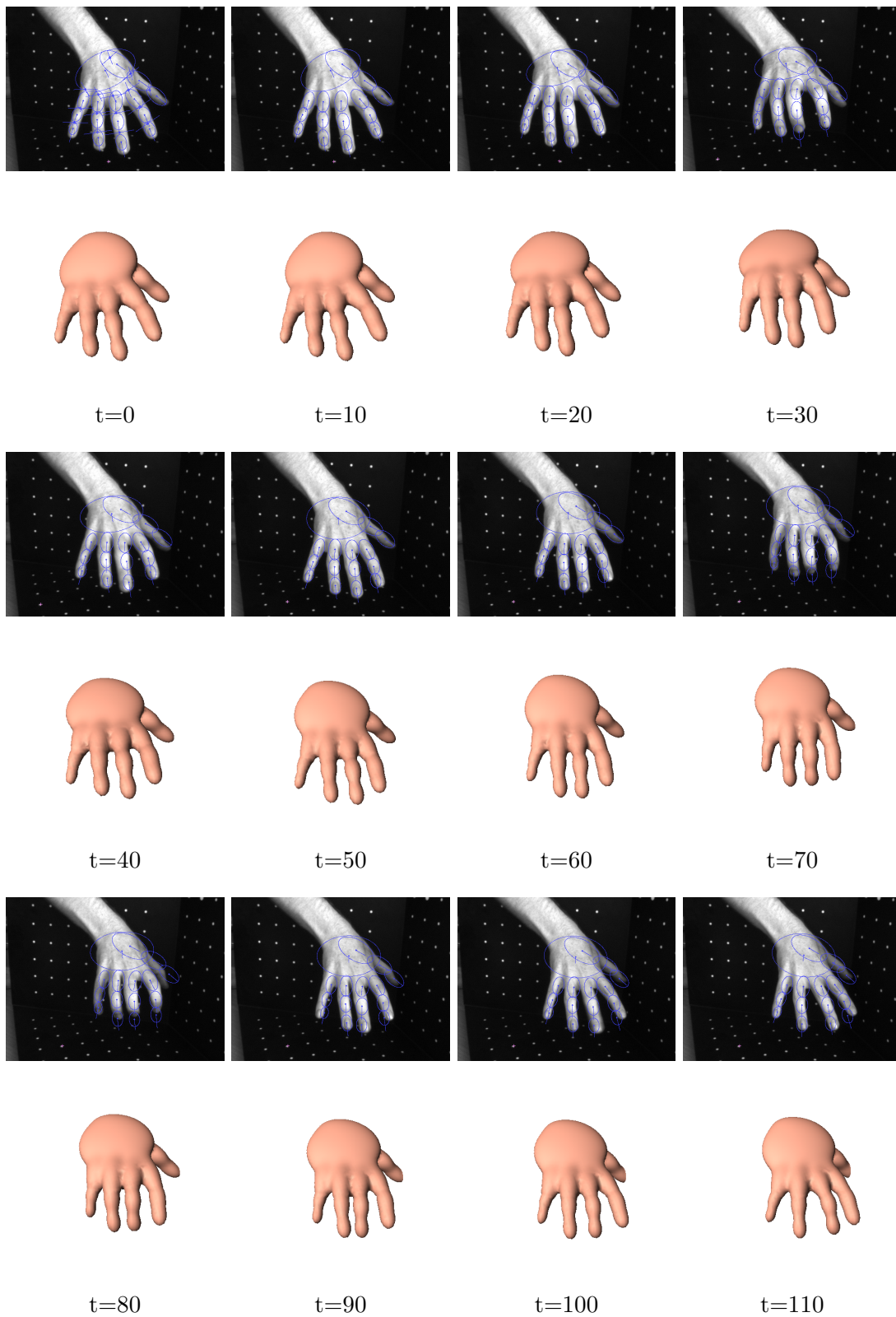


FIG. 8.6 – Plusieurs mouvements de prise successifs, vus cette fois depuis le dessus de la main. Un manque de points occasionnel lorsque les doigts sont repliés cause une erreur, qui est cependant corrigée lorsque les doigts se déplient à nouveau.

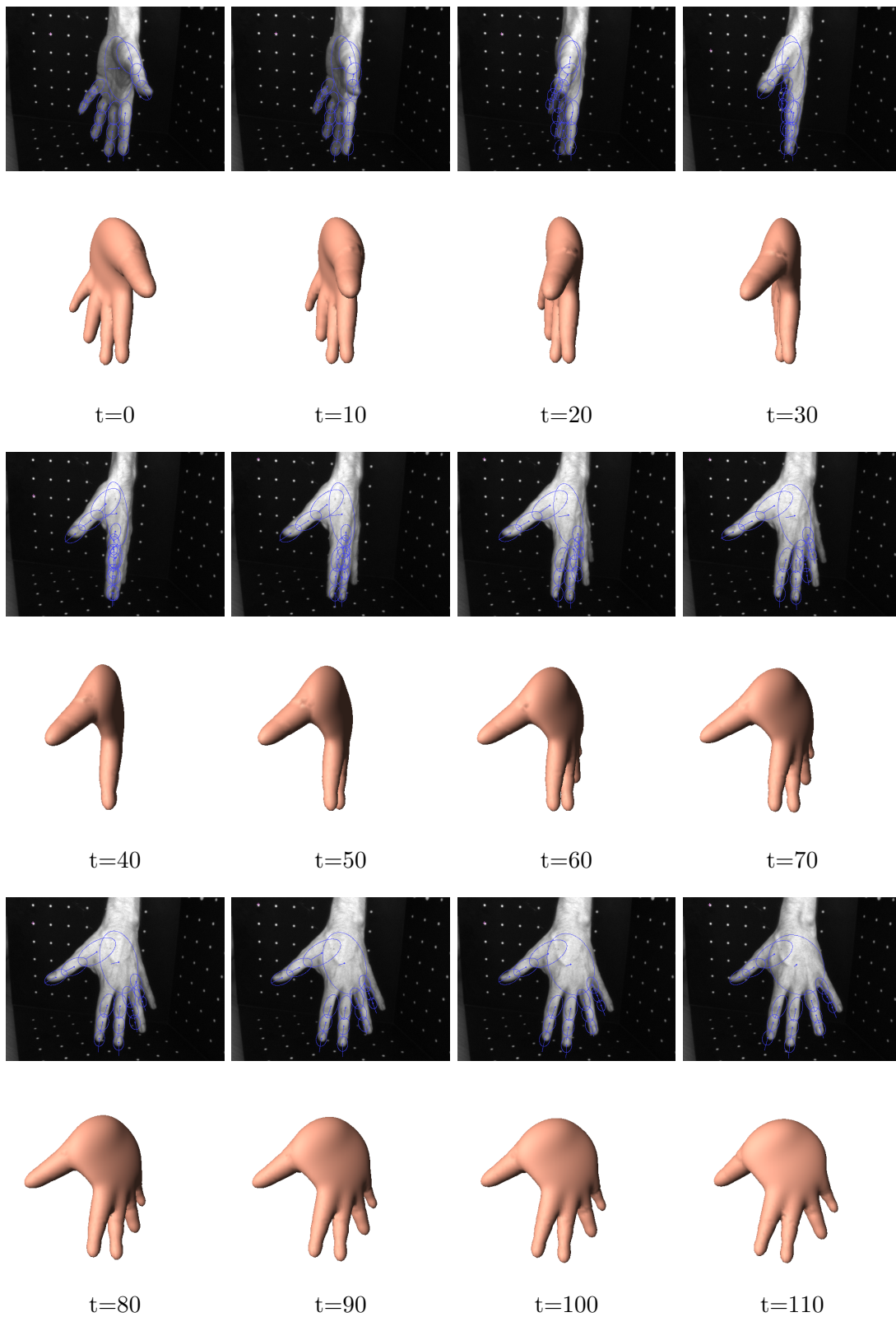


FIG. 8.7 – Un mouvement de retournement de la main cause un grand nombre d'occlusions, ce qui conduit ici à une erreur dans l'estimation de la position de l'annulaire et de l'auriculaire, le modèle se calant du mauvais côté des données juste avant que les doigts ne disparaissent à la vue.

Conclusion

Dans ce document, nous avons présenté le modèle d'argile virtuelle temps réel, ainsi qu'un dispositif de suivi robuste des mouvements de la main. Ces outils ont été développés dans le but de proposer un système dans lequel un artiste interagirait avec une argile virtuelle en effectuant, face à une caméra, les gestes qu'il emploierait avec un matériau réel.

Avant d'atteindre cet objectif, il reste divers problèmes à résoudre. La méthode de suivi de mouvement que nous avons développée nous a semblé être en mesure de déterminer correctement les mouvements de la main, et de corriger d'éventuelles erreurs. Une seconde paire de caméras, placées selon un point de vue différent, serait probablement utile pour permettre un meilleur suivi lorsque l'on est confronté à des occlusions.

Cela mis à part, il faudra inclure l'ensemble des éléments de la chaîne de suivi (extraction des points d'intérêt et suivi proprement dit) dans un processus temps réel. Il faudra également mettre au point un système permettant l'initialisation du suivi, problème difficile en soi que nous n'avons pas, à ce jour, étudié en détail. Ce système permettrait également de reprendre le suivi de mouvement lorsque, suite à une difficulté, il a été interrompu.

Un suivi des points d'intérêt d'une image sur l'autre, enfin, même s'il comporte de nombreuses erreurs dues à la texture, très répétitive et déformable, de la main, nous semble être une piste intéressante. Les méthodes statistiques que nous avons développées dans ce manuscrit peuvent être aisément adaptées à ce cas particulier où l'on dispose d'informations sur les mises en correspondances possibles.

L'argile virtuelle que nous avons développée est, quant à elle, utilisable dès à présent. Il est possible de retrouver l'essentiel des caractéristiques d'une argile véritable dans notre modèle, telles que la conservation du volume, les changements de topologie et les déformations à petite et grande échelle. On peut interagir avec notre modèle en temps réel avec un nombre quelconque d'outils. Cependant, en l'absence d'une interface permettant de contrôler aisément plusieurs outils, il est difficile de savoir dans quelle mesure il est réellement simple et naturel de travailler avec ce matériau virtuel.

Les principaux développements que l'on pourrait faire sur ce matériau sont liés à des questions d'optimisation. Les modèles, bien que simples, sont rapidement gourmands en ressources lorsque l'on travaille sur des objets comportant plusieurs milliers de cellules. Un schéma multirésolution efficace pour chacune des trois couches de simulation, ébauché dans ce manuscrit,

apporterait sans doute beaucoup à ce modèle.

