



UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE

Numéro d'ordre :

Estimation de propriétés mécaniques d'objets complexes à partir de séquences d'images

Thèse de doctorat (spécialité Informatique) présentée le 5 mars 2010

par Cédric Syllebranque

Composition du jury

Rapporteurs : Yannick Remion, professeur Université de Reims
Marie Odile Berger, CR Inria HDR, Nancy

Examineurs : Mohamed Daoudi, professeur Telecom Lille1
Stéphane Cotin, DR Inria, Lille
Marc des Rieux, PDG de Didhaptic
Maud Marchal, MC Irisa Rennes

Directeurs de thèse : Christophe Chaillou, professeur Lille1
Samuel Boivin, CR Inria, Saclay

« À mon père, qui ne verra pas le fruit d'un travail qu'il a contribué à lancer... »

Remerciements

Je tiens en premier lieu à remercier les membres de mon jury, Marie-Odile Berger, Yannick Remion, Mohamed Daoudi, Stéphane Cotin, Marc des Rieux et Maud Marchal pour avoir accepté d'y figurer.

Je remercie également Christophe Chaillou pour avoir encadré ce travail, Claude Puech pour l'avoir autorisé, et Samuel Boivin pour son suivi et ses idées.

Je tiens également à remercier Frederic Giraud et Yohann Dhont pour leur aide précieuse sur la conception et la réalisation des dispositifs de mesure d'efforts.

Un grand merci à Christian Duriez pour son aide sur la simulation mécanique et ses nombreuses idées, Stéphane Cotin, Adrien Thetheen et Nicolas Martin pour leur soutien, et tout le reste de l'équipe Alcove pour leur bonne humeur et pour m'avoir supporté toutes ces années.

Bien évidemment, je n'oublie pas ici mes amis, Bertrand Gressier, Alexis Muller, Damien Marchal, Lukasz Kokot, Sophie Pochwalski, Angélique Muller, Iovka Boneva, Laurent Smadja, Christophe Picard et François Ingelrest, ainsi que ceux que je ne cite pas ici mais qui me sont chers, sans qui ces travaux seraient vite devenus insupportables.

Ces lignes n'existeraient pas sans le soutien inconditionnel de mes parents, Marc et Élisabeth, de ma famille, Sébastien, Florent, Nadège ainsi que mes neveux et nièces, belle famille, cousins, oncles, tantes et grand-mère. C'est le cœur lourd que je mets un point final à ce document que tu ne liras jamais papa.

Il ne me reste qu'une personne à remercier, et non des moindres, ma compagne, Vanessa. Impossible de tout exprimer ici. Je n'écrirais donc qu'un seul mot pour ton soutien durant toutes ces années et pour m'avoir aidé à traverser tous ces moments difficiles : Merci.

Table des matières

| | | |
|----------|---|-----------|
| 1 | Introduction | 12 |
| 1.1 | Contexte | 13 |
| 1.2 | Problématique | 14 |
| 1.3 | Proposition | 14 |
| 1.4 | Plan du document | 15 |
| 2 | Lois physiques des matériaux déformables | 17 |
| 2.1 | Caractérisation des déformations | 17 |
| 2.1.1 | Phases de comportement | 18 |
| 2.1.2 | Matériaux élastiques et viscoélastiques | 18 |
| 2.1.3 | Phénomène de préconditionnement | 19 |
| 2.1.4 | Matériaux isotropes et anisotropes | 20 |
| 2.1.5 | Homogénéité | 20 |
| 2.2 | Modélisations mathématiques | 20 |
| 2.2.1 | Généralités | 21 |
| 2.2.2 | Élasticité linéaire | 21 |
| 2.2.3 | Viscoélasticité linéaire | 23 |
| 2.2.4 | Élasticité non-linéaire | 24 |
| 2.3 | Simulation numérique | 24 |
| 2.3.1 | Simulation en temps interactif | 25 |
| 2.3.1.1 | Modèles masses-ressorts | 26 |
| 2.3.2 | Simulation par éléments finis | 28 |
| 2.3.2.1 | Description | 29 |
| 2.3.2.2 | Simulation statique | 32 |
| | Déformations linéaires | 32 |
| | Déformations non-linéaires | 33 |
| | Modèle corotationnel | 33 |
| 2.3.2.3 | Simulation dynamique | 35 |
| 2.3.2.4 | Contraintes | 35 |
| | Points fixes : Conditions de Dirichlet | 35 |

| | | |
|----------|---|-----------|
| | Conditions d'inégalité | 36 |
| 2.4 | Module de Young et Coefficient de Poisson | 37 |
| 2.4.1 | Module de Young | 37 |
| 2.4.1.1 | Unité | 38 |
| 2.4.2 | Coefficient de Poisson | 39 |
| 2.4.2.1 | Unité | 39 |
| 2.4.2.2 | Auxétisme | 40 |
| 2.4.3 | Relation avec les coefficients de Lamé | 40 |
| 3 | Mesures et estimations de paramètres | 43 |
| 3.1 | Le concept de mécanique inverse | 44 |
| 3.1.1 | Méthode d'estimation inverse générale | 44 |
| | L'exemple du rendu inverse | 44 |
| 3.1.2 | Mécanique inverse | 45 |
| | Exemple du ressort à spires | 46 |
| 3.1.3 | Mécanique inverse à partir de vidéos | 48 |
| | Exemple des tissus | 49 |
| 3.2 | Optimisation | 51 |
| 3.2.1 | Quelques méthodes d'optimisation continue | 52 |
| 3.2.1.1 | Quelques algorithmes locaux | 53 |
| 3.2.1.2 | Quelques algorithmes globaux | 54 |
| | Recuit simulé | 54 |
| | Recherche avec tabous | 55 |
| | Algorithmes évolutionnaires | 55 |
| 3.3 | Capture de données | 56 |
| 3.3.1 | Dispositifs de mesure mécaniques | 56 |
| 3.3.1.1 | Dispositifs automatiques | 57 |
| | La machine de Kawabata | 57 |
| | Indenteurs | 57 |
| | Ultrasons | 61 |
| 3.3.1.2 | Dispositifs interactifs | 63 |
| 3.3.2 | Dispositifs incluant l'approche visuelle | 65 |
| 3.4 | Identification de paramètres | 68 |
| 3.4.1 | Médical | 70 |
| 3.4.2 | Non médical | 72 |
| 3.5 | Proposition | 73 |
| 4 | Le dispositif expérimental | 77 |
| 4.1 | Introduction | 77 |

| | | |
|----------|--|------------|
| 4.2 | Dispositif optique | 78 |
| 4.2.1 | Description | 78 |
| 4.2.2 | Généralités sur le calibrage | 78 |
| 4.2.3 | Calibrage de la caméra vidéo : <i>paramètres intrinsèques</i> | 80 |
| 4.2.4 | Calibrage du vidéo-projecteur : <i>paramètres intrinsèques</i> | 81 |
| 4.2.4.1 | Quelques méthodes existantes | 81 |
| 4.2.4.2 | Notre méthode | 82 |
| 4.2.5 | Positionnement virtuel des dispositifs optiques : <i>paramètres extrinsèques</i> | 83 |
| | Marqueurs ARToolkit | 83 |
| 4.2.5.1 | Méthode découplée | 84 |
| 4.2.5.2 | Méthode séquentielle | 85 |
| 4.3 | Dispositif de mesure mécanique | 86 |
| 4.3.1 | Capteur de force | 87 |
| | Calibrage | 87 |
| 4.3.2 | Conditionnement et évolution | 87 |
| 4.3.2.1 | Prototype | 88 |
| 4.3.2.2 | Montage « vissé » | 89 |
| 4.3.2.3 | Montage « segmenté » | 89 |
| | Touche | 91 |
| 4.3.3 | Positionnement virtuel | 92 |
| 4.3.3.1 | Capture magnétique | 93 |
| 4.3.3.2 | Capture mécanique | 94 |
| 4.3.3.3 | Capture visuelle | 95 |
| 4.4 | Mise en œuvre | 97 |
| 5 | Méthodes et algorithmes d'estimation | 101 |
| 5.1 | Simulation mécanique | 101 |
| 5.2 | Rendu graphique | 103 |
| | Distorsion optique | 104 |
| 5.3 | Estimations séparées | 105 |
| 5.3.1 | Étude de corrélation/décorrélation | 106 |
| | Procédure | 106 |
| 5.3.2 | Analyse et ordre d'estimation | 108 |
| 5.4 | Estimation du coefficient de Poisson | 111 |
| 5.4.1 | Méthode | 111 |
| 5.4.2 | Traitements d'images préliminaires | 111 |
| 5.4.2.1 | Détourage automatique | 111 |
| 5.4.2.2 | Masquage automatique de l'outil | 113 |
| 5.4.2.3 | Recadrage semi-automatique | 114 |

| | | |
|----------|---|------------|
| 5.4.2.4 | Extraction de la position de l'outil | 114 |
| 5.4.3 | Métriques d'erreur | 115 |
| 5.4.3.1 | Silhouette | 118 |
| 5.4.3.2 | Différence absolue | 118 |
| 5.4.3.3 | Différence au carré | 118 |
| 5.4.3.4 | Flou | 119 |
| 5.4.3.5 | Corrélation inverse | 119 |
| 5.4.3.6 | Métriques composées | 120 |
| 5.4.3.7 | <i>Angular Map</i> | 121 |
| 5.4.3.8 | Classes de métriques | 122 |
| 5.4.4 | Optimisation | 123 |
| 5.4.4.1 | Méthode « exhaustive » | 123 |
| 5.4.4.2 | Choix de l'algorithme d'optimisation | 123 |
| 5.5 | Estimation du module de Young | 127 |
| 5.5.1 | Méthode directe | 127 |
| 5.5.2 | Méthode inverse | 129 |
| 5.5.2.1 | Paramètres initiaux | 130 |
| 5.5.2.2 | Comparaison des forces | 130 |
| 5.5.2.3 | Optimisation | 132 |
| 6 | Résultats | 135 |
| 6.1 | Matériaux testés | 135 |
| 6.1.1 | Silicones | 135 |
| | Mesures mécaniques du module de Young | 136 |
| 6.1.2 | Éponges | 137 |
| 6.1.3 | Peluches | 138 |
| 6.2 | Estimations du coefficient de Poisson | 139 |
| 6.2.1 | Silicones | 139 |
| 6.2.1.1 | Autres objets silicones | 143 |
| | Cylindres | 143 |
| | Hémisphères | 145 |
| 6.2.1.2 | Métrique de projection | 147 |
| 6.2.1.3 | Optimisation par recuit simulé | 147 |
| 6.2.2 | Éponges | 149 |
| 6.2.3 | Peluches | 152 |
| | Problème de densité | 152 |
| 6.3 | Estimations du module de Young | 154 |
| 6.3.1 | Silicones | 154 |
| 6.3.1.1 | Validation par mesures mécaniques | 156 |

| | | |
|----------|--|------------|
| 6.3.2 | Éponges | 156 |
| 6.3.3 | Peluches | 157 |
| 6.4 | Bilan et réutilisation | 158 |
| 7 | Conclusion et perspectives | 161 |
| 7.1 | Conclusion | 162 |
| 7.2 | Travaux futurs | 164 |
| 7.2.1 | Dispositif matériel | 164 |
| 7.2.2 | Estimation du coefficient de Poisson | 164 |
| 7.2.3 | Estimation du module de Young | 165 |
| | Bibliographie personnelle | 167 |
| | Bibliographie | 168 |
| | Index | 179 |
| | Quelques symboles | 183 |

Premier Chapitre

INTRODUCTION

1.1 Contexte

De nombreux phénomènes physiques sont aujourd'hui grandement étudiés par simulation numérique. Ces études ont amené une meilleure compréhension du comportement de notre environnement. Issus de modèles mathématiques, les implémentations informatiques permettent d'obtenir des prédictions météorologiques, des images synthétiques réalistes à partir de l'observation du comportement de la lumière, ou encore des environnements mécaniques complexes, par exemple en réalité virtuelle.

Si l'on s'attarde sur l'évolution de la modélisation informatique des comportements mécaniques en particulier, on s'aperçoit que ce domaine n'a cessé de se développer ces dernières décennies. De grandes avancées ont été réalisées et de plus en plus de modèles informatiques voient le jour, avec une complexité toujours croissante. Chacun de ces modèles est défini dans un but précis. Certains, très complexes, seront destinés à obtenir un comportement toujours plus réaliste mais à un coût en temps de calcul très important, malgré la puissance actuelle des machines. Ces modèles informatiques permettent, entre autres, des simulations au comportement réaliste dans le génie civil, par exemple pour tester la résistance de ponts suspendus aux forces naturelles extrêmes, comme les séismes ou les typhons. Les enjeux de ces prévisions sont colossaux. Elles permettent de limiter au maximum les risques de destructions et les pertes humaines que celles-ci pourraient engendrer.

D'autres modèles, simplifiés pour être plus rapides, permettent de créer des simulations en temps-réel où l'utilisateur peut interagir avec un environnement virtuel. C'est le cas par exemple dans les nombreux simulateurs d'apprentissage, et le domaine le plus florissant dans ce secteur est l'apprentissage au geste chirurgical. Là aussi, les enjeux sont importants et on imagine les répercussions que pourrait avoir une erreur de simulation du comportement d'organes humains lors de la préparation d'une opération réelle.

Il existe bon nombre d'autres secteurs également très demandeurs de modélisation mécanique, principalement pour accroître leur productivité. On citera par exemple l'industrie du cinéma pour les productions de films d'animations (ex : simulation de vêtements) ou d'effets spéciaux réalistes (ex : fumée, explosions, feux, tornades, etc...). En effet, les infographistes ne doivent dès lors plus modéliser image par image chaque comportement mais lancer une simulation globale avant de peaufiner, le cas échéant, le résultat obtenu. Le gain de temps est alors colossal et les résultats semblent plus précis, mieux intégrés, en somme, plus réalistes.

1.2 Problématique

Afin de concevoir tous ces modèles mécaniques, de nombreux travaux se basent sur des mesures réalisées à partir d'objets réels. Ces mesures sont alors analysées et les équipes de recherche tentent de proposer des solutions pour réaliser des simulateurs suivant les buts recherchés, que ce soit en termes de réalisme ou de temps de calcul. Mais il existe un point commun à tous ces modèles, du plus simple au plus complexe : *le paramétrage*. Derrière cette notion générale se cache la clé de la réussite d'une simulation réaliste. En effet, il est difficile, voire impossible, même pour un expert, de trouver les bons paramètres de ces simulateurs pour produire un effet donné, et tout particulièrement sur des objets réels complexes comme un œil ou un foie par exemple. L'adéquation de ces paramètres avec le monde réel est très loin d'être évidente, et dans le meilleur des cas il s'agit toujours de « tâtonner » pour obtenir le résultat voulu, et ce parfois pendant des heures. Cette difficulté de réglage est encore plus évidente lorsque les paramètres recherchés n'ont aucune signification physique.

De plus, les résultats attendus des simulations numériques ont parfois des enjeux capitaux. Dans ce genre d'exercice, l'exactitude et la rigueur sont les maîtres mots. Un mauvais paramétrage des modèles mécaniques peut alors avoir des conséquences désastreuses, qu'elles soient matérielles, intellectuelles ou humaines.

Il existe des méthodes pour tenter d'extraire ces paramètres à partir des mesures mécaniques effectuées sur des objets réels. Malheureusement, soit les machines utilisées pour ces mesures sont massives, complexes et/ou coûteuses (cf section 3.3.1), soit les pré-requis sont très contraignants. Par exemple, dans bon nombre des cas, il sera nécessaire d'obtenir un échantillon cylindrique d'un objet fait du matériau que l'on veut tester. Cette tâche peut parfois s'avérer très complexe, tout particulièrement sur des objets hautement déformables. De nombreux organes, comme le foie ou les reins, font notamment partie de cette catégorie qui, en outre, subissent des changements moléculaires modifiant leurs propriétés mécaniques dès qu'ils sont prélevés (*ex-vivo*).

Une approche plus récente consiste à estimer ces paramètres à partir d'images ou de vidéos d'objets réels. Dans le cas d'objets déformables, on filme l'objet lorsqu'une contrainte lui est appliquée afin d'analyser son comportement de réaction. Pour estimer les divers paramètres du modèle mécanique, la méthode la plus utilisée est une technique dite « inverse ». Cette technique a été utilisée avec succès pour l'estimation des propriétés photométriques d'objets à partir de photographies (*inverse rendering* [Yu 99][Boivin 01]). Cette thèse s'inscrit dans le cadre cette approche inverse que nous détaillerons en 3.1.1.

1.3 Proposition

Nous proposons ici une méthode simple et un ensemble matériel peu coûteux permettant d'obtenir un jeu de paramètres à partir de séquences d'images. Ces images sont obtenues en filmant, à l'aide d'une caméra standard, la déformation progressive d'un objet.

Pour minimiser le coût de la solution totale et pour offrir une plus grande liberté de mouvement, nous choisissons d'appliquer la déformation via un utilisateur et non à l'aide d'une machine robotisée.

De plus, comme nous le verrons par la suite, tous les paramètres ne peuvent pas être estimés à partir de vidéos. En effet, ceux-ci se caractérisent par des notions haptiques uniquement,

par exemple par les forces que le matériau génère en réponse à l'application d'une contrainte. En conséquence, il est nécessaire d'introduire un outil spécial qui permet de mesurer les forces appliquées par l'utilisateur. Ce type d'outil n'étant pas disponible sous la forme désirée, nous concevons également un dispositif de capture de force. Cet outil est de type « stylet » pour une utilisation simplifiée et une bonne prise en main.

Enfin, pour un meilleur contrôle des conditions d'éclairage, nous utilisons un video-projecteur classique, qui, avec la caméra numérique standard et notre nouvel outil de capture de force, constitue une solution matérielle simple et peu onéreuse.

1.4 Plan du document

Ce document est scindé en deux grandes parties.

Tout d'abord, les deux premiers chapitres décrivent le contexte et réalisent un état de l'art des travaux déjà effectués sur le sujet. Dans un premier temps, nous effectuons donc un survol des notions essentielles à la compréhension du document. Nous y expliquons les différentes lois physiques des matériaux déformables à l'aide de multiples illustrations ainsi que le vocabulaire associé à leurs comportements. Nous caractérisons également les différents types de déformations et donnons une définition des modèles mathématiques généralement utilisés ainsi que les méthodes de modélisation et de résolution numérique. Nous nous attardons sur une définition précise des paramètres mécaniques que nous nous proposons d'estimer dans ces travaux, le *module de Young* et le *coefficient de Poisson*. Dans un second temps, nous introduisons le concept de *mécanique inverse* avant de dresser un état de l'art dans les différents domaines en jeu dans ces travaux. Il s'agit de l'optimisation numérique, de la capture de données réelles et de l'estimation de paramètres à partir de ces données. À la fin de cette première partie, nous discutons des choix que nous avons opérés et détaillons notre proposition de contribution à ce sujet de recherche.

La seconde partie du document est composée de trois chapitres décrivant cette contribution. Un chapitre est dédié à la présentation d'un nouveau dispositif expérimental de capture. Les composants de ce dispositif sont classés dans deux parties distinctes. La première partie est dédiée au dispositif d'acquisition visuelle. Nous y présentons la caméra vidéo utilisée et le vidéo projecteur que nous avons choisi pour projeter des motifs sur les objets déformés. Nous y détaillons également les méthodes originales utilisées pour leur calibrage et leur positionnement dans notre environnement virtuel. La seconde partie de ce chapitre présente quant à elle l'évolution du dispositif de capture de forces interactif que nous avons développé. Dans le chapitre suivant, nous proposons nos méthodes et algorithmes d'estimation de paramètres à partir des données capturées. Nous y exposons tout d'abord les étapes préliminaires qui nous permettent d'estimer séquentiellement deux paramètres, le *module de Young* et le *coefficient de Poisson*. Nous détaillons ensuite les différentes *métriques d'erreur* (ou fonctions objectives) qui sont au cœur de notre algorithme de *mécanique inverse* à partir de séquences d'images. Enfin, un chapitre est consacré aux résultats obtenus et à leur analyse, avant de conclure sur les différentes perspectives qu'offrent ces travaux.

Deuxième Chapitre

LOIS PHYSIQUES DES MATÉRIAUX DÉFORMABLES

Dans ce chapitre, nous introduisons les notions de matériau déformable et de mécanique continue. Nous expliquons les lois mathématiques qui permettent d'exprimer les diverses déformations rencontrées. Cela nous permettra dans un premier temps de nous familiariser avec les propriétés mécaniques des matériaux déformables et les divers paramètres et méthodes de simulation qui permettent de les reproduire.

2.1 Caractérisation des déformations

On peut qualifier de déformable un objet solide dont la forme géométrique change sous une action externe. Cette action peut avoir des natures très diverses, comme l'application d'une force (par exemple la gravité universelle) ou encore la modification de sa température. Le qualificatif « déformable » de l'objet peut aussi s'appliquer au matériau qui le compose. Potentiellement, tout type de matériau connu est déformable, dès lors que l'action qui lui est appliquée est suffisamment importante.

La nature de ces déformations peut être très variée. En effet, l'objet en question peut par exemple se comprimer, se dilater, plier ou encore vriller. Dans des cas extrêmes non étudiés ici, l'état du matériau peut même changer, par exemple passer de l'état solide à l'état liquide.

Tous ces comportements peuvent être expliqués à l'échelle moléculaire voire atomique. Cependant, on comprend aisément qu'il est actuellement impossible de modéliser ces interactions à l'échelle moléculaire lorsqu'on désire simuler informatiquement et à échelle humaine ce type de modifications.

Le domaine qui s'intéresse à la déformation des solides et à l'écoulement des fluides se nomme la *mécanique des milieux continus*. L'hypothèse faite ici consiste à considérer les propriétés mécaniques de ces milieux (élasticité, compressibilité, etc...) comme continues. Il devient alors possible d'utiliser les outils mathématiques reposant sur les fonctions continues et/ou dérivables pour caractériser les lois qui les régissent.

Bien évidemment, il est impossible de concevoir une loi mathématique permettant de représenter la totalité des comportements observables. La *mécanique des milieux continus* se divise en deux sous domaines : la *mécanique des fluides* et celle des *déformations solides*. Nous nous intéressons dans nos travaux à ces dernières.

2.1.1 Phases de comportement

Lorsque l'on sollicite un matériau en appliquant sur celui-ci un effort croissant, sa réaction se divise en trois *phases comportementales* :

- *Déformation élastique* : Lors de cette phase, si l'action exercée est relâchée, l'objet retourne à son état initial.
- *Déformation plastique* : Dès le début de cette phase, la structure du matériau est irrémédiablement modifiée et l'objet ne peut retourner à son état initial.
- *Rupture* : On observe une cassure et l'objet peut se scinder.

La figure 2.1 illustre ces trois phases lors d'un test de traction sur un objet cylindrique.

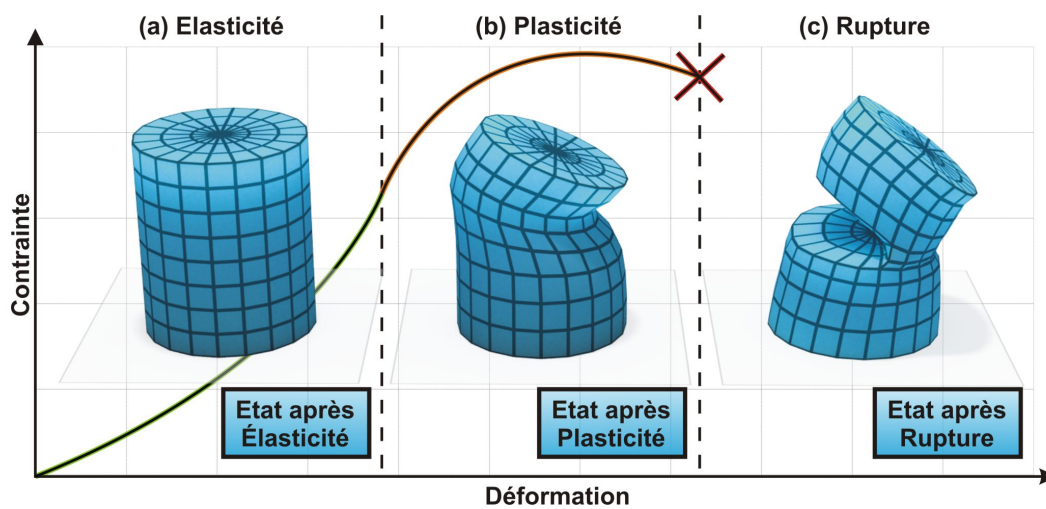


Figure 2.1 - Phases de déformation solide - Cette courbe montre la relation déformation/contrainte pour un test de traction sur un objet cylindrique. Les domaines de déformation (élasticité, plasticité, rupture) sont illustrés par l'état de l'objet dans le cas d'un relâchement de la traction lors de chacune de ces phases. On note que lors de la phase élastique l'objet revient à l'état normal, qu'il est irrémédiablement altéré à partir de la phase plastique et que la topologie peut changer lors de la rupture.

2.1.2 Matériaux élastiques et viscoélastiques

Les phases obtenues précédemment sont issues de l'application d'un effort croissant. On appelle ce procédé le *chargement* (*loading*). Si l'on stoppe cet effort, le matériau entre en phase de *relaxation* (*unloading*).

Dans le cas d'un matériau purement élastique, les efforts générés lors de ces deux phases sont identiques. Cependant, dans la nature, il n'existe que très peu de matériaux de ce type. En général, on observe une *dissipation d'énergie* lors de la phase de relaxation : on dit alors que le matériau a un comportement *viscoélastique*.

Si l'on trace une courbe de ces efforts (cf Fig. 2.2), on obtient un effet d'*hystérésis* : la différence entre les courbes de chargement et de relaxation représente l'*énergie dissipée* (principalement sous forme de chaleur).

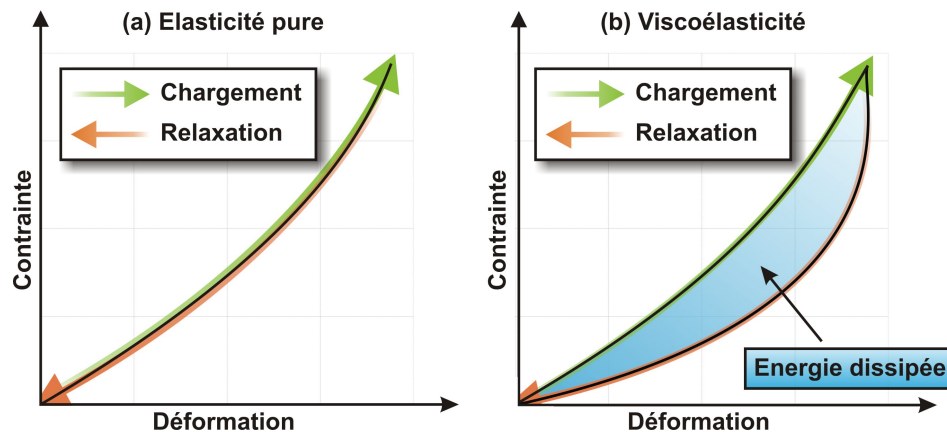


Figure 2.2 - Comportement élastique (a) et viscoélastique (b) : Dans le cas élastique, le comportement est strictement identique pendant la phase de chargement (en vert) et de relaxation (en orange). En revanche, on observe un hystérésis dans le cas viscoélastique. Cet écart correspond à l'énergie dissipée (en bleu).

2.1.3 Phénomène de préconditionnement

En plus d'un comportement *viscoélastique*, certains matériaux présentent des courbes de réponse différentes pour un même cycle de chargement/relaxation. En effet, lorsque ce même cycle est appliqué plusieurs fois successivement, on observe un décalage progressif des efforts générés avant une certaine stabilisation. On appelle ce phénomène le *pré-conditionnement*. Ce type de comportement, illustré sur la figure 2.3, est particulièrement fréquent pour les tissus organiques.

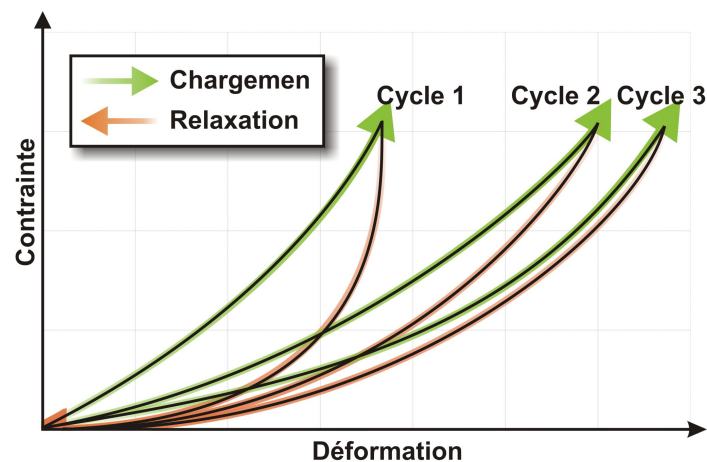


Figure 2.3 - Préconditionnement du matériau : Les cycles de chargement/relaxation successifs induisent un préconditionnement des matériaux. A partir d'un certain nombre de cycles, ce phénomène s'atténue.

2.1.4 Matériaux isotropes et anisotropes

Il existe une autre caractéristique pour les matériaux. Si la contrainte mesurée lors d'une déformation est la même quelle que soit la direction d'application de cette déformation, on dit que le matériau est *isotrope*. Ce type de matériau étant très rare dans la nature, la plupart des matériaux ayant cette propriété sont synthétiques (silicones par exemple).

La majorité des matériaux est donc *anisotrope* car bien souvent, les efforts générés par une déformation diffèrent selon l'orientation de la contrainte appliquée. La figure 2.4 illustre la différence entre ces deux caractéristiques.

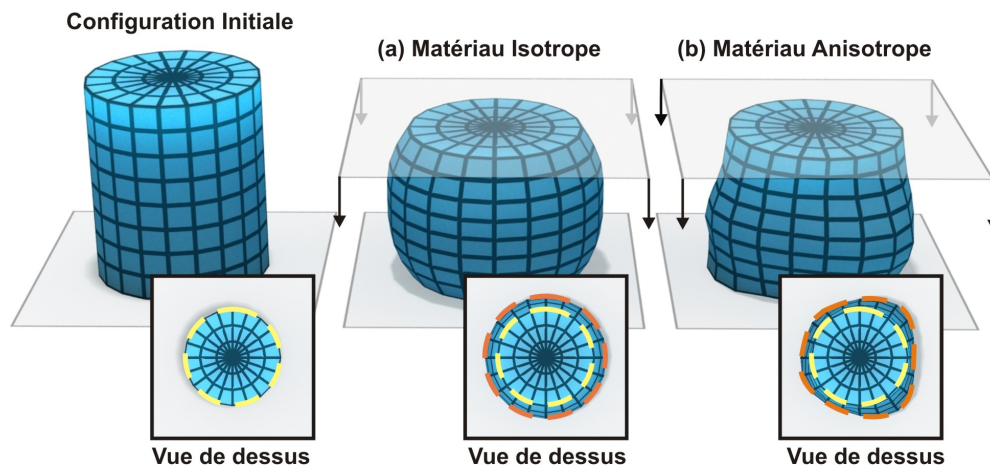


Figure 2.4 - Matériaux isotropes et anisotropes : L'image de gauche illustre la configuration initiale d'un objet. On applique une force identique sur deux objets géométriquement semblables mais fabriqués à partir de matériaux différents : l'un isotrope (a) et l'autre anisotrope (b). On note clairement que la déformation en (a) est identique dans toutes les directions alors que la déformation en (b) ne l'est pas. Les vues de dessus permettent de mieux voir ce phénomène en mettant en évidence les différences entre le contour initial (pointillés jaunes) et après la déformation (pointillés oranges).