Les deuxièmes et troisièmes couches de notre modèle d'argile, correspondant respectivement aux déformations à grande échelle et à la tension de surface, pourraient également être quelque peu affinées. Soit, dans le premier cas, pour tendre vers un comportement un peu plus naturel dans certaines situations, soit, dans le second cas, pour accroître sa stabilité.

Annexes

Annexe A

Éléments d'analyse vectorielle

Dans le domaine de la physique des milieux continus, comme la mécanique des fluides ou bien encore l'électromagnétisme, les opérateurs différentiels jouent un rôle important. Cette annexe a pour but de présenter rapidement leurs expressions ainsi que certaines de leurs propriétés, ainsi que la forme qu'on leur donne usuellement lorsque l'on travaille sur des données discrétisées dans une grille. Bien que ces objets mathématiques soient définies dans un cadre bien plus général, nous nous restreindrons ici à les exprimer dans le cadre de fonctions scalaires ou de champ de vecteur définis sur tout ou partie de \mathbf{R}^3 .

A.1 Différentielle et gradient d'une fonction

Considérons f , une fonction réelle, définie et continûment dérivable dans une partie de \mathbf{R}^3 . On peut exprimer sa différentielle sous la forme :

$$df = f'_u du + f'_v dv + f'_w dw \quad (\text{A.1})$$

On appelle *gradient* de f , et on note $\mathbf{grad}(f)$, le champ vectoriel défini, pour tout déplacement élémentaire $d\mathbf{r}$, par la relation suivante :

$$df = \mathbf{grad}(f) \cdot d\mathbf{r}. \quad (\text{A.2})$$

Dans un système de coordonnées cartésiennes, les composantes du gradient correspondent très exactement à la dérivée de f par rapport aux trois coordonnées.

$$\mathbf{grad}(f)_x = \frac{\partial f}{\partial x} \quad \mathbf{grad}(f)_y = \frac{\partial f}{\partial y} \quad \mathbf{grad}(f)_z = \frac{\partial f}{\partial z} \quad (\text{A.3})$$

Il est utile de noter qu'un déplacement $d\mathbf{r}$ sur une surface de type $f(x, y, z) = cste$ n'entraîne aucune variation de f , et donc :

$$df = \mathbf{grad}(f) \cdot d\mathbf{r} = 0. \quad (\text{A.4})$$

Le gradient est donc normal aux isosurfaces. Il est orienté suivant les valeurs croissantes de f .

A.2 Flux d'un champ de vecteurs

Considérons une surface \mathcal{S} et un champ de vecteurs \mathbf{A} . Par convention, on appellera *flux* de \mathbf{A} à travers \mathcal{S} la quantité :

$$\Phi = \int_{\mathcal{S}} \mathbf{A} \cdot \mathbf{n} dS \quad (\text{A.5})$$

dans laquelle le vecteur \mathbf{n} est un vecteur normal à la surface.

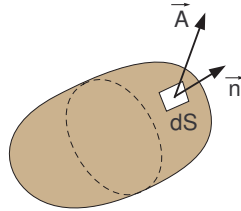


FIG. A.1 – Flux d'un champ de vecteurs à travers une surface fermée.

La divergence d'un champ de vecteur en un point \mathbf{R} correspond à la limite du flux du champ de vecteur \mathbf{A} à travers une surface fermée \mathcal{S} entourant \mathbf{R} lorsque la taille de \mathcal{S} tend vers 0. De façon équivalente, on peut introduire la divergence au moyen de la formule d'Ostrogradsky, reliant le flux d'un champ de vecteurs \mathbf{A} à travers une surface fermée \mathcal{S} à l'intégrale de la divergence dans un volume \mathcal{V} délimité par \mathcal{S} :

$$\oint_{\mathcal{S}} \mathbf{A} \cdot \mathbf{n} dS = \int_{\mathcal{V}} \text{div}(\mathbf{A}) dV. \quad (\text{A.6})$$

Si l'on considère un petit volume parallélépipédique \mathcal{V} , de taille (dx, dy, dz) autour du point (x, y, z) , le flux $d\Phi$ à travers sa frontière s'écrit :

$$\begin{aligned} d\Phi &= \mathbf{A}^x(x + \frac{dx}{2}, y, z) dy dz - \mathbf{A}^x(x - \frac{dx}{2}, y, z) dy dz \\ &\quad + \mathbf{A}^y(x, y + \frac{dy}{2}, z) dx dz - \mathbf{A}^y(x, y - \frac{dy}{2}, z) dx dz \\ &\quad + \mathbf{A}^z(x, y, z + \frac{dz}{2}) dx dy - \mathbf{A}^z(x, y, z - \frac{dz}{2}) dx dy \end{aligned}$$

soit

$$\begin{aligned} d\Phi &= \frac{\partial \mathbf{A}^x}{\partial x} dx dy dz + \frac{\partial \mathbf{A}^y}{\partial y} dx dy dz + \frac{\partial \mathbf{A}^z}{\partial z} dx dy dz \\ &= \left(\frac{\partial \mathbf{A}^x}{\partial x} + \frac{\partial \mathbf{A}^y}{\partial y} + \frac{\partial \mathbf{A}^z}{\partial z} \right) dV. \end{aligned}$$

En coordonnées cartésiennes, la divergence d'un champ de vecteurs \mathbf{A} s'écrit donc :

$$\text{div}(\mathbf{A}) = \frac{\partial \mathbf{A}^x}{\partial x} + \frac{\partial \mathbf{A}^y}{\partial y} + \frac{\partial \mathbf{A}^z}{\partial z}. \quad (\text{A.7})$$

Il est relativement aisé d'associer un sens physique à la divergence d'un champ de vecteurs. La quantité $\int_{\mathcal{S}} \mathbf{A} \cdot \mathbf{n} d\mathcal{S}$ représente d'une certaine façon la bilan entre les entrées et les sorties du volume \mathcal{V} délimité par \mathcal{S} .

Si l'on s'intéresse à un fluide incompressible, la quantité de fluide à l'intérieur du volume reste constant, et le bilan des échanges de fluide au travers d'une quelconque surface fermée \mathcal{S} doit donc être nul. On retrouve donc de cette façon (moyennant quelques précautions) la condition $div(\mathbf{u}) = 0$ en tout point du fluide.

A.3 Circulation d'un champ de vecteurs

Une autre grandeur intéressante, particulièrement en physique, est la circulation d'un champ de vecteurs le long d'un contour fermé \mathcal{C} . Elle s'exprime par :

$$\oint_{\mathcal{C}} \mathbf{A} \cdot d\mathbf{l} \quad (\text{A.8})$$

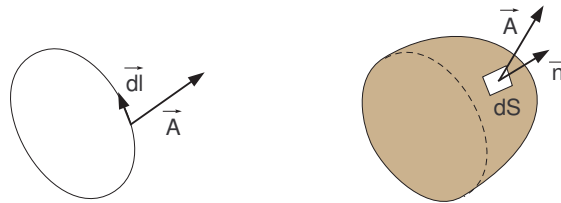


FIG. A.2 – Circulation le long d'un contour fermé \mathcal{C} et rotationnel d'un champ de vecteurs.

De la même façon que pour la divergence, le rotationnel d'un champ de vecteur peut être défini par une équivalence, la formule de Stokes. La circulation d'un champ de vecteurs \mathbf{A} le long d'une courbe fermée \mathcal{C} est égale au flux de son rotationnel à travers une surface \mathcal{S} qui s'appuie sur le contour \mathcal{C} :

$$\oint_{\mathcal{C}} \mathbf{A} \cdot d\mathbf{l} = \int_{\mathcal{S}} \mathbf{rot}(\mathbf{A}) \cdot \mathbf{n} d\mathcal{S}. \quad (\text{A.9})$$

Contrairement à la divergence, le rotationnel d'un champ de vecteur est une grandeur vectorielle. Nous ne nous étendrons pas ici sur ses propriétés, car nous n'en aurons guère l'usage dans ce manuscrit. Précisons quand même que son expression cartésienne est :

$$\mathbf{rot}(\mathbf{A}) = \left(\frac{\partial \mathbf{A}^z}{\partial y} - \frac{\partial \mathbf{A}^y}{\partial z} \right) \mathbf{e}_x + \left(\frac{\partial \mathbf{A}^x}{\partial z} - \frac{\partial \mathbf{A}^z}{\partial x} \right) \mathbf{e}_y + \left(\frac{\partial \mathbf{A}^y}{\partial x} - \frac{\partial \mathbf{A}^x}{\partial y} \right) \mathbf{e}_z. \quad (\text{A.10})$$

A.4 Expressions du second ordre : le laplacien

Nous allons introduire un dernier opérateur vectoriel très utile, le laplacien. Le laplacien d'une fonction scalaire f des coordonnées de l'espace est défini comme la divergence du vecteur $\mathbf{grad}(f)$:

$$\Delta f \equiv div \mathbf{grad}(f) \quad (\text{A.11})$$

Le laplacien est une grandeur scalaire. Notons au passage qu'il est possible de définir un laplacien vectoriel associé à un champ de vecteur \mathbf{A} , en utilisant le laplacien de chacune des composantes du champ de vecteurs \mathbf{A} .

Le laplacien s'écrit donc en coordonnées cartésiennes :

$$\Delta f = \frac{\partial}{\partial x} \left(\frac{\partial f}{\partial x} \right) + \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial y} \right) + \frac{\partial}{\partial z} \left(\frac{\partial f}{\partial z} \right) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}. \quad (\text{A.12})$$

A.5 Expression sur une grille

Il est fréquent, en physique, que les problèmes exprimés sous la forme d'équations faisant intervenir de l'analyse vectorielle ne soient pas solubles de façon analytique. Une des solutions que l'on adopte consiste à discrétiser les champs scalaires et vectoriels du problème sur une grille régulière aussi fine que possible, et de travailler sur les valeurs rangées dans cette grille plutôt qu'avec des fonctions continues dans l'espace. Il est alors nécessaire de déterminer les expressions que prennent ces différents opérateurs lorsque l'on travaille dans une grille. Nous supposons dans notre cas que l'on travaille sur un réseau cubique et régulier.

Revenons pour un temps sur le cas plus simple d'une fonction f définie et dérivable sur \mathbf{R} . La dérivée de f en a est définie comme :

$$\lim_{h \rightarrow 0} \frac{f(a+h) - f(a-h)}{2h}. \quad (\text{A.13})$$

Cette limite existe bien lorsque la fonction f est dérivable en a . Supposons à présent que l'on discrétise cette fonction f sur un ensemble de valeurs x_i réparties régulièrement sur \mathbf{R} , et intéressons-nous à la dérivée de f en un de ces point x_i . On ne dispose plus, à présent, de valeurs x aussi proches qu'on le souhaite de x_i pour lesquelles on connaisse $f(x)$. Il n'est donc plus possible de calculer la limite précédente, et une des approximations raisonnables que l'on puisse faire de $f'(x_j)$ est donc :

$$\frac{f(x_{j+1}) - f(x_{j-1}))}{x_{j+1} - x_{j-1}} = \frac{f(x_{j+1}) - f(x_{j-1}))}{2d} \quad \text{où } d \text{ représente le pas de la grille} \quad (\text{A.14})$$

Il existe d'autres formes pour estimer la dérivée qui nous intéresse, symétriques ou non, faisant éventuellement intervenir les valeurs que prend la fonction f en x_{j-2} et x_{j+2} , etc. Dans le domaine du traitement du signal, on parlera d'estimateurs, et ces expressions présentent diverses qualités en terme de biais (erreur systématique commise dans l'estimation de la dérivée), de résistance au bruit, etc. L'expression précédente reste cependant la plus utilisée dans le domaine des simulations numériques.

L'expression discrète de la dérivée seconde est à peine plus délicate à établir. On choisit généralement la forme suivante :

$$\frac{f(x_{j+1}) - f(x_{j-1}) - 2f(x_j)}{4d^2}. \quad (\text{A.15})$$

Ces expressions de la dérivée et de la dérivée seconde, nous allons simplement les réintroduire dans les définitions des opérateurs vectoriels afin d'avoir leur expression dans le cadre d'une grille cubique régulière. Ainsi, les différents opérateurs précités prennent les formes suivante :

$$\mathbf{grad}_{i,j,k}(f) = \frac{1}{2d} \begin{bmatrix} f_{i+1,j,k} - f_{i-1,j,k} \\ f_{i,j+1,k} - f_{i,j-1,k} \\ f_{i,j,k+1} - f_{i,j,k-1} \end{bmatrix} \quad (\text{A.16})$$

$$\mathit{div}_{i,j,k}(\mathbf{A}) = \frac{1}{2d} ((\mathbf{A}^x_{i+1,j,k} - \mathbf{A}^x_{i-1,j,k}) + (\mathbf{A}^y_{i,j+1,k} - \mathbf{A}^y_{i,j-1,k}) + (\mathbf{A}^z_{i,j,k+1} - \mathbf{A}^z_{i,j,k-1})) \quad (\text{A.17})$$

$$\mathbf{rot}_{i,j,k}(\mathbf{A}) = \frac{1}{2d} \begin{bmatrix} (\mathbf{A}^z_{i,j+1,k} - \mathbf{A}^z_{i,j-1,k}) - (\mathbf{A}^y_{i,j,k+1} - \mathbf{A}^y_{i,j,k-1}) \\ (\mathbf{A}^x_{i,j,k+1} - \mathbf{A}^x_{i,j,k-1}) - (\mathbf{A}^z_{i+1,j,k} - \mathbf{A}^z_{i-1,j,k}) \\ (\mathbf{A}^y_{i+1,j,k} - \mathbf{A}^y_{i-1,j,k}) - (\mathbf{A}^x_{i,j+1,k} - \mathbf{A}^x_{i,j-1,k}) \end{bmatrix} \quad (\text{A.18})$$

$$\Delta_{i,j,k}(f) = \frac{1}{4d^2} (f_{i+1,j,k} + f_{i-1,j,k} + f_{i,j+1,k} + f_{i,j-1,k} + f_{i,j,k+1} + f_{i,j,k-1} - 6f_{i,j,k}). \quad (\text{A.19})$$

Annexe B

Pseudo-distance

Dans cette annexe, nous allons revenir sur la pseudo-distance à un ellipsoïde que nous avons brièvement introduite dans le chapitre 6. Nous nous efforcerons en particulier de montrer que, pour une excentricité bornée des ellipsoïdes, cette distance d' est équivalente à la distance euclidienne d , ce qui devrait nous assurer de bonnes conditions de convergence des algorithmes¹.

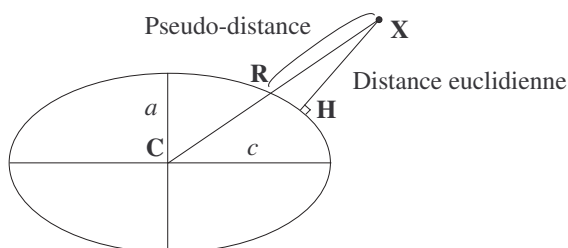


FIG. B.1 – La pseudo-distance d' que nous allons utiliser correspond à la longueur du segment \mathbf{XR} porté par la droite reliant le point \mathbf{X} au centre de l'ellipsoïde.

Par définition de la distance euclidienne, il est évident que $d' \geq d$. Mais on peut remarquer également que la distance euclidienne d est nécessairement plus grande que la distance entre X et le plan tangent à l'ellipsoïde en R , qui vaut $\frac{d'}{\sin(\phi)}$ (cf figure). Nous avons donc l'encadrement :

$$d \leq d' \leq \frac{d}{\sin(\phi)}.$$

Comme nous travaillons avec une symétrie de révolution, nous pouvons aisément replacer le problème dans le plan. On peut définir deux vecteurs directeurs décrivant l'angle ϕ , \mathbf{u} et \mathbf{v} , l'un correspondant à la direction \mathbf{RX} , l'autre symbolisant le plan tangent à l'ellipsoïde

¹Rappelons que deux distances d et d' sont équivalentes si et seulement si on peut trouver deux coefficients non nuls k_1 et k_2 vérifiant $k_1 d \leq d' \leq k_2 d$.

en \mathbf{R} . Cela permet d'en déduire ϕ :

$$\cos(\phi) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|} \quad \text{où } \mathbf{u} \begin{cases} c \cos(\theta) \\ a \sin(\theta) \end{cases} \quad \text{et } \mathbf{v} \begin{cases} c \sin(\theta) \\ -a \cos(\theta) \end{cases} .$$

On a noté respectivement c et a le demi-grand-axe et le demi-petit-axe de notre ellipsoïde. A partir de là, nous pouvons en déduire :

$$\begin{aligned} \cos(\phi) &= \frac{(c^2 - a^2)\cos(\theta)\sin(\theta)}{\sqrt{c^2\cos^2(\theta) + a^2\sin^2(\theta)}\sqrt{c^2\sin^2(\theta) + a^2\cos^2(\theta)}} \\ &= \frac{(c^2 - a^2)\frac{\sin(2\theta)}{2}}{\sqrt{a^2 + (c^2 - a^2)\cos^2(\theta)}\sqrt{a^2 + (c^2 - a^2)\sin^2(\theta)}} \\ &= \frac{(c^2 - a^2)\frac{\sin(2\theta)}{2}}{\sqrt{c^2a^2 + (c^2 - a^2)^2\left(\frac{\sin(2\theta)}{2}\right)^2}} \\ \cos^2(\phi) &= \frac{(c^2 - a^2)^2\left(\frac{\sin(2\theta)}{2}\right)^2}{c^2a^2 + (c^2 - a^2)^2\left(\frac{\sin(2\theta)}{2}\right)^2} \\ \sin^2(\phi) &= \frac{c^2a^2}{c^2a^2 + (c^2 - a^2)^2\left(\frac{\sin(2\theta)}{2}\right)^2} \\ 1/\sin(\phi) &= \sqrt{1 + (c/a - a/c)^2\left(\frac{\sin(2\theta)}{2}\right)^2} \\ &\leq \sqrt{1 + \frac{(c/a - a/c)^2}{4}} \\ 1/\sin(\phi) &\leq \frac{c/a + a/c}{2}. \end{aligned}$$

Et en définitive :

$$d \leq d' \leq d \frac{c/a + a/c}{2}.$$

Bibliographie

- [AAS01] V. ATHITSOS, J. ALON et S. SCLAROFF : 3d hand pose estimation by finding appearance-based matches in a large database of training views. *In IEEE Workshop on Cues in Communication*, pages 100–106, décembre 2001.
- [AASK04] V. ATHITSOS, J. ALON, S. SCLAROFF et G. KOLLIOS : Boostmap : A method for efficient approximate similarity rankings. *In Computer Vision and Pattern Recognition*, 2004.
- [ACWK04] Alexis ANGELIDIS, Marie-Paule CANI, Geoff WYVILL et Scott KING : Swirling-sweepers : Constant volume modeling. *In Pacific Graphics'2004*, Korea, octobre 2004.
- [AGB04] A. ALEXE, V. GAILDRAT et L. BARTHE : Interactive modelling from sketches using spherical implicit functions. *In AFRIGRAPH '04 : Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 25–34, New York, NY, USA, 2004. ACM Press.
- [AHB87] K.S. ARUN, T.S. HUANG et S.D. BLOSTEIN : Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:698–700, septembre 1987.
- [AMH02] T. AKENINE-MÖLLER et E. HAINES, éditeurs. *Real-Time Rendering, 2nd edition*. A.K. Peters Ltd., 2002.
- [AS96] Ricardo S. AVILA et Lisa M. SOBIERAJSKI : A haptic interaction method for volume visualization. *In VIS '96 : Proceedings of the 7th conference on Visualization '96*, pages 197–ff., Los Alamitos, CA, USA, 1996. IEEE Computer Society Press.
- [AS03] Vassilis ATHITSOS et Stan SCLAROFF : Estimating 3D hand pose from a cluttered image. *In CVPR '03 : Proceedings of Conference in Computer Vision and Pattern Recognition*, juin 2003.
- [ATTY99] H. ARATA, Y. TAKAI, N. K. TAKAI et T. YAMAMOTO : Free-form shape modeling by 3d cellular automata. *In International Conference on Shape Modeling and Applications*, pages 242–247, 1999.
- [Bær98] J. A. BÆRENTZEN : Octree-based volume sculpting. *In Proceedings of IEEE Visualization*, pages 9–12, 1998.
- [Bar84] Alan H. BARR : Global and local deformations of solid primitives. *In SIGGRAPH '84 : Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 21–30, New York, NY, USA, 1984. ACM Press.