2.1.5 Homogénéité

Un matériau est dit *homogène* si la matière qui le compose a le même comportement et les mêmes caractéristiques en tout point de l'objet. Dans le cas contraire, on parle de matériau *hétérogène*. On peut noter qu'un matériau hétérogène favorise un comportement global anisotrope. Cette notion d'*hétérogénéité* est utilisée dans le domaine médical, en *mammographie* (cf Fig. 2.5), ou il s'agit de déterminer les zones plus rigides pouvant se révéler être des tumeurs cancéreuses [Revell 05].

2.2 Modélisations mathématiques

Comme on peut le voir, il existe dans le « monde réel » un nombre très important de comportements différents. De plus, la plupart des matériaux rencontrés montrent une combinaison de ces différents phénomènes. Nous allons voir ici certaines formulations qui permettent de les représenter mathématiquement. Les cas que nous traitons au cours de nos travaux restant

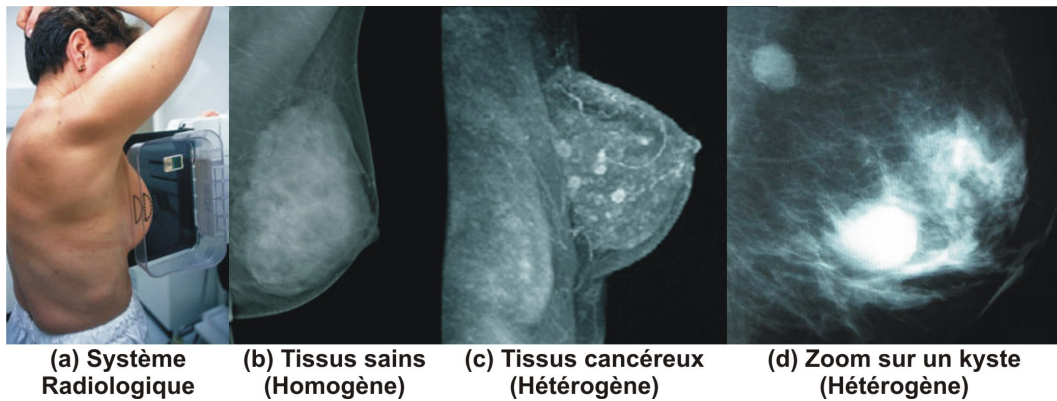


Figure 2.5 - Tissus homogènes et hétérogènes : exemple en mammographie. La figure (a) montre le système en radiologie utilisé pour réaliser l'examen mammaire. Le résultat de cet examen pour des tissus sains (b) révèle un ensemble assez homogène. En revanche, dans le cas de tumeurs (c) ou de kystes (d), on note de grandes disparités.

dans le cadre de la déformation élastique des matériaux (cf Fig 2.1), nous nous limiterons à cette phase de déformation dans la suite de ce document.

2.2.1 Généralités

Il existe deux *représentations mathématiques* pour décrire les milieux continus. Tout d'abord la *représentation Lagrangienne* qui définit les mouvements au niveau de chaque particule. La position de chaque particule sera donc fonction de sa position initiale et du temps écoulé ($\mathbf{x} = \mathbf{x}(\mathbf{m}_0, t)$). La discrétisation de l'objet modélisé suit le corps représenté dans son mouvement. La *représentation Eulérienne* considère quant à elle chaque point de l'espace fixe et représente donc la valeur pour la particule qui occupe le point m à l'instant t donné ($\mathbf{x} = \mathbf{x}(\mathbf{m}, t)$). Dans ce cas, l'objet traverse une discrétisation de l'espace.

De manière générale, la *représentation Eulérienne* est utilisée pour décrire les fluides : un découpage de l'espace en une « grille » permet de suivre l'évolution du fluide à chaque noeud de cette grille fixe. En revanche, la *représentation Lagrangienne* est mieux adaptée à la représentation des solides car la plupart des méthodes nécessitent une référence à l'état initial de l'objet. Nous utilisons donc cette représentation.

Les lois de comportement ont pour but de modéliser les relations qui existent entre les *déplacements*, les *déformations*, les *forces* et les *contraintes*. Un déplacement représente le changement de position d'une particule élémentaire du milieu continu. Si l'on considère une petite parcelle volumique à l'intérieur de ce milieu continu, le mouvement de la particule en modifie la configuration. Cette modification est mesurée par un *tenseur de déformation* (en pratique une matrice 3x3) qui permet de quantifier les *déformations élémentaires* appliquées au petit élément volumique (cf Fig 2.6). De la même manière, on mesure l'effet de forces externes appliquées sur cet élément de volume par un *tenseur de contrainte*.

2.2.2 Élasticité linéaire

L'*élasticité linéaire* est la méthode la plus simple pour exprimer les relations entre ces entités. Les lois qui la composent sont en effet très faciles à implémenter lors de la *résolution*

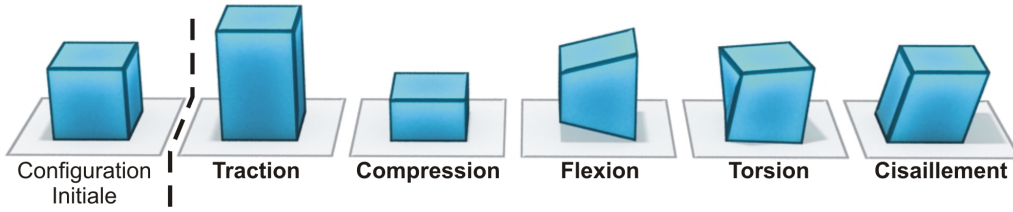


Figure 2.6 - Déformations élémentaires d'un élément volumique

numérique (calcul informatique de la déformation, cf 2.3). Cela fait de cette méthode l'une des plus utilisées dans les applications en temps-réel.

Une *déformation élastique* est dite linéaire si la relation entre la mesure de la déformation (*tenseur de déformation, strain*) et l'effort généré par le matériau (mesuré par le *tenseur de contrainte, stress*) est linéaire. En réalité, ce comportement n'est observable que pour très peu de matériaux. Cependant, on peut admettre que ce phénomène est vrai pour tous les matériaux lorsque ces déplacements et ces déformations sont relativement petits par rapport à la taille de l'objet.

Dans ce cas, on peut exprimer le tenseur des déformations $[\boldsymbol{\varepsilon}]$ (aussi nommé *tenseur de Green*) par rapport au déplacement \mathbf{u} de la particule par la formule :

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\delta u_i}{\delta x_j} + \frac{\delta u_j}{\delta x_i} \right) \text{ avec } i \in [1, 2, 3], j \in [1, 2, 3] \text{ et } \mathbf{u} = [u_1, u_2, u_3]^T \quad (2.1)$$

Ce qui donne sous forme matricielle, pour un système de coordonnées cartésien ($x = x_1, y = x_2, z = x_3$) :

$$[\boldsymbol{\varepsilon}] = \begin{bmatrix} \varepsilon_x & \frac{\gamma_{xy}}{2} & \frac{\gamma_{xz}}{2} \\ \frac{\gamma_{yx}}{2} & \varepsilon_y & \frac{\gamma_{yz}}{2} \\ \frac{\gamma_{zx}}{2} & \frac{\gamma_{zy}}{2} & \varepsilon_z \end{bmatrix} \quad (2.2)$$

où $\varepsilon_i = \frac{\delta u_i}{\delta x_i}$, différentiel de \mathbf{u} sur l'axe i , est la partie linéaire de la déformation et $\gamma_{ij} = \frac{\delta u_i}{\delta x_j} + \frac{\delta u_j}{\delta x_i}$ la partie angulaire.

Soit $[\mathbf{C}^{el}]$ un tenseur d'ordre 4 (81 coefficients réduits à 21 en tenant compte des symétries $C_{ijkl}^{el} = C_{klij}^{el} = C_{jikl}^{el} = C_{ijlk}^{el}$) définissant les propriétés élastiques du matériau, on peut écrire la *loi de Hooke* reliant le *tenseur de contrainte* $[\boldsymbol{\sigma}]$ et le *tenseur de déformation* $[\boldsymbol{\varepsilon}]$ sous sa forme généralisée, en tenant compte des matériaux anisotropes (cf 2.1.4) :

$$\sigma_{ij} = \sum_{k=1}^3 \sum_{l=1}^3 C_{ijkl} \cdot \varepsilon_{kl} \text{ ou sous forme matricielle } [\boldsymbol{\sigma}] = [\mathbf{C}][\boldsymbol{\varepsilon}] \quad (2.3)$$

Si l'on ne considère que les matériaux isotropes, l'équation 2.3 peut s'écrire sous la forme :

$$\sigma_{ij} = \frac{E}{1 + \nu} \left(\varepsilon_{ij} + \frac{\nu}{1 - 2\nu} \varepsilon_{kk} \delta_{ij} \right) \quad (2.4)$$

où ε_{kk} est la somme des termes diagonaux du *tenseur de déformation* ($\varepsilon_{kk} = \varepsilon_x + \varepsilon_y + \varepsilon_z$, aussi appelé *trace du tenseur*), δ_{ij} le *symbole de Kronecker*, E est le **module de Young** (ou *module d'élasticité*) et ν le **coefficient de Poisson**.

Au cours de cette thèse, nous nous proposons de retrouver ces deux derniers paramètres. En effet, ils sont utilisés par les lois les plus courantes dans les diverses implémentations en temps interactif. De plus, comme nous le verrons par la suite lors de l'explication de la méthode d'*estimation inverse*, nous ne pouvons nous permettre d'employer des méthodes non-linéaires (cf 2.2.4). En effet, bien que nous n'ayons pas réellement besoin d'une méthode temps-réel, la complexité algorithmique devra rester raisonnable. Ces deux paramètres ayant une explication physique concrète, nous les détaillerons plus tard dans la section 2.4.

2.2.3 Viscoélasticité linéaire

Comme nous l'avons vu en 2.1.2, le caractère *viscoélastique* d'un matériau est un phénomène temporel, que l'on ne peut observer que sur une période de la déformation. Nous verrons qu'au cours de ces travaux, nous ne nous attachons pas à caractériser ces propriétés viscoélastiques. Cependant, bon nombre de travaux tentent de les modéliser et nous allons ici en voir les formulations les plus connues.

Ces formulations tiennent compte de la *viscoélasticité linéaire* uniquement. Tout comme pour l'*élasticité linéaire*, on peut faire l'hypothèse d'une linéarité spatiale du comportement viscoélastique dans le cadre de *petites déformations*. Dans ce cas, on peut le représenter par une *équation de Volterra*, séparable pour les phénomènes de *fluage* et de *relaxation*, qui relie les tenseurs de contrainte et de déformation au cours du temps :

$$[\boldsymbol{\sigma}](t) = E_{inst,relax}[\boldsymbol{\varepsilon}](t) + \int_0^t \mathbf{Relax}(\mathbf{t}-\mathbf{t}')[\boldsymbol{\varepsilon}](t')dt' \quad (2.5)$$

où $[\boldsymbol{\sigma}](t)$ est le *tenseur de contrainte*, $[\boldsymbol{\varepsilon}](t)$ le *tenseur de déformation*, $E_{inst,relax}$ est le *module d'élasticité* instantanée pour la relaxation et $\mathbf{Relax}(\mathbf{t})$ la *fonction de relaxation*. On peut inverser la relation en définissant $E_{inst,fluage}$ le *module d'élasticité* instantanée pour le fluage et $\mathbf{Flu}(\mathbf{t})$ la *fonction de fluage* :

$$[\boldsymbol{\varepsilon}](t) = \frac{[\boldsymbol{\sigma}](t)}{E_{inst,fluage}} + \int_0^t \mathbf{Flu}(\mathbf{t}-\mathbf{t}')[\boldsymbol{\sigma}](t')dt' \quad (2.6)$$

Plusieurs modèles sont proposés pour ce comportement. Ils sont constitués d'un assemblage de ressorts et d'amortisseurs. Nous ne nous attardons pas ici à décrire ces modèles en détail mais nous contentons de citer le *modèle de Maxwell* (un amortisseur et un ressort en série), le *modèle Kelvin-Voigt* (un amortisseur et un ressort en parallèle), le *modèle linéaire standard* (un amortisseur et un ressort en série le tout en parallèle avec un autre ressort) et le *modèle de Maxwell généralisé* ou *Maxwell-Weichert* (n couples parallèles de ressort/amortisseurs en série plus un autre ressort en parallèle). Ces modèles sont représentés sur la figure 2.7.

1. <http://en.wikipedia.org/wiki/Viscoelasticity>

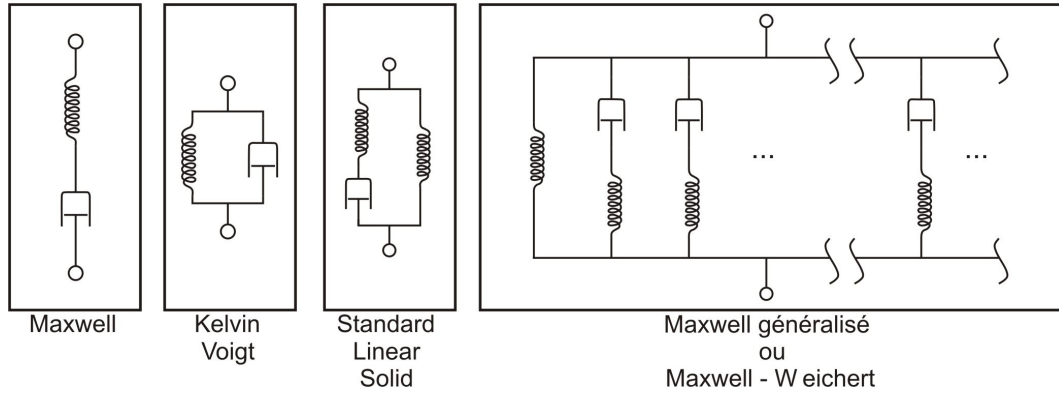


Figure 2.7 - Schémas des différents modèles mécaniques de viscoélasticité¹

2.2.4 Élasticité non-linéaire

Il existe bon nombre de formulations des lois de comportement et des tenseurs correspondants pour formuler l'*élasticité non-linéaire*. Nous ne retiendrons ici que deux formulations du tenseur de contrainte. Tout d'abord, celle du *modèle Neo-Hookean*, extension de la *loi de Hooke* en *élasticité linéaire*, qui suppose que les contraintes en dehors du champ linéaire sont proportionnelles au *tenseur de Finger* noté $[B]$. Soit $[P]$ la *matrice de pression*, $[I]$ la *matrice identité* et $[G]$ le *module de cisaillement*, on peut récrire le tenseur de contraintes :

$$[\sigma_{NH}] = -[P][I] + [G][B] \quad (2.7)$$

Cette formulation est généralement utilisée pour les *matériaux incompressibles*². Outre le fait que sa complexité soit supérieure à celle des équations 2.3 et 2.4, cette particularité la rend inadaptée aux objectifs de cette thèse. En effet, nous désirons caractériser des propriétés mécaniques aussi bien sur des matériaux compressibles qu'incompressibles.

Une généralisation de ce modèle est connue sous le nom de *modèle de Mooney-Rivlin* :

$$[\sigma_{MR}] = -[P][I] + 2C_1[B] + 2C_2[B]^{-1} \quad (2.8)$$

où C_1 et C_2 sont des variables empiriques du modèle (si $C_1 = \frac{1}{2}G$ et $C_2 = 0$, alors $[\sigma_{MR}] = [\sigma_{NH}]$, *modèle Neo-Hookean*). Ce modèle donne généralement de meilleurs résultats que le *modèle Neo-Hookean* (cf fig. 2.8). Cependant, le principal désavantage est le manque de signification physique concrète des constantes C_1 et C_2 .

De manière générale, ces formulations élastiques non-linéaires sont trop coûteuses pour être résolues numériquement en temps réel, particulièrement pour les cas en trois dimensions qui sont utilisés des cette thèse. Cependant, nous verrons dans la section suivante que certaines approches permettent une approximation de ces comportements non-linéaires.

2.3 Simulation numérique

Les phénomènes physiques décrits dans la section 2.1 sont observables dans de nombreuses applications. On peut les rencontrer par exemple dans la construction automobile et aéronau-

2. Les objets composés d'un matériau incompressible gardent un volume constant au cours de la déformation

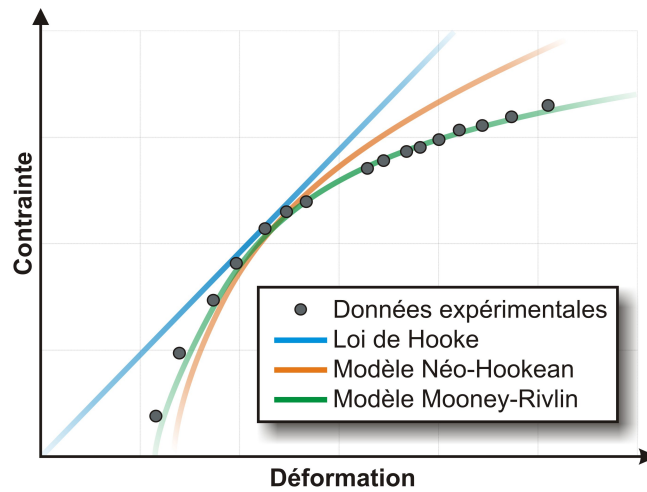


Figure 2.8 - Prédictions de la loi de Hooke (eq. 2.4), du modèle Neo-Hookean (eq. 2.7) et du modèle de Mooney-Rivlin (eq. 2.8) pour des mesures obtenues lors d'un essai en traction/compression.

tique (crash-tests, conception d'avions, ...) ou dans le génie civil (test de résistance des ponts aux forces naturelles par exemple). Mais on peut également avoir besoin de les prendre en compte pour des objets bien moins rigides comme des organes humains lors de *simulations interactives* d'opérations chirurgicales.

La diversité de ces applications engendre des besoins différents. En effet, certaines auront besoin d'une simulation physique très rigoureuse. Au contraire, d'autres auront besoin d'un temps de calcul beaucoup plus rapide, bien souvent au détriment de la qualité de simulation.

Dans un premier temps, nous illustrerons les méthodes généralement utilisées pour réaliser des *simulations en temps-interactif* par la méthode des *masses-ressort*.

Nous décrirons ensuite la méthode de *résolution par éléments finis*. Nous verrons tout d'abord comment elle est utilisée pour résoudre des problèmes suivant les *lois de déformations linéaires* puis des *lois de déformations non-linéaires*. Nous mettrons en évidence son utilisation dans des cas *statiques* puis *dynamiques*.

Enfin, nous introduirons les deux paramètres d'élasticité linéaire que nous nous proposons de retrouver dans cette thèse : le *module de Young* et le *coefficient de Poisson* ainsi que les raisons qui ont motivé ce choix.

2.3.1 Simulation en temps interactif

Depuis plusieurs années, l'évolution de la puissance de calcul a permis d'envisager la création de simulations d'objets déformables en temps interactif. Il s'agit de capturer les mouvements de l'utilisateur sur le système à l'aide d'un dispositif spécifique ([[Immersion Corporation](#)] [[Vicon](#)] par exemple), de les intégrer au calcul de la simulation des objets déformables et de résoudre le système avant d'envoyer le résultat vers l'utilisateur. Ce résultat peut prendre plusieurs formes. Il s'agit principalement de l'affichage des objets déformés et de sons d'interaction (ces sons peuvent être simulés physiquement [[van den Doel 01](#)] ou encore pré-enregistrés, par exemple un son mat lors d'un contact avec un objet assez rigide comme un os maxillaire en chirurgie dentaire). Mais d'autres méthodes complémentaires ont vu le jour.

En effet, de nouveaux dispositifs dits *haptiques*, permettent une immersion plus importante en générant un *retour d'effort* vers l'utilisateur [technologies].

On trouve l'un des domaines les plus actifs dans la *simulation interactive* en médecine. En effet, de nombreux travaux concernent la simulation destinée à l'apprentissage du geste chirurgical. La figure 2.9 illustre le principe global de la *simulation interactive*, appliquée ici à l'entraînement des chirurgiens.

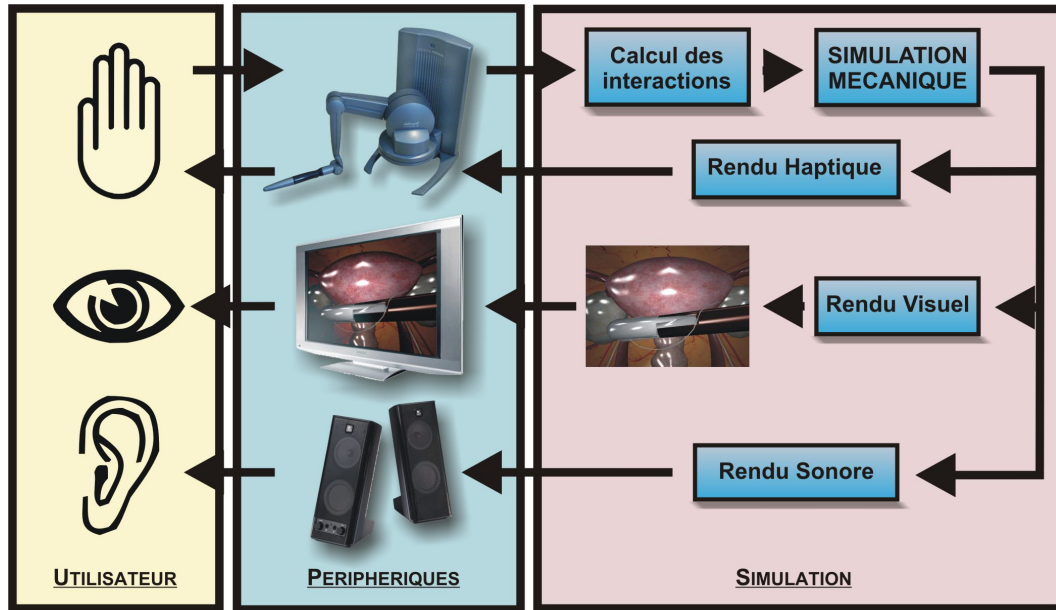


Figure 2.9 - Schéma d'une simulation interactive avec retour haptique, visuel et sonore

Pour être réaliste, le temps de réponse d'un tel système doit être très court. Cependant, la puissance de calcul actuelle est insuffisante pour générer tous les comportements mécaniques d'objets aussi complexes que des organes de manière physiquement rigoureuse. C'est pourquoi nombre de travaux tentent de trouver le meilleur compromis entre temps de calcul et précision.

2.3.1.1 Modèles masses-ressorts

Un modèle de type *masses-ressorts* consiste à utiliser un ensemble de particules reliées entre elles par des ressorts. Dans le cas le plus simple, chaque particule a une masse m donnée et chaque ressort une raideur k et une longueur initiale l_0 . Une particule \mathbf{p}_i est alors reliée à des particules dites « voisines » définies par $\mathbf{p}_j \forall j \in \mathbf{V}_i$. On décrit donc le mouvement de chaque particule \mathbf{p}_i par l'équation dynamique :

$$m\ddot{\mathbf{p}}_i = -\beta\dot{\mathbf{p}}_i - \sum_{j \in \mathbf{V}_i} k^{ij} \left(l_0^{ij} - |\mathbf{p}_i \mathbf{p}_j| \right) \frac{\mathbf{p}_i \mathbf{p}_j}{|\mathbf{p}_i \mathbf{p}_j|} \quad (2.9)$$

où k^{ij} est la raideur du ressort reliant les particules i et j , l_0^{ij} la longueur initiale de ce ressort et β l'amortissement global du système.

Comme on peut le voir sur cette équation, le mouvement d'une particule dépend uniquement de ses voisins directs. Il est donc très simple d'implémenter un tel système. C'est pour

cette raison que cette technique est très répandue, que ce soit en simulation surfacique (pour des tissus par exemple [Provot 95] [Provot 97]) ou volumique [Meseure 02] [Lenoir 02].

En général, il s'agit d'utiliser une méthode de résolution locale : on itère sur chaque particule en utilisant la position des voisins au pas de temps précédent. Cette méthode de résolution est donc *explicite*. Le principal avantage de cet algorithme est sa rapidité (de complexité linéaire, $\mathcal{O}(n)$ pour n particules). De plus, il est très simple de modifier la topologie de l'objet en supprimant des ressorts pour simuler la découpe de l'objet.

Cependant, ce genre de résolution introduit de manière générale des instabilités dans le maillage qui se répercutent sur les itérations suivantes. La parade à ce type de problèmes est d'augmenter l'*amortissement global* β du système pour atténuer la création de forces erronées (ce qui revient à augmenter la *dissipation d'énergie*). Malheureusement, le comportement obtenu ne peut être considéré comme représentatif de la réalité, bien qu'il soit perceptuellement valide. Certains travaux tentent une approche de résolution *implicite* [Meseure 02], mais la complexité du calcul augmente grandement, particulièrement pour des objets volumiques.

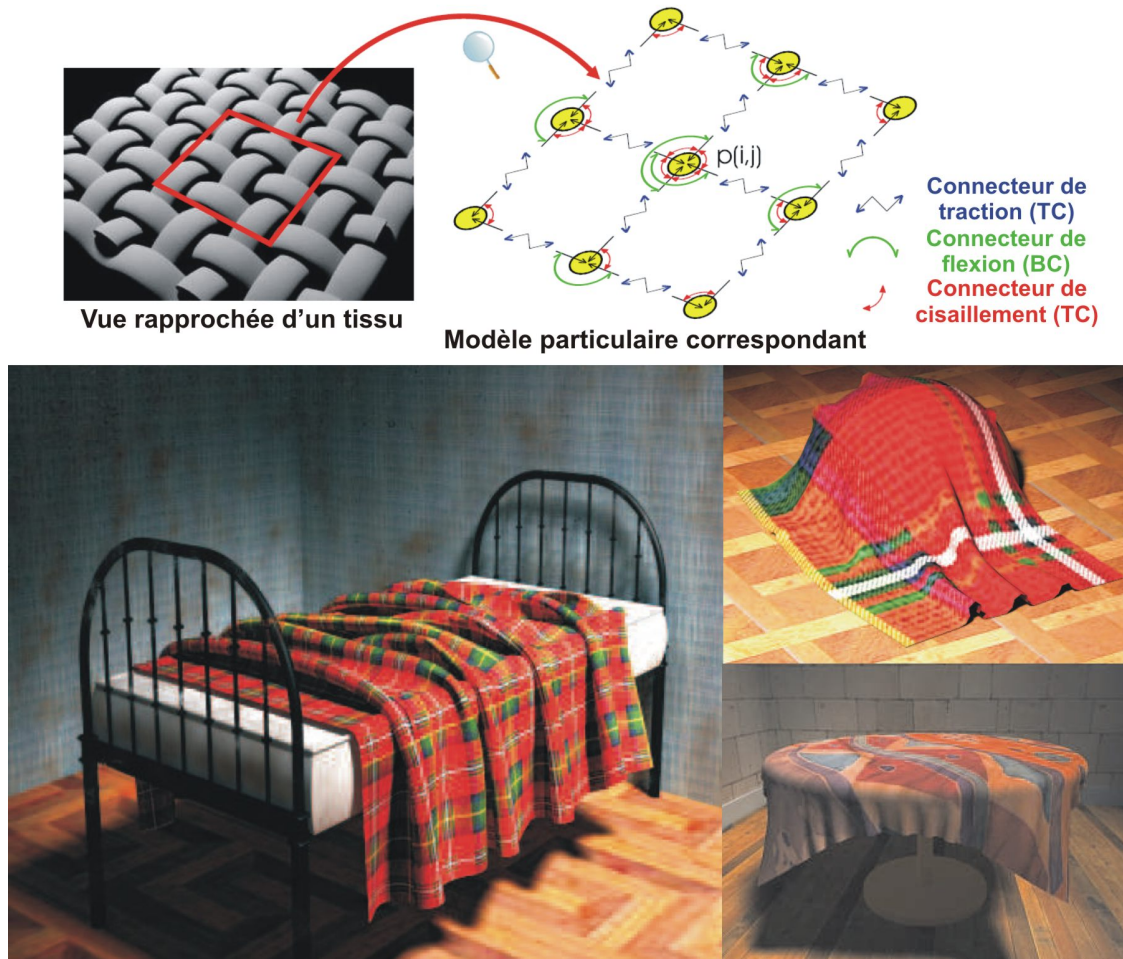


Figure 2.10 - Comportement réaliste de tissus à base de méthode masses-ressort évoluée [NgoNgoc 04].

De nombreuses extensions de cette méthode ont vu le jour pour améliorer le comportement des systèmes masses-ressorts. Par exemple, Ngo Ngoc et Boivin [NgoNgoc 04] définissent un

modèle mécanique de tissus basé sur les masses/ressorts et y introduisent des comportements mécaniques plus complexes que ceux décrits par l'équation 2.9. Ainsi, plusieurs types de ressorts, ici des connecteurs, sont définis (traction, cisaillement, flexion) et leurs raideurs k ainsi que leur amortissement β ne sont plus des constantes mais des fonctions à plusieurs paramètres. Cette méthode permet en outre d'utiliser les mesures faites sur des tissus réels à l'aide de la machine de Kawabata [Kawabata 80] et d'obtenir ainsi des comportements de tissus très réalistes (cf Fig. 2.10).



Figure 2.11 - Illustration du problème de configuration du maillage dans les systèmes masse-ressort (cas surfacique). Un treillis simple (a) ne génère pas le même comportement qu'un treillis croisé (b). En effet, même en diminuant la rigidité des ressorts, la simulation (b) semble moins flexible. Cette non-généricité géométrique rend l'identification très complexe lorsque les paramètres de chaque ressort doivent être évalués.

Cependant, dans les cas plus généraux comme pour les objets déformables volumiques, il est très difficile de connaître les liaisons adéquates entre les ressorts et leur voisinage. En effet, le comportement de la simulation est très différent pour des configurations différentes, même si la position des particules à l'état de repos reste la même. Pour des valeurs de raideurs identiques, on peut alors obtenir des comportements totalement différents. La figure 2.11 illustre ce problème pour des objets surfaciques de type textile (modèle mécanique simple de l'équation 2.9).

Il s'agit alors de tester de nombreuses configurations différentes pour obtenir le mouvement désiré. Le problème se complexifie encore lorsqu'il est nécessaire de choisir des rigidités différentes pour chaque ressort. Certaines approches tentent de les approximer [Louchet 95] [Joukhadar 97] [Bianchi 04] [Burion 08]. Dans ce cas, le nombre de paramètres dépend du nombre de particules et de ressorts, ce qui rend l'identification de chacun très complexe. Certaines méthodes essaient également de retrouver ces différents paramètres par comparaison avec des méthodes par *élément finis* [Choi 04] que nous allons maintenant détailler.

2.3.2 Simulation par éléments finis

Cette technique par *éléments finis* est très utilisée dans la simulation mécanique d'objets déformables pour le calcul scientifique (*off-line*, ou non-interactif). En effet, bon nombre de logiciels professionnels dédiés à la conception d'ouvrages d'art civils, comme des ponts ou des routes, bénéficient d'une implémentation de cette technique de calcul puissante et robuste. Sa modularité permet de tester les divers éléments modélisés sous des contraintes extrêmes. On citera par exemple les logiciels de tests virtuels pour s'assurer de la résistance des ponts sous diverses contraintes (passage des camions, forces naturelles, etc... , cf Fig. 2.12), ou encore la solidité des matériaux et assemblages mécaniques dans le domaine aéronautique ou le secteur automobile (Fig. 2.13).

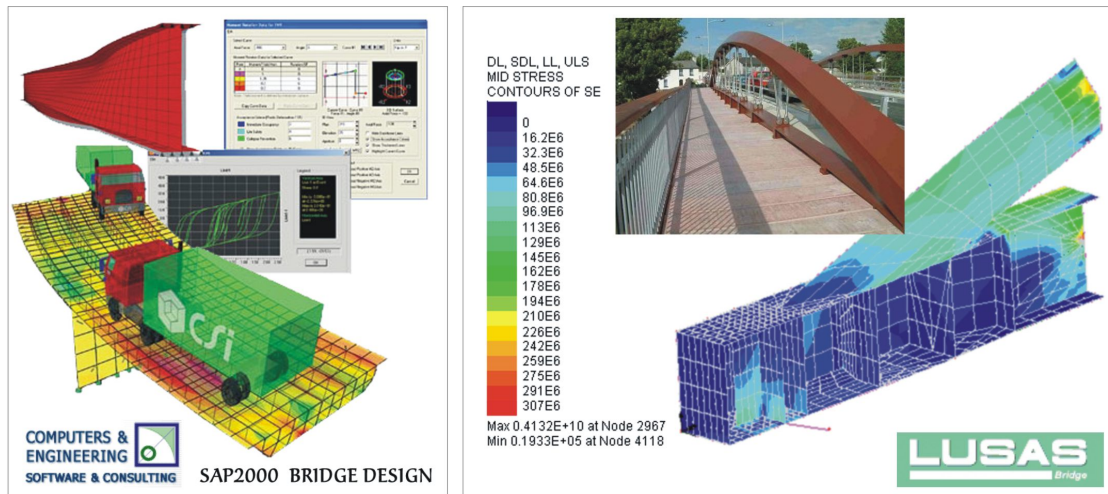


Figure 2.12 - Utilisation des éléments finis dans le génie civil. À gauche, quelques captures d'écran illustrent la conception d'un pont à l'aide du logiciel SAP 2000 : Bridge Design (COMPUTERS & ENGINEERING)⁵. On peut voir notamment un test des contraintes subies lors du passage de camions. À droite, un autre logiciel dédié à la conception de ponts est illustré par un test de contraintes sur l'arche du nouveau pont Watergate Bridge au dessus de la River Boyne à Trim, County Meath, Irlande (LUSAS Bridge, réalisé par Roughan & O'Donovan)⁶.

De plus, la méthode de résolution par éléments finis bénéficie depuis quelques années d'un grand nombre de travaux pour l'optimiser et l'adapter à la résolution de problèmes interactifs. Cette méthode est donc devenue incontournable dans la résolution des problèmes posés en mécanique des milieux continus.

2.3.2.1 Description

Le but des *éléments finis* est de résoudre de manière discrète une *équation aux dérivées partielles*, comme par exemple l'étude des déformations d'un objet continu.

La méthode consiste à discrétiser le domaine en petits éléments. On pourra utiliser par exemple des triangles ou encore des rectangles dans un cas en deux dimensions et des tétraèdres ou des cubes dans un cas tri-dimensionnel pour approximer la géométrie d'un objet continu. Par convention, nous définissons une constante *dim* qui représente la *dimension du problème* (2 en deux dimensions, 3 en trois dimensions, ...) et t_e le nombre de noeuds d'un élément (3 pour un triangle, 4 pour un tétraèdre, ...).

Ensuite, on définit pour chacun des éléments qui constitue la discrétisation de l'objet, une *matrice élémentaire* notée $[K_e]$. Cette matrice, de taille $(dim \times t_e)^2$ est définie comme suit :

$$[K_e] = [B] \cdot [C][B]^T \quad (2.10)$$

5. http://www.comp-engineering.com/announce/bridge/bridge_en.html

6. <http://www.lusas.com/case/bridge/index.html>

11. <http://www.hks.com/>

12. http://en.wikipedia.org/wiki/Finite_element_analysis

13. <http://www.3ds.com/>

14. <http://www.am-inc.com/software.shtml>

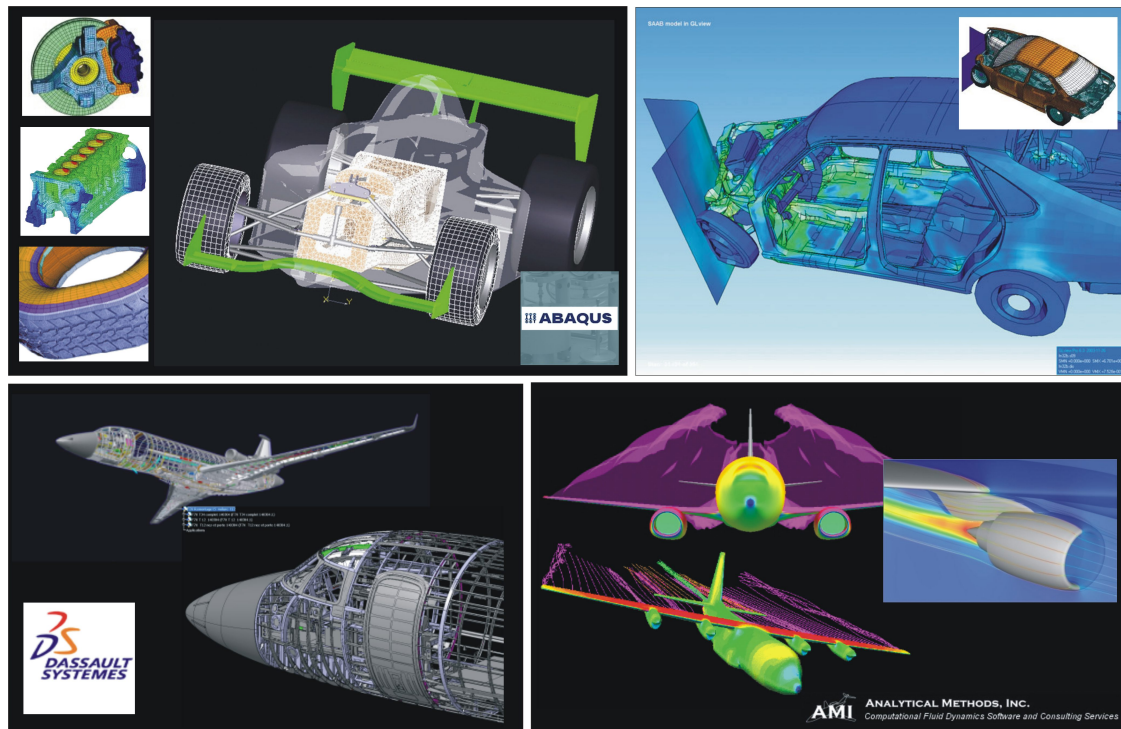


Figure 2.13 - Utilisation de la méthode des éléments finis dans les secteurs automobiles et aéronautiques. En haut à gauche : Modélisation et simulation automobile sous ABAQUS ¹¹. En haut à droite : Simulation d'un crash test pour une SAAB, visualisation sous GLview ¹². En bas à gauche : simulation aérodynamiques pour divers appareils avec les solutions Dassault Systems ¹³. En bas à droite : Modélisations aéronautiques, Analytical Methods, Inc. ¹⁴.

où $[C]$, construite à partir des paramètres de la *loi de comportement* utilisée et des tenseurs correspondants (cf. 2.2), représente le matériau et $[B]$ traduit le déplacement des sommets de l'élément pour passer de l'*élément de référence* (ex. un triangle isocèle en deux dimensions) à sa forme actuelle (cf Fig 2.14).

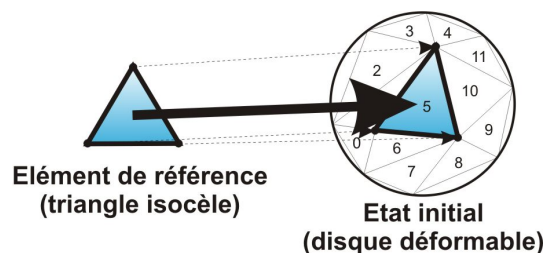


Figure 2.14 - Illustration de la construction des matrices élémentaires depuis un élément de référence en deux dimensions : Un cercle est discrétisé par 12 éléments. Le passage de l'élément de référence à l'élément 5 se fait par la matrice B_5 . Ensuite, on construit $[K_{e5}] = [B_5] \cdot [C_5] \cdot [B_5]^T$. Cette opération est répétée pour chaque élément.

Évidemment, certains éléments ont des sommets communs. Ainsi, la déformation d'un élément implique la déformation d'autres éléments. Pour modéliser ces interactions, on procède à un assemblage de toutes ces matrices élémentaires dans une matrice globale de taille $(dim \times Nb_{noeuds})^2$, nommée *matrice de rigidité globale* et notée $[K]$. Pour ce faire, on part

d'une matrice nulle puis on procède à l'opération itérative suivante pour chaque élément :

$$\begin{aligned} [K_{(L,M)}] &= [K_{(L,M)}] + [K_e (i+t_e(k-1), j+t_e(l-1))] \\ L &= k + \dim(\mathbf{Ind}(\mathbf{i}) - 1) \\ M &= l + \dim(\mathbf{Ind}(\mathbf{j}) - 1) \end{aligned} \quad (2.11)$$

où $k, l \in [1 \dots \dim]$ sont les indices du nœud dans la matrice $[K_e]$, $i, j \in [1 \dots t_e]$ sont les indices du nœud dans la matrice $[K]$, et $\mathbf{Ind}(\mathbf{i})$ est l'indice du i^{eme} sommet de l'élément dans le tableau des sommets (cf illustration Fig. 2.15).

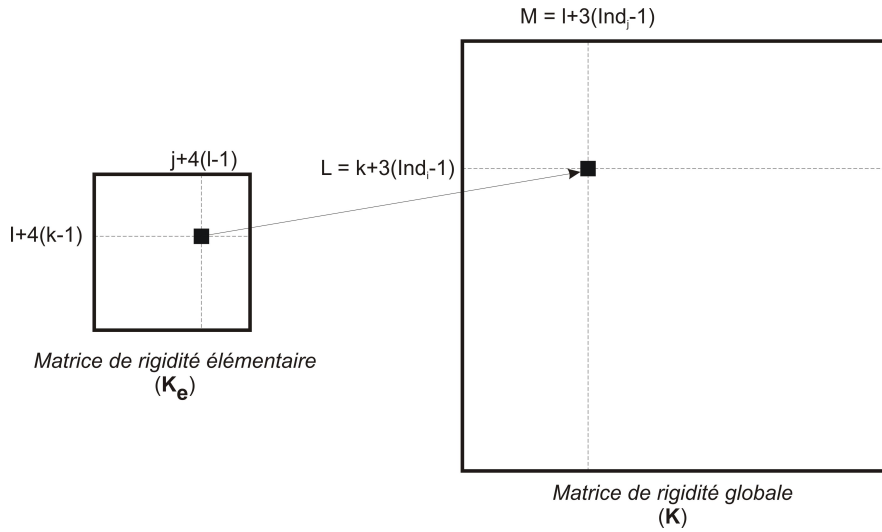


Figure 2.15 - Illustration de la méthode d'assemblage de la matrice de rigidité globale $[K]$ (pour un problème en trois dimensions avec des tétraèdres), d'après [Cotin 97].

Dans la plupart des cas, un élément n'est en relation qu'avec quelques autres éléments. La matrice $[K]$ est donc souvent une *matrice creuse à bande* (et symétrique).

Nous désirons utiliser la méthode des *éléments finis* pour résoudre des problèmes liés à la déformation d'objets. La *déformation géométrique* d'un objet discrétisé est caractérisée par le déplacement de chacun des sommets qui le compose, noté \mathbf{u}_i pour chaque sommet i . \mathbf{u}_i est constitué de \dim composantes. La position du sommet i après déformation est alors $\mathbf{x}_i^{def} = \mathbf{x}_i^0 + \mathbf{u}_i$ (\mathbf{x}_i^0 est la position initiale du sommet). On définit alors le vecteur \mathbf{U} , de taille $\dim \times Nb_{noeuds}$, qui caractérise le déplacement géométrique des sommets constituant la déformation globale de l'objet :

$$\mathbf{U} = [\mathbf{u}_0 \mathbf{u}_1 \mathbf{u}_2 \dots \mathbf{u}_{Nb_{noeuds}-1}] \quad (2.12)$$

De la même manière, on peut définir le vecteur \mathbf{F} , de taille identique à \mathbf{U} , qui correspond à toutes les forces \mathbf{f}_i appliquées sur chaque nœud du maillage :

$$\mathbf{F} = [\mathbf{f}_0 \mathbf{f}_1 \mathbf{f}_2 \dots \mathbf{f}_{Nb_{noeuds}-1}] \quad (2.13)$$

Ces définitions de $[K]$, $[U]$ et $[F]$ permettent de linéariser le problème de la déformation. Ce problème peut être formulé de différentes manières selon le mode de résolution souhaité : *statique* ou *dynamique*.

2.3.2.2 Simulation statique

La *formulation statique* du problème est utilisée lorsque la simulation des effets définis dans le temps n'est pas nécessaire.

Il existe plusieurs systèmes d'équations dépendants du type de déformation désiré et des hypothèses formulées pour simplifier ces systèmes et leurs résolutions :

Déformations linéaires Nous avons vu que la formulation de la matrice $[K]$ dépend de la position des sommets du maillages. Si ces sommets se déplacent selon le vecteur U , $[K]$ est dépendante de ce mouvement ($[K(U)]$). Or, dans l'hypothèse de *petites déformations*, on peut s'affranchir de la dépendance de la matrice $[K]$ aux déplacements U . Ainsi, cette matrice ne dépend que de la topologie initiale. On cherche alors l'*état d'équilibre* du système que l'on peut représenter par la formulation matricielle :

$$F = [K]U \quad (2.14)$$

Il s'agit donc ici de résoudre ce système pour l'inconnue U . De nombreuses méthodes sont ici applicables. Parmi celles-ci, on peut noter l'*inversion de la matrice* $[K]$ pour obtenir directement le vecteur U :

$$U = [K^{-1}]F \quad (2.15)$$

Lorsque l'on veut faire une animation composée de suites d'états statiques, on peut, dans le cas de petits déplacements, précalculer $[K^{-1}]$ puisque $[K]$ est supposée invariable. Cependant, lors d'une *formulation statique* par éléments finis pour un objet continu, les objets n'ont pas de masse. Autrement dit, si on applique une force sur le système, U est infini. Mathématiquement, cette condition se traduit par le fait que $[K]$ n'est pas inversible. En effet, pour cela, le rang de cette matrice doit-être égal au nombre de nœuds n . Le système n'est donc pas encore défini et il est nécessaire d'ajouter des *contraintes de points fixes*, traduit par des modifications de $[K]$, pour obtenir une matrice inversible (cf. 2.3.2.4).

On peut également citer d'autres méthodes de résolution directe comme une décomposition $[L][U]$ ou $[Q][R]$ de $[K]$. A ces solutions viennent s'ajouter les méthodes itératives par relaxation (*Jacobi*, *Gauss-Seidel*) ou par projection (gradient conjugué, Lanczos).

De nombreuses optimisations et formulations ont vu le jour. Par exemple, Bro-Nielsen et Cotin [Bro-Nielsen 96] suggèrent de séparer, dans l'équation 2.14 les informations relatives aux nœuds de surface et aux nœuds internes à l'objet :

$$\begin{bmatrix} [K_{ss}] & [K_{si}] \\ [K_{is}] & [K_{ii}] \end{bmatrix} \begin{bmatrix} U_s \\ U_i \end{bmatrix} = \begin{bmatrix} F_s \\ F_i \end{bmatrix}$$

soit, en substituant U_i

$$[K_{ss} - K_{si}K_{ii}^{-1}K_{is}] U_s = F_s - [K_{si}K_{ii}^{-1}] F_i \quad (2.16)$$

avec U_s , U_i les déplacements, et F_s , F_i les forces extérieures respectivement sur les nœuds de surface et sur les nœuds internes. Les auteurs ne cherchent alors à imposer les forces appliquées à leur modèle que sur les nœuds de surface (soit $F_i = 0$) et donc ramener le

modèle aux nœuds de surface uniquement, ce qui accélère considérablement le processus de résolution.

Déformations non-linéaires Dans le cas linéaire, nous faisons la supposition que la *matrice de rigidité* $[K]$ est indépendante des mouvements des nœuds U , ce qui n'est valide que pour de petites déformations. Si l'on ne tient pas compte de cette hypothèse, la formulation devient non-linéaire :

$$F = [K(U)]U \quad (2.17)$$

Pour résoudre ce système, on traduit généralement le problème non-linéaire par une succession itérative de problèmes linéaires (*Broyden* ou *Newton-Raphson* par exemple). Cependant, comme nous l'avons déjà évoqué, de telles techniques sont très coûteuses en temps de calcul, particulièrement dans le cas tri-dimensionnel dès que le nombre d'éléments devient élevé, et donc non adaptées à des *simulations interactives*.

La taille des maillages est un facteur important pour la précision du résultat de la simulation. Or, la complexité algorithmique est directement liée au nombre de nœuds et d'éléments du maillage. Conjugée à la complexité des méthodes de résolution non-linéaire, le temps de calcul total devient alors très vite énorme, tout particulièrement lorsque la dimension du problème augmente. Au cours de nos travaux, nous voulons rendre raisonnable ce temps de calcul pour que l'*identification* d'un jeu de paramètres soit réalisable en quelques heures au maximum sur une machine seule (ici Bi-processeur Pentium 4 3.2GHz et 2Go de RAM). Par conséquent, nous n'utilisons pas la *formulation non-linéaire*. Nous avons néanmoins constaté que les matériaux que nous traitons montraient ce type de comportement, particulièrement lors de grands déplacements. Nous avons donc choisi une approche plus récente : la *formulation corotationnelle*.

Modèle corotationnel Initialement, la méthode des *éléments finis* était réduite au domaine des *petits déplacements* (*tenseur de déformation* linéaire $[\epsilon]$) et des *petites déformations* (*tenseur de contrainte* linéaire $[\sigma]$). Le principal problème rencontré dans ces cas est qu'une partie de la rotation des objets est interprétée comme une déformation. Cela entraîne la production de forces non désirées et les déformations obtenues sont rapidement incorrectes. On peut voir par exemple un changement de volume aberrant (cf Fig 2.16).

Pour résoudre ce problème, des solutions ont été proposées. Il s'agit d'intégrer les rotations directement dans le modèle, sous forme de *mouvement rigide*. Il existe deux approches. La première, globale, consiste à donner à l'objet total un mouvement rigide qui n'induit aucune déformation. Les déformations restent alors locales et en *petits déplacements*. Il devient alors très simple de déplacer l'objet dans l'espace.

La méthode locale proposée par Felippa [Felippa 00] permet d'étendre cette approche au niveau de chaque élément. La non-linéarité des déplacements est principalement due aux rotations des éléments. Le *modèle corotationnel* propose alors d'extraire le calcul de la rotation de chaque élément $[R_e]$ de celui de la déformation réelle subie par chaque élément autour de cette valeur de rotation (cf Fig. 2.17). Soit, $[K_e]$ la *matrice élémentaire* d'un élément e et δU_e le petit déplacement de cet élément, on peut alors supposer :

$$F_{[K_e]}(\delta U_e) \approx ([R_e][K_e][R_e]^T)\delta U_e \quad (2.18)$$

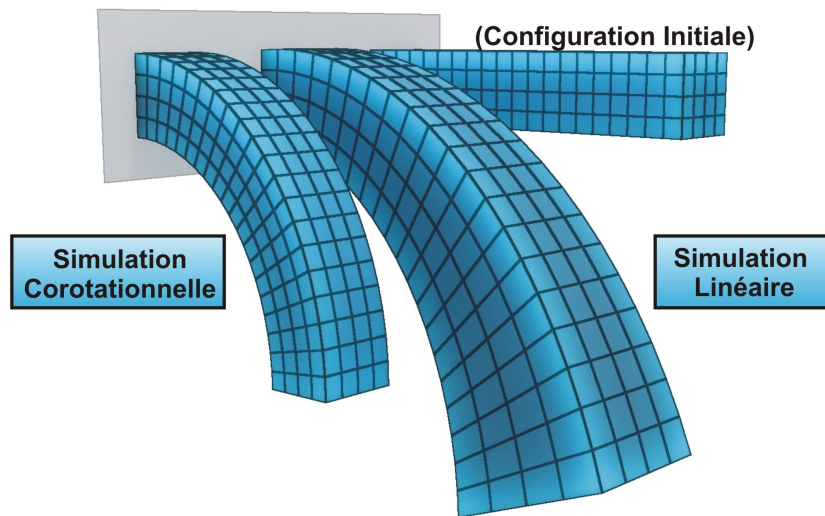


Figure 2.16 - Illustration du problème des grands déplacements en simulation linéaire par un exemple de poutres encastées. La simulation corotationnelle résout ce problème. Pour se faire une idée des déplacements en jeu, la configuration initiale de la poutre est donnée.

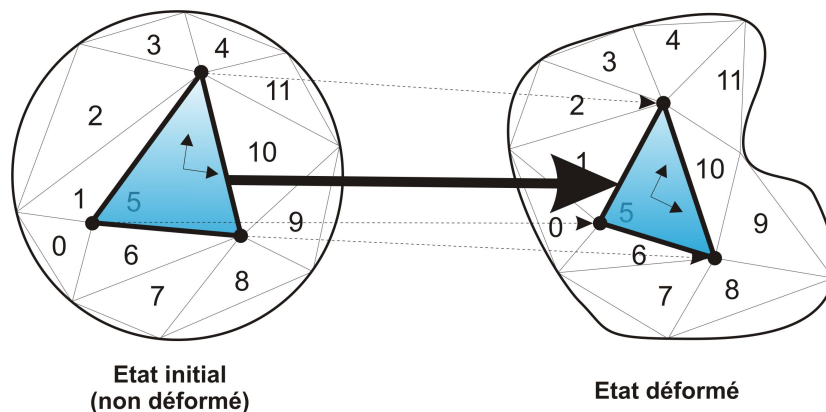


Figure 2.17 - Illustration de la décomposition d'une transformation élémentaire : L'élément 5 passe de l'état initial à l'état déformé en deux phases. La matrice $[R_5]$ traduit tout d'abord le mouvement rigide qui permet de passer du repère O_{5init} au repère O_{5def} . La déformation effective de l'élément peut alors être exprimée localement autour de ce repère et donc rester dans le cadre des petites déformations.

Le comportement global de l'objet devient donc quasiment non-linéaire bien que les déformations locales restent linéaires. On qualifie également cette technique de *grands déplacements* ou de *grandes rotations*. Le temps de calcul est alors bien plus raisonnable en trois dimensions et les déplacements géométriques ont un comportement beaucoup plus réaliste qu'avec une méthode purement linéaire (cf Fig 2.16). Nous avons donc opté pour cette méthode qui nous semble le meilleur compromis entre le temps de calcul et le réalisme de la simulation qui, comme nous le verrons par la suite, est capital pour l'*estimation de paramètres* depuis des séquences d'images réelles.

Pour plus d'informations sur la *formulation corotationnelle*, le lecteur pourra se référer aux travaux de Felippa [Felippa 00].

2.3.2.3 Simulation dynamique

De nombreux comportements des matériaux dépendent du temps. Par exemple, la compression de matériaux très mous, proches de l'état liquide, introduit en réalité des oscillations de la matière (cf. 2.1.2). Ce phénomène étant temporel, la *formulation statique* ne suffit plus. La *formulation dynamique* introduit donc des termes temporels comme la *vitesse* ou l'*accélération*, afin de reproduire les phénomènes transitoires, les oscillations et les propagations d'ondes :

$$\mathbf{F} = [\mathbf{M}]\ddot{\mathbf{U}} + [\mathbf{D}]\dot{\mathbf{U}} + [\mathbf{K}]\mathbf{U} \quad (2.19)$$

avec $[\mathbf{M}]$ la *matrice de masses* et $[\mathbf{D}]$ la *matrice d'amortissement*. La résolution se fait selon le mode d'intégration choisi. Par exemple, dans une *intégration Euler explicite*, on utilise les déplacements du pas de temps précédent ($t - 1$) :

$$\mathbf{F}_t = [\mathbf{M}]\ddot{\mathbf{U}}_t + \left[[\mathbf{D}](\dot{\mathbf{U}}_{t-1} + \ddot{\mathbf{U}}_{t-1}dt) + [\mathbf{K}](\mathbf{U}_{t-1} + \dot{\mathbf{U}}_{t-1}dt) \right] \quad (2.20)$$

Le problème de cette formulation est que les erreurs d'approximation ont tendance à s'ajouter au fil des itérations. Il est par conséquent nécessaire de réduire au maximum le pas de temps afin d'avoir une précision optimale. Cependant, diminuer ce pas de temps revient à ralentir considérablement la simulation.

Les approches implicites ont quant à elles le raisonnement inverse : faire un calcul plus complexe mais plus précis pour pouvoir augmenter ce pas de temps. Par exemple, le schéma d'*intégration Euler implicite* pour une meilleure prédiction :

$$\mathbf{F}_t = \left(\frac{[\mathbf{M}]}{dt^2} + \frac{[\mathbf{D}]}{dt} + [\mathbf{K}] \right) \mathbf{U}_t \left[-\frac{[\mathbf{M}]\mathbf{U}_{t-1}}{dt^2} - \frac{[\mathbf{M}]\dot{\mathbf{U}}_{t-1}}{dt} - \frac{[\mathbf{D}]\mathbf{U}_{t-1}}{dt} \right] \quad (2.21)$$

Le choix du type d'intégration dépendra donc grandement du problème à traiter, toujours dans un souci de compromis entre rapidité et précision de la simulation.

2.3.2.4 Contraintes

Un système non contraint est inutilisable pour une simulation mécanique dont le but est de modéliser les comportements du modèle en réaction à des forces externes ou dans un environnement donné. Il faut donc modéliser les *interactions* avec le monde extérieur à l'objet. Il existe plusieurs méthodes pour intégrer ces contraintes à la résolution par *éléments finis*. Nous ne détaillons ici que les contraintes appliquées dans le cas statique bien que ces méthodes soient aussi valides dans le cas dynamique.

Points fixes : Conditions de Dirichlet Nous avons vu que la résolution du système de l'équation 2.14 est impossible sans l'application de contraintes. En effet, par défaut, la matrice K est une matrice de rang $n - 6$. Or, pour être capable de l'inverser, celle-ci doit être de rang n . Autrement dit, toutes les équations linéaires qui composent le système doivent être indépendantes. Pour ce faire, il est possible de fixer des nœuds du maillage en les bloquant en déplacement. On dit alors qu'on leur applique une *condition de Dirichlet*.

Il existe deux méthodes. Tout d'abord, il est possible de modéliser ces contraintes en supprimant pour chaque nœud visé les dim lignes et colonnes correspondantes dans la matrice \mathbf{K} . Les déplacements \mathbf{U} de ces nœuds ne sont alors pas définis. Il est possible de contraindre le déplacement de ces nœuds sur un plan en ne supprimant qu'un couple ligne/colonne ou sur un axe en supprimant deux par nœuds.

La seconde méthode est la méthode des *multiplieurs de Lagrange*. Elle consiste à construire une matrice constituant des conditions dites d'égalité. Cette matrice $[\mathbf{A}_{eq}]$, de taille $dimNb_{noeuds} \times dimNb_{fixes}$ contient une matrice identité $[\mathbf{I}]$ (de taille dim^2) pour chaque nœud fixé. Voici un exemple pour fixer les nœuds 1 et 3 dans un maillage de 4 nœuds :

$$[\mathbf{A}_{eq}] = \begin{bmatrix} [\mathbf{I}] & [\mathbf{0}] & [\mathbf{0}] & [\mathbf{0}] \\ [\mathbf{0}] & [\mathbf{0}] & [\mathbf{I}] & [\mathbf{0}] \end{bmatrix}$$

Le système de l'équation 2.14 s'écrit alors :

$$\begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} [\mathbf{K}] & [\mathbf{A}_{eq}]^T \\ [\mathbf{A}_{eq}] & [\mathbf{0}] \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{\Lambda} \end{bmatrix} \quad (2.22)$$

où $\mathbf{\Lambda}$ est la valeur de la réaction par *multiplieur de Lagrange*.

Dans ces deux cas, l'inversion de $[\mathbf{K}]$ devient possible si l'on ajoute la contrainte fixe pour trois nœuds indépendants (par exemple non alignés pour former un plan et bloquer une rotation dans un sens).

Conditions d'inégalité Contrairement aux conditions de Dirichlet qui visent à supprimer des degrés de liberté pour un certain nombre de nœuds, les *conditions d'inégalités* servent à interdire les nœuds dans une zone définie. Par exemple, on voudra leur imposer de rester au dessus d'un plan sans pour autant les lier à ce plan.

La méthode est semblable à la construction de la matrice $[\mathbf{A}_{eq}]$. On définit pour chaque nœud une valeur b_{noeud} qui sera le déplacement maximal autorisé pour ce nœud dans une direction donnée. Par exemple, pour une *contrainte planaire*, cette direction sera donnée par la normale du plan à ne pas pénétrer. On définira alors $[\mathbf{A}_{ineq}]$ et \mathbf{b}_{ineq} comme suit pour exprimer la contrainte (ici un exemple pour contraindre les nœuds 1 et 3 dans un maillage de 4 nœuds à ne pas passer outre un plan de normale \mathbf{n}) :

$$[\mathbf{A}_{ineq}] = \begin{bmatrix} \mathbf{n} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{n} & \mathbf{0} \end{bmatrix} \text{ et } \mathbf{b}_{ineq} = \begin{bmatrix} b_1 \\ b_3 \end{bmatrix}$$

où b_i est la distance minimale entre le nœud i et le plan. Il faut alors résoudre :

$$\begin{cases} [\mathbf{K}] & \mathbf{U} = \mathbf{F} \\ [\mathbf{A}_{ineq}] & \mathbf{U} \leq \mathbf{b}_{ineq} \\ [\mathbf{A}_{eq}] & \mathbf{U} = \mathbf{0} \text{ s'il existe en outre des conditions de Dirichlet} \end{cases} \quad (2.23)$$

Ce type de système peut être résolu par *programmation quadratique* (QP).

Grâce à ce type de contrainte, il est possible de modéliser des contacts avec un objet extérieur (par exemple le bout d'une sonde). Nous verrons que nous pouvons utiliser cette formulation pour contraindre un objet sur un plan (Dirichlet, contrainte de points fixes) et

pour simuler le contact avec une sphère rigide (inégalité qui interdit aux sommets « d’entrer » dans la sphère).

Cependant, ce contact est pour le moment uniquement de type glissant. Pour prendre en compte le *frottement* dans ces contacts, il est nécessaire d’introduire la *loi de Coulomb* et de formuler le problème dans l’espace des contacts pour obtenir un *problème de complémentarité linéaire* (LCP, voir [Duriez 04] pour plus d’informations sur la résolution de ce type de contact pour des objets déformables).