- [BF95] Jules BLOOMENTHAL et Keith FERGUSON : Polygonization of non-manifold implicit surfaces. In *SIGGRAPH '95 : Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 309–316, New York, NY, USA, 1995. ACM Press.
- [BFK84] Wolfgang BÖHM, Gerald FARIN et Jürgen KAHMANN : A survey of curve and surface methods in cagd. *Computer Aided Geometric Design*, 1(1):1–60, 1984.
- [BL95] J. BILL et S. LODHA : Sculpting polygonal models using virtual tools. In *Proceedings of Graphics Interface'95*, pages 272–279, 1995.
- [BL99] Jules BLOOMENTHAL et Chek LIM : Skeletal methods of shape manipulation. In *SMI '99 : Proceedings of the International Conference on Shape Modeling and Applications*, page 44, Washington, DC, USA, 1999. IEEE Computer Society.
- [Bli82] James F. BLINN : A generalization of algebraic surface drawing. *ACM Trans. Graph.*, 1(3):235–256, 1982.
- [BLL02] L. BRETZNER, I. LAPTEX et T. LINDBERG : Hand gesture recognition using multiscale colour features, hierarchical models and particle filtering. In *FG '02 : Proceedings of the International Conference on Automatic Face and Gesture Recognition*, pages 423–428, Washington, DC, USA, 2002. IEEE Computer Society.
- [Blo88] J. BLOOMENTHAL : Polygonization of implicit surfaces. *Computer Aided Geometric Design*, 5(4):341–355, 1988.
- [Blo02] Jules BLOOMENTHAL : Medial-based vertex deformation. In *SCA '02 : Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 147–151, New York, NY, USA, 2002. ACM Press.
- [BM92] Paul J. BESL et Neil D. MCKAY : A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, février 1992.
- [BM98] C. BREGLER et J. MALIK : Tracking people with twists and exponential maps. In *CVPR '98 : Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 8, Washington, DC, USA, 1998. IEEE Computer Society.
- [BMP03] C. BREGLER, J. MALIK et K. PULLEN : Twist based acquisition and tracking of animal and human kinematics. *IJCV*, 2003.
- [BR94] Paul BORREL et Ari RAPPOPORT : Simple constrained deformations for geometric modeling and interactive design. *ACM Trans. Graph.*, 13(2):137–155, 1994.
- [BS91] Jules BLOOMENTHAL et Ken SHOEMAKE : Convolution surfaces. In *SIGGRAPH '91 : Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 251–256, New York, NY, USA, 1991. ACM Press.
- [BW90] Jules BLOOMENTHAL et Brian WYVILL : Interactive techniques for implicit modeling. In *Proceedings of the 1990 Symposium on Interactive 3D graphics*, pages 109–116, New York, NY, USA, 1990. ACM Press.
- [BW98] David BARAFF et Andrew WITKIN : Large steps in cloth simulation. *Computer Graphics*, 32(Annual Conference Series):43–54, 1998.
- [Can93] Marie-Paule CANI : An implicit formulation for precise contact modeling between flexible solids. In *Computer Graphics*, pages 313–320. ACM, août 1993.

- [CG91] G. CELNIKER et D. GOSSARD : Deformable curve and surface finite-element for free-form shape design. *In SIGGRAPH '91 : Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 257–266, New York, NY, USA, 1991. ACM Press.
- [CHP89] J. E. CHADWICK, D. R. HAUMANN et R. E. PARENT : Layered construction for deformable animated characters. *In SIGGRAPH '89 : Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 243–252, New York, NY, USA, 1989. ACM Press.
- [CJRF97] S. CHEN, D. JOHNSON, P RAAD et D FADDA : The surface marker and microcell method. *International Journal for Numerical Methods in Fluids*, 25:749–778, 1997.
- [CKBH00] K. CHEUNG, T. KANADE, J. BOUGUET et M. HOLLER : A real time system for robust 3d voxel reconstruction of human motions. *In CVPR*, volume 2, pages 714–720, June 2000.
- [CMBT02] Mark CARLSON, Peter J. MUCHA, R. BROOK VAN HORN III et Greg TURK : Melting and flowing. *In SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2002. ACM Press.
- [CMT04] Mark CARLSON, Peter J. MUCHA et Greg TURK : Rigid fluids : Animating the interplay between rigid bodies and fluid. *In SIGGRAPH '04 : Proceedings of the 31th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2004. ACM Press.
- [Coq90] Sabine COQUILLART : Extended free-form deformation : a sculpturing tool for 3d geometric modeling. *In SIGGRAPH '90 : Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 187–196, New York, NY, USA, 1990. ACM Press.
- [CR03] H. CHUI et A. RANGARAJAN : A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, février 2003.
- [Cre99] B. CRESPIAN : Implicit free-form deformations. *In IS'99 : Proceedings of Implicit Surfaces 99*, pages 17–23, 1999.
- [CW92] G. CELNIKER et W. WELCH : Linear constraints for deformable b-spline surfaces. *In Proceedings of the 1992 Symposium on Interactive 3D graphics*, pages 165–170, New York, NY, USA, 1992. ACM Press.
- [CW96] Y. CUI et J. J. WENG : Hand segmentation using learning-based prediction and verification for hand sign recognition. *In Proc. of Conf. Computer Vision and Pattern Recognition*, pages 88–93, juin 1996.
- [CW00] Y. CUI et J. J. WENG : Appearance-based hand sign recognition from intensity image sequences. *Computer Vision and Image Understanding*, 78:157–176, 2000.
- [DBR00] J. DEUTSCHER, A. BLAKE et I.D. REID : Articulated body motion capture by annealed particle filtering. *In CVPR*, pages II : 126–133, 2000.
- [DC95] Matthieu DESBRUN et Marie-Paule CANI : Animating soft substances with implicit surfaces. *In SIGGRAPH '95 : Proceedings of the 22nd annual conference*

- on Computer graphics and interactive techniques*, New York, NY, USA, 1995. ACM Press.
- [DC99] Mathieu DESBRUN et Marie-Paule CANI : Space-time adaptive simulation of highly deformable substances. Rapport technique 3829, INRIA, décembre 1999.
- [DC01] T. DRUMMOND et R. CIPOLLA : Real-time tracking of highly articulated structures in the presence of noisy measurements. *In ICCV*, pages II : 315–320, 2001.
- [DC03] Guillaume DEWAELE et Marie-Paule CANI : Interactive global and local deformations for virtual clay. *In PG '03 : Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, page 131, Washington, DC, USA, 2003. IEEE Computer Society.
- [DC04a] Guillaume DEWAELE et Marie-Paule CANI : Interactive global and local deformations for virtual clay. *Graphical Models (GMOD)*, 66:352–369, septembre 2004.
- [DC04b] Guillaume DEWAELE et Marie-Paule CANI : Virtual clay for direct hand manipulation. *In Eurographics (short papers)*, septembre 2004.
- [DCH] Guillaume DEWAELE, Marie-Paule CANI et Radu HORAUD : Argile virtuelle temps-reel : le cas bidimensionnel. *In Neuvieme reunion du groupe de travail*.
- [DD02] D. DEMIRDJIAN et T. DARRELL : 3-d articulated pose tracking for untethered diectic reference. *In ICMI*, page 267. IEEE Computer Society, 2002.
- [DDCB01] Gilles DEBUNNE, Mathieu DESBRUN, Marie-Paule CANI et Alan H. BARR : Dynamic real-time deformations using space & time adaptive sampling. *In SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 31–36, New York, NY, USA, 2001. ACM Press.
- [DDH04] Guillaume DEWAELE, Frédéric DEVERNAY et Radu HORAUD : Hand motion from 3d point trajectories and a smooth surface model. *In T. PAJDLA et J. MATAS, éditeurs : Proceedings of the 8th European Conference on Computer Vision*, volume I de *LNCS 3021*, pages 495–507. Springer, mai 2004.
- [Deb00] Gilles DEBUNNE : *Animation multirésolution d'objets déformables en temps-réel, Application à la simulation chirurgicale*. Thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble, France, décembre 2000.
- [Del03] Quentin DELAMARRE : *Suivi du mouvement d'objets articulés dans des séquences d'images vidéo*. Thèse de doctorat, Université de Nice-Sophia Antipolis, décembre 2003.
- [Des97] Matthieu DESBRUN : *Modélisation et Animation de Matériaux Hautement Déformables en Synthèse d'Images*. Thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble, France, décembre 1997.
- [Dev97] Frédéric DEVERNAY : *Vision stéréoscopique et propriétés différentielles des surfaces*. Thèse de doctorat, École Polytechnique, Palaiseau, France, février 1997.
- [DF98] Quentin DELAMARRE et Olivier FAUGERAS : Finding pose of hand in video images : a stereo-based approach. *In Third International Conference on Automatic Face and Gesture Recognition*, pages 585–590, avril 1998.
- [DF99] Quentin DELAMARRE et Olivier FAUGERAS : 3d articulated models and multi-view tracking with silhouettes. *In ICCV '99 : Proceedings of the International*

- Conference on Computer Vision-Volume 2*, page 716, Washington, DC, USA, 1999. IEEE Computer Society.
- [DGLR89] B. DIU, C. GUTHMAN, D. LEDERER et B. ROULET : *Physique Statistique*. Hermann, Paris, France, 1989.
- [DKT98] Tony DEROSE, Michael KASS et Tien TRUONG : Subdivision surfaces in character animation. In *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 85–94, New York, NY, USA, 1998. ACM Press.
- [DLR77] A. P. DEMPSTER, N. M. LAIRD et D. B. RUBIN : Maximum likelihood estimation from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B*, 39:1–38, 1977.
- [Dru04] Sebastien DRUON : *Modélisation et recalage d'objets et d'environnements en déformation viscoplastique : Aide au geste pour les métiers matière*. Thèse de doctorat, Université de Montpellier II, Montpellier, France, 2004.
- [DTC95] Mathieu DESBRUN, Nicolas TSINGOS et Marie-Paule CANI : Adaptive sampling of implicit surfaces for interactive modeling and animation. In *IS'95 : Proceedings of Implicit Surfaces 95*, avril 1995.
- [DUS01] K. DORFMÜLLER-ULHAAS et D. SCHMALSTIEG : Finger tracking for interaction in augmented environments. In *ISAR '01 : Proceedings of the IEEE and ACM International Symposium on Augmented Reality*, page 55, Washington, DC, USA, 2001. IEEE Computer Society.
- [EMF02] Douglas ENRIGHT, Steve MARSHNER et Ronald FEDKIW : Animating and rendering of complex water surfaces. In *SIGGRAPH '02 : Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2002. ACM Press.
- [FB88] David R. FORSEY et Richard H. BARTELS : Hierarchical b-spline refinement. In *SIGGRAPH '88 : Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 205–212, New York, NY, USA, 1988. ACM Press.
- [FB02] Ronan FABLET et Michael J. BLACK : Automatic detection and tracking of human motion with a view-based representation. In *ECCV (1)*, pages 476–491, 2002.
- [FCG99] Eric FERLEY, Marie-Paule CANI et Jean-Dominique GASCUEL : Virtual sculpture. In *Short Papers Proceedings of Eurographics '99*, septembre 1999. Short paper.
- [FCG01] Eric FERLEY, Marie-Paule CANI et Jean-Dominique GASCUEL : Resolution adaptive volume sculpting. *Graphical Models*, 63(6):459–478, 2001.
- [Fer02] Eric FERLEY : *Sculpture Virtuelle*. Thèse de doctorat, Institut National Polytechnique de Grenoble, septembre 2002.
- [FF01] Nick FOSTER et Ronald FEDKIW : Practical animation of liquids. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 23–30, New York, NY, USA, 2001. ACM Press.