2.4 Module de Young et Coefficient de Poisson

Nous venons de voir qu’il est possible de modéliser de nombreux comportements à l’aide de la simulation mécanique par éléments finis. Néanmoins, toutes les implémentations ne permettent pas de tous les reproduire, notamment pour la grande majorité des simulations mécaniques en temps réel ou interactif en trois dimensions où généralement l’élasticité linéaire est le seul phénomène reproduit. Cependant, le *module de Young* et le *coefficient de Poisson* sont deux paramètres incontournables et très largement utilisés pour la simulation de matériaux déformables.

Identifier ces paramètres permet d’améliorer significativement le réalisme des comportements des objets simulés. Nous verrons dans le chapitre suivant les travaux qui proposent des méthodes pour tenter de les mesurer ou de les estimer. Cependant, il est nécessaire de bien comprendre leur signification et leur influence sur les déformations et les forces en jeu de manière à déterminer les expérimentations discriminantes que nous aurons à réaliser par la suite.

Contrairement à d’autres paramètres que nous avons évoqués dans les sections précédentes, le *module de Young* et le *coefficient de Poisson* ont une signification physique.

2.4.1 Module de Young

Le *module de Young*, généralement noté E et aussi nommé *module d’élasticité*, est une mesure de la raideur d’un matériau donné en traction/compression (ou *élasticité longitudinale*, car il existe d’autres modules d’élasticité, le *bulk modulus* ou le *shear modulus*).

Il définit le rapport entre la contrainte appliquée au matériau et la déformation qui en résulte, et reste constant tant que cette déformation est petite. Il n’est défini que dans la phase élastique de la déformation (cf 2.1.1).

L’illustration la plus simple est celle du ressort. Si l’on soumet un ressort de longueur au repos l_0 et de raideur k à une force f , on a $f = k(l - l_0)$ où l est la longueur observée après déformation. Dans ce cas simple, $E = k$. Pour une forme quelconque, on divise la force par l’aire de la section de la pièce (s) et on utilise l’*allongement relatif* $\frac{l-l_0}{l_0}$, ce qui donne :

$$\frac{f}{s} = E \cdot \frac{l - l_0}{l_0} \quad (2.24)$$

Lorsque l’on trace la courbe déformation/contrainte d’un essai en traction ou compression, le *module de Young* est représenté par la pente de la partie linéaire (cf. Figure 2.18).

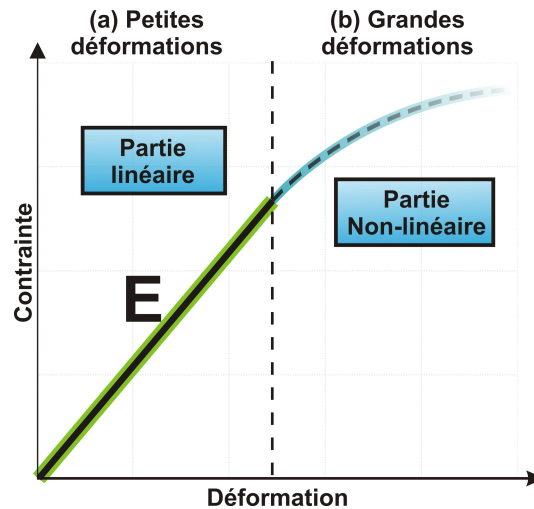


Figure 2.18 - Courbe contrainte/déformation lors d'un essai en traction. Le module de Young E est représenté par la pente de la partie linéaire.

2.4.1.1 Unité

L'unité du *module de Young* est le *Pascal*, noté Pa , est équivalent à un Newton par mètre carré (N/m^2). Lorsque la valeur du *module de Young* est élevée, on dit que l'objet est « rigide », ce qui est le cas par exemple pour les métaux. On utilisera alors par commodité les kilo-Pascal (kPa), méga-Pascal (MPa) et parfois des giga-Pascal (GPa). Le tableau 2.1 recense quelques valeurs connues pour certains types de matériaux.

| Matériau | module de Young (MPa) |
|-----------------------|-----------------------|
| Aluminium | 69 000 |
| Acier de construction | 210 000 |
| Argent | 83 000 |
| Tungstène | 406 000 |
| Uranium | 208 000 |
| Diamant | 1 000 000 |
| Saphir | 420 000 |
| Acajou | 12 000 |
| Chêne | 12 000 |
| Papier | 3 000 à 4 000 |
| Caoutchoucs | 700 à 4 000 |
| Nylon | 2 000 à 5 000 |
| Cartilage | 24 |
| Soie d'araignée | 60 000 |
| Vertèbre cervicale | 230 |

Table 2.1 - Valeurs du module de Young pour quelques matériaux¹⁵.

15. Une liste plus complète pourra être consultée sur http://fr.wikipedia.org/wiki/Module_de_Young.

2.4.2 Coefficient de Poisson

Dans la majorité des cas, lorsqu'un matériau est étiré dans une direction axiale, on constate une contraction de celui-ci dans la direction transversale à l'axe de traction. Le *coefficient de Poisson*, noté ν , est le rapport entre cette contraction transversale et l'allongement axial.

Soit une pièce cylindrique de longueur au repos l_0 et de diamètre au repos d_0 , on observe pendant une traction parfaite (sans frottement) suivant l'axe du cylindre une déformation de la pièce donnant un nouveau cylindre de longueur l et de diamètre d (cf Fig. 2.19).

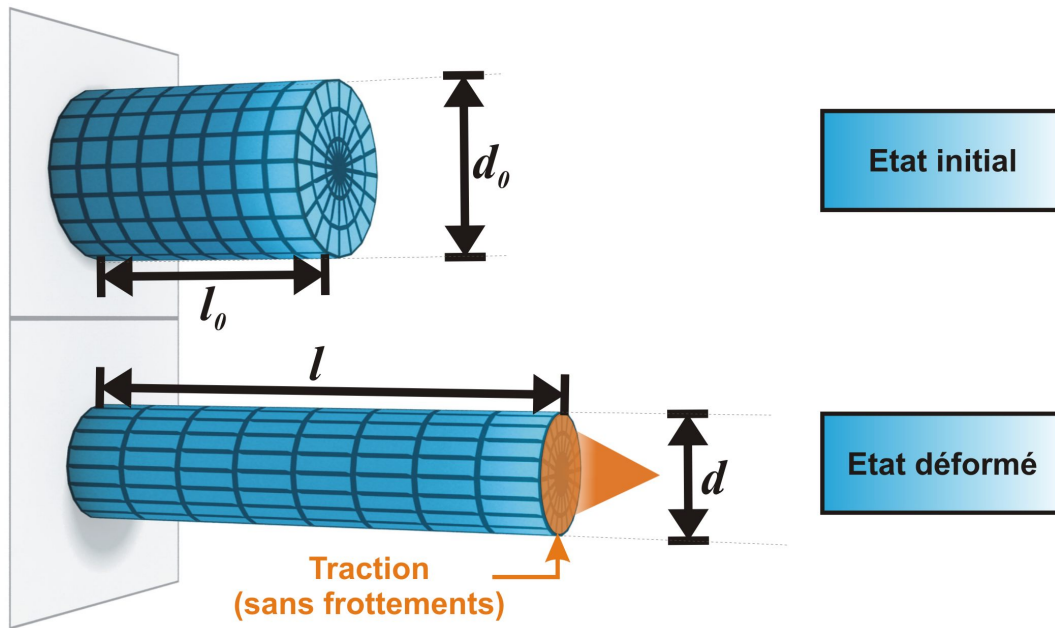


Figure 2.19 - Schéma de l'équation 2.25 pour une pièce cylindrique déformée par un test de traction sans frottement (vue latérale).

Le *coefficient de Poisson* ν est alors défini par :

$$\nu = \frac{\text{contraction transversale}}{\text{allongement axial}} = \frac{(d_0 - d)/d_0}{(l_0 - l)/l_0} \quad (2.25)$$

Il peut également être utilisé pour le calcul du changement relatif de volume (v_0 est le volume initial et v est le volume observé après déformation) :

$$\frac{v - v_0}{v_0} = (1 - 2\nu) \frac{l_0 - l}{l_0} \quad (2.26)$$

2.4.2.1 Unité

Le *coefficient de Poisson* n'a pas d'unité. Il est défini dans l'intervalle $[-1; 0.5]$. A partir de l'équation 2.26, il est trivial de voir que si $\nu = 0.5$ alors le matériau totalement *incompressible*, c'est-à-dire qu'il est impossible d'en modifier le volume. Certains matériaux, comme les caoutchoucs, ont cette propriété. De la même manière, si $\nu = 0$, alors le matériau est totalement *compressible* (le volume peut être nul). Tous les matériaux existant, à quelques

exceptions près, ont un *coefficient de Poisson* positif ($0.0 \leq \nu \leq 0.5$). Beaucoup, comme les métaux, ont un *coefficient de Poisson* $\nu \approx 0.3$. Le tableau 2.2 recense le *coefficient de Poisson* pour certains matériaux.

| Matériau | Coefficient de Poisson |
|----------------------|------------------------|
| Aluminium-alliage | 0.33 |
| Béton | 0.20 |
| Fonte | 0.21-0.26 |
| Verre | 0.24 |
| Glaise | 0.30-0.45 |
| Glaise saturée | 0.40-0.50 |
| Cuivre | 0.33 |
| Liège | 0.00 |
| Magnesium | 0.35 |
| Acier inoxydable | 0.30-0.31 |
| Caoutchouc | 0.50 |
| Acier | 0.27-0.30 |
| Mousse | 0.10 à 0.40 |
| Titane | 0.34 |
| Sable | 0.20-0.45 |
| Matériaux auxétiques | <0 |

Table 2.2 - Valeurs du coefficient de Poisson pour quelques matériaux¹⁶.

2.4.2.2 Auxétisme

Si ν est positif, le matériau se rétracte transversalement lors d'une traction axiale et, à l'inverse, se dilate lors d'une compression axiale. Cependant, quelques matériaux, dits *auxétiques*, ont un *coefficient de Poisson* $\nu < 0$ (certains cristaux, mousses de polymère ou fibres de matériaux composites). Dans ce cas, le comportement observé lors d'une traction/compression axiale est opposé : l'objet se dilate lors d'une traction et se rétracte lors d'une compression. Ce phénomène s'explique au niveau micro-structural : il ne s'agit pas d'une structure en nid d'abeille mais de cellules ré-entrantées (cf Fig. 2.20).

2.4.3 Relation avec les coefficients de Lamé

Il existe certaines formulations de la *loi de Hooke* qui utilisent les *coefficients de Lamé* λ et μ . Par exemple, une reformulation matricielle de l'équation 2.4 avec les *coefficients de Lamé* s'écrit :

$$[\sigma] = 2\mu[\varepsilon] + \lambda[\varepsilon]_{kk}[\mathbf{I}] \quad (2.27)$$

Voici les relations entre les *coefficients de Lamé* λ et μ , le *module de Young* E et le *coefficient de Poisson* ν :

16. http://en.wikipedia.org/wiki/Poisson_ratio

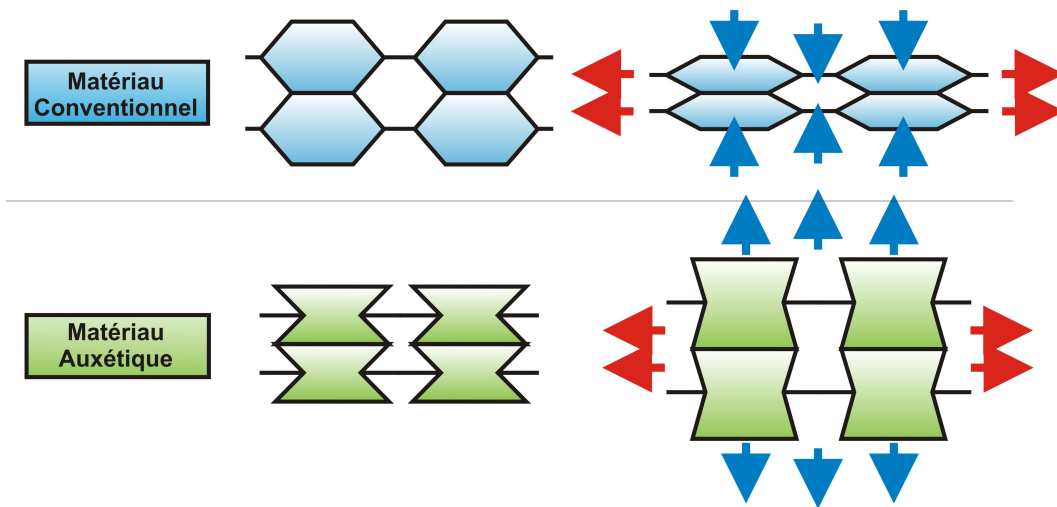


Figure 2.20 - Schéma explicatif des matériaux auxétiques. Le niveau micro-structural n'est pas en forme de nid d'abeille mais de cellules ré-entrantes : le comportement observé est inversé.

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)} \quad (2.28)$$

$$\mu = \frac{E}{2(1 + \nu)} \quad (2.29)$$

$$E = \mu \frac{3\lambda + 2\mu}{\lambda + \mu} \quad (2.30)$$

$$\nu = \frac{\lambda}{2(\lambda + \mu)} \quad (2.31)$$

Troisième Chapitre

MESURES ET ESTIMATIONS DE PARAMÈTRES

Ce chapitre présente les travaux existants qui permettent d'appréhender l'estimation de paramètres à partir de séquences d'images. Dans un premier temps, il décrit le concept de mécanique inverse. Nous y illustrons l'évolution de cette méthode jusqu'à l'estimation à partir de séquences vidéos en s'appuyant sur quelques travaux représentatifs de ce domaine. L'une des clés de ce concept étant l'optimisation au sens large, nous recensons quelques méthodes nécessaires à une convergence rapide de l'algorithme d'estimation inverse. En outre, identifier des paramètres depuis des données réelles suppose implicitement qu'il existe une interface avec le monde réel pour capturer ces données. Les données collectées sont ensuite utilisées pour réaliser l'estimation elle-même. Nous décrivons donc tout d'abord les différentes méthodes de mesure qui ont été appliquées à ce jour dans le but de capturer des données utiles à l'estimation de propriétés mécaniques d'objets déformables puis nous expliquons les différentes méthodes qui utilisent les mesures effectuées afin d'estimer des propriétés mécaniques. Enfin, nous récapitulons et expliquons les choix que nous avons fait dans le cadre de ces travaux.

3.1 Le concept de mécanique inverse

3.1.1 Méthode d'estimation inverse générale

La méthode d'estimation inverse est un processus itératif qui a pour but d'estimer un jeu de np paramètres P^{np} utilisé par une fonction déterministe $D = \mathbf{Fun}(P^{np})$ lorsque le résultat D de cette fonction est connu. Dans ce cas, l'estimation de P^{np} est envisageable même lorsque \mathbf{Fun} est inconnue ou qu'il est impossible d'en définir l'inverse \mathbf{Fun}^{-1} ¹. Le fonctionnement de base est très simple. Soit :

$$\begin{aligned} D &= \mathbf{Fun}(P^{np}) \\ \epsilon &= \Delta(D_A, D_B) \\ P_{\mathbf{Var}}^{np} &= \mathbf{Var}(P^{np}) \end{aligned}$$

La fonction Δ , appelée *fonction objective* ou *métrique d'erreur*, permet de quantifier la différence entre deux résultats D_A et D_B de la fonction \mathbf{Fun} . La fonction $\mathbf{Var}(P^{np})$ est une *fonction de variation* qui retourne une petite variation du jeu de paramètres. Ces deux fonctions forment le cœur du processus et nous verrons quelles méthodes d'optimisations peuvent être appliquées et comment les choisir (cf 3.2).

Le schéma général de l'estimation inverse est décrit par l'algorithme 1. Il s'agit d'essayer un jeu de paramètres, de vérifier si le résultat est satisfaisant, et, dans le cas contraire, de recommencer avec un nouveau jeu de paramètres (cf Fig. 3.1).

Algorithm 1 Algorithme général d'estimation inverse

```

Inconnue du système :  $P_R^{np}$ 
Données :  $\mathbf{Fun}$ ,  $D_R$ , seuil

 $P_S^{np}$  = initialisation (aléatoire, supposition,...)
 $\epsilon = +inf$ 
while  $\epsilon > \textit{seuil}$  do
  Calculer  $D_S = \mathbf{Fun}(P_S^{np})$ 
  Calculer  $\epsilon = \Delta(D_R, D_S)$ 
   $P_S^{np} = \mathbf{Var}(P_S^{np}) + P_S^{np}$ 
end while
 $P_{\mathbf{optim}}^{np} = P_S^{np} \approx P_R^{np}$ 

```

Dans le cas idéal, à la fin du processus itératif (quand $\epsilon < \textit{seuil}$, *seuil* défini par l'utilisateur) , $\epsilon = 0$ et le jeu de paramètres $P_{\mathbf{optim}}^{np}$ est alors optimal. Cependant, lorsque \mathbf{Fun} est inconnue, il est impossible d'affirmer que $P_{\mathbf{optim}}^{np}$ est la solution unique du problème. Lorsque cela est nécessaire, il faut alors avoir recours à une étude de cette fonction pour s'assurer de l'unicité de la solution.

L'exemple du rendu inverse Le terme « *rendu* » est utilisé en image de synthèse. Il s'agit du calcul d'une image par un ordinateur. D'innombrables algorithmes ont été définis, depuis les techniques de projection utilisées dans les cartes graphiques jusqu'aux méthodes plus rigoureuses physiquement comme le lancer de rayon (ou *raytracing*). Celles-ci n'ont cessé d'évoluer

1. Même lorsque \mathbf{Fun}^{-1} est calculable, il est parfois plus rapide d'utiliser cette méthode d'estimation inverse.

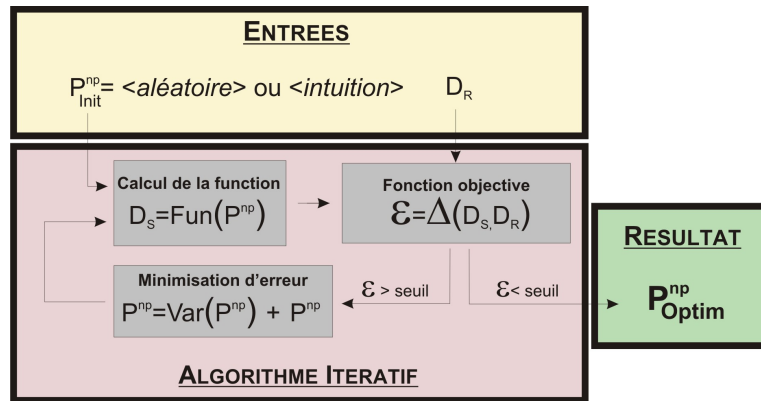


Figure 3.1 - Schéma de fonctionnement d'un algorithme d'estimation inverse

et aujourd'hui, de nombreuses méthodes permettent d'obtenir un rendu *photo-réaliste*, notamment en *infographie*. Les résultats obtenus se rapprochent toujours plus de la réalité, chaque algorithme ajoutant de nouvelles fonctionnalités (par exemple le *photon-mapping* qui permet, entre autres, de simuler la diffusion de photons sous la surface des objets, ou *Subsurface-scattering*). Cependant, l'accroissement du nombre de fonctionnalités entraîne également un accroissement du nombre de paramètres de ces algorithmes. Le *rendu inverse* est une technique de vision assistée par ordinateur dont le but est de retrouver certains de ces paramètres à partir d'images du monde réel.

Certains travaux [Yu 99][Boivin 01] proposent par exemple de retrouver les paramètres de fonctions de réflectance des objets (comme les *BRDF* [Cook 81][He 91][Ward 92][Koenderink 99]). À partir d'une photographie, d'un algorithme de synthèse d'image et des éléments en trois dimensions qui composent l'environnement des objets présents sur la photographie², il est possible d'estimer les divers paramètres nécessaires à la reproduction en image de synthèse de la photographie originale. Pour cela, les auteurs utilisent la méthode générale d'estimation inverse en posant :

- P_R^{np} : Les paramètres des *BRDF* à retrouver
- $D = \text{Fun}(P^{np})$: L'algorithme de synthèse d'image produisant une image D
- $\Delta(D_A, D_B)$: Une fonction quantifiant la différence entre deux images
- $\text{Var}(P^{np})$: Une fonction modifiant légèrement les paramètres P^{np}

Ce type d'estimation permet d'obtenir d'excellents résultats et l'image synthétique est alors très proche de l'image réelle (cf fig. 3.2).

3.1.2 Mécanique inverse

Par analogie avec le *rendu inverse*, la *mécanique inverse* est définie comme l'application de la méthode d'estimation inverse sur les algorithmes de simulation mécanique. Pour illustrer cette définition, nous utilisons la simulation d'un ressort simple.

2. Selon les différentes recherches sur le sujet, de nombreux autres paramètres doivent être connus comme par exemple la position de l'appareil photo et des sources de lumières.



Figure 3.2 - Résultats obtenus par rendu inverse après estimation des paramètres de fonctions de réflectance : à gauche, les images réelles, à droite, les images synthétiques correspondantes (d'après [Boivin 01]).

Exemple du ressort à spires Un algorithme simple permet de simuler l'extension d'un ressort à spires lorsqu'une extrémité est fixe et qu'une masse est accrochée à l'autre. L'équation d'un tel ressort suit la *loi de Hooke* : $F = -Kx$, où, dans ce cas simple, F est la force de réponse du ressort, x est l'élongation ($l - l_0$), et K est le coefficient de raideur du ressort. A l'état d'équilibre (statique), on sait que $F = -mG$, où m est la masse de l'objet fixé sur le ressort, et G est l'accélération de la gravité³. On a donc :

$$m = K \frac{x}{G}$$

Supposons que m et x soient connues et que le but du problème soit de trouver la raideur du ressort K correspondant (cf Fig. 3.3). En suivant la description donnée en 3.1.1, ce problème peut-être posé comme suit :

3. G est constante sur terre (≈ 9.81).

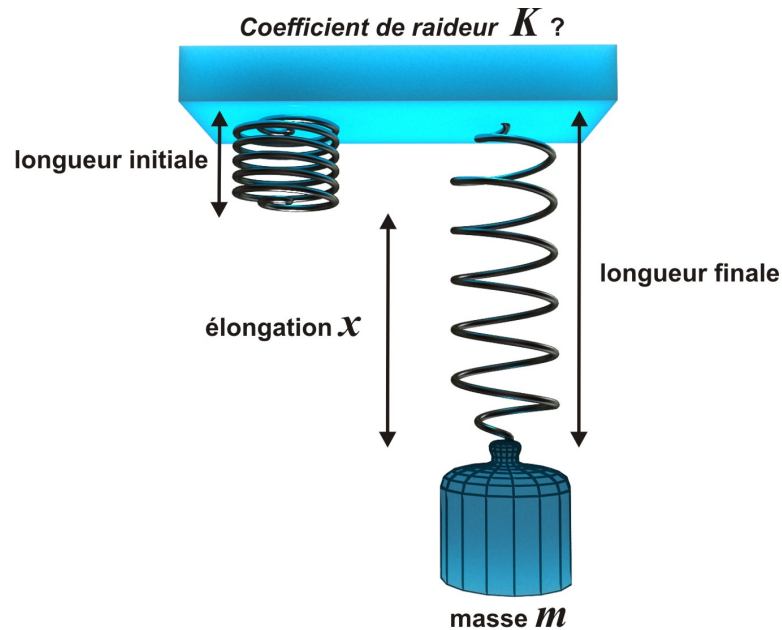


Figure 3.3 - Schéma du problème d'estimation de la raideur d'un ressort à spires.

$$\begin{aligned}
 np &= 1 \\
 \mathbf{P}^{np} &= K \\
 D &= m \\
 \mathbf{Fun}(\mathbf{P}^{np}) &= P_1 \frac{x}{G}
 \end{aligned}$$

Il ne reste plus qu'à définir la *fonction objective* et la *fonction de variation*. Pour cet exemple, nous prendrons l'association de la méthode des *moindres carrés* et de la *descente de gradient* (cf 3.2.1.1) :

$$\begin{aligned}
 \Delta(D_A, D_B) &= (D_A - D_B)^2 \\
 \mathbf{Var}(\mathbf{P}^{np}) &= -\gamma \nabla \mathbf{Fun}(\mathbf{P}^{np})
 \end{aligned}$$

Ces définitions permettent de lancer l'algorithme décrit en 3.1.1 et la raideur K est estimée en quelques itérations avec une précision déterminée par la valeur *seuil*. Il est également possible d'inclure d'autres inconnues dans cette formulation. Par exemple, si l'élongation x n'est pas mesurable, en posant $np = 2$ et $\mathbf{P}^{np} = [K, x]$, l'algorithme déterminera un couple solution⁴.

Evidemment, appliquer une méthode d'estimation inverse dans ce cas est contre-performante puisqu'il est possible d'obtenir la réponse directement ($K = m \frac{G}{x}$). Mais dans certains cas plus complexes, il peut être plus intéressant en termes de temps de calcul d'évaluer n fois la fonction \mathbf{Fun} que d'en calculer l'inverse \mathbf{Fun}^{-1} .

4. Dans ce cas précis, il en existe une infinité. La solution dépendra des valeurs d'initialisation et de $\mathbf{Var}(\mathbf{P}^{np})$

3.1.3 Mécanique inverse à partir de vidéos

En étendant cette définition de *mécanique inverse*, on peut faire une autre analogie avec le *rendu inverse*. En effet, si l'on reprend l'exemple de la section 3.1.1, le but est de retrouver les paramètres des fonctions de réflectance des objets (ici les *BRDF*). Ainsi, à partir d'une photographie et d'autres paramètres connus (comme la position de la caméra et la géométrie des objets en jeu par exemple), il est possible d'estimer les divers paramètres nécessaires à la reproduction en image de synthèse de la photographie originale.

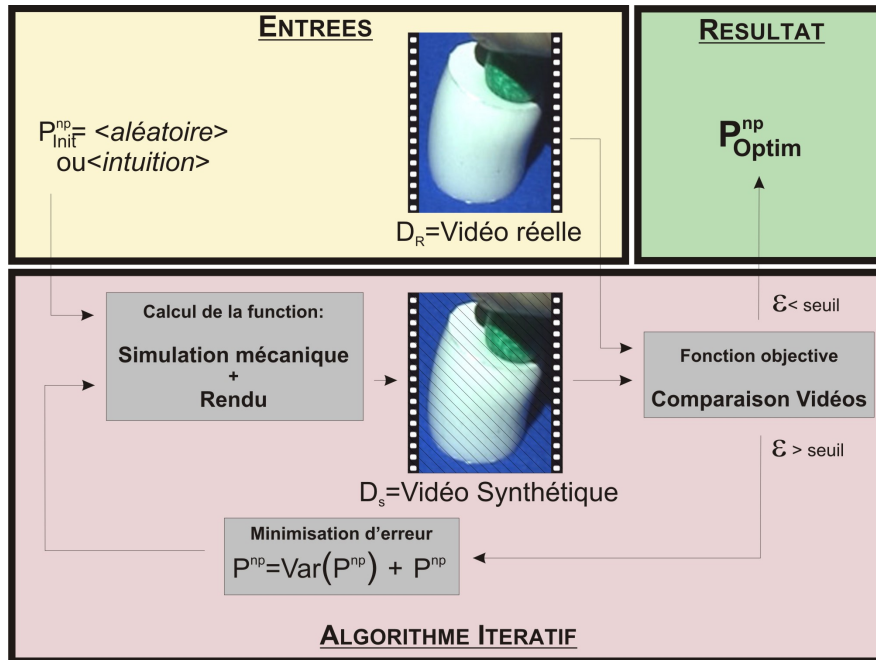


Figure 3.4 - Schéma de la mécanique inverse à partir de séquences d'images.

On peut alors considérer la *mécanique inverse* comme une extension du rendu inverse. En effet, si ces paramètres de fonctions de réflectance sont connus et que l'on dispose en outre d'un algorithme de simulation mécanique, il est possible de générer non plus une image statique mais une séquence d'images synthétiques composant une animation. Par exemple, si la simulation mécanique est capable de recréer le comportement physique d'un objet déformable lorsqu'une contrainte lui est appliquée, on peut générer une séquence vidéo synthétique qui pourra être comparée à une vidéo réelle d'un objet soumis à une même contrainte. Ainsi, en utilisant la boucle d'estimation inverse, on sera capable d'approximer les propriétés physiques de l'objet que peut reproduire l'algorithme de simulation mécanique (cf Fig. 3.4).

En utilisant la définition de la méthode inverse décrite dans le cas général en 3.1.1, on peut définir la mécanique inverse à partir de séquences d'images par :

- \mathbf{P}^{np} , les paramètres mécaniques du simulateur (exemple : *module de Young E , coefficient de Poisson ν , friction de Coulomb, ...*)
- \mathbf{D}_R , la séquence vidéo réelle et les mesures des forces appliquées sur l'objet déformable
- \mathbf{Fun} , une fonction qui :
 - Lance la simulation mécanique utilisant les paramètres \mathbf{P}^{np} et qui génère une géométrie animée et les efforts synthétiques internes

- Réalise le rendu d’une vidéo synthétique à partir de la géométrie animée et de paramètres externes (exemple : position de la caméra et des sources de lumières, fonctions de réflectance, etc. . .)
- Δ , une fonction capable de quantifier les différences entre deux séquences vidéos (réelle et synthétique)

Au cours de ces travaux, nous développons une évolution de cette définition pour estimer les paramètres mécaniques d’objets déformables à partir de séquences d’images (cf Chapitres 4 et 5).

Exemple des tissus Il existe une machine qui est la référence en terme de mesure des propriétés mécaniques des tissus : le système de Kawabata [Kawabata 80]. Bien que cette machine soit très précise et performante, elle est également très encombrante et onéreuse (cf fig. 3.8). C’est pourquoi certains auteurs ont tenté de retrouver ces paramètres à partir de vidéos de tissus déformés. Bhat (*et al.*) [Bhat 03] proposent par exemple une méthode pour estimer divers paramètres utilisés lors de la simulation mécanique de tissus à l’aide de l’algorithme de Baraff et Witkin [Baraff 98]. Elle consiste à filmer un morceau de tissu rectangulaire sur lequel est projeté un motif lumineux structuré. Le mouvement imposé est capturé par un système infrarouge (cf 3.3.2). Il appliquent ensuite la méthode d’estimation inverse en posant :

- P^{np} , les paramètres mécaniques du simulateur de tissus (flexion (*bend*), étirement (*stretch*), cisaillement (*shear*), amortissement en flexion (*bend damping*), amortissement en étirement (*stretch damping*), amortissement en cisaillement (*shear damping*), traînée linéaire (*linear drag*), traînée au carré (*quadratic drag*), traînée dégradée (*drag degradation*))
- D_R , la séquence vidéo réelle et les positions tridimensionnelles des deux coins du carré de tissus (boules reflectives aux infrarouges)
- \mathbf{Fun} , une fonction qui :
 - Lance la simulation mécanique du tissu utilisant les paramètres P^{np} et qui génère une animation tridimensionnelle du tissu
 - Réalise le rendu d’une vidéo synthétique à partir de cette géométrie animée et des positions de la caméra et du vidéo projecteur, ainsi que du motif projeté
- Δ , une métrique d’erreur comparant les « plis » observés et les silhouettes respectives des vidéos réelles et synthétiques
- $\mathbf{Var}(P^{np})$ une fonction de variation faisant appel à la méthode d’optimisation du « recuit simulé » (cf 3.2)

À l’aide de cette définition, les auteurs estiment les paramètres de quatre tissus différents (lin, laine, satin et tricot) à partir de plusieurs séquences d’images et peuvent ainsi recréer informatiquement les comportements de ces différents tissus (cf Fig. 3.5)⁵.