- [FH86] O. D. FAUGERAS et M. HÉBERT : The representation, recognition, and locating of 3d shapes from range data. *The International Journal of Robotics Research*, 5:27–52, 1986.
- [FM96] Nick FOSTER et Dimitri METAXAS : Realistic animation of liquids. *Graphical Models and Image Processing*, 58(5), septembre 1996.
- [Fow92] Barry FOWLER : Geometric manipulation of tensor product surfaces. In *SI3D'92 : Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 101–108, New York, NY, USA, 1992. ACM Press.
- [GBO04] Tolga G. GOKTEKIN, Adam W. BARGTEIL et James F. O'BRIEN : A method for animating viscoelastic fluids. *ACM Trans. Graph.*, 23(3):463–468, 2004.
- [GD96] D. M. GAVRILA et L. S. DAVIS : 3-d model-based tracking of humans in action : a multi-view approach. In *CVPR '96 : Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, page 73, Washington, DC, USA, 1996. IEEE Computer Society.
- [GH91] Tinsley A. GALYEAN et John F. HUGHES : Sculpting : an interactive volumetric modeling technique. *Computer Graphics*, 25(4):267–274, 1991.
- [GMW81] Philip E. GILL, Walter MURRAY et Margaret H. WRIGHT : *Practical Optimization*. Academic Press, New York, NY, 1981.
- [GOP99] Carlos GONZALEZ-OCHOA et Jorg PETERS : Localized-hierarchy surface splines (less). In *SI3D '99 : Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 7–15, New York, NY, USA, 1999. ACM Press.
- [Gra03] Sébastien GRANGER : *Une approche statistique multi-échelle au recalage rigide de surfaces : Application à l'implantologie dentaire*. Thèse de doctorat, Ecole des Mines de Paris, April 2003.
- [Gui02] Claire GUILBAUD : *Modélisation et visualisation de phénomènes naturels simulés pas système physique particulière*. Thèse de doctorat, Institut National Polytechnique de Grenoble, Grenoble, France, septembre 2002.
- [Har58] H. HARTLEY : Maximum likelihood estimation from incomplete data. *Biometrics*, 14:174–194, 1958.
- [HH96] T. HEAP et D. HOGG : Towards 3d hand tracking using a deformable model. In *FG '96 : Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)*, page 140, Washington, DC, USA, 1996. IEEE Computer Society.
- [HHD00] I. HARITAOGLU, D. HARWOOD et L.S. DAVIS : W4 : Real-time surveillance of people and their activities. *PAMI*, 22(8):809–830, August 2000.
- [HHK92] William M. HSU, John F. HUGHES et Henry KAUFMAN : Direct manipulation of free-form deformations. In *SIGGRAPH '92 : Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 177–184, New York, NY, USA, 1992. ACM Press.
- [HKD93] Mark HALSTEAD, Michael KASS et Tony DEROSE : Efficient, fair interpolation using catmull-clark surfaces. In *SIGGRAPH '93 : Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 35–44, New York, NY, USA, 1993. ACM Press.

- [HML99] Gentaro HIROTA, Renee MAHESHWARI et Ming C. LIN : Fast volume-preserving free form deformation using multi-level optimization. *In SMA '99 : Proceedings of the fifth ACM symposium on Solid modeling and applications*, pages 234–245, New York, NY, USA, 1999. ACM Press.
- [Hog83] D. HOGG : Model-based vision : A program to see a walking person. *IVC*, 1(1):5–20, février 1983.
- [HOR99] E. HOLDEN, R. OWENS et G. ROY : 3d hand tracker for visual sign recognition. Rapport technique, Department of Computer Science, 1999.
- [HW65] F. HARLOW et J. WELCH : Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids*, 8:2182–2189, 1965.
- [HW88] Berthold K. P. HORN et E. J. WELDON : Direct methods for recovering motion. *International Journal on Computer Vision*, 2:51–76, 1988.
- [IB98] M. ISARD et A. BLAKE : Icondensation : Unifying low-level and high-level tracking in a stochastic framework. *In ECCV '98 : Proceedings of the 5th European Conference on Computer Vision*, pages 893–908, 1998.
- [IH01] Takeo IGARASHI et John F. HUGHES : A suggestive interface for 3d drawing. *In UIST '01 : Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 173–181, New York, NY, USA, 2001. ACM Press.
- [IMH05] Takeo IGARASHI, Tomer MOSCOVITCH et John F. HUGHES : As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, 24(3), 2005.
- [IMT99] Takeo IGARASHI, Satoshi MATSUOKA et Hidehiko TANAKA : Teddy : A sketching interface for 3d freeform design. *In SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [Jou96] Ammar JOUKHADAR : Adaptative time step for fast converging dynamic simulation system. *In IEEE-RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 418–424, New York, NY, USA, novembre 1996. ACM Press.
- [KB89] D. KALRA et A. H. BARR : Guaranteed ray intersections with implicit surfaces. *In SIGGRAPH '89 : Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 297–306, New York, NY, USA, 1989. ACM Press.
- [KCVS98] Leif KOBBELT, Swen CAMPAGNA, Jens VORSATZ et Hans-Peter SEIDEL : Interactive multi-resolution modeling on arbitrary meshes. *In SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 105–114, New York, NY, USA, 1998. ACM Press.
- [KJFH02] Olga KARPENKO et Ramesh Raskar JOHN F. HUGHES : Free-form sketching with variational implicit surfaces. *In Proceedings of Eurographics '02*, 2002.
- [KM96] I. A. KAKADIARIS et D. METAXAS : Model-based estimation of 3D human motion with occlusion based on active multi-viewpoint selection. *In Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 81–87, Los Alamitos, California, U.S.A., 18–20 1996. IEEE Computer Society.

- [KS99] Andrei KHODAKOVSKY et Peter SCHRÖDER : Fine level feature editing for subdivision surfaces. *In SMA '99 : Proceedings of the fifth ACM symposium on Solid modeling and applications*, pages 203–211, New York, NY, USA, 1999. ACM Press.
- [LC86] A. LUCIANI et C. CADOZ : Application of mechanical and geometrical modeling to generate sequences of motion in animated pictures. *In CESTA*, 1986.
- [LC87] William E. LORENSEN et Harvey E. CLINE : Marching cubes : A high resolution 3d surface construction algorithm. *In SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, 1987. ACM Press.
- [LCJ94] F. LAZARUS, S. COQUILLART et P. JANCENE : Axial deformations : an intuitive deformation technique. *Computer Aided Geometric Design*, 26(8), 1994.
- [LCN00] J. LEWIS, M. CORDNER et N.FONG : Unified approach to shape interpolation and skeleton-driven deformation. *In SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2000. ACM Press.
- [LF02] R. LOCKTON et A. W. FITZGIBBON : Real-time gesture recognition using deterministic boosting. *In Proc. of the British Machine Vision Conference*, volume 2, pages 817–826, septembre 2002.
- [LF03] Jason LAWRENCE et Thomas FUNKHOUSER : A painting interface for interactive surface deformations. *In PG '03 : Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, page 141, Washington, DC, USA, 2003. IEEE Computer Society.
- [LH00] Fabienne LATHUILLIÈRE et Jean-Yves HERVÉ : Visual hand posture tracking in a gripper guiding application. *In ICRA*, pages 1688–1694, 2000.
- [LL86] L. LANDAU et E. LIFCHITZ : *Mécanique des fluides*. Mir Moscou, second edition, 1986.
- [LMSO03] S. LU, D. METAXAS, D. SAMARAS et J. OLIENSIS : Using multiple cues for hand tracking and model refinement. *In CVPR '03 : Proceedings of Conference in Computer Vision and Pattern Recognition*, volume 2, pages 443–450, juin 2003.
- [Mah97] Diana Phillips MAHONEY : Visualizing volumes. *Computer Graphics World*, 20(7):42–ff., juillet 1997.
- [MCG03] Matthias MÜLLER, David CHARYPAR et Markus GROSS : Particle-based fluid simulation for interactive applications. *In SCA '03 : Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 154–159, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [Men00] A. MENACHE : *Understanding Motion Capture for Computer Animation and Video Games*. Morgan Kaufmann Publishers, 2000.
- [MG01] T.B. MOESLUND et E. GRANUM : A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.
- [MI00] J. MACCORMICK et M. ISARD : Partitionned sampling, articulated objects, and interface-quality hand tracking. *In ECCV '00 : Proceedings of the 6th European Conference on Computer Vision*, volume 2, pages 3–19, juin 2000.

- [Mil88] Gavin S. P. MILLER : The motion dynamics of snakes and worms. *In SIGGRAPH '88 : Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 169–173, New York, NY, USA, 1988. ACM Press.
- [MJ96] Ron MACCRACKEN et Kenneth I. JOY : Free-form deformations with lattices of arbitrary topology. *In SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 181–188, New York, NY, USA, 1996. ACM Press.
- [MK93] L. MOCCOZET et P. KALRA : Interactive and controlled synthesis of 3d irregular shapes. *Virtual World and Multimedia*, pages 179–198, 1993.
- [MM02] Greg MORI et Jitendra MALIK : Estimating human body configurations using shape context matching. *In ECCV '02 : Proceedings of the 7th European Conference on Computer Vision-Part III*, pages 666–680, London, UK, 2002. Springer-Verlag.
- [MN78] D. MARR et H. K. NISHIHARA : Representation and recognition of the spatial organization of three dimensional structure. *Journal of the Royal Statistical Society*, 200:269–294, 1978.
- [MQ00] Kevin T. MCDONNELL et Hong QIN : Dynamic sculpting and animation of free-form subdivision solids. *In CA '00 : Proceedings of the Computer Animation*, page 126, Washington, DC, USA, 2000. IEEE Computer Society.
- [MQ02] Kevin T. MCDONNELL et Hong QIN : Dynamic sculpting and animation of free-form subdivision solids. *The Visual Computer*, 18(2):81–96, 2002.
- [MQW01] Kevin T. MCDONNELL, Hong QIN et Robert A. WLODARCZYK : Virtual clay : a real-time sculpting system with haptic toolkits. *In SI3D '01 : Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 179–190, New York, NY, USA, 2001. ACM Press.
- [MS92] Henry P. MORETON et Carlo H. SÉQUIN : Functional optimization for fair surface design. *In SIGGRAPH '92 : Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 167–176, New York, NY, USA, 1992. ACM Press.
- [MT93] D.N. METAXAS et D. TERZOPOULOS : Shape and nonrigid motion estimation through physics-based synthesis. *PAMI*, 15(6):580–591, June 1993.
- [MT97] Laurent MOCCOZET et Nadia Magnenat THALMANN : Dirichlet free-form deformations and their application to hand simulation. *In CA '97 : Proceedings of the Computer Animation*, page 93, Washington, DC, USA, 1997. IEEE Computer Society.
- [MTHC01] I. MIKIC, M. TRIVEDI, E. HUNTER et P. COSMAN : Articulated body posture estimation from multi-camera voxel data. *In IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, December 2001*, 2001.
- [MTHC03] Ivana MIKIC, M. TRIVEDI, E. HUNTER et P. COSMAN : Human body model acquisition and tracking using voxel data. *IJCV*, 53(3):199–223, July 2003.
- [Nay90] B. NAYLOR : Sculpt : an interactive solid modeling tool. *In Proceedings on Graphics interface '90*, pages 138–148, Toronto, Ont., Canada, Canada, 1990. Canadian Information Processing Society.

- [NB93] Paul NING et Jules BLOOMENTHAL : An evaluation of implicit surface tilers. *IEEE Comput. Graph. Appl.*, 13(6):33–41, 1993.
- [NBH05] Matti NISKANEN, Edmond BOYER et Radu HORAUD : Articulated motion capture from 3-d points and normals. In Torr CLOCKSIN, Fitzgibbon, éditeur : *British Machine Vision Conference*, volume 1, pages 439–448, Oxford, UK, September 2005. BMVA, British Machine Vision Association.
- [NSACO05] Andrew NEALEN, Olga SORKINE, Marc ALEXA et Daniel COHEN-OR : A sketch-based interface for detail-preserving mesh editing. In *SIGGRAPH '05 : Proceedings of the 32th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2005. ACM Press.
- [NSMO96] K. NIREI, H. SAITO, M. MOCHIMARU et S. OZAWA : Human hand tracking from binocular image sequences. In *22th Int'l Conf. on Industrial Electronics, Control, and Instrumentation*, pages 297–302, août 1996.
- [OB80] Joseph O'ROURKE et Norman I. BADLER : Model-based image analysis of human motion using constraint propagation. *IEEE Trans. Pattern Analysis and Machine Intell.*, 2(6):522–536, novembre 1980.
- [OC97] Agata OPALACH et Marie-Paule CANI : Local deformation for animation of implicit surfaces. In *SCCG'97*, juin 1997.
- [OH98] H. OUHADDI et P. HORAIN : Conception et ajustement d'un modèle 3d articulé de la main. In *Actes des 6èmes journées du Groupe de Travail Réalité Virtuelle*, volume 12/13, pages 83–90, 1998.
- [ON03] Koichi ONOUE et Tomoyuki NISHITA : Virtual sandbox. In *PG '03 : Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, page 252, Washington, DC, USA, 2003. IEEE Computer Society.
- [Par77] Richard E. PARENT : A system for sculpting 3-d data. In *SIGGRAPH '77 : Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pages 138–147, New York, NY, USA, 1977. ACM Press.
- [PASS95] A. PASKO, V. ADSHIEV, A. SOURIN et V. SAVCHENKO : Function representation in geometric modeling : concepts, implementation and applications. *Visual Computer*, pages 429–446, 1995.
- [PB81] S. PLATT et N. BADLER : Animating facial expressions. In *SIGGRAPH '81 : Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, volume 15, pages 245–252, New York, NY, USA, 1981. ACM Press.
- [PB88] John C. PLATT et Alan H. BARR : Constraints methods for flexible models. In *SIGGRAPH '88 : Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 279–288, New York, NY, USA, 1988. ACM Press.
- [PF01a] Ronald N. PERRY et Sarah F. FRISKEN : Kizamu : a system for sculpting digital characters. In *SIGGRAPH '01 : Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 47–56, New York, NY, USA, 2001. ACM Press.
- [PF01b] Ralf PLÄNKERS et Pascal FUA : Articulated soft objects for video-based body modeling. In *ICCV '01 : Proceedings of International Conference on Computer Vision*, Vancouver, Canada, 2001. IEEE Computer Society Press.

- [PF03] Ralf PLÄNKERS et Pascal FUA : Articulated soft objects for multi-view shape and motion capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10), 2003.
- [PFTV93] William H. PRESS, Brian P. FLANNERY, Saul A. TEUKOLSKY et William T. VETTERLING : *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, 1993.
- [PRC00] Vladimir PAVLOVIC, James M. REHG et Tat-Jen CHAM : A dynamic bayesian network approach to tracking using learned switching dynamic models. In *HSCC '00 : Proceedings of the Third International Workshop on Hybrid Systems : Computation and Control*, pages 366–380, London, UK, 2000. Springer-Verlag.
- [PSH97] V. PAVLOVIĆ, R. SHARMA et T. HUANG : Visual interpretation of hand gestures for human-computer interaction : A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:677–695, juillet 1997.
- [PSS01] A. PASKO, V. SAVCHENKO et A. SOURIN : Synthetic carving using implicit surface primitives. *Computer Aided Design*, 33(5):379–388, 2001.
- [PTB⁺03] Simon PREMOZE, Tolga TASDIZEN, James BIGLER, Aaron LEFOHN et Ross T. WHITAKER : Particle-based simulation of fluids. In *Proceedings of Eurographics 2003*, pages 401–410, 2003.
- [PW89] A. PENTLAND et J. WILLIAMS : Good vibrations : model dynamics for graphics and animation. In *SIGGRAPH '89 : Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 215–222, New York, NY, USA, 1989. ACM Press.
- [RASS01] R. ROSALES, V. ATHITSOS, L. SIGAL et S. SCARLOFF : 3d hand pose reconstruction using specialized mappings. In *ICCV '01 : Proceedings of 8th International Conference on Computer Vision*, pages 378–385, Vancouver, Canada, 2001.
- [RCB97] Anand RANGARAJAN, Haili CHUI et Fred L. BOOKSTEIN : The softassign procrustes matching algorithm. In *IPMI*, pages 29–42, 1997.
- [RE99] Alon RAVIV et Gershon ELBER : Three dimensional freeform sculpting via zero sets of scalar trivariate functions. In *SMA '99 : Proceedings of the fifth ACM symposium on Solid modeling and applications*, pages 246–257, New York, NY, USA, 1999. ACM Press.
- [RK94] J. REHG et T. KANADE : Visual tracking of high dof articulated structures : an application to human hand tracking. In *ECCV '94 : Proceedings of the European Conference on Computer Vision*, pages 35–46, Washington, DC, USA, mai 1994. IEEE Computer Society.
- [RK95] J. REHG et T. KANADE : Model-based tracking of self-occluding articulated objects. In *ICCV '95 : Proceedings of the International Conference on Computer Vision*, pages 612–617, Washington, DC, USA, 1995. IEEE Computer Society.
- [SBS02] Hedvig SIDENBLADH, Michael J. BLACK et Leonid SIGAL : Implicit probabilistic models of human motion for synthesis and tracking. In *ECCV '02 : Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 784–800, London, UK, 2002. Springer-Verlag.
- [SF98] Karan SINGH et Eugene FIUME : Wires : a geometric deformation technique. In *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer*

- graphics and interactive techniques*, pages 405–414, New York, NY, USA, 1998. ACM Press.
- [SH97] Barton T. STANDER et John C. HART : Guaranteeing the topology of an implicit surface polygonization for interactive modeling. In *SIGGRAPH '97 : Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 279–286, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [SKS01] N. SHIMADA, K. KIMURA et Y. SHIRAI : Real-time 3-d hand posture estimation based on 2-d appearance retrieval using monocular camera. In *FG '01 : Proceedings of the International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems*, pages 23–30, juillet 2001.
- [SMC01] B. STENGER, P. R. S. MENDONÇA et R. CIPOLLA : Model based 3D tracking of an articulated hand. In *CVPR '01 : Proceedings of Conference in Computer Vision and Pattern Recognition*, pages 310–315, 2001.
- [Sou01] Alexei SOURIN : Functionally based virtual computer art. In *SI3D '01 : Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 77–84, New York, NY, USA, 2001. ACM Press.
- [SP86] Thomas W. SEDERBERG et Scott R. PARRY : Free-form deformation of solid geometric models. In *SIGGRAPH '86 : Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 151–160, New York, NY, USA, 1986. ACM Press.
- [SS96] N. SHIMADA et Y. SHIRAI : 3d hand pose estimation and shape model refinement from a monocular image sequence. In *Proc. of Int. Conf. on Virtual Systems and Multimedia*, pages 423–428, septembre 1996.
- [SSKK98] H. SUZUKI, Y. SAKURAI, T. KANAI et F. KIMURA : Interactive mesh dragging with adaptive remeshing technique. In *PG '98 : Proceedings of the 6th Pacific Conference on Computer Graphics and Applications*, page 188, Washington, DC, USA, 1998. IEEE Computer Society.
- [ST03] Cristian SMINCHISESCU et Bill TRIGGS : Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, 2003.
- [Sta98] Jos STAM : Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. In *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 395–404, New York, NY, USA, 1998. ACM Press.
- [Sta99] Jos STAM : Stable fluids. In *SIGGRAPH '99 : Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [SVD03] Gregory SHAKHNAROVICH, Paul VIOLA et Trevor DARRELL : Fast pose estimation with parameter-sensitive hashing. In *ICCV '03 : Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 750, Washington, DC, USA, 2003. IEEE Computer Society.
- [SWP98] T. STARNER, J. WEAVER et A. PENTLAND : Real-time american sign language recognition using desk and wearable computer-based video. In *IEEE Trans.*

- Pattern Analysis and Machine Intell.*, volume 20, pages 1371–1375, décembre 1998.
- [TdM01] J. TRIESCH et C. Von der MALSBURG : A system for person-independent hand posture recognition against complex backgrounds. In *IEEE Trans. Pattern Analysis and Machine Intell.*, volume 23, pages 1449–1453, 2001.
- [TF88] Demetri TERZOPOULOS et Kurt FLEISCHER : Modeling inelastic deformation : viscoelasticity, plasticity, fracture. In *SIGGRAPH '88 : Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 269–278, New York, NY, USA, 1988. ACM Press.
- [TMSS02] C. THEOBALT, M. MAGNOR, P. SCHUELER et H. P. SEIDEL : Combining 2d feature tracking and volume reconstruction for online video-based human motion capture. In *Pacific Graphics*, pages 96–103, 2002.
- [Ton98] David TONNESEN : *Dynamically Coupled Particle Systems for Geometric Modeling, Reconstruction, and Animation*. Thèse de doctorat, Department of Computer Science, University of Toronto, 1998.
- [TPBF87] D. TERZOPOULOS, J. PLATT, A. BARR et K. FLEISCHER : Elastically deformable models. *Computer Graphics (Proc. SIGGRAPH'87)*, 21(4):205–214, 1987.
- [TPF91] D. TERZOPOULOS, J. PLATT et K. FLEISCHER : Heating and melting deformable models. *The Journal of Visualization and Computer Animation*, 2(2):68–73, 1991.
- [TPS03] Carlo TOMASI, Slav PETROV et Arvind SASTRY : 3d tracking = classification + interpolation. In *ICCV '03 : Proceedings of 9th International Conference on Computer Vision*, 2003.
- [Ume91] S. UMEYAMA : Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:376–380, 1991.
- [UO99] Akira UTSUMI et Jun OHYA : Multiple-hand-gesture tracking using multiple cameras. In *CVPR '99 : Proceedings of Conference in Computer Vision and Pattern Recognition*, pages 1473–1478, 1999.
- [WADP97] Christopher Richard WREN, Ali AZARBAYEJANI, Trevor DARRELL et Alex PENTLAND : Pfunder : Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.
- [WE98] R. WESTERMANN et T. ERTL : Efficiently using graphics hardware in volume rendering applications. In *SIGGRAPH '98 : Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 169–177, New York, NY, USA, 1998. ACM Press.
- [WH94] A. WITKIN et P. HECKBERT : Using particles to sample and control implicit surfaces. In *SIGGRAPH '94 : Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 269–278, New York, NY, USA, 1994. ACM Press.
- [WH99] Ying WU et Thomas S. HUANG : Capturing articulated human hand motion : A divide-and-conquer approach. In *ICCV '99 : In Proc. of the IEEE International Conference on Computer Vision*, pages 606–611, 1999.