Cet exemple démontre la faisabilité d’une estimation de propriétés mécaniques d’objets complexes à partir de séquences d’images réelles.

5. Plus de détails sur ces travaux sont disponibles en 3.4.

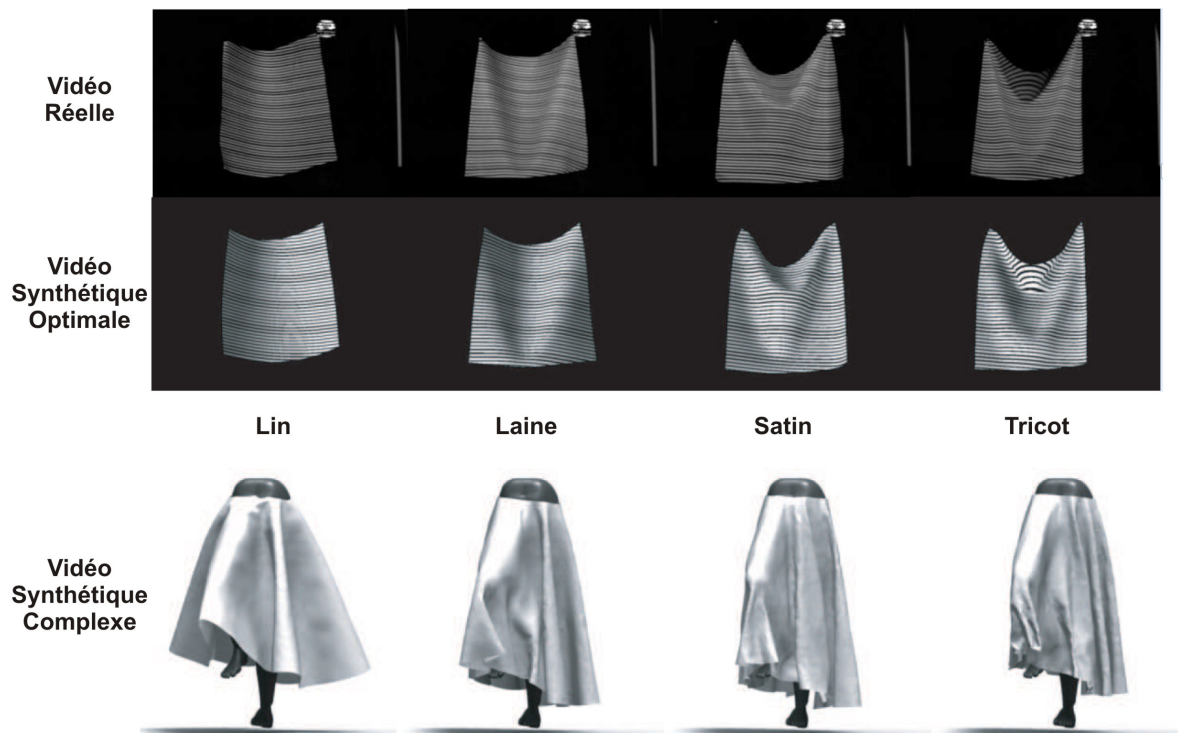


Figure 3.5 - Résultats obtenus par mécanique inverse après estimation des paramètres du modèle de Baraff et Witkin [Baraff 98] pour quatre tissus différents (lin, laine, satin et tricot) : en haut une image de chaque séquence vidéo réelle capturée, au centre, les images synthétiques correspondantes après optimisation des paramètres, en bas, l'utilisation de ces paramètres optimaux pour une scène plus complexe (d'après [Bhat 03]).

3.2 Optimisation

La section précédente montre la technique générale d'estimation inverse. Celle-ci met en place une étape appelée « minimisation d'erreur ». Nous avons vu que le but de cette étape est de faire varier le jeu de paramètres \mathbf{P}^{np} pour trouver la ou les solution(s) du problème (où le résultat de la fonction Δ est minimal). Par exemple, la méthode dite de « force brute » consiste à parcourir de manière exhaustive tout l'espace de validité des paramètres \mathbf{P}^{np} , à calculer à chaque fois le résultat de Δ après l'évaluation de $\mathbf{Fun}(\mathbf{P}^{np})$ puis de choisir le jeu de paramètres $\mathbf{P}_{\text{optim}}^{np}$ qui a permis d'obtenir le résultat optimal. Lorsque l'espace de validité est continu, on réalise un échantillonnage de celui-ci avec une précision donnée. Cependant, il est facile de voir que le nombre d'appels à la fonction peut vite devenir faramineux, tout particulièrement lorsque la taille de \mathbf{P}^{np} augmente ou quand la précision nécessaire est importante. C'est pourquoi il est généralement nécessaire de faire appel à des techniques « d'optimisation ».

L'*optimisation* au sens large du terme peut être définie comme une méthode permettant de trouver la ou les solution(s) d'un problème donné de la manière la plus efficace possible. Ce problème à résoudre peut souvent être exprimé par une *fonction objectif*, ou *fonction de coût*, que l'on cherche à minimiser ou maximiser par rapport à un ou plusieurs paramètres, sur lesquels peuvent venir s'ajouter un certain nombre de contraintes. On parle alors d'*optimisation mono-objectif* lorsque l'on recherche une solution unique au problème ou que celui-ci n'en a qu'une, et d'*optimisation multi-objectif* dans les cas contraires. Ce domaine de recherche est particulièrement actif car il apporte de nombreuses solutions dans des secteurs très variés comme la recherche opérationnelle, la conception de systèmes mécaniques, le traitement d'image ou encore l'électronique.

Les algorithmes existants sont légions, d'autant que beaucoup sont dérivés de grandes familles existantes, voire appliquant des méthodes hybrides. Il est donc difficile de dresser une liste exhaustive des travaux actuels. Cependant, on peut tenter de regrouper les recherches dans un organigramme. Nous ne nous attardons pas ici sur l'*optimisation multi-objectif*. Le lecteur pourra néanmoins consulter les travaux de Collette (*et al.*) [Collette 02] pour de plus amples informations sur le sujet. On trouve par ailleurs dans ces travaux un organigramme de classification générale des méthodes d'optimisation mono-objectif, publié également par la suite dans d'autres documents [Dréo 03]. Nous avons reporté cet organigramme sur la figure 3.6.

On distingue tout d'abord l'*optimisation combinatoire* de l'*optimisation continue*. L'optimisation combinatoire relève des problèmes dont les variables sont discrètes. Un exemple bien connu est le *problème du voyageur de commerce* : étant donné un certain nombre de villes et leurs positions géographiques, quel est le plus court chemin pour que le sujet les traverse toutes une et une seule fois chacune avant de revenir à la ville de départ ? À l'inverse, l'optimisation continue relève des problèmes dont les variables sont continues dans l'espace de recherche. C'est le cas notamment de la plupart des problèmes d'identification de paramètres à partir de données du monde réel. Nous ne nous attardons donc pas ici sur l'optimisation combinatoire, bien que certains algorithmes ont été adaptés pour passer d'une classe à l'autre, notamment parmi les *métaheuristiques* comme le *recuit simulé* par exemple.

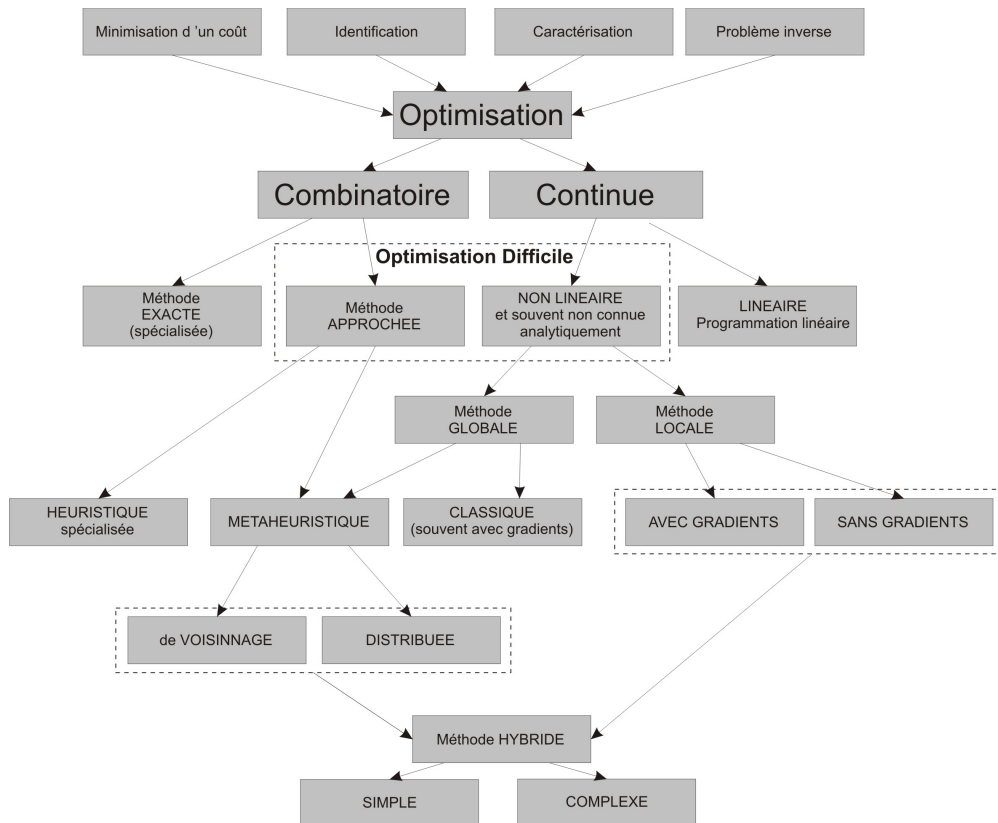


Figure 3.6 - Organigramme de classification générale des méthodes d'optimisation mono-objectif [Collette 02] [Dréo 03].

3.2.1 Quelques méthodes d'optimisation continue

Le but de ce type d'optimisation est de trouver le minimum (ou maximum) d'une fonction objectif $\mathbf{f}(\mathbf{x})$, où \mathbf{x} peut être un vecteur de longueur N qui peut être soumis à des conditions d'égalité comme $Ax = b$ ou d'inégalités comme $Ax < b$ par exemple.

Si la fonction $\mathbf{f}(\mathbf{x})$ est linéaire (c'est le cas de notre formulation linéaire pour la simulation mécanique par éléments finis, cf 2.3.2), la résolution du problème relève de la *programmation linéaire* et peut être résolu par exemple par la *méthode de Jacobi*, du *gradient conjugué* ou encore de *Gauss-Seidel*.

Dans le cas contraire, la résolution entre dans le cadre de l'*optimisation difficile*. On distingue quatre critères qui définissent un problème d'optimisation continue dite « difficile » :

1. $\mathbf{f}(\mathbf{x})$ n'est pas connue analytiquement
2. $\mathbf{f}(\mathbf{x})$ est bruitée, par exemple à la suite de mesures expérimentales issues du monde réel, ou encore par des problèmes de précision de calcul numérique
3. $\mathbf{f}(\mathbf{x})$ est non-linéaire
4. Il existe des corrélations entre certaines variables du vecteur \mathbf{x}

L'évaluation de ces critères permet de choisir entre les méthodes dites « locales » et les méthodes « globales », pour la plupart itératives. Les critères 1 et 2 interdisent l'accès à des algorithmes utilisant les gradients de $\mathbf{f}(\mathbf{x})$ (dérivées partielles). Les critères 2, 3 et 4

contribuent à générer un « paysage d'énergie » (réponse de la fonction sur l'espace de validité de \mathbf{x}) perturbé et comportant un grand nombre d'*extrema locaux* (cf. Fig. 3.7).

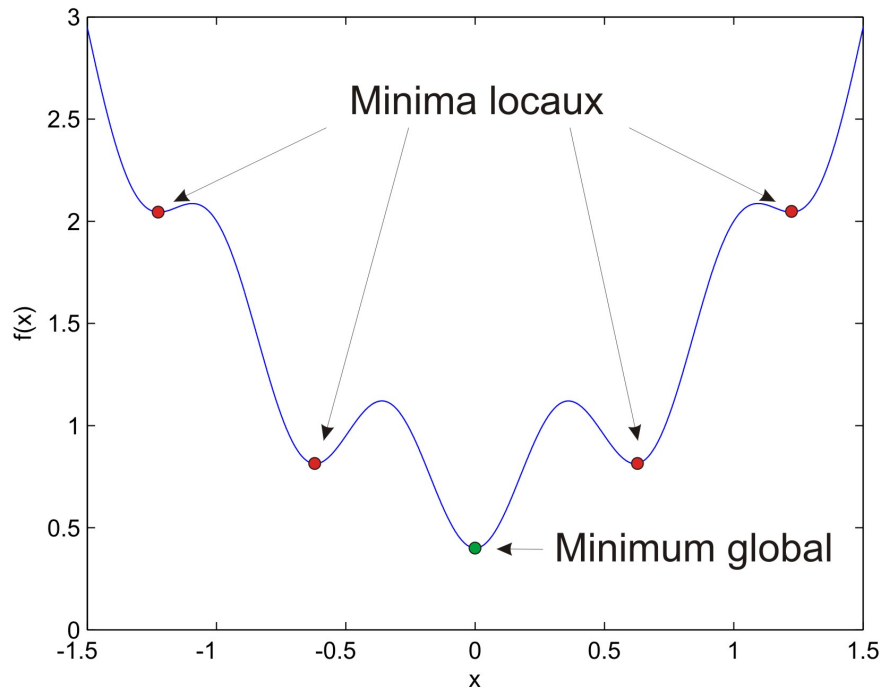


Figure 3.7 - Minima locaux et minimum global pour une fonction $\mathbf{f}(\mathbf{x})$ à un paramètre.

3.2.1.1 Quelques algorithmes locaux

Un des algorithmes de recherche d'extrema locaux les plus simples à mettre en œuvre est la *descente de gradient* (ou *steepest descent*). Il s'agit de suivre, à chaque itération, le gradient (dérivée \mathbf{f}') de \mathbf{f} pour converger vers un maximum local (l'inverse du gradient pour le minimum local). La méthode des *sous-gradients* peut-être utilisée lorsque les fonctions ne sont pas dérivables en tous points (par exemple la fonction « valeur absolue » $|x|$).

La *méthode de Newton*, algorithme itératif bien connu pour trouver la racine d'une équation à une ou plusieurs variables, peut également être utilisée pour trouver les minima (ou maxima) locaux d'une fonction \mathbf{f} . En effet, si un point x^* est un extremum local, la dérivée en ce point est nulle $\mathbf{f}'(x^*) = 0$. On peut donc utiliser la méthode de Newton pour trouver la racine x^* de la dérivée \mathbf{f}' . Il faut donc pouvoir également calculer la dérivée seconde \mathbf{f}'' . Cette méthode converge généralement plus vite que la descente de gradient car elle utilise la courbure locale pour calculer un pas plus efficace à chaque itération. Les variantes de cet algorithme, appelées méthodes quasi-Newton, s'affranchissent du calcul de la dérivée seconde en l'estimant à partir de l'évolution de la dérivée première.

La *méthode de Gauss-Newton* est utilisée pour résoudre le problème de *data fitting* non-linéaires par moindres carrés. Contrairement à la méthode de Newton, c'est la seule utilisation possible. En revanche, il n'est pas nécessaire de calculer une dérivée seconde, ce qui n'est pas

toujours possible. Un autre algorithme très connu pour ce problème est celui de *Levenberg-Marquard*. Il peut également être utilisé en programmation linéaire. Il est plus robuste que l'algorithme Gauss-Newton, c'est à dire qu'il trouvera le bon résultat plus souvent même si le point d'initialisation (première valeur de \mathbf{x} dans le processus itératif, souvent choisi au hasard, aussi appelé *initial guess*) en est éloigné. En revanche, il est en général moins rapide.

La méthode *golden section* [Kiefer 53], ou « section en or », est une méthode pour rechercher un extremum local sur une fonction à une seule variable. La technique consiste à choisir un intervalle $[\mathbf{x}_a \dots \mathbf{x}_b]$, dans lequel on sait a priori que se trouve l'extremum et d'évaluer $\mathbf{f}(\mathbf{x}_a)$, $\mathbf{f}(\mathbf{x}_b)$ et $\mathbf{f}(\mathbf{x}_c)$. \mathbf{x}_c est choisi tel que $\mathbf{x}_a < \mathbf{x}_c < \mathbf{x}_b$ et $\frac{(\mathbf{x}_b - \mathbf{x}_a) + (\mathbf{x}_c - \mathbf{x}_a)}{(\mathbf{x}_b - \mathbf{x}_a)} = \frac{(\mathbf{x}_b - \mathbf{x}_a)}{(\mathbf{x}_c - \mathbf{x}_a)} = \frac{1 + \sqrt{5}}{2} = \varphi$, où φ est le *nombre d'or* ou *golden ratio*, d'où le nom de la méthode. Selon le résultat des évaluations de \mathbf{f} , on divise de la même manière soit $[\mathbf{x}_a \dots \mathbf{x}_c]$, soit $[\mathbf{x}_c \dots \mathbf{x}_b]$. En procédant itérativement jusqu'à ce que la taille de l'intervalle soit inférieur à un seuil prédéfini, l'algorithme converge vers la solution. Le principal avantage de cette méthode est qu'elle n'a pas recours au calcul du gradient, ce qui est très utile lorsque l'expression analytique de $\mathbf{f}(\mathbf{x})$ est inconnue.

3.2.1.2 Quelques algorithmes globaux

Lorsque les critères décrits ci-dessus montrent une fonction comportant de nombreux extrema locaux et que le but est de trouver l'extremum global de la fonction (cf Fig. 3.7), plusieurs solutions sont possibles. Tout d'abord, l'utilisation répétée de méthodes locales à divers endroits de l'espace de recherche peut parfois porter ses fruits. Cependant, dans le cas où un très grand nombre d'extrema locaux sont présents, il est préférable d'adopter une *métaheuristique*, bonne alternative aux méthodes d'optimisation classiques.

On peut classer les métaheuristiques en deux catégories : les méthodes dites « de voisinage », qui font évoluer une solution unique au cours des itérations, et les méthodes « distribuées », qui manipulent en parallèle une population entière de solutions. Parmi ces deux solutions, l'un des critères de choix est le coût en temps de calcul de l'évaluation de la fonction $\mathbf{f}(\mathbf{x})$, les stratégies distribuées étant beaucoup plus coûteuses à chaque itération⁶.

Recuit simulé L'une des métaheuristiques les plus répandues est le *recuit simulé* (ou *simulated annealing*). Elle est issue des travaux de Kirkpatrick (*et al.*) [Kirkpatrick 83] qui ont adapté les résultats de l'observation d'un phénomène physique en métallurgie. En effet, ils se sont aperçu que lorsqu'un métal refroidit trop rapidement, notamment par la technique de la « trempe », la structure du métal entre dans un état solide amorphe, dans lequel l'écart entre les molécules n'est pas optimal (minimum local de l'énergie). En revanche, par un refroidissement plus lent, appelé le « recuit », le métal entre dans un état solide cristallin où l'agencement des molécules est optimal, d'où une solidité accrue (minimum global de l'énergie).

Le principe de l'adaptation de cette technique à l'optimisation est relativement simple. Tout d'abord, à chaque itération, on choisit un point \mathbf{x}_t dans le voisinage $\Delta\mathbf{x}$ de la solution courante \mathbf{x} ($\mathbf{x}_t = \mathbf{x} + \Delta\mathbf{x}$), on évalue la fonction $\mathbf{f}(\mathbf{x}_t)$ puis on choisit ce point comme solution courante ($\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_t)$) si $\mathbf{f}(\mathbf{x}_t) < \mathbf{f}(\mathbf{x})$. Mais pour éviter d'être « piégé » dans un minimum local, l'algorithme accepte parfois une solution plus « coûteuse » (si $\mathbf{f}(\mathbf{x}_t) > \mathbf{f}(\mathbf{x})$) selon une probabilité définie par un critère (critère de Métropolis) qui utilise une valeur de « température

6. Cependant, celles-ci sont de par nature très facilement adaptables à une architecture de calcul parallèle.

courante ». Cette température décroît avec le temps et les itérations : plus la température baisse, moins la probabilité d'accepter une solution plus « coûteuse » est grande. Ainsi, la métaheuristique autorise au début de l'algorithme plus de flexibilité que les algorithmes dits « gloutons » afin de s'approcher plus facilement du minimum global.

Cette technique a l'avantage d'être très simple à implémenter. En revanche, elle est très difficile à paramétrer, d'autant que des paramètres supplémentaires sont nécessaires pour la version continue de l'algorithme [Dréo 03]. Certains travaux en cours tentent de publier une liste des valeurs qui ont le plus de succès pour diverses applications et fonctions objectif. Cet algorithme a été abondamment modifié dans divers travaux en proposant des variantes [Creutz 83] [Aluffi-Pentini 85] [Geman 86] [Dueck 90] [Bertocchi 91] [Dueck 93].

Recherche avec tabous Une autre méthode de voisinage est également très répandue. Il s'agit de la *recherche avec tabous*. L'idée de cette méthode de recherche globale est de doter l'algorithme d'une certaine « mémoire à court terme ». En effet, à chaque itération, on évalue la fonction pour un certain nombre de voisins de la solution courante \mathbf{x} , et on choisit la meilleure de ces évaluations. Il est à noter que celle-ci peut être moins bonne que la solution courante : c'est ce mécanisme qui évite de rester coincé dans un minimum local. En revanche, sans amélioration, l'algorithme reviendrait dans des zones déjà explorées et ne pourrait pas vraiment converger vers le minimum global. À chaque fois que l'on a choisi une solution voisine \mathbf{x}' , on stocke en mémoire la transition $\mathbf{x} \rightarrow \mathbf{x}'$ dans une liste des dernières transitions qui seront par la suite interdites, les « tabous ».

Cette méthode est très simple à appliquer et donne de bons résultats, et ce, avec moins de paramètres de réglage que le recuit simulé. En ajoutant une notion de fréquence des occurrences des transitions, on peut également doter l'algorithme d'une « mémoire à long terme » par des mécanismes d'« intensification » et de « diversification ». Cela permet par exemple d'intensifier les recherches dans des zones jugées prometteuses, ou au contraire d'orienter l'algorithme vers les zones les moins explorées. Cependant, la complexité de l'algorithme et des réglages est alors bien plus importante.

Algorithmes évolutionnaires Parmi les métaheurstiques distribuées, on peut noter les *algorithmes évolutionnaires*, inspirées par l'évolution biologique des espèces [Fraser 57]. Les plus représentatives de ces méthodes sont sans doute les *algorithmes génétiques* [Goldberg 89] [Goldberg 94]. L'idée est de faire évoluer une population d'individus initiale, chaque individu étant doté d'une notion de « performance ». Par exemple, pour la recherche d'un minimum global, plus $\mathbf{f}(\mathbf{x})$ est plus petit, plus \mathbf{x} est performant. Au cours des itérations successives, appelées ici « générations », l'algorithme tente de faire évoluer progressivement la composition des individus pour tenter d'améliorer globalement leurs performances tout en en conservant un nombre constant. Pour ce faire, on passe par deux étapes issues de la théorie de Charles Darwin [Darwin 59]. Tout d'abord, on procède à une « sélection » qui permet de choisir les individus les plus performants susceptibles de se reproduire et de survivre. L'étape suivante est la « reproduction », au cours de laquelle les individus sélectionnés sont croisés pour donner aux descendants des potentialités nouvelles en fonction des caractères héréditaires des parents et d'éventuelles « mutations génétiques ».

Concrètement, une représentation d'un individu est choisie. Par exemple, pour l'optimisation combinatoire, un individu peut être représenté par un entier. Le croisement d'individus peut être alors représenté par une opération booléenne sur certains bits de ces entiers, alors

que la mutation est représentée par l'altération aléatoire de certains bits. Un critère d'arrêt doit également être défini, par exemple un seuil sur l'évaluation de la fonction objectif au delà duquel on stoppe l'algorithme dès qu'il est franchi par le meilleur individu.

Ces algorithmes sont de par leur nature très simples à paralléliser. En revanche, le coût en temps de calcul pour l'évaluation de la fonction objectif doit être faible car elle est effectuée pour chaque individu.

3.3 Capture de données

Le résultat d'une simulation mécanique d'un objet déformable se caractérise généralement par la déformation géométrique de celui-ci et par les efforts mis en jeu pour obtenir cette déformation. Les outils permettant la capture de données réelles dans le cadre de l'estimation de propriétés mécaniques peuvent donc se diviser en deux catégories : ceux qui permettent de mesurer les forces appliquées sur l'objet ainsi que les forces générées par celui-ci et ceux qui permettent de quantifier la déformation géométrique⁷.

Cependant, cette distinction est bien souvent difficile à faire. C'est notamment le cas des dispositifs à *indenteur* ou des machines universelles de test en traction/compression. Un *indenteur* est un composant mobile en contact avec l'objet qui applique la déformation. Bien souvent, il est relié à un couple capteur de force/capteur de déplacement. Une machine de test en traction/compression est composée de deux plaques, l'une fixe et l'autre mobile (déplacement vertical uni-axial), entre lesquelles est placé l'échantillon (mesures *in-vitro*). Là aussi, la mesure de force est réalisée conjointement à la mesure de déplacement.

Ces deux types de dispositif réalisent donc une mesure *mécanique*. D'autres dispositifs permettent de mesurer la déformation de manière *visuelle*. Il existe une multitude de travaux définissant des dispositifs de mesure. La principale raison est que peu ont su s'imposer comme référence et qu'il est bien souvent plus simple de concevoir un dispositif spécifique aux travaux de recherches menés que d'adapter un dispositif existant. Cependant, nous tentons de dresser tout d'abord une liste de plusieurs dispositifs de mesure mécanique permettant de survoler la majorité des techniques actuelles. Nous recensons également quelques dispositifs incluant une approche visuelle pour donner un aperçu des technologies utilisées dans ce domaine.

3.3.1 Dispositifs de mesure mécaniques

Les dispositifs de mesure de force existants peuvent être classés en deux sous-catégories : les *dispositif automatiques*, motorisés et contrôlés par ordinateur, et les *dispositifs interactifs*, contrôlés par l'utilisateur. Les caractéristiques principales de ces outils utilisant un *indenteur* ou une machine de test en traction/compression [Instron 07] [Zwick] sont⁸ :

- f_{max} la force maximale mesurable en Newtons (N) ou en grammes (g)
- u_{max} le déplacement maximal autorisé de l'*indenteur* ou de la plaque
- v_{max} la vitesse maximale du déploiement de l'*indenteur* ou de la plaque
- \varnothing^{ind} le diamètre de l'*indenteur* (si celui-ci est sphérique ou cylindrique)

A ces caractéristiques viennent s'ajouter d'autres paramètres selon le type de dispositif.

7. On peut noter ici le parallèle avec les formulations mathématiques qui mettent en rapport le *tenseur des contraintes* $[\sigma]$ et le *tenseur des déformations* $[\epsilon]$

8. Les symboles sont utilisés dans la suite de ce chapitre pour une meilleure lisibilité.

L'un des gros challenges étant la simulation d'opérations chirurgicales, la grande majorité des dispositifs présentés ici sont destinés à mesurer les efforts générés par des tissus biologiques sous plusieurs conditions :

- *in-vitro* ou *ex-vivo* : les organes ont été préalablement prélevés avant d'effectuer la mesure en laboratoire
- *in-vivo* : la mesure s'effectue directement sur le sujet dans son environnement initial
- *in-situ* : la mesure s'effectue directement dans l'environnement initial mais après la mort du sujet

Nous utiliserons également les termes *invasif*, *peu invasif* et *non-invasif*. Une mesure invasive nécessite une large coupure de la peau pour accéder aux organes alors que lors d'une mesure non-invasive, aucune incision n'est réalisée (ex : Imagerie à Résonance Magnétique (IRM)). Une mesure peu invasive nécessite quant à elle une incision relativement petite (maximum $\approx 2\text{cm}^2$ en *laparoscopie* par exemple). Cette technique est de plus en plus utilisée car elle permet un rétablissement plus rapide des patients tout en réduisant les risques de dommages, d'infections et de douleurs. Les outils de mesures les plus récents se basent donc sur le schéma des outils d'opération non-invasives.

3.3.1.1 Dispositifs automatiques

Les dispositifs automatiques, contrôlés par ordinateur, ont de nombreux avantages. Le principal est la répétabilité d'une mesure. En effet, il est possible, par exemple dans le cas d'un indenteur, de réaliser plusieurs fois une mesure au même endroit. De plus, le schéma de déformation induite peut être contrôlé en vitesse ce qui permet de mesurer l'effet de cette vitesse sur les matériaux à comportement viscoélastique et, plus généralement, les phénomènes dynamiques comme le préconditionnement (cf 2.1.3).

La machine de Kawabata L'un des premiers systèmes de mesure complets est défini en 1980 par Kawabata [Kawabata 80]. Il conçoit un système d'évaluation en plusieurs modules permettant de réaliser des tests sur les textiles en *flexion*, *traction*, *compression* et *cisaillement*. Le tissu est fixé entre deux mâchoires mécaniques commandées par ordinateur (cf Fig. 3.8). Le résultat est une courbe de la force mesurée soit par rapport au temps, soit par rapport au déplacement ou l'angle de rotation.

Les tests de flexion permettent de déterminer la résistance produite par le textile lorsque la machine tente de le plier. Lors des essais en traction, le système permet de déterminer les efforts générés par le tissu et l'extensibilité de celui-ci. À l'inverse, les tests de compression permettent de connaître son comportement lorsque la machine tente de le comprimer. Le cisaillement est lui déterminé par un test où les mâchoires se déplacent parallèlement l'une par rapport à l'autre. Pour tous ces tests, il est possible d'obtenir les valeurs statiques à un instant t donné mais également le comportement dynamique du tissu. La précision et la gamme complète des mesures possibles a fait de cette machine la référence dans le domaine des mesures textiles.

Indenteurs Ce type de dispositifs est généralement utilisé dans le domaine médical pour caractériser les comportements mécaniques des tissus biologiques. Miller (*et al.*) [Miller 97] utilisent une machine de test en traction/compression (« Tensilon UTM-10T », © *Orientec Corporation*) pour effectuer des mesures *in-vitro* d'échantillons de tissus cérébraux de porc.