- [WH00] Y. WU et T. S. HUANG : View-independent recognition of hand postures. *In Proc. of Conf. Computer Vision and Pattern Recognition*, volume 2, pages 88–94, juin 2000.
- [WH01] Y. WU et T. S. HUANG : Human hand modeling, analysis and animation in the context of human computer interaction. *IEEE Signal Processing Magazine, Special issue on Immersive Interactive Technology*, 18:51–60, mai 2001.
- [WK95] Sidney W. WANG et Arie E. KAUFMAN : Volume sculpting. *In SI3D '95 : Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 151–ff., New York, NY, USA, 1995. ACM Press.
- [WLH01] Ying WU, John Y. LIN et Thomas S. HUANG : Capturing natural hand articulation. *In ICCV '01 : Proceedings of International Conference on Computer Vision*, pages 426–432, Vancouver, Canada, 2001. IEEE Computer Society Press.
- [WMW86] B. WYVILL, C. MCPHEETERS et G. WYVILL : Data structure for soft objects. *Visual Computer*, 4(2):227–234, août 1986.
- [WSV91] M. W. WALKER, L. SHAO et R. A. VOLZ : Estimating 3-D location parameters using dual number quaternions. *CVGIP : Image Understanding*, 54:358–367, novembre 1991.
- [WW89] B. WYVILL, et G. WYVILL : Field functions for implicit surfaces. *Visual Computer*, 5:75–82, décembre 1989.
- [WW92] William WELCH et Andrew WITKIN : Variational surface modeling. *In SIGGRAPH '92 : Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 157–166, New York, NY, USA, 1992. ACM Press.
- [ZH03] Hanning ZHOU et Thomas S. HUANG : Tracking articulated hand motion with eigen-dynamics analysis. *In ICCV '03 : Proceedings of 9th International Conference on Computer Vision*, pages 1102–1109, Nice, France, 2003.
- [Zha93] Z. ZHANG : Recalage de deux nuages de points 3D. *Traitement du Signal*, 10:263–281, 1993.
- [Zha94] Z. ZHANG : Iterative point matching for registration of free-form curves and surfaces. *International Journal on Computer Vision*, 13:119–152, 1994.
- [ZHH96] Robert C. ZELEZNIK, Kenneth P. HERNDON et John F. HUGHES : Sketch : an interface for sketching 3d scenes. *In SIGGRAPH '96 : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 163–170, New York, NY, USA, 1996. ACM Press.
- [ZSD+00] D. ZORIN, P. SCHRÖDER, T. DEROSE, L. KOBELT, A. LEVIN et W. SWELDENS : Subdivision for modeling and animation, siggraph'00 course notes 23. *Computer Graphics*, juillet 2000.

Table des figures

1.1	Les bases de la CSG : union, différence et intersection.	8
1.2	Exemples de fonctions d'influence.	11
1.3	Exemples de déformation de maillages issus de [KCVS98].	11
1.4	Peinture d'un modèle pour en déformer le maillage [LF03].	12
1.5	Déformations d'un bras au moyen d'un squelette [Blo02].	13
1.6	Fonctions de pondération pour contrôler les déformations d'un squelette.	14
1.7	Axe médian d'un volume.	15
1.8	Construction d'une B-spline d'ordre 3.	16
1.9	Déformation d'une surface B-Spline par un polygone tangent selon [Fow92].	17
1.10	Changements de topologie dans [GOP99].	18
1.11	Un exemple de surface de subdivision [DKT98].	20
1.12	Obtention de plis avec des surfaces de subdivision.	20
1.13	Exemples de déformations issues de [MQ02].	23
1.14	Fusion de volumes élémentaires pour la construction d'un volume implicite.	24
1.15	Exemples d'estampes virtuelles d'après [Sou01].	26
1.16	Déformation de surface implicite à squelette [OC97].	27
1.17	Quelques objets sculptés par déformations de l'espace, présentés dans [Bar84].	30
1.18	Déformation d'un modèle de girafe au moyen des FFD [Deb00].	31
1.19	Déformation FFD basée sur un réseau non parallélépipédique [Coq90].	31
1.20	Quelques déformations d'un modèle, présentées dans [MT97].	32
1.21	Un outil de déformation à volume constant présenté dans [ACWK04].	33
1.22	Création de volumes simples avec SKETCH [ZHH96].	34

1.23	Exemples de volumes construits avec CHATEAU [IH01].	35
1.24	Divers exemples d'objets obtenus avec des modeleurs par esquisse.	36
2.1	Modélisation 3D en deux temps, en scannant un modèle en argile.	40
2.2	L'argile permet des déformations et la création d'empreintes.	44
2.3	Le modelage d'une argile autorise des changements de topologie	44
2.4	Déformation d'un cube élastique d'après le modèle de [TPBF87].	46
2.5	Des exemples de simulations de fluides présentés dans [EMF02].	51
2.6	interaction entre fluide et solides d'après le modèle de [CMT04].	52
2.7	Fonte d'un bunny obtenue par simulation d'un fluide visqueux d'après [CMBT02].	53
2.8	Exemples de simulations de fluides selon [PTB ⁺ 03] et [MCG03].	55
2.9	Déformations locales calculées grâce à un automate cellulaire [ATTY99, Dru04].	56
3.1	La matière est stockée par le modèle dans une grille régulière.	60
3.2	Échanges de matière entre une cellule et sa voisine.	62
3.3	Voisinages de Margolus d'une cellule.	65
3.4	Influence du sens de parcours de la grille sur la relaxation.	66
3.5	Influence de l'ordre de traitement des cellules.	71
3.6	Résultats obtenus par déversement direct des excès.	72
3.7	Correction des résultats obtenus par déversement direct.	73
3.8	Interaction de l'outil avec la matière.	74
3.9	Directions privilégiées pour l'écoulement dans l'outil.	78
3.10	Cône des directions permises pour l'écoulement.	79
3.11	Résultats obtenus avec la première couche de simulation.	80
3.12	Déplacement de la matière par la seconde couche.	82
3.13	Construction du réseau masses-ressorts pour le calcul des déformations. . . .	85
3.14	Action des outils sur le réseau masses-ressorts.	87
3.15	Comportement incorrect du réseau masses-ressorts.	87
3.16	Problème des collisions entre outils et nœuds du réseau.	88
3.17	Exemples de cas simples de cinématique des fluides.	91

3.18	Principe de la pseudo-distance à un outil.	93
3.19	Erreur commise avec une simple pondération barycentrique.	95
3.20	Distances utiles au calcul des influences des outils.	96
3.21	Déformation d'une barre d'argile par la seconde couche.	96
3.22	Défauts de la seconde couche.	97
3.23	Comparaison des résultats de notre méthode avec un matériau réel.	99
3.24	Exemples de perçage de différents types de matériaux.	101
4.1	Déformation d'une barre d'argile sous l'action de quatre outils.	107
4.2	Action d'un outil en rotation sur l'argile.	108
4.3	Torsion d'une barre d'argile.	109
4.4	Effets d'une fluidité non homogène sur les déformations.	111
4.5	La couleur fait partie des caractéristiques importantes de la pâte à modeler.	113
4.6	Résultats obtenus par mélange de plusieurs teintes.	114
4.7	Un essai de déformation d'une pièce colorée d'argile avec notre modèle.	115
4.8	Échanges avec des cellules de taille différente.	116
4.9	Dissymétrie issue de l'usage d'une grille non régulière.	117
4.10	Solution proposée au problème de dissymétrie.	118
4.11	Illustration de la méthode de tracé rapide.	124
5.1	Quelques exemples d'interfaces héritées de la souris.	127
5.2	Bras articulés et gants de données.	128
6.1	Les trois "couches" du modèle utilisé pour le suivi.	144
6.2	La liaison pivot.	146
6.3	La liaison de type "Cardan".	146
6.4	Un exemple de chaîne articulaire	149
6.5	Angle entre deux pièces articulées autour d'une liaison pivot.	151
6.6	Mesure des paramètres d'une liaison "Cardan"	151
6.7	Le squelette de la main, où l'on identifie carpe, métacarpe et phalanges.	153

6.8	Modèle articulaire de main à 27 degrés de libertés.	154
6.9	Repères associés au éléments dans un fichier BVH.	155
6.10	Quelques exemples de modélisation volumique de la main.	158
6.11	Comparaison des distances euclidienne et pseudo-euclidienne.	161
6.12	Distance à une isosurface.	163
6.13	Différentes approches de fusion des ellipsoïdes	164
6.14	Notre modèle de main, obtenu par la fusion de seize éléments.	165
6.15	Prédiction du mouvement des points à la surface du modèle.	168
7.1	Matrice d'appariement tenant compte des points aberrants.	187
7.2	Évolution des probabilités d'appariement $\alpha_{i,j}$ avec σ_p	189
7.3	Rayon de recherche en fonction de σ	191
7.4	Sélection des points qui serviront de modèle.	192
7.5	Plate-forme de test des différents algorithmes.	196
7.6	Dérive de l'erreur lors d'un suivi par points d'intérêt.	197
7.7	Disparition de la dérive en l'absence d'occlusions.	197
7.8	Dérive due à une mise en correspondance incertaine.	198
7.9	Probabilité pour un point d'être associé à la surface.	201
7.10	Incertitude en position lors d'un ajustement.	202
7.11	Un suivi en deux étapes.	205
7.12	Évolution de l'erreur dans le suivi d'un ellipsoïde.	206
7.13	Comparaison de l'erreur pour différents algorithmes.	208
7.14	Suivi d'un objet rigide avec ICP.	208
7.15	Suivi d'un objet rigide avec la méthode EM.	209
7.16	Suivi d'un objet rigide avec la méthode Soft-assign.	209
7.17	Suivi d'un objet rigide notre méthode de suivi point à point robuste basée sur EM.	210
7.18	Suivi d'un objet rigide par simple ajustement d'un ellipsoïde aux données. . .	211
7.19	Suivi d'un objet rigide combinant suivi de points robuste et ajustement. . . .	212