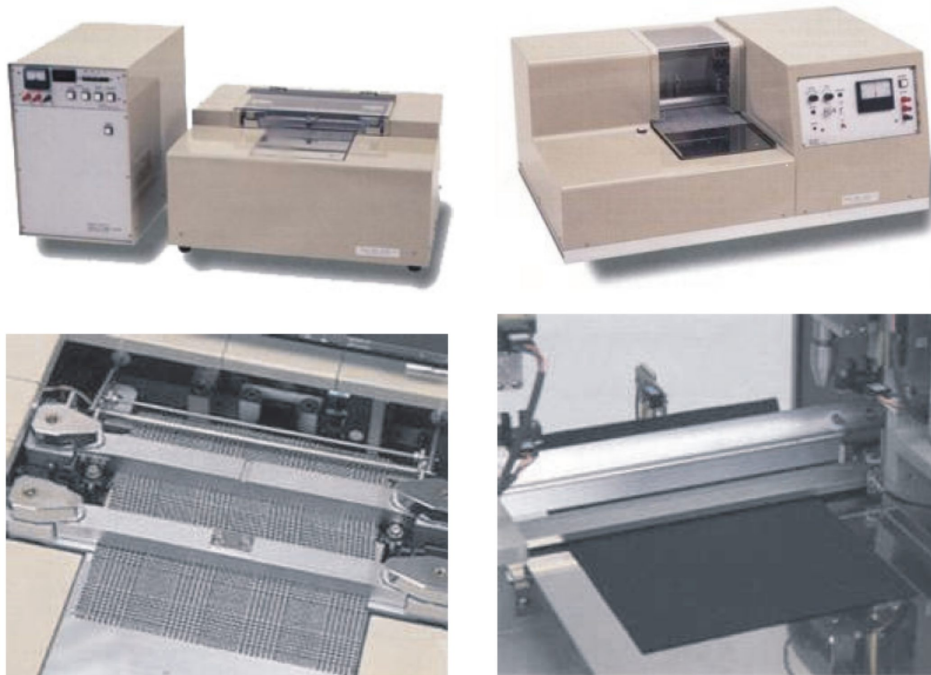


Figure 3.8 - Différentes vues du système d'évaluation de kawabata (d'après [NgoNgoc 04])

La machine n'est utilisée ici qu'en compression. L'échantillon est placé entre une plaque fixe et une plaque mobile verticalement (uni-directionnelle) contrôlée par ordinateur. Le déplacement et la force sont mesurés. Pour cette machine, $f_{max} = 9.0N$ et $v_{max} \approx 8.33mm/s$. Par ailleurs, un dispositif de mesure au laser (« LB-02/LB-62 », © Keyence Corp) est utilisé pour mesurer le déplacement radial du tissu. La vitesse de déploiement réglable permet d'étudier le comportement visco-élastique des tissus cérébraux *in-vitro*.

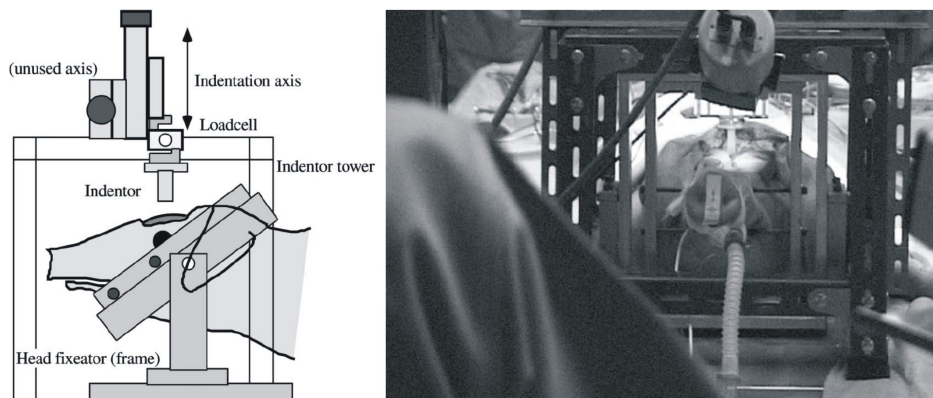


Figure 3.9 - D'après Miller (et al.) [Miller 00] (*in-vivo, Invasif*) : À gauche, vue schématique du dispositif. À droite, photo du dispositif en situation.

A la suite de ces travaux, Miller (et al.) [Miller 00] utilisent un dispositif volumineux (environ $50cm^3$) pour mesurer les forces *in-vivo* et *in-situ* générées par les tissus cérébraux de porcs de manière invasive (cf Fig. 3.9). Un *indenteur* cylindrique ($\varnothing^{ind} = 10mm$) est placé sur une partie mobile motorisée elle-même placée sur un bâti fixe. Grâce à la partie mobile,

l'*indenteur* peut être déplacé en deux dimensions au dessus de la zone d'intérêt avant de se déplacer verticalement à l'aide d'un moteur. La tête du sujet est placée sous le bâti et une partie du crâne est ouverte pour accéder aux tissus cérébraux. Le déplacement maximal de l'*indenteur* est réglé sur $u_{max} = 3.9mm$ pour éviter au sujet des dommages irréversibles. La force maximale mesurable est $f_{max} = 0.98N$ et la vitesse de déploiement maximale est fixée à $v_{max} = 1mm/s$. Ce dispositif permet un contrôle précis de la zone de mesure et de la force mesurée.

Pai (*et al.*) [Pai 00] [Lang 02] proposent une solution de mesure complète pour capturer *in-vitro* des informations de déformation. Parmi les outils composant le système nommé *ACME*, on trouve un capteur de force/moment de torsion (« mini40 », © *ATI Industrial Automation Inc.*, 6-axes, $f_{max} = 120N$) monté sur un bras robotisé (« Puma 260 »). Ce bras est lui-même monté sur un axe motorisé permettant un déplacement horizontal supplémentaire de $1.267m$, ce qui accroît encore l'espace de travail. Des mesures peuvent donc être faites sur des objets de taille relativement importantes. L'ensemble est commandé par ordinateur (cf Fig. 3.10).

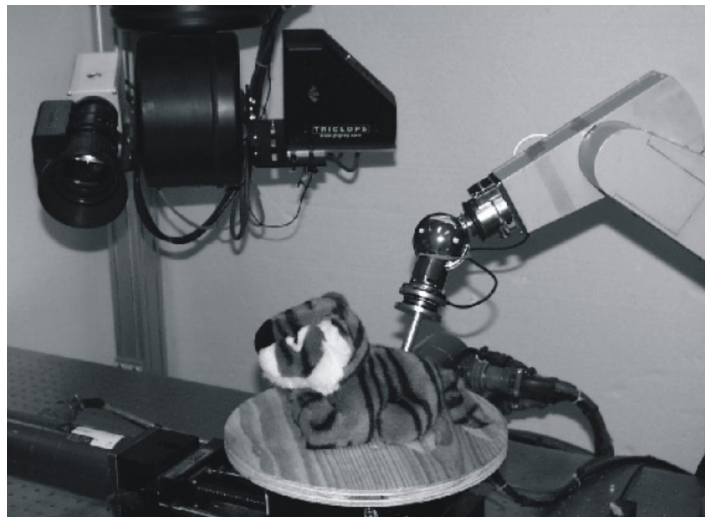


Figure 3.10 - D'après Lang (*et al.*) [Lang 02] : application d'une déformation et mesures sur une peluche à l'aide du dispositif *ACME*.

Brouwer (*et al.*) [Brouwer 01] décrivent plusieurs dispositifs de mesures. Parmi eux se trouve un instrument destiné à réaliser un étirement uniaxial (horizontalement) des tissus *in-vivo*, *in-situ* et *in-vitro*. Deux pinces, l'une fixe, l'autre mobile, sont montées sur un axe horizontal. La pince mobile est déplacée sur l'axe à l'aide d'un moteur qui mesure également le déplacement. L'ensemble peut être fixé sur une table d'opération. Un capteur de force/moment de torsion (6-axes) est placé sur la partie mobile pour mesurer l'effort de traction. La résolution du capteur est de $0.0125N$. Un autre capteur de force est placé sur la pince mobile elle-même pour mesurer l'effort de pression. Ce type de système est très utile pour mesurer les efforts induits par l'appareil de mesure lui-même (cf Fig. 3.11). Pour les tissus plus fragiles qui ne supportent pas les efforts de traction de ce dispositif, ils proposent de monter un *indenteur* ($\varnothing^{ind} = 20mm$) sur un dispositif haptique (« Phantom 1.5 », © *Sensable technologies*). Le « Phantom » peut être contrôlé par ordinateur pour réaliser une indentation à vitesse constante afin de déterminer des propriétés viscoélastiques. La mesure ne se fait pas *in-situ* mais les organes sont déplacés hors du corps sans être incisés, puis remis en place après la

mesure (invasif).

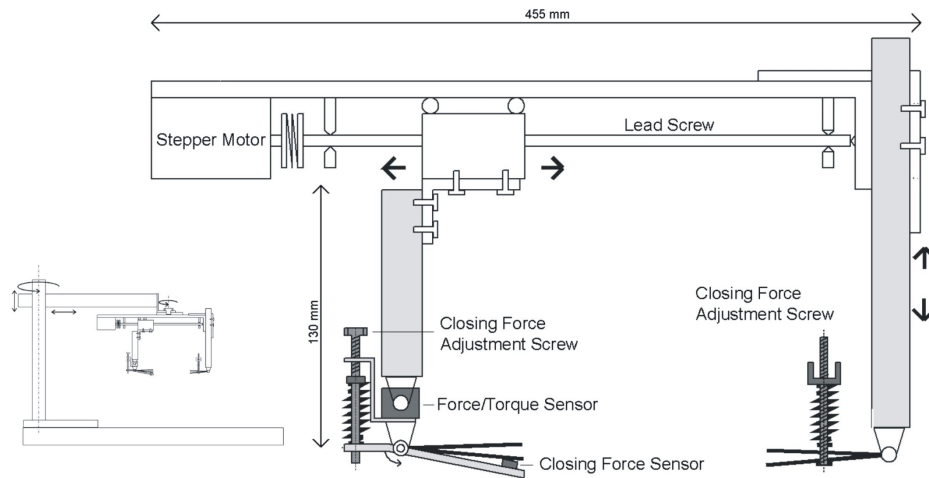


Figure 3.11 - D'après Brouwer (et al.) [Brouwer 01] (in-vivo, in-situ et in-vitro, Invasif) : Schémas du dispositif

Ottensmeyer (et al.) [Ottensmeyer 01] [Bruyns 02] [Ottensmeyer 02] [Ottensmeyer 04] proposent un dispositif à indenteur nommé « TeMPeST 1-D » (cf Fig. 3.12, à gauche). Les caractéristiques de cet outil sont $\varnothing^{ind} = 5\text{mm}$, $u_{max} = \pm 0.5\text{mm}$ et $f_{max} = 0.3\text{N}$. Le test d'excitation des tissus peut être fait de plusieurs manières : le profil du déplacement est contrôlé par ordinateur. Des oscillations de l'indenteur peuvent également être programmées. Le diamètre de l'outil complet est de 12mm ce qui le rend *peu invasif*. Il est utilisé dans toutes conditions de mesures, *in-vivo*, *in-situ* et *in-vitro*. Dans [Ottensmeyer 04], un autre dispositif est utilisé (cf Fig. 3.12, à droite). Il s'agit également d'un indenteur, appelé « VESPI » ($\varnothing^{ind} = 6\text{mm}$, $u_{max} \approx 10\text{mm}$ et $f_{max} = 100\text{g}$). Il est monté sur bras muni d'un système de contrepoids qui permet de limiter le stress initial à 3g . Les déformations induites par cet outil sont plus importantes que pour le « TeMPeST 1-D ». Il est utilisé pour des mesures sur des organes *in-vitro* mais qui sont raccordés à des solutions salines pour limiter l'altération de leurs propriétés mécaniques.

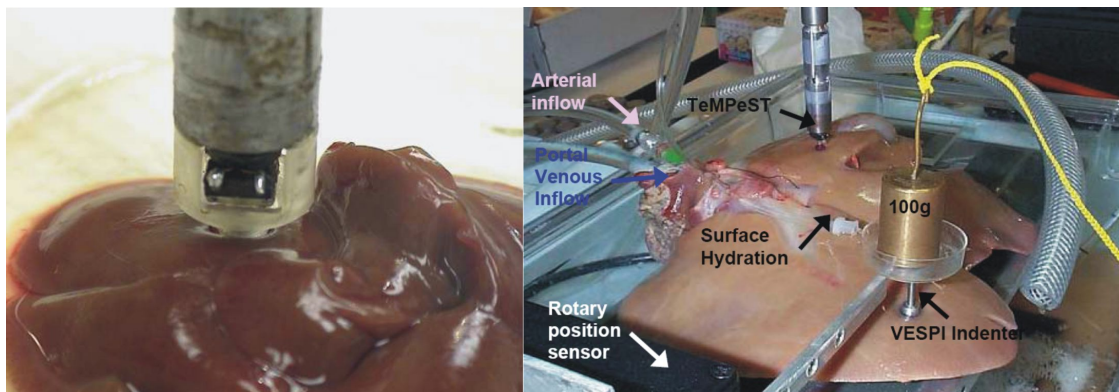


Figure 3.12 - À gauche : Mesure in-vitro sur le foie d'un rat à l'aide du « TeMPeST 1-D » (D'après [Bruyns 02]). À droite : Mesure in-vitro sous perfusion d'un foie de porc à l'aide du « TeMPeST 1-D » et du « VESPI » (D'après [Ottensmeyer 04])

Vacalebri (*et al.*) [Vacalebri 03] ont conçu un dispositif à *indenteur* basé sur le fonctionnement des machines de test en traction/compression (cf Fig. 3.13). Le but de cet outil est de réduire les dimensions importantes de ces machines en général ($\approx 1m \times 0.5 \times 0.5m$) pour effectuer des tests de compression uniaxiale. Leur dispositif de forme générale cylindrique est réduit à une longueur de $185mm$ pour un poids total de $720g$. Le capteur de force (« L1650 », © *Futek*) est monté sur un piston en « Teflon », © *DuPont* sur lequel est fixé l'indenteur. Les caractéristiques de l'ensemble sont $f_{max} = 50N$, $u_{max} = 35mm$ et $v_{max} = 5mm/s$. Un boîtier de contrôle et d'enregistrement des données pilote l'engin pour en faire une solution portable et indépendante d'un quelconque système informatique.

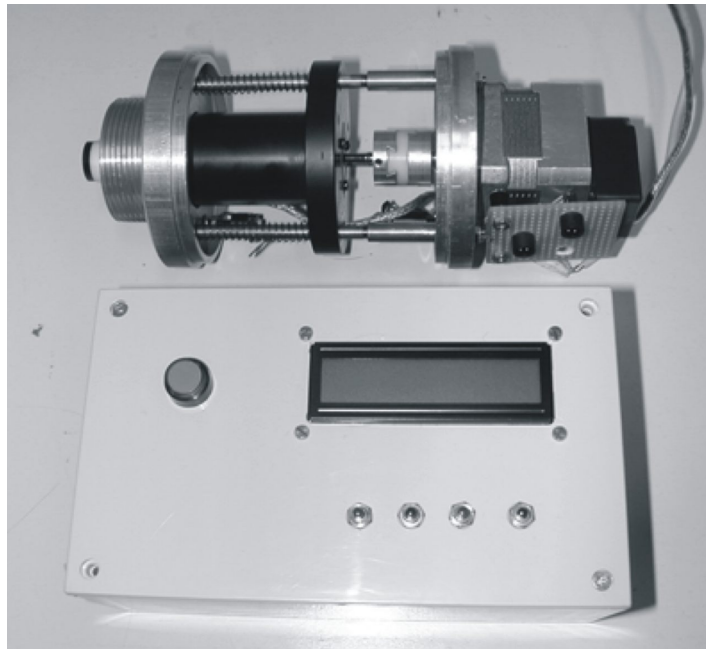


Figure 3.13 - D'après Vacalebri (*et al.*) [Vacalebri 03] : en haut, le dispositif à indenteur, en bas, le boîtier de contrôle autonome

Delalleau (*et al.*) [Delalleau 05] utilisent un *indenteur* motorisé à embout sphérique ($\varnothing^{ind} = 6.22mm$). La précision de la mesure du déplacement est $0.1\mu m$ et la précision de la mesure de force $0.1mN$. Ils utilisent cet outil pour mesurer *in-vivo* les déformations de la peau et des forces que celle-ci génère de manière *non-invasive* (cf Fig. 3.14).

Samur (*et al.*) [Samur 05] [Sedef 06] [Samur 07] conçoivent un dispositif à *indenteur* en modifiant un périphérique haptique (« Phantom 1.0 », © *Sensable technologies*) piloté par un logiciel. Un capteur de force/moment de torsion est fixé sur le « Phantom » (« Nano 17 », © *ATI Industrial Automation Inc.* , $f_{max} = 70N$). Une longue tige ($\varnothing^{ind} = 4mm$) à bout arrondi est fixée à l'autre extrémité du capteur et l'ensemble est maintenu par un capot en deux parties démontables pour stériliser l'outil. L'ensemble peut s'insérer dans un trocart. Il est utilisé pour mesurer *in-vivo* et *in-situ* les réponses haptiques d'organes de porc en chirurgie *peu invasive* (cf Fig. 3.15).

Ultrasons Une autre approche est illustrée par Chen (*et al.*) [Chen 96] qui valident la technique de l'imagerie à ultrasons. Pour cela, ils effectuent un test en une dimension en induisant et en mesurant des micro-déplacements à l'aide d'un couple capteur de fréquence/balance. Les

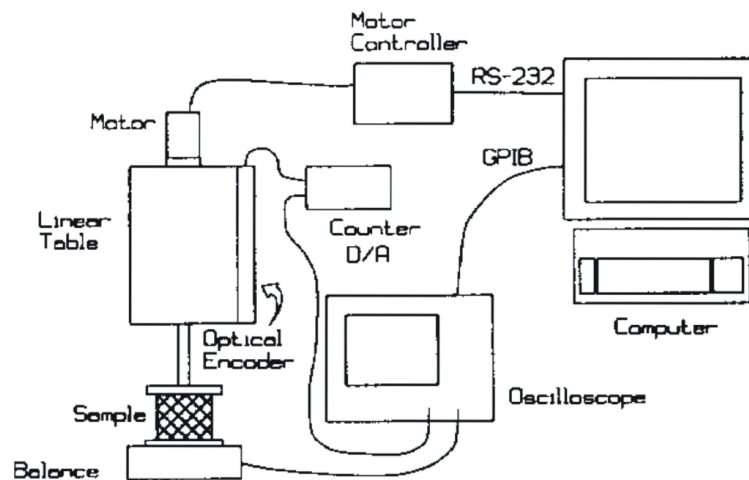


Figure 3.16 - D'après Hall (et al.) [Hall 97] : Schéma de fonctionnement

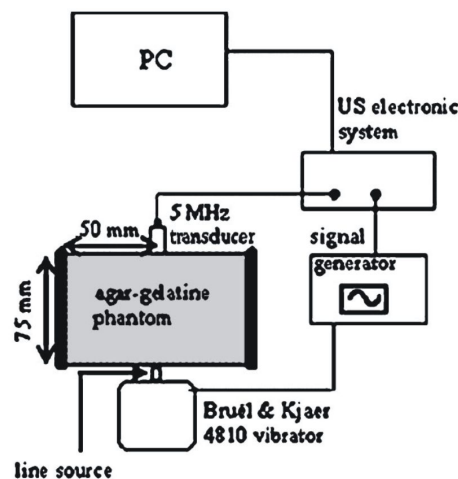


Figure 3.17 - D'après Benech et Negreira [Benech 05] : Schéma de fonctionnement

3.3.1.2 Dispositifs interactifs

Les dispositifs interactifs, contrôlés par l'utilisateur, ont l'inconvénient de ne pas pouvoir répéter précisément une mesure. En revanche, ils sont bien souvent moins encombrants, plus maniables et proposent une plus grande liberté de mouvement que les dispositifs automatiques.

Ostrander et Lee [Ostrander 92] proposent un outil de mesure de forces destiné à capturer le comportement viscoélastique des tissus humains. Le but est d'évaluer la pression exercée par les prothèses, utilisées suite à la perte d'un membre, sur les tissus environnants. Le dispositif est composé d'un bâti sur lequel est placé un *indenteur* relié à un capteur de force et dont le déplacement est mesuré à l'aide de trois capteurs de position linéaire. Des poids sont utilisés pour appliquer la déformation. Les mesures de force et du déplacement de l'*indenteur* sont uni-directionnelles par rapport au bâti. Les caractéristiques sont $f_{max} = 1kg(\pm 1g)$ et $u_{max} = 25mm(\pm 0.1mm)$. Des *indenteurs* de formes et de tailles différentes peuvent être utilisés. L'*indenteur* de taille maximale utilisé dans les exemples a une section rectangulaire

de $12\text{mm} \times 17\text{mm}$. Ce dispositif a l'avantage de pouvoir supporter différents *indenteurs* ce qui peut s'avérer très utile selon le site de mesure et pour le calcul des déformations lors de l'identification des paramètres.

Parmi les dispositifs de mesures décrits par Brouwer (*et al.*) [Brouwer 01], on trouve trois instruments contrôlés manuellement. Il s'agit d'outils chirurgicaux existants modifiés, sur lesquels sont placés des capteurs de force. Tout d'abord, un capteur de force/moment de torsion (6-axes) (résolution de 0.0125N) est placé sur une pince permettant de saisir une aiguille de suture (cf Fig. 3.18, à gauche). Le capteur mesure les forces ressenties par le chirurgien lors de cette suture. Ensuite, un capteur de force uniaxial est fixé sur une paire de ciseaux chirurgicaux (cf Fig. 3.18, au centre) et sur une pince à clamper (cf Fig. 3.18, à droite). Dans les deux cas, le capteur est placé dans l'anneau de l'outil pour mesurer la force exercée par le pouce du chirurgien.



Figure 3.18 - D'après Brouwer (*et al.*) [Brouwer 01] : outils chirurgicaux modifiés pour mesurer les interactions outils/tissus et chirurgiens/outils

D'autres travaux proposent des modifications d'outils chirurgicaux. Brown (*et al.*) [Brown 03] proposent un outil que l'on peut qualifier d'« hybride » entre un dispositif interactif et automatique. Il s'agit une pince à clamper endoscopique (« Babcock », © Karl Storz). Cette pince s'utilise également à travers un trocart lors d'une opération peu invasive. Un moteur est adapté sur la pince (cf Fig. 3.19) . Celui-ci gère la fermeture des deux mâchoires pour un test des propriétés du matériau en compression. Cependant, le point d'application de la contrainte de compression est choisi manuellement. La mesure de la force est réalisée par un capteur de force ($f_{max} = 70\text{N}$). La déformation que peut produire l'outil est $u_{max} < 30\text{mm}$. Ce dispositif est un bon compromis entre le contrôle de la déformation appliquée par un moteur et la grande liberté de mouvements procurée par le positionnement interactif de la pince.

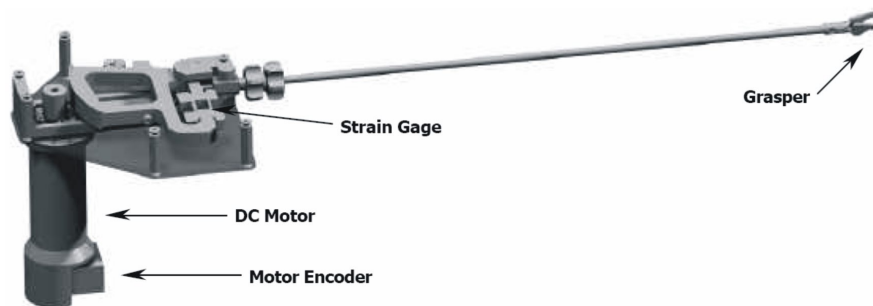


Figure 3.19 - D'après Brown (*et al.*) [Brown 03] (in-vivo, peu invasif) : outil chirurgical modifié (rendu 3D sans le capot de protection)

Fiene (*et al.*) [Fiene 06] mettent au point un dispositif interactif permettant de mesurer les forces ressenties lors d'un impact avec un outil de type « stylet » et une surface pour améliorer le réalisme du retour haptique (cf Fig. 3.20). Selon les auteurs, ce type d'impact permet à l'utilisateur de caractériser au mieux la dureté du matériau. L'outil est composé d'un capteur de force (« mini-40 », © *ATI Industrial Automation Inc.*) et d'un accéléromètre (« ADXL321 », © *Analog Devices* , $\pm 18g$). En outre, un second capteur de forces (© *Flexi-force*) permet également de mesurer la force de pression que l'utilisateur déploie pour maintenir l'outil. Le but de ces mesures est de créer une base de donnée de différents profils de forces et d'accélération pour moduler les signaux classiques de retour haptique.

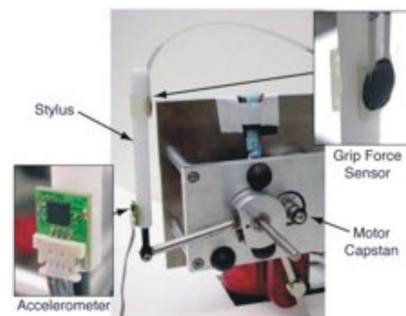


Figure 3.20 - D'après Fiene (*et al.*) [Fiene 06] : dispositif de mesure des forces d'impact entre un stylet et une surface

Kry et Paï [Kry 06] conçoivent un système permettant de mesurer les efforts en jeu lors de la saisie manuelle d'objets. Les positions des doigts et de la main sont capturées à l'aide d'un système infrarouge (« Motion capture system », © *Vicon*). Les efforts de pression exercés par les doigts sont mesurés à l'aide de capteurs utilisant des résistances sensibles aux efforts (*force sensitive resistors (FSRs)*, cf Fig. 3.21 a). Le diamètre de ces résistances est de 5mm et seuls les efforts normaux à la surface sont capturés. Ces capteurs, couplés au système de capture de mouvements, permettent l'acquisition de données lors de la saisie d'un objet (cf Fig. 3.21 b et c). Ces données sont alors réutilisées lors de la simulation de saisie d'un objet semblable (cf Fig. 3.21 d). Les auteurs définissent un algorithme permettant d'utiliser ces données pour simuler la saisie d'objets ayant une forme très différente (cf Fig. 3.21 e).

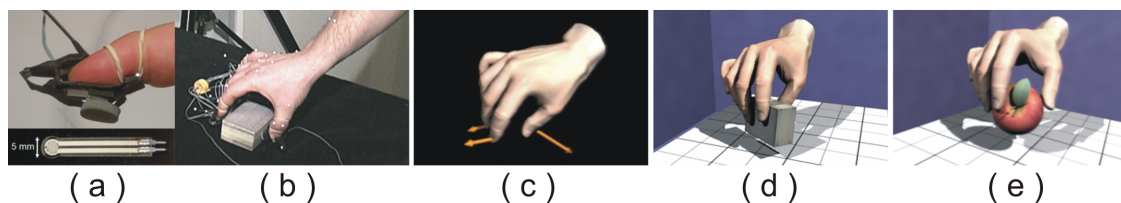


Figure 3.21 - D'après Kry et Paï [Kry 06] : dispositif de mesure des forces de pression lors de la saisie d'objet.

3.3.2 Dispositifs incluant l'approche visuelle

Nous tentons ici de regrouper des exemples de travaux incluant une approche de mesure visuelle au sens large du terme. En effet, que ce soit en imagerie classique (caméra ou appareil photo standard), en imagerie médicale (par IRM ou Tomographie), ou encore par visualisation

infrarouge, de nombreux travaux font appel à des techniques de visualisation et il est difficile d'en dresser ici une liste exhaustive. De plus, nous restreignons ces exemples aux travaux dont le résultat des visualisations effectuées est utilisé pour déterminer des paramètres mécaniques, bien souvent par méthode inverse (la partie estimation à partir des mesures effectuées est détaillée en 3.4).

En 1980, Courbon relate une méthode expérimentale visuelle pour déterminer le *coefficient de Poisson* à partir d'une poutre à section rectangulaire en flexion pure ([Courbon 80], page 8). Lorsque sa face supérieure est polie et que l'on place sur elle une plaque de verre coïncidant avec cette face, il se produit, après flexion, un vide d'épaisseur variable entre la plaque de verre et la surface déformée de la face supérieure. Cette épaisseur variable peut être mesurée par des procédés optiques : en dirigeant perpendiculairement à la plaque un faisceau de lumière monochromatique, on obtient des franges hyperboliques, analogues aux anneaux de Newton, qui dessinent les lignes de niveaux de la face supérieure de la poutre et permettent d'estimer le *coefficient de Poisson* du matériau.

Shi (*et al.*) [Shi 99] [Liu 04] [Shi 04] utilisent un scanner IRM pour faire la reconstruction surfacique en 3D d'une partie du coeur d'un chien. Un autre type d'imagerie, les *phases contrast images*, permet également d'obtenir la vitesse de chaque pixel à certain instants donnés. Les auteurs utilisent ensuite un modèle à éléments finis pour reconstruire l'animation entre ces instants. D'autres travaux se basent également sur l'imagerie IRM, par exemple Samani (*et al.*) [Samani 01] pour estimer les zones à risque lors d'une mammographie.

Vuskovic (*et al.*) [Vuskovic 01] [Kauer 01a] [Kauer 01b] se basent sur les outils microscopiques de type « pipette » utilisés dans les expérimentations à l'échelle cellulaire [Hochmuth 93]. Ils décrivent un outil interactif, destiné à faire des mesures *in-vivo peu invasives* et *in-vitro*, permettant d'aspirer une partie du matériau tout en filmant la déformation occasionnée (cf Fig. 3.22). La dépression appliquée est mesurée à l'aide d'un capteur de pression pendant toute la durée de l'expérimentation, ce qui génère un profil de la dépression appliquée (cf Fig. 3.22, en bas à droite). Conjugué au profil de la déformation extrait de chaque image filmée (fréquence $\approx 25Hz$), une courbe 3D représentant la déformation est générée (cf Fig. 3.22, en haut à droite). L'outil est utilisé pour des expérimentations *in-vivo* sur des utéri humains et *in-vitro* après *hystérectomie* (ablation de l'utérus). Ce dispositif est également utilisé par Nava (*et al.*) [Nava 03].

Kerdok (*et al.*) [Kerdok 01] proposent une méthode basée sur l'*imagerie par tomographie* (CT) pour caractériser la déformation d'un cube de silicone, appelé « TruthCube ». Lors du moulage du cube, des billes en « Teflon », © DuPont sont incorporées à la mixture selon un quadrillage donné et par couches successives. Le résultat est une grille 3D de billes moulées dans le silicone : cette grille se déforme en même temps que le cube. Des poids sont alors ajoutés manuellement sur une plaque positionnée sur la face supérieure du cube. La déformation résultante est « filmée » à l'aide d'une séquence d'images CT d'où est extraite une animation de grille 3D (cf Fig. 3.23). Le but de ce dispositif est de constituer une base de données des déformations pour permettre la validation de modèles mécaniques.

Le système de mesure ACME de Pai (*et al.*) [Pai 00] [Lang 02] inclus également un dispositif visuel. En effet, la déformation appliquée est également filmée par une caméra haute définition et par un système tri-oculaire vidéo couleur (« Color Triclops », © Point Grey Research , cf Fig. 3.24). Ce dernier permet de calculer une *carte de profondeur* (ou *depth map*) pour chaque image de l'animation et ainsi faire une reconstruction partielle en 3D de la déformation. Cet ensemble est également visible sur la figure 3.10, en haut à gauche.

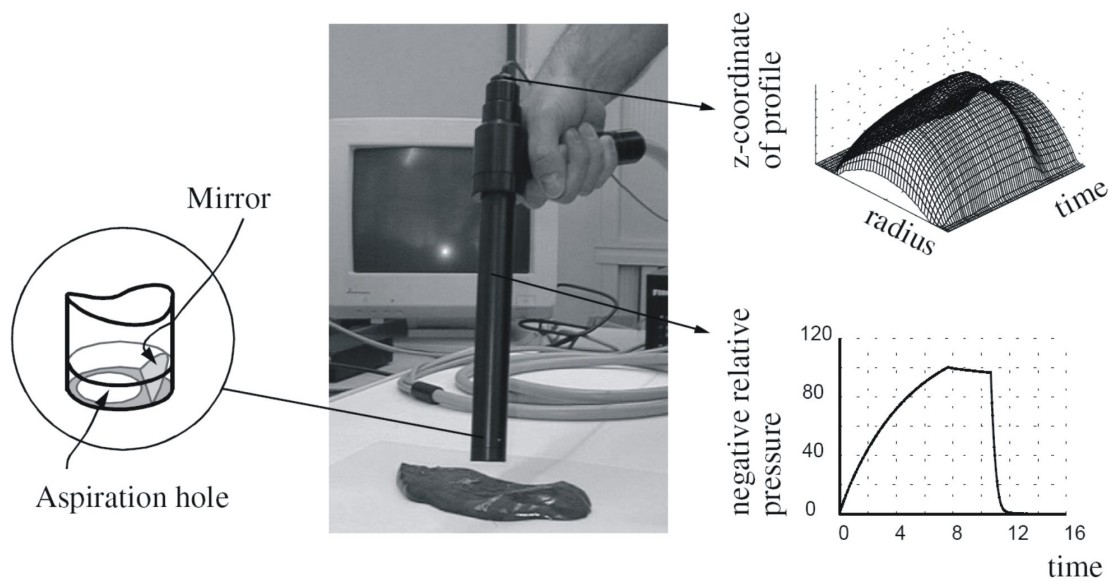


Figure 3.22 - D'après Kauer (et al.) [Kauer 01a] : Dispositif d'aspiration des tissus : la déformation de la partie aspirée est filmée grâce à un système de miroir et de micro-caméra (à gauche). Le profil de la déformation couplé à la dépression appliquée dans le temps (en bas à droite) génère un profil 3D de la déformation (en haut à droite).

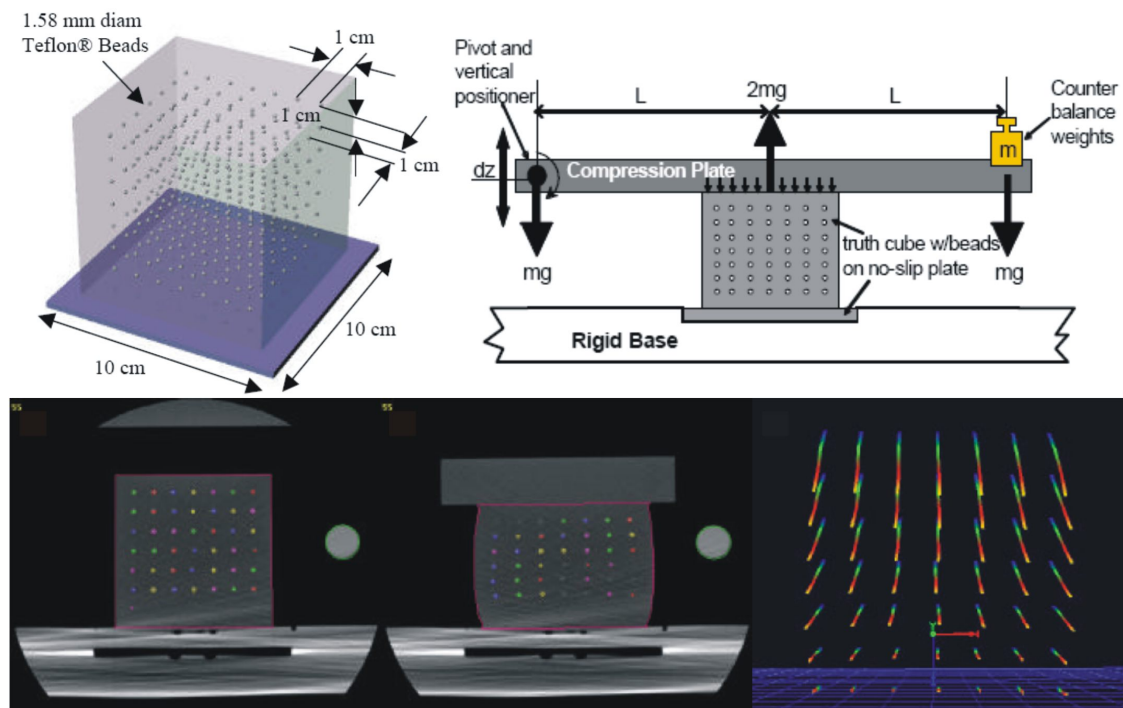


Figure 3.23 - D'après Kerdok (et al.) [Kerdok 01] : En haut - à gauche, le schéma 3D du « Truth-Cube », à droite, le schéma de l'expérimentation. En bas - à gauche et au centre, les images CT et l'extraction 3D de la grille respectivement avant et après l'application des poids, à droite, la visualisation 3D de la déformation.



Figure 3.24 - D'après Pritchard et Heidrich [Pritchard 03b] : Photo du système tri-oculaire « Color Triclops », © Point Grey Research .

Pritchard et Heidrich [Pritchard 03b] [Pritchard 03a] utilisent eux aussi le système « Color Triclops » (cf Fig. 3.24). Ils extraient ainsi la *carte de profondeur* pour chaque image d'une animation mettant en scène la déformation d'un rectangle de tissu. Ces données permettent d'estimer le mouvement de ce tissu lors d'une simulation à l'aide du modèle mécanique de Baraff et Witkin [Baraff 98].

Pour estimer les paramètres de ce même modèle mécanique, Bhat (*et al.*) [Bhat 03] proposent une nouvelle méthode. Ils emploient une caméra vidéo standard pour filmer les déformations appliquées à la main par un utilisateur sur un morceau rectangulaire de tissu. Le tissu est « secoué » devant la caméra. Pour réaliser l'acquisition du déplacement imposé, les auteurs utilisent un système de capture de mouvement infrarouge (« Motion capture system », © *Vicon*). De plus, un motif est projeté grâce à un vidéo projecteur pour pouvoir détecter par la suite les plis du tissu.

Toujours à l'aide d'une caméra vidéo, Greenspan et Fraser [Greenspan 03] proposent de capturer les mouvements d'un dipôle à sphères (tige sur laquelle sont fixées deux sphères, une à chaque extrémité). En connaissant les paramètres intrinsèques de la caméra et par extraction de contour, les auteurs déduisent la position 3D de l'outil. L'avantage de cette méthode est que l'outil est très simple, facile à construire et très peu onéreux. Cependant, les conditions d'éclairage doivent être maîtrisées pour retrouver la position 3D avec une précision suffisante.

Enfin, Samani et Plewes [Samani 07] font évoluer leur technique de mesure du module élastique de tumeurs cancéreuses du sein. Cette méthode, *ex-vivo*, permet, après prélèvement, découpe, et coloration des échantillons, de photographier des deux côtés de la coupe transversale le tissu et de recréer, après segmentation manuelle des images, un modèle volumique de l'échantillon qui sera utilisé dans une simulation mécanique par éléments finis.

3.4 Identification de paramètres

La section 3.3 expose de nombreux dispositifs, interactifs ou manuels pour acquérir les données du monde réel. On peut noter que nombre d'entre eux sont destinés au monde

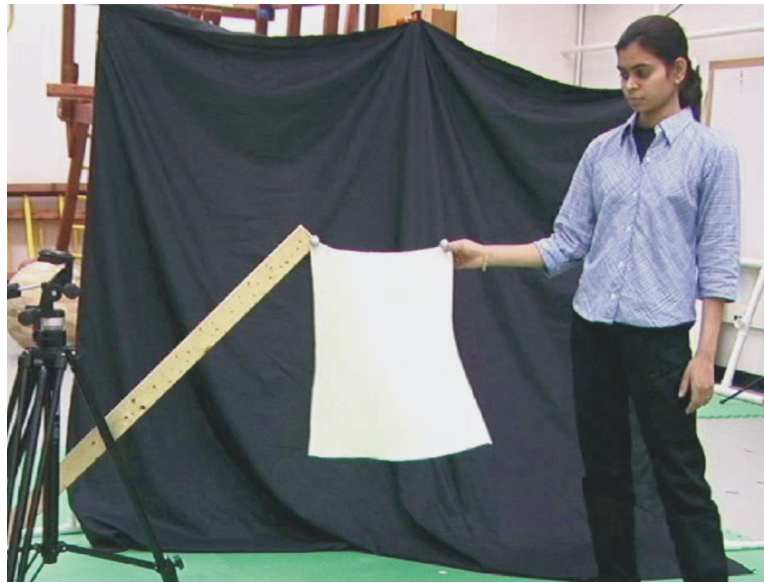


Figure 3.25 - D'après Bhat (et al.) [Bhat 03] : Les oscillations appliquées sur le carré de tissu par l'utilisateur sont capturées à l'aide d'un système de capture de mouvement infrarouge (« Motion capture system », © Vicon, les deux sphères grises sont les marqueurs). Le tout est filmé par une caméra standard (extrait de la vidéo fournie avec l'article).

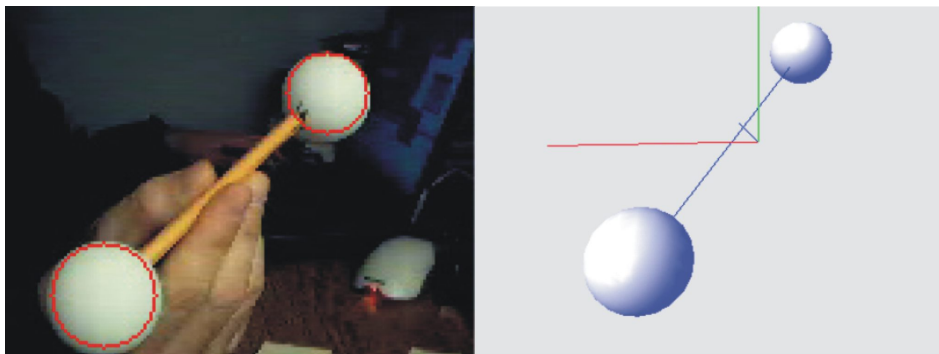


Figure 3.26 - D'après Greenspan et Fraser [Greenspan 03] : La position 3D d'un dipôle à sphère est estimée à l'aide d'une extraction de contour depuis une caméra standard.

médical en général [Miller 00] [Brouwer 01] [Ottensmeyer 01] [Bruyns 02] [Ottensmeyer 02] [Ottensmeyer 04] [Delalleau 05] [Samur 05] [Sedef 06] [Samur 07] [Benech 05] [Ostrander 92] [Brouwer 01] [Brown 03] [Shi 99] [Liu 04] [Shi 04] [Vuskovic 01]. En effet, une grande partie de la littérature sur l'estimation de paramètres porte sur ce domaine [Kauer 01a] [Nava 03] [Liu 04] [Samani 01] [Samani 07]. Les types de comportements que l'on y rencontre sont très souvent similaires puisque le but est généralement d'estimer les propriétés mécaniques d'organes, la plupart de ceux-ci étant composés en grande partie d'eau. Les algorithmes qui tentent de les reproduire ont par conséquent bon nombre de paramètres en communs et les techniques qui tentent de les estimer peuvent être classées dans un groupe commun. Il est donc possible de diviser en deux parties ce survol des techniques d'estimation de propriétés mécaniques à partir de données réelles : dans le domaine médical puis les autres travaux, comme par exemple l'estimation des propriétés de tissus ou l'animation synthétique en général.

3.4.1 Médical

Les travaux de Kauer (*et al.*) [Kauer 01a] [Kauer 01b] proposent une méthode très complète pour estimer de nombreux paramètres à partir de mesures sur des organes effectuées par le dispositif de Vuskovic [Vuskovic 01]. Ils constituent donc une contribution majeure que nous détaillons ici avant de présenter d'autres méthodes existantes dans le monde médical.

Les auteurs représentent tout d'abord les organes par un modèle à éléments finis suivant la loi de Sussman et Bathe [Sussman 87], une autre extension du *modèle Neo-Hookean* (cf. 2.2.4). Cette loi permet de modéliser les matériaux proches de l'incompressibilité (Coefficient de Poisson $\nu \rightarrow 0.5$ ou Bulk modulus $\rightarrow \infty$). Cette hypothèse est donc valable pour la plupart des organes, composés en majeure partie d'eau. De plus, ce modèle permet de simuler des comportements non-linéaires. La complexité de cette non-linéarité est représentée par un *degré de non-linéarité* N . Les auteurs déterminent ce nombre par expérimentations. Le modèle permet également de simuler la visco-élasticité 2.1.2. Les auteurs doivent alors fixer N temps de relaxation τ_i . Au total, $2 \times N$ paramètres doivent être estimés (μ_i et c_i). Ces paramètres n'ont pas de représentations physiques concrètes, excepté $\mu_1 \approx$ *module de Young* E .

Pour ce faire, les auteurs utilisent une méthode d'estimation inverse. La simulation mécanique s'effectue sur un modèle axisymétrique en deux dimensions permettant une simulation mécanique en temps « acceptable » (pas de détails fournis). Cela est possible car les contraintes imposées par le dispositif d'aspiration génère une déformation « contrôlée » de forme partiellement ovoïde (cf. Fig. 3.22). Cependant, cette approximation induit plusieurs hypothèses. Tout d'abord, le matériau est considéré isotrope (cf 2.1.4). Ensuite, l'épaisseur sous-jacente doit être fixée par les auteurs. Enfin, le pré-stress du matériau est négligé.

L'estimation inverse est réalisée en se basant sur une métrique de comparaison de contours. Les contours sont dans un premier temps extraits de la vidéo réelle. Ensuite, à chaque itération de la simulation mécanique, le contour du « bulbe » issu de cette simulation est également extrait avant une comparaison par moindres carrés. Tous les paramètres sont ainsi estimés à la fois et l'optimisation est réalisée à l'aide de l'algorithme de Levenberg-Marquard (cf 3.2.1.1). Les auteurs valident expérimentalement cette méthode sur un objet en silicone « Sil-gel ». Quatre tests préliminaires sont effectués pour estimer N puis, pour les τ_i fixés, quatre paramètres sont estimés ($\mu_1, c_0, c_1, c_2, c_3$). Enfin, des mesures et estimations sont réalisées sur 5 utéri, in-vivo et ex-vivo à trois endroits différents. Cinq paramètres sont alors estimés : $\mu_1, \mu_2, c_0, c_1, c_2$. Les résultats montrent une perte de raideur entre les expérimentations in-vivo et ex-vivo. Ils montrent également une grande variabilité entre différents utéri, plus importante que sur un même utérus à différents endroits. Les auteurs notent cependant que la force de contact semble perturber les mesures.

Un peu plus tard, Nava (*et al.*) [Nava 03] s'inspirent de ces travaux. Les auteurs utilisent également le dispositif développé par Vuskovic [Vuskovic 01] pour pratiquer des expérimentations ex-vivo sur deux reins et trois foies (post-mortem). Ils utilisent cette fois-ci deux modèles. Le premier est un modèle « simple » à une dimension. Il s'agit d'un ressort amorti en série avec un modèle de friction amorti (Coulomb). Celui-ci permet de déterminer la raideur du ressort et de classer les échantillons en trois catégories : les reins, plus « mous », les foies « normaux », plus durs, et les foies « malades » de raideurs intermédiaires. Une seconde estimation sur un modèle plus complexe, avec un modèle mécanique par éléments finis uniaxial semblable à celui utilisé par Kauer (*et al.*) [Kauer 01a] révèle, pour des expérimentations de traction et de relaxation, des comportements similaires (60 heures pour une simulation de 400

itérations sur un P4 2GHz, optimisation Nelder-Mead simplex).

Une autre technologie est utilisée dans plusieurs travaux pour estimer le module de Young dans un matériau non-homogène (cf 2.1.5). Il s'agit d'imagerie par scanner à résonance magnétique (IRM). Ce type de dispositif permet non seulement d'obtenir plusieurs « images » représentant les coupes du volume scanné et une estimation des densités des matériaux mais également les « champs de déplacements » correspondant. Liu (*et al.*) [Liu 04] proposent une formulation dynamique d'un modèle par éléments finis suivant une loi d'élasticité linéaire ainsi que quatre méthodes d'estimation du module de Young dans un matériau non-homogène (pour un coefficient de Poisson $\nu = 0.47$) : une estimation directe par moindres carrées (LS), une estimation itérative par filtre de Kalman étendu (EKF) et deux autres estimations itératives par deux filtres \mathcal{H}_∞ (FSDI et NPFSI). Ils valident tout d'abord ces modèles sur des données synthétiques. Un modèle géométrique par éléments finis est créé en deux dimensions. Celui-ci comporte deux inclusions où le module de Young est plus important et plus faible que sur le reste du maillage. Ils génèrent ensuite un jeu de données en perturbant ce modèle par deux niveaux de bruit gaussien (30dB et 20dB) et deux niveaux de bruit de Poisson (30dB et 20dB). La méthode LS donne d'excellents résultats quand il n'y a pas ou peu de bruit. Cependant, elle échoue lorsque le bruit est important et n'est donc pas fiable lors de cas réels. La méthode EKF donne de moins bons résultats quand le bruit est nul mais est très performante pour l'estimation lorsque les données sont perturbées par un bruit Gaussien. Malheureusement, la convergence est assez lente et le résultat dépend de la valeur d'initialisation du module de Young. La méthode FSDI échoue à retrouver les paramètres quand le bruit est fort. En revanche, la méthode NPFSI donne les meilleurs résultats quand le bruit n'est pas Gaussien, converge très rapidement quand c'est le cas, et n'est pas affectée par la valeur d'initialisation du module de Young. Cette dernière semble donc la mieux adaptée à des estimations à partir de données réelles et donc perturbées par un bruit de type non déterminé. Les auteurs appliquent cette méthode sur des données issues d'un scanner IRM à contraste de phase permettant d'obtenir en outre la vitesse des tissus de la cloison cardiaque d'un chien. La distribution du module de Young retrouvée correspond à un examen post-mortem du myocarde où la région de l'infarctus est mise en évidence par le laboratoire d'imagerie.

Samani et al. [Samani 01] utilisent également des images IRM pour reconstruire une carte de distribution non homogène du module de Young pour une mammographie. Ils utilisent la *loi de Hooke* en fixant le coefficient de Poisson à 0.499 pour simuler la quasi incompressibilité du milieu mammaire. Grâce à une modélisation par éléments finis recréant les conditions d'une mammographie (compression entre deux plaques) et une estimation inverse en se basant sur les images issues du scanner, ils parviennent à déterminer les zones qui risquent de contenir une tumeur cancéreuse, où le module Young est plus important que dans le milieu environnant. La suite de ces travaux [Samani 07] montre, à partir de tissus prélevés et de segmentation manuelle des images obtenues, que ce rapport est d'environ 5 pour des lésions bénignes et d'environ 10 pour les tumeurs malignes.

Les estimations des propriétés mécaniques des matériaux peuvent également être appliquées à des échelles très petites comme le prouvent Boudou et al. [Boudou 06]. Ils réalisent une estimation du module de Young et du coefficient de Poisson pour des tissus biologiques fins et adhérents par simulation de l'expérimentation de succion à l'aide de micropipettes. Pour cela, ils n'utilisent pas de méthode inverse à proprement parler mais une « base d'expérimentation ». Ils réalisent la création d'une base de données par simulation mécanique pour environ 2000 pipettes différentes et développent une formule permettant de *fitter* ces

paramètres. Ils définissent alors une procédure pour estimer le coefficient de Poisson à partir de la force de suction appliquée à la pipette et la longueur d'échantillon aspirée mesurée par imagerie microscopique. Ils en déduisent alors le module de Young en utilisant la formule issue de la base de données.

3.4.2 Non médical

On peut également dénombrer plusieurs travaux d'estimation de paramètres à partir d'images dans d'autres domaines que la médecine. Lang (*et al.*) [Lang 02] proposent par exemple une méthode d'acquisition de modèles déformables élastiques pour la simulation interactive. Le but est d'identifier itérativement des parties de la matrice des fonctions de Green (élasticité linéaire, cf 2.2.2) qui relie les forces aux déplacements des points de la surface de l'objet. Ils utilisent le système ACME [Pai 00] (cf Fig.3.10) pour mesurer automatiquement les forces et les déplacements locaux sous plusieurs angles (par un capteur 6-axes monté sur un bras robotisé). Les déplacements globaux sont ensuite estimés par un flux optique stéréo (ici réalisé depuis trois caméras « Triclops ») et de leur méthode « range-flow ». Celle-ci combine les flux optiques avec les cartes de profondeur issues de la vision stéréoscopique. Pour cela, l'objet déformé doit montrer suffisamment d'informations visuelles (« texture »). Les données manquantes dans la matrice des fonctions de Green sont ensuite estimées par interpolation. Les auteurs ont réalisé ces mesures sur une peluche et sur un modèle de bras anatomique médical déformable.

Plus récemment, Bickel (*et al.*) [Bickel 09] proposent un autre modèle de simulation mécanique d'objets déformables à partir de données du monde réel. Des mesures préliminaires sont effectuées en plusieurs endroits d'un objet déformable réel à l'aide d'un capteur de forces et d'un système de trois caméras. À chaque mesure correspond un échantillon dans l'espace déformations/déplacements, selon la loi d'élasticité linéaire. Les auteurs utilisent ensuite des fonctions d'interpolation dans cet espace pour synthétiser et recréer des comportements non-linéaires.

Dans un autre registre, Tao (*et al.*) [Tao 02] définissent un modèle volumique d'un visage par deux surfaces de Bézier (PVBD) (une couche interne et une couche externe). Celui-ci permet de synthétiser des expressions et de recréer une séquence complète d'un visage synthétique animé lisant une phrase (par phonèmes) ainsi que plusieurs expressions (sourire, peur, ...). Pour s'adapter à la morphologie et aux expressions d'une personne réelle, les auteurs développent une méthode d'estimation des paramètres à partir de vidéo. Un grand nombre de paramètres doit alors être « retrouvé ». En effet, aux paramètres de déformation rigide (matrice de translation et de rotation 3D globale) viennent s'ajouter les « paramètres déformables », c'est à dire 4×2 points de contrôle (surfaces de bezier cubiques sur deux couches) par élément volumique multiplié par le nombre d'éléments composant le visage synthétique. Les points « d'intérêt » (coins des yeux, etc) sont sélectionnés par l'utilisateur à la main sur la première image de la vidéo (visage en position neutre). Ils estiment ensuite par méthode inverse tous les paramètres à partir de la vidéo en utilisant une fonction objective par moindres carrés. Celle-ci s'avère efficace car le PVBD est un système linéaire. Les résultats peuvent être comprimés et décompressés à l'aide d'un codec développé par les auteurs pour une compatibilité MPEG4 ou, par exemple, effectuer un dialogue à distance sans flux vidéo continu.

Toujours dans le domaine de l'animation humaine, Liu *et al.* [Liu 05] définissent un mo-

dèle dynamique du mouvement pour la simulation de la marche. Pour paramétrer ce modèle pouvant comporter un grand nombre de variables (jusqu'à 147), les auteurs utilisent un apprentissage par estimation inverse à partir de données issues d'un système de capture de mouvements.

La section 3.1.3 montre un exemple d'estimation de paramètres par méthode inverse à partir de vidéos. Il s'agit des travaux de Bhat (*et al.*) [Bhat 03] qui proposent une méthode pour tenter d'estimer les paramètres mécaniques d'un modèle de simulation de tissus [Baraff 03]. Les auteurs filment dans un premier temps des morceaux de tissu rectangulaires sur lesquels est projeté un motif lumineux structuré. Ils tentent d'approcher le comportement des tissus filmés en générant itérativement des séquences d'images synthétiques à partir du modèle de simulation mécanique. À l'aide d'une méthode d'optimisation par recuit simulé (cf 3.2.1.2), la méthode approxime les différents paramètres de l'algorithme (*flexion, traction, compression, cisaillement, amortissement, ...*). Les paramètres ainsi estimés pour plusieurs tissus (lin, laine, satin et tricot) permettent dans un second temps de transposer les comportements obtenus à partir de simples morceaux rectangulaires de tissus dans des scènes complexes comme la jupe d'une danseuse. Cependant, ces résultats ne font pas l'objet d'une validation objective, par exemple avec système de mesure de Kawabata [Kawabata 80]. C'est pourquoi, plus récemment, de nombreux travaux proposent des méthodes pour capturer et recréer les mouvements de tissus directement depuis des scènes complexes mais sans tenter de retrouver les paramètres de modèles mécaniques sous-jacents [White 07] [Scholz 05] [Hasler 06] [Bradley 08].

3.5 Proposition

Dans ces travaux, nous désirons réaliser l'estimation de paramètres à partir de séquences d'images montrant un objet déformable volumique quelconque sous une contrainte externe. Le dispositif de Vuskovic [Vuskovic 01] permet à la fois d'appliquer une déformation par succion et de filmer cette déformation pour pouvoir traiter les informations ultérieurement. Ce mode de fonctionnement réalise de bonnes conditions pour l'estimation de propriétés mécaniques [Kauer 01a] [Nava 03]. Cependant, ce dispositif est conçu pour une application *in-vivo* lors d'un acte opératoire en chirurgie. Bien que l'application de la contrainte par aspiration soit très adaptée aux matériaux humides comme c'est le cas pour les organes, celle-ci nous paraît relativement limitée dans d'autres contextes. En effet, il est impossible par exemple d'utiliser ce matériel sur des blocs de mousse. Nous lui préférons donc une méthode par compression, notamment utilisée par les dispositifs de mesure à indenteurs.

Cependant, à notre connaissance, aucun dispositif interactif de mesure d'effort par compression n'est disponible actuellement. C'est pourquoi nous proposons ici un outil de type « stylet », ergonomique, peu encombrant, et permettant d'appliquer et de mesurer une force sans aucune hypothèse sur le point d'application de l'effort. L'un des buts de ce travail étant de proposer une solution matérielle peu onéreuse et facile à mettre en place, nous préférons un outil interactif à un outil automatique dont le montant ou le coût de fabrication est en général plus important à cause de l'intégration d'actionneurs. De plus, nous utilisons un capteur uniaxial bien moins coûteux que son homologue à trois ou six dimensions [Pai 00]. Nous concevons notamment un montage original incluant une « bille de transmission » permettant de réduire de manière significative la transmission au capteur des efforts tangentiels parasites. De plus, nous utilisons une touche sphérique qui permettra par la suite de simplifier le modèle géométrique utilisé dans la simulation mécanique (contrairement à Kauer

(*et al.*) [Kauer 01a]). Par ailleurs, nous ajoutons à ce dispositif la possibilité de s'adapter à plusieurs périphériques de positionnement virtuel, ou *tracking 6D*, puis nous testons trois méthodes existantes : magnétique (très grande liberté de mouvements), mécanique (montage sur un périphérique haptique disponible au laboratoire, très précis) et visuelle (par marqueurs, très facile à mettre en place et gratuit). Nous préférons cette dernière méthode aux coûteux dispositifs à capteurs infrarouge [Bhat 03] [Liu 05].

Pour filmer la déformation obtenue nous utilisons une caméra standard contrairement à de nombreux travaux utilisant des caméras stéréoscopiques moins bon marché [Pritchard 03b] [Pai 00] [Lang 02]. Pour contrôler la source lumineuse, nous nous basons sur les travaux de Bhat (*et al.*) [Bhat 03] et utilisons un vidéo projecteur standard. Cela nous permet en outre de projeter des motifs structurés sur la scène filmée. Nous proposons une méthode interactive originale pour calibrer les paramètres intrinsèques de ce vidéo projecteur ainsi qu'une méthode couplée pour le positionnement de ces deux dispositifs optiques.

De nombreux travaux portant sur l'estimation de paramètres ou l'extraction d'informations à partir de séquences d'images reposent sur un modèle de simulation mécanique sous-jacent [Kauer 01a] [Bhat 03] [Nava 03] [Hasler 06] [Liu 05]. Lorsqu'il s'agit d'objet volumiques, les auteurs utilisent généralement une modélisation par éléments finis (cf 2.3.2). Dans ce cas, les travaux proposant des méthodes d'estimation de paramètres d'objets déformables à partir du dispositif de Vuskovic [Vuskovic 01] bénéficient de conditions aux limites très bien définies. En effet, l'aspiration crée un « bulbe » qui peut être très facilement simulé. Le modèle géométrique par éléments finis peut être très simplifié et surtout reste le même, quel que soit l'objet sur lequel est réalisé l'expérimentation. De plus, si on fait l'hypothèse d'un matériau isotrope (cf 2.1.4), il est possible de ramener ce modèle à une formulation en deux dimensions puis d'effectuer une résolution axi-symétrique. Cela accélère grandement les calculs et il est alors possible d'utiliser des lois de comportement relativement complexes, simulant par exemple des comportements non-linéaires (cf 2.2.4). Nous voyons cependant deux problèmes majeurs à ce type d'estimation. Le premier est que dans ces modèles de comportement complexe, très peu de paramètres ont une signification physique réelle (par exemple les constantes empiriques du modèle de *Mooney-Rivlin*, cf 2.2.4). Les valeurs trouvées pour ces paramètres ne sont donc en aucune façon réutilisables avec d'autres modèles de simulation. Le second problème est que dans de nombreux cas, et tout particulièrement dans le domaine médical, les paramètres ainsi estimés servent à configurer des logiciels de simulation mécanique en temps réel, comme les logiciels d'apprentissage aux gestes chirurgicaux. Or, les lois de comportements complexes sont à l'heure actuelle très difficilement intégrables dans ce type de simulateur.

Nous proposons donc d'estimer deux des paramètres les plus utilisés dans les modèles par éléments finis simulant des matériaux homogènes et isotropes : le *module de Young* et le *coefficient de Poisson*. En effet ces paramètres sont pratiquement indispensables dans toutes les simulations en temps interactif à base d'éléments finis [Cotin 97] [Bro-Nielsen 96] [Felippa 00]. De plus, ces paramètres ont une signification physique réelle (cf 2.4.1 et 2.4.2) et peuvent donc être en général utilisés par plusieurs modèles mécaniques différents. Nous utilisons ici le modèle décrit par Duriez [Duriez 04] qui propose un bon compromis entre réalisme, rapidité d'exécution et généralité des paramètres. Il inclut notamment la résolution par méthode corotationnelle qui permet de conserver une certaine intégrité lors de grands déplacements en formulation statique (cf 2.3.2.2). Ce modèle mécanique n'est pas basé directement sur des données du monde réel (*data-driven models*, [Lang 02] [Bickel 09]). Par conséquent, les esti-

mations de paramètres réalisées peuvent être réutilisées dans d'autres modèles géométriques sans effectuer d'autres mesures sur des objets réels.

Tous les travaux actuels sont, à notre connaissance, basés sur une estimation unique de l'ensemble des paramètres à la fois [Bhat 03] [Kauer 01a] [Nava 03]. Or, l'espace de recherche peut être alors très grand (de l'ordre de np^{np} pour np paramètres) et le nombre d'itérations nécessaires à l'estimation très important, même avec une bonne méthode d'optimisation. Or, il nous semble préférable, lorsque cela est possible, de réaliser np estimations successives pour réduire l'espace de recherche à $np \times np$. Nous prouvons dans notre cas la faisabilité de cette technique et proposons une méthode pour réaliser cette étude dans un cas d'estimation inverse quelconque. De plus, nous montrons qu'il est indispensable de déterminer un ordre dans lequel les estimations successives doivent être effectuées et à partir de quelles données les réaliser.

Le coefficient de Poisson influence la géométrie de l'objet déformé. Cependant, nous avons choisi, pour plus de liberté pour l'utilisateur, de se baser sur une déformation « manuelle » de l'objet par compression. De plus, pour plus de généralité, nous ne faisons pas d'hypothèses sur la géométrie initiale de cet objet, celui-ci pouvant être quelconque. Nous ne pouvons donc estimer directement ce paramètre comme le font Kerdok (*et al.*) [Kerdok 01] sur le *Truthcube* par exemple. Nous proposons donc un algorithme d'estimation par méthode inverse depuis une scène en trois dimensions complète. Pour ce faire, nous définissons plusieurs *métriques d'erreur* nous permettant de quantifier la différence entre deux séquences d'images. Nous testons ces métriques de manière exhaustive sur des scènes relativement simples où des objets de silicone sont déformés. Pour accélérer ce processus d'estimation, nous choisissons ensuite une méthode d'optimisation globale robuste avant d'appliquer ce processus à des scènes plus complexes.

Le module de Young influence quant à lui les efforts générés par le matériau en réponse à la contrainte appliquée. Puisque nous avons défini un nouveau dispositif d'application et de mesure de contrainte, nous définissons une méthode d'estimation du module de Young adaptée. Nous proposons tout d'abord une méthode originale d'estimation directe et robuste aux bruits de mesure (utilisation d'un *RanSaC* [Fischler 81]). Cette méthode faisant de nombreuses hypothèses sur l'objet déformé, nous définissons ensuite une seconde méthode, basée sur une estimation inverse, qui nous permet de s'affranchir de ces hypothèses. Ces deux méthodes sont validées sur des objets en silicone de diverses forme par des mesures sur le système Zwick [Zwick]. Nous appliquons ensuite ces techniques sur d'autres types de matériaux et d'objets.

Quatrième Chapitre

LE DISPOSITIF EXPÉRIMENTAL

Ce chapitre décrit le dispositif expérimental que nous avons conçu. Il est composé de trois parties. La première présente le dispositif optique que nous avons choisi, basé sur une caméra vidéo et un vidéo-projecteur standard. Nous y décrivons la mise en œuvre de ce dispositif et les méthodes de calibrage des paramètres intrinsèques et extrinsèques. Une méthode de calibrage originale pour vidéo-projecteur est proposée. La seconde partie décrit le dispositif interactif de mesure des forces appliquées par l'utilisateur. Nous proposons un nouveau montage basé sur un capteur de forces pour obtenir un outil ergonomique de type « stylet ». Enfin, la dernière partie expose les fonctionnalités du module d'acquisition que nous avons développé pour permettre de synchroniser les informations provenant de ces différentes sources matérielles.

4.1 Introduction

Lors de l'élaboration de ce projet, nous nous sommes fixé pour directive principale une méthode ne nécessitant qu'un ensemble matériel peu coûteux, facile à mettre en œuvre ou à construire et simple d'accès. Nous avons vu dans les chapitres précédents que les données nécessaires à l'estimation de paramètres d'objets déformables sont :

- une séquence d'images capturant visuellement la déformation de l'objet
- une mesure de la contrainte appliquée en termes de position et d'« intensité »

Pour permettre de réaliser une estimation uniquement de manière visuelle, la position et l'intensité de la contrainte doivent nécessairement être visible sur la séquence d'image. Nous verrons que l'évolution de notre dispositif de capture permet de retrouver la position de la contrainte à partir de la vidéo. Cependant, il reste difficile de traduire visuellement l'intensité de l'effort appliqué par l'utilisateur. Il est possible par exemple d'utiliser un ressort dont la raideur est connue, mais nous verrons que la précision de la mesure est importante.

En prenant en compte ces considérations, nous avons choisi de débiter ces travaux par un dispositif composé d'une part d'un ensemble de capture vidéo et d'autre part, d'un matériel de capture de position et d'effort. Nous décrivons ici ces deux ensembles et leurs évolutions, ainsi que les techniques que nous avons utilisées pour mettre en relation les différentes parties qui les composent.

4.2 Dispositif optique

4.2.1 Description

Pour nous permettre d'estimer visuellement des paramètres, nous avons fait le choix de nous baser sur un dispositif de capture d'images standard. Nous avons opté pour une *caméra vidéo* grand public (« Handycam DCR-TRV950E », © *Sony*)¹. Cette caméra, accessible pour un budget relativement modeste (moins de 1000 euros), propose des performances que nous jugeons suffisantes pour nos travaux en termes de définition d'image et de fréquence d'acquisition (capteur 3CCD, $3 \times 1.070.000 \text{ pixels}$, acquisition au format DV, 720×576 entrelacé, 60 Hz).

De plus, dans un premier temps, nous décidons de contrôler les conditions d'illumination. Pour simplifier au maximum la complexité du dispositif, nous choisissons d'utiliser une source unique. Pour être efficace, celle-ci doit être assez puissante pour que les sources ambiantes (lumière du jour) soient négligeables ($Intensite^{proj} \gg Intensite^{amb}$). Nous choisissons ici un *vidéo-projecteur*. L'avantage d'un vidéo-projecteur sur une source classique (spot par exemple) est que la couleur de la lumière est paramétrable. De plus, il est également possible de projeter des motifs complexes par programmation. Nous utilisons de tels motifs pour le calibrage du vidéo projecteur virtuel (cf 4.2.4). Par ailleurs, le vidéo projecteur nous permettra par la suite de projeter de motifs spéciaux dédiés à l'estimation (cf 7.2).

La méthode que nous développons ne nécessite pas de caractéristiques techniques contraignantes pour le vidéo-projecteur. Outre une résolution de projection raisonnable ($\approx 1024 \times 768$ pixels), la seule contrainte est le taux de rafraîchissement vertical qui doit être supérieur ou égal à celui de la caméra vidéo. Dans le cas contraire, des artefacts visuels (trames colorées par exemple) peuvent apparaître sur la séquence vidéo enregistrée. Notre choix s'est porté sur un vidéo-projecteur standard également (« MP-3130 », © *Hewlett Packard*)² qui propose de plus une option de fixation sur un trépied vidéo. Cette option est très utile car elle permet une projection plongeante (cf Fig. 4.1).

4.2.2 Généralités sur le calibrage

Filmer uniquement la déformation de l'objet n'est pas suffisant. En effet, pour réaliser l'estimation par méthode inverse (cf 3.1.1) il est nécessaire de disposer d'un minimum d'informations sur les positions de chaque partie du dispositif. De cette manière, il sera possible de recréer virtuellement l'environnement **avec précision**. Une des clés de la réussite de l'estimation est cette précision : l'adéquation entre l'environnement virtuel et l'environnement réel doit être maximale.

Cependant, nous nous sommes fixé pour but une simplification maximale de la manipulation. Pour que l'ensemble puisse être utilisé de la manière la plus rapide possible, le nombre d'étapes d'une manipulation doit être réduit au minimum. Pour atteindre la précision nécessaire et minimiser le temps de préparation d'une acquisition, nous utilisons des bibliothèques de *positionnement automatique à partir d'images* (ou *tracking*) : *ARToolkit*³ et *ARToolkitPlus*⁴ (basées sur [Zhang 98][Kato 99]).

1. <http://www.sony.fr/view/ShowProduct.action?product=DCR-TRV950E>

2. <http://h10032.www1.hp.com/ctg/Manual/c00063469.pdf>

3. http://www.hitl.washington.edu/projects/shared_space/

4. http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php



Figure 4.1 - Vidéo projecteur sur trépied (« MP-3130 », © Hewlett Packard) : projection plongeante.

Le calibrage d'un dispositif visuel consiste à retrouver, par mesure ou par calcul, tous les éléments qui permettront de reproduire virtuellement les images capturées. Ces paramètres dépendent du modèle virtuel. En général, il s'agit d'un modèle « trou d'épingle » (ou *pinhole*), valide pour les caméras vidéos comme pour les vidéo-projecteurs. Ces paramètres peuvent être séparés en deux catégories :

- Les **paramètres intrinsèques** : Ce sont les différents éléments internes au dispositif, par exemple la *focale* ou, le *centre optique*. Selon le modèle virtuel, d'autres paramètres peuvent venir complexifier cet ensemble, par exemple la *distortion optique* ou la *distortion radiale* pour un modèle *pinhole* évolué, prenant en compte la distortion. En règle générale, ces paramètres sont fixes lors de la capture d'une séquence d'image. Ils peuvent donc être calculés lors d'une étape préalable (*off-line*). Cependant, certains paramètres intrinsèques peuvent varier. C'est notamment le cas de la *focale* lorsque le zoom est modifié. Nous choisissons donc d'imposer une focale fixe (zoom fixe). Ceci peut être obtenu très simplement en se positionnant sur le zoom minimum. Il est ainsi facile de revenir à la focale voulue lorsque celle-ci a été modifiée par une autre utilisation du dispositif.
- Les **paramètres extrinsèques** : Ce sont les mouvements rigides de la caméra qui représentent la position et l'orientation de celle-ci dans l'environnement. Ils sont donc variables si la caméra se déplace (calcul *on-line*) lors de la capture d'images et fixes si celle-ci ne bouge pas. Dans ce dernier cas ils devront cependant être déterminés à l'initialisation de la capture (calcul *off-line*).

4.2.3 Calibrage de la caméra vidéo : *paramètres intrinsèques*

Les bibliothèques nécessitent un calibrage des *paramètres intrinsèques* avant de pouvoir retrouver les *paramètres extrinsèques*. Il est donc nécessaire de procéder à une étape de calibrage préalable de ces paramètres pour la caméra. Il existe plusieurs méthodes. L'une, assez simple, consiste à repérer les points d'une cible sous plusieurs angles : les points sont entrés manuellement par l'utilisateur (cf Fig. 4.2, à gauche). Une autre, plus contraignante, nécessite un positionnement précis (mesuré) de la caméra par rapport à la même cible. À chaque itération, la caméra doit être reculée précisément de 10cm. Des lignes horizontales et verticales doivent être superposées à l'image. Cette dernière opération est réalisée après la manipulation de la méthode « simple » (cf Fig. 4.2, à droite).

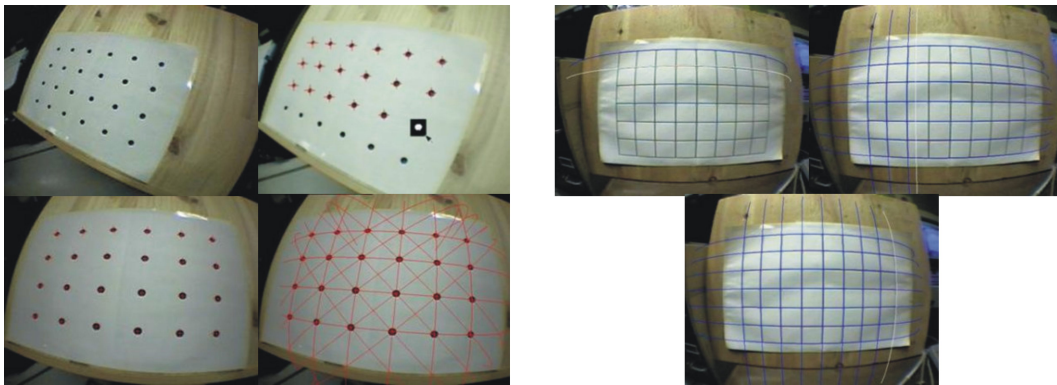


Figure 4.2 - Calibrage de la caméra à l'aide d'ARToolkit : à gauche, une illustration la méthode simplifiée et son résultat, à droite, une itération de l'étape suivante pour la méthode plus précise.⁶

Il existe également une alternative : il est possible de retrouver ces paramètres à l'aide d'un algorithme implémenté pour le logiciel « Matlab », © The MathWorks : le *Camera Calibration Toolbox*. Le fonctionnement de cet algorithme, développé par Bouguet (*et al.*) [Bouguet 04] est sensiblement identique à la version simplifiée dans *ARToolkit* mais plus précis et plus complet : on présente à la caméra un quadrillage sous différents angles de vue, on pointe manuellement les quatre intersections extérieures, on corrige manuellement le résultat de la détection d'intersections si besoin, puis on lance un processus itératif qui calcule les *paramètres intrinsèques* de la caméra (cf Fig. 4.3). De plus, les résultats de ce calcul sont utilisables avec la bibliothèque *ARToolkitPlus*, une évolution d'*ARToolkit*.

Nous utilisons cette dernière méthode pour calibrer les paramètres intrinsèques de notre caméra vidéo. Nous présentons un damier en noir et blanc pour capturer des images sous plusieurs angles et entrons manuellement les intersections dans le repère 2D de la caméra à l'aide du *Camera Calibration Toolbox*. Nous obtenons donc la focale, le centre optique et plusieurs paramètres de distortion optique.

6. Images tirées de <http://www.hitl.washington.edu/artoolkit/documentation/usercalibration.htm>

8. Images tirées de http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html. Plus d'étapes sont disponibles sur cette page.

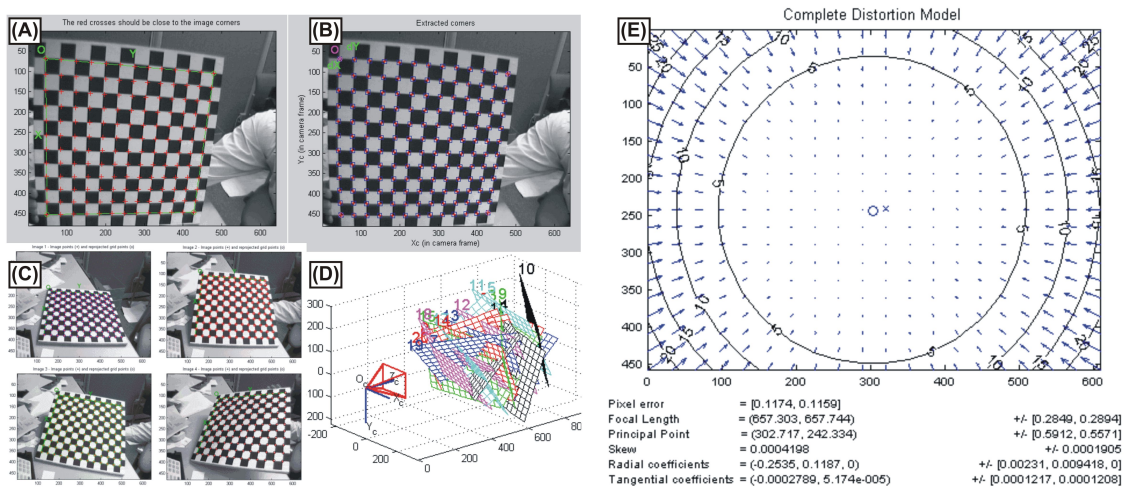


Figure 4.3 - Calibrage de la caméra à l'aide du Camera Calibration Toolbox⁸ : (A) sélection des quatre coins externes - (B) Détection automatique des intersections - (C) résultats sous quatres angles de vue différents - (D) résultats du positionnement de la caméra sous 20 points de vue - (E) Illustration des paramètres de distortion finale (radiale + tangentielle)

4.2.4 Calibrage du vidéo-projecteur : paramètres intrinsèques

Un vidéo-projecteur peut être modélisé de la même manière qu'une caméra vidéo. Les paramètres, intrinsèques et extrinsèques, sont donc identiques à ceux d'une caméra lors d'une modélisation en *trou d'épingle* (*pinhole*), parfait ou tenant compte de la distortion. Cependant, celui-ci ne capturant pas d'images, le calibrage ne peut se faire directement par les techniques de calibrage de caméra.

4.2.4.1 Quelques méthodes existantes

Il existe plusieurs travaux sur le calibrage de vidéo-projecteur. Nous ne donnons ici que deux exemples représentatifs (voir [Shen 02] pour un état de l'art plus complet sur le sujet).

Pour retrouver les paramètres intrinsèques, Shen et Menq [Shen 02] proposent une technique semi-automatique. Chaque « pixel » de projection est modélisé par un rayon 3D. Le calibrage est effectué par un recoupement entre les rayons projetés sur un plan mobile et le pointage (manuel) des spots résultants sur ce plan réalisé à l'aide d'une « machine de mesure de coordonnées » (*Coordinate Measuring Machine*). Grâce à la correspondance entre ces points mesurés en 3D et les coordonnées 2D dans la grille de projection, les auteurs retrouvent les paramètres intrinsèques du vidéo projecteur en tenant compte de la distortion. Cette technique est très précise grâce au dispositif de mesure de points 3D mais engendre un besoin matériel supplémentaire (dispositif de pointage 3D).

Ashdown et Sato [Ashdown 05] utilisent quant à eux une caméra dont les *paramètres intrinsèques* sont retrouvés à l'aide d'un damier imprimé en blanc et cyan. Ce choix de couleur permet de calibrer le vidéo-projecteur à partir d'une même image en projetant un damier bleu. En séparant l'image capturée selon les différents canaux de couleur, le calibrage du vidéo projecteur peut se faire à partir du canal bleu et la caméra à partir du canal rouge où la contribution de l'image projetée est négligeable (cf Fig. 4.4). Les données sont utilisées

pour calculer les paramètres à l'aide du *Camera Calibration Toolbox*. Contrairement à l'approche précédente, cette méthode, plus récente, ne nécessite pas de dispositif de pointage 3D et génère un jeu de paramètres devenu plus standard grâce au *Camera Calibration Toolbox*.

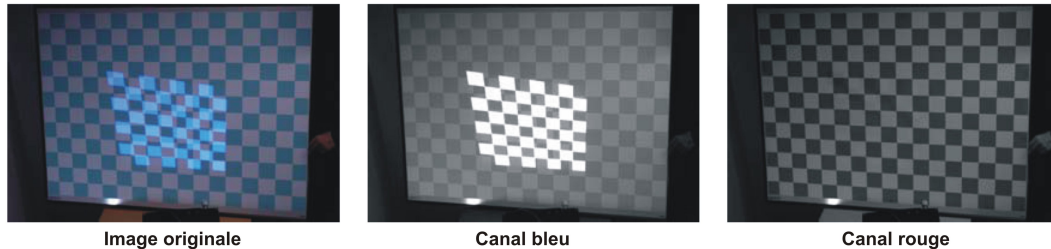


Figure 4.4 - D'après Ashdown et Sato[Ashdown 05] : Séparation de la contribution lumineuse du vidéo-projecteur par canaux de couleur.

4.2.4.2 Notre méthode

Nous nous basons donc sur cette dernière approche pour effectuer le calibrage des paramètres intrinsèques de notre vidéo-projecteur. Cependant, nous voulons nous affranchir de l'utilisation de la caméra vidéo pour cette étape. En effet, il nous semble que la mise en place de la caméra alourdit la procédure, d'autant plus qu'il est toujours nécessaire d'entrer manuellement les intersections dans le *Camera Calibration Toolbox*.

Nous avons donc développé une méthode reprenant le concept des intersections mais en pointant celles-ci non plus à partir d'une image vidéo mais **directement sur une grille imprimée**. Le pointage des points se fait à la souris dans l'espace de projection (2D)(Fig. 4.5 a) et l'utilisateur doit faire correspondre chaque point de la grille projetée avec la grille imprimée (Fig. 4.5 b).

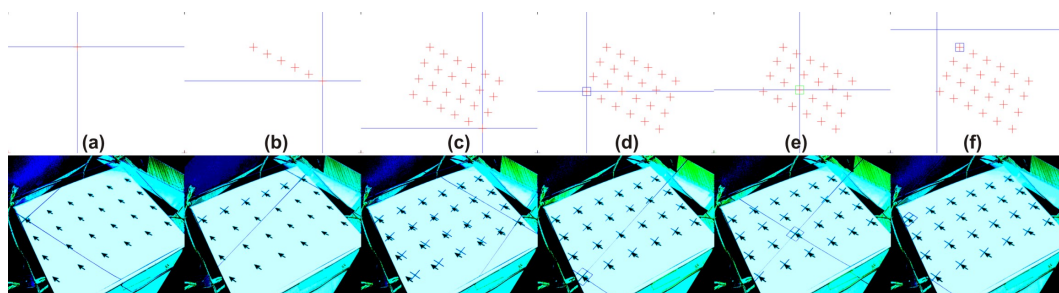


Figure 4.5 - Calibrage des paramètres intrinsèques du vidéo projecteur : illustration de notre méthode par une séquence d'utilisation pour une étape du calibrage. En haut, les images projetées par le vidéo projecteur. En bas, des captures de la projection sur le motif imprimé (Ces captures sont ici montrées pour une meilleure compréhension de la méthode. À aucun moment elles ne sont utilisées pour le calibrage du vidéo-projecteur.).

Pour accélérer la procédure, seuls les quatre points extérieurs sont nécessaires dans une première phase (Fig. 4.5 c). La grille est ensuite positionnée automatiquement depuis ces quatre points (Fig. 4.5 d). De cette manière, seules les intersections ne correspondant pas exactement à la grille imprimée doivent être déplacées manuellement. En effet, la distortion optique étant généralement plus importante sur la périphérie de la projection, il est bien

souvent inutile de déplacer les points centraux, ceux-ci correspondant parfaitement par défaut (Fig. 4.5 e).

Ces étapes sont répétées pour chaque angle de projection. Les variations de cet angle de projection peuvent être obtenues soit en déplaçant le vidéo projecteur, soit en déplaçant la grille imprimée. Pour chaque angle de projection, les points correspondant à la grille projetée (2D) sont enregistrés et sont utilisés comme données, soit pour l'utilitaire de calibrage *ARToolkit*, soit pour le *Camera Calibration Toolbox* (Fig. 4.6). Nous obtenons ainsi les *paramètres intrinsèques* du vidéo-projecteur.

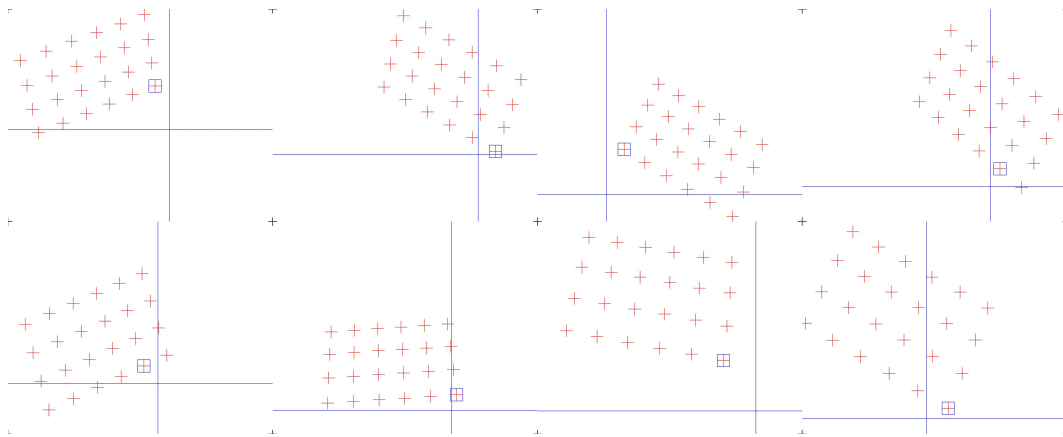


Figure 4.6 - Calibration des paramètres intrinsèques du vidéo projecteur : différents angles

4.2.5 Positionnement virtuel des dispositifs optiques : *paramètres extrinsèques*

Nous avons développé deux méthodes pour le positionnement virtuel des deux appareils. La première, découplée, permet un calibrage des paramètres extrinsèques de chaque appareil séparément à l'aide d'un motif imprimé sur un support. La seconde méthode, séquentielle, consiste à utiliser le calibrage du vidéo projecteur pour positionner automatiquement la caméra vidéo. Les deux techniques se basent sur la librairie *ARToolkitPlus*.

Marqueurs ARToolkit Les librairies *ARToolkit* et *ARToolkitPlus* utilisent des *marqueurs visuels*. Ces marqueurs sont planaires et bicolores, en noir et blanc pour un contraste maximal. Ils sont composés d'un bord carré noir au centre duquel est placé un motif. Ces motifs doivent être imprimés puis collés sur un support planaire. Le bord noir est détecté dans l'image filmée à l'aide d'un algorithme de détection de contours. Il sert à positionner le motif et, selon sa taille réelle, sa taille dans l'image capturée et aux *paramètres intrinsèques* de la caméra (cf 4.2.2), à connaître la distance qui le sépare de la caméra. Le motif central est quant à lui utilisé pour détecter l'orientation du marqueur.

Avec *ARToolkit*, quelques marqueurs sont fournis (cf Fig. 4.7, à gauche). Il s'agit d'images bitmap binaires. En revanche, un nouveau type de marqueurs a fait son apparition avec *ARToolkitPlus* : l'*ARTag*. Il s'agit également de marqueurs binaires noir et blanc, mais dont le motif est constitué d'une série de carrés blancs ou noirs (cf Fig. 4.7, à droite) : les marqueurs peuvent donc être générés facilement lors de l'exécution et le nombre de marqueurs disponibles



Figure 4.7 - Exemples de marqueurs : à gauche, Marqueur ARToolkit , à droite, Marqueur ARToolkitPlus (ou ARTag)

est alors bien plus élevé ($2^{6 \times 6}$ pour une grille 6×6 en général). De plus, la taille de la bordure peut être diminuée ce qui permet d'imprimer des marqueurs plus petits.

4.2.5.1 Méthode découplée

Dans un premier temps, nous avons utilisé un support planaire sur lequel était imprimé un marqueur *ARToolkit*. Lorsque nous avons pu disposer d'*ARToolkitPlus*, nous avons préféré utiliser un *ARTag* en raison d'un meilleur taux de réussite de la détection. Ce marqueur définit l'origine du repère virtuel.

La caméra, fixée sur un trépied, est pointée sur ce marqueur. Le marqueur doit être totalement visible dans l'image filmée par la caméra. Grâce aux *paramètres intrinsèques* retrouvés dans la partie *off-line* du calibrage, la librairie calcule la matrice 3D (4×4) permettant de passer du repère caméra au repère du marqueur. L'inversion de cette matrice permet de passer du repère du marqueur, que nous appellerons *marqueur origine*, au repère de la caméra : la caméra est alors positionnée virtuellement.

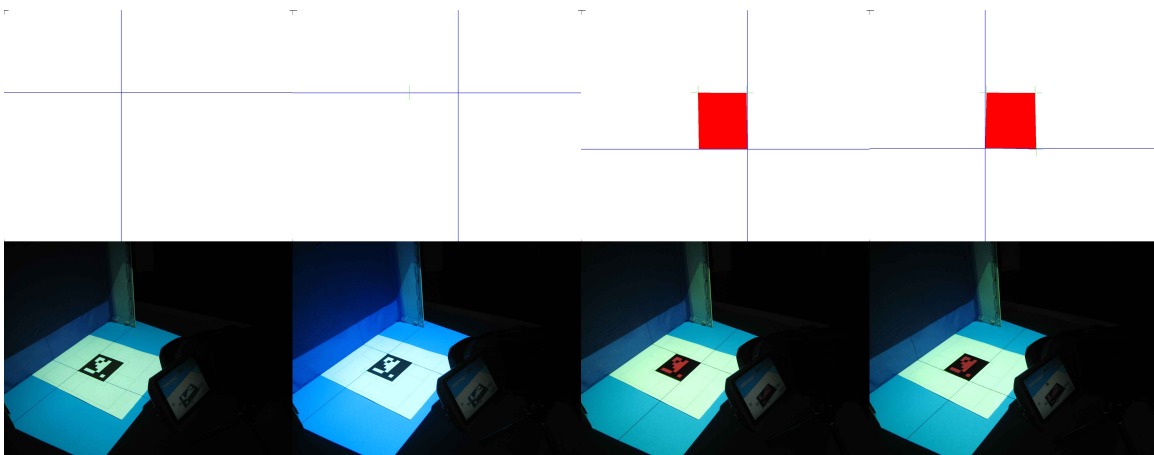


Figure 4.8 - Positionnement du vidéo-projecteur avec la première méthode : pointer les quatre coins du marqueur origine imprimé sur un support physique.

Dans un second temps, l'utilisateur est appelé à effectuer quelques opérations simples pour positionner le vidéo-projecteur. À l'aide de la souris, il doit pointer les quatre coins du *marqueur origine*, par projection de deux lignes (cf Fig. 4.8). L'ordre requis est d'abord le coin supérieur gauche puis les trois autres dans le sens des aiguilles d'une montre. La correspondance de ces quatre sommets dans le repère 2D du vidéo-projecteur permet de calculer la matrice de passage du *marqueur origine* vers le repère du vidéo-projecteur à l'aide de la librairie : le vidéo-projecteur est également positionné dans l'environnement virtuel.

4.2.5.2 Méthode séquentielle

L'inconvénient de cette méthode est qu'il est nécessaire d'imprimer un motif complexe et de le fixer sur un plan rigide. Puisque nous disposons ici d'un vidéo-projecteur, nous nous basons sur Fiala [Fiala 05] qui projette les marqueurs utilisés dans *ARToolkitPlus* devant une caméra pré-calibrée. À l'inverse, nous projetons à un endroit prédéterminé un motif *ARTag* qui nous permet de positionner la caméra à l'aide d'*ARToolkitPlus* sans impression de motif. Seul un carré, plus large, est nécessaire sur le support. Ce support est supposé planaire.

L'utilisateur est donc appelé à pointer, toujours à l'aide la souris, les quatre coins de ce carré dans le sens des aiguilles d'une montre. Tout comme dans la première méthode, *ARToolkitPlus* est utilisé pour positionner le vidéo-projecteur dans l'environnement virtuel. La seule différence ici est que le « marqueur », autrement dit le carré, a une taille différente de celui imprimé dans la méthode précédente.

Les informations 2D dans le repère du vidéo projecteur sont ensuite utilisées pour projeter un motif *ARTag*. Cette projection sur le plan (nous utilisons une simple table) est perçue par la caméra comme le *marqueur origine* : *ARToolkitPlus* renvoie alors une matrice de positionnement de la caméra par rapport à ce marqueur projeté (cf Fig. 4.9). La seule restriction ici est que la couleur du support doit être la plus claire possible pour un contraste maximum.

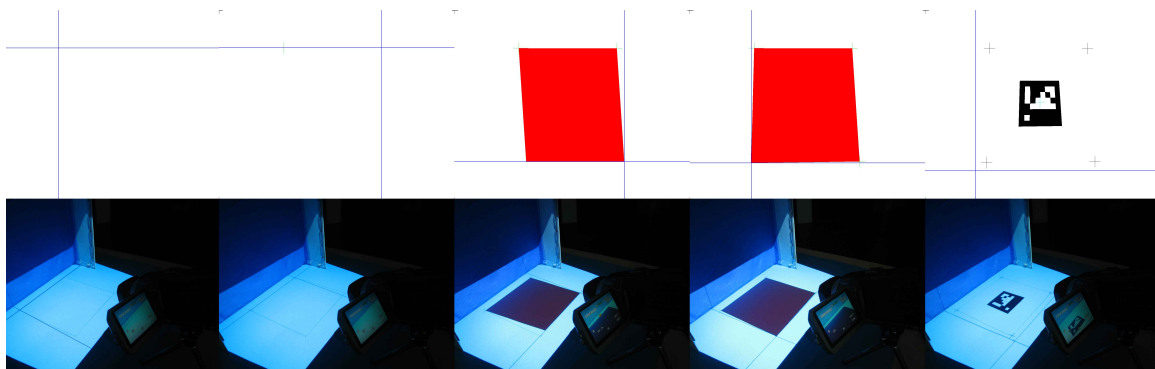


Figure 4.9 - Positionnement du vidéo-projecteur avec la seconde méthode : pointer les quatre coins du carré puis projeter le marqueur origine au centre.

Les avantages de cette méthode sont nombreux :

- **Précision** : Il n'est pas nécessaire que le carré tracé sur le support soit visible dans l'image filmée par caméra. Il doit simplement être compris dans le trapèze