<i>TABLE DES FIGURES</i>	261
8.1 Champ de déplacement pour un objet déformable	215
8.2 Sélection des points qui serviront à construire le modèle.	216
8.3 Suivi grâce à EM d'un objet déformable en éliminant les points aberrants. . .	218
8.4 Résultats obtenus sur une séquence synthétique.	225
8.5 Résultats pour un mouvement de prise vu côté paume.	228
8.6 Résultats pour un mouvement de flexion des doigts.	229
8.7 Résultats obtenus lors d'un retournement de la main.	230
A.1 Flux d'un champ de vecteurs à travers une surface fermée.	236
A.2 Circulation et rotationnel d'un champ de vecteurs.	237
B.1 Pseudo-distance à un ellipsoïde.	241

Table des matières

Introduction	1
I Sculpture	5
1 Modélisation géométrique : état de l'art	7
1.1 Introduction	7
1.2 Les débuts	8
1.2.1 Manipulation de volumes simples	8
1.2.2 Vers des maillages polygonaux	9
1.3 Édition de maillages polygonaux	10
1.3.1 Manipulation explicite du maillage	10
1.3.2 Vers une abstraction du maillage	10
1.3.3 Autres méthodes de manipulation	12
1.3.4 Rectification automatique du maillage	12
1.3.5 Déformations par squelette	13
1.3.6 Bilan	15
1.4 Surfaces lisses de forme libre	15
1.4.1 Les surfaces Splines paramétriques	16
1.4.2 Les surfaces d'optimisation	18
1.4.3 Les surfaces de subdivision	19
1.4.4 Bilan	21
1.5 Modélisation par surfaces implicites	21

1.5.1	Présentation des surfaces implicites	21
1.5.2	Expressions possibles du champ scalaire	22
1.5.3	Isovolumes exprimés directement	22
1.5.4	Fusion de volumes élémentaires	24
1.5.5	Des surfaces implicites discrétisées	27
1.5.6	Bilan	28
1.6	Sculpture par déformation de l'espace	29
1.6.1	Utilisation d'opérateurs de déformation	29
1.6.2	Les volumes FFD (Free Form Deformations)	30
1.6.3	Déformations à volume constant	32
1.6.4	Bilan	33
1.7	Modèles à base d'esquisses	34
1.7.1	Premiers pas	34
1.7.2	Création de volumes courbes	34
1.7.3	Des esquisses pour déformer les maillages	36
1.7.4	Bilan	36
1.8	Modélisation 3D pour le grand public	37
2	Modèles de matériau pour la sculpture	39
2.1	Introduction	39
2.1.1	Vers un modèle physique d'argile	39
2.1.2	Avantages d'une argile virtuelle	40
2.2	Comportement et caractéristiques d'une argile	42
2.2.1	À mi-chemin entre les solides et les liquides	42
2.2.2	Les caractéristiques recherchées	43
2.3	État de l'art sur les modèles physiques de matériau pour l'animation	45
2.3.1	Les matériaux élastiques et plastiques	45
2.3.2	Les modèles de fluides	48
2.3.3	Bilan	56

3	Un modèle d'argile virtuelle	57
3.1	Un modèle multi-couches	57
3.1.1	Retour sur l'équation de Stokes	57
3.1.2	Solution proposée : un modèle en trois couches	59
3.1.3	Une représentation sous-jacente commune	59
3.2	Première couche : un modèle de matériau fluide	61
3.2.1	Comportement du matériau	61
3.2.2	Bilan pour une des cellules de la grille	62
3.2.3	Un automate pour simuler la diffusion	63
3.2.4	Subtilités de la méthode	68
3.2.5	Déversement direct	72
3.3	Interaction de la première couche avec les outils	73
3.3.1	Introduction	73
3.3.2	Description des outils	74
3.3.3	La relaxation pour chasser le fluide de l'outil	75
3.3.4	Guidage de la relaxation	75
3.3.5	Définition de l'outil	76
3.3.6	Limitations de l'interaction	79
3.4	Objectifs de la seconde couche	80
3.4.1	Insuffisance de la seule couche fluide	80
3.4.2	Une seconde couche pour les déformations globales	81
3.4.3	Utilisation du champ de déformation	81
3.4.4	Remarque sur la terminologie	82
3.5	Une première approche	82
3.5.1	Un modèle dérivé des solides élastiques et plastiques	82
3.5.2	Construction du réseau de déformation	84
3.5.3	Animation du réseau	86
3.5.4	Limitations de cette approche	88
3.6	Retour sur les modèles de fluides	89

3.6.1	Le second terme de l'équation de Stokes	89
3.6.2	Approches pour la résolution	90
3.6.3	Quelques exemples	91
3.6.4	Idée générale	92
3.7	Seconde couche : un modèle pour les déformations globales	92
3.7.1	Calcul de la distance à un outil	92
3.7.2	Influence d'un seul outil	94
3.7.3	Champ de déplacement dans le cas de deux outils	95
3.7.4	Résultats et limitations	96
3.7.5	Collaboration des deux couches	98
3.8	Troisième couche : une "tension de surface" limitant la dispersion	101
3.8.1	Nécessité d'une troisième couche	101
3.8.2	Mise en œuvre de la troisième couche	102
3.8.3	Un ramasse-miettes	103
3.8.4	Résultats et limitations	103
4	Enrichissement du modèle	105
4.1	Introduction	105
4.2	Améliorations de la seconde couche	106
4.2.1	Extension à un nombre quelconque d'outils	106
4.2.2	Rotation des outils pour une interaction plus riche	107
4.3	Des paramètres intrinsèques supplémentaires	108
4.3.1	Introduction	108
4.3.2	Fluidité du matériau	109
4.3.3	Utilisation de la fluidité dans la simulation	110
4.3.4	Évolution des paramètres	111
4.4	Un matériau en couleurs	112
4.4.1	De l'origine des couleurs	112
4.4.2	Adaptation de notre modèle	113

<i>TABLE DES MATIÈRES</i>	267
4.4.3 Résultats obtenus	114
4.5 Vers une approche multirésolution	115
4.5.1 Introduction	115
4.5.2 Maillage multirésolution	116
4.5.3 Correction des dissymétries dues au maillage non régulier	116
4.5.4 Bilan	118
4.6 Quelques considérations sur le rendu	119
4.6.1 Rendu d'une surface implicite	119
4.6.2 Une méthode de tracé rapide	122
II Capture de gestes par vidéo	125
5 La capture de gestes	127
5.1 Introduction	127
5.2 Suivi de mouvements par vidéo	129
5.2.1 Introduction	129
5.3 Suivi de la main	130
5.3.1 Approches bidimensionnelles	130
5.3.2 Approches basées sur un modèle	131
5.3.3 Approches par classification	135
5.3.4 Bilan sur le suivi de la main	137
5.4 Suivi du corps humain	138
5.4.1 Introduction	138
5.4.2 Modèles articulaires	139
5.4.3 Confrontation du modèle aux mesures	139
5.4.4 Prédiction du mouvement	140
5.4.5 Bilan	140
6 Modèle articulé et déformable de la main	143
6.1 Introduction	143

6.2	Construction d'un squelette	144
6.2.1	Objectif poursuivi	144
6.2.2	Contraintes articulaires	145
6.2.3	Paramétrisation des liaisons	147
6.2.4	Construction de la chaîne articulaire	148
6.2.5	Limitation des mouvements	150
6.2.6	La cinématique de la main	152
6.3	Habillage du squelette	157
6.3.1	Modélisation du volume	157
6.3.2	Solutions usuellement retenues	157
6.3.3	Les surfaces implicites	159
6.3.4	Distance à une surface implicite	160
6.3.5	Calcul de la pseudo-distance à un ellipsoïde	162
6.3.6	Modèle géométrique de main	164
6.4	Habillage de l'isosurface	165
6.4.1	Nécessité de l'habillage	165
6.4.2	Principe du "skinning"	165
6.4.3	Application à la prédiction de mouvement d'un point	166
6.4.4	Déformations de la peau	166
7	Suivi robuste d'objets rigides	169
7.1	Introduction au problème du suivi de mouvement	169
7.1.1	Deux approches différentes	169
7.1.2	Organisation du chapitre	169
7.2	Présentation du dispositif	170
7.2.1	Un système multi-caméras	170
7.2.2	Efficacité de la reconstruction 3D sur la main	171
7.2.3	Points 3D parasites	171
7.2.4	Enrichissement des informations	172

7.3	Suivi point à point	172
7.3.1	Introduction	172
7.3.2	Cas des points non appariés	173
7.3.3	L'algorithme ICP (ou <i>Iterative Closest Point</i>)	176
7.3.4	L'algorithme EM (ou <i>Expectation Maximization</i>)	179
7.3.5	Utilisation de EM dans le cadre du suivi de mouvements	184
7.3.6	Solution proposée pour un suivi robuste avec EM	188
7.3.7	Mise en œuvre pratique	192
7.3.8	Avantages et inconvénients de la méthode	194
7.4	Suivi par ajustement de modèle	198
7.4.1	Introduction	198
7.4.2	Objectif et critère de minimisation	199
7.4.3	Élimination des outliers	200
7.4.4	Avantages et inconvénients de la méthode	202
7.5	Combinaison des deux approches	203
7.5.1	Motivation	203
7.5.2	Stratégie de minimisation	203
7.5.3	Résultats sur des données simulées	205
7.5.4	Utilisation conditionnelle	207
7.5.5	Initialisation	207
8	Suivi d'objets articulés et déformables	213
8.1	Introduction	213
8.2	Modèles d'objets articulés et déformables	213
8.3	Suivi point à point	215
8.3.1	Introduction	215
8.3.2	Mise en œuvre	216
8.3.3	Évaluation des coefficients $\alpha_{i,j}$	217
8.4	Suivi par ajustement au modèle	220

8.5	Combinaison des méthodes	222
8.6	Cas de la main	224
8.6.1	Application du schéma de suivi	224
8.6.2	Résultats obtenus	224
Conclusion		231
Annexes		235
A Éléments d'analyse vectorielle		235
A.1	Différentielle et gradient d'une fonction	235
A.2	Flux d'un champ de vecteurs	236
A.3	Circulation d'un champ de vecteurs	237
A.4	Expressions du second ordre : le laplacien	237
A.5	Expression sur une grille	238
B Pseudo-distance		241
Bibliographie		243
Table des figures		257
Table des matières		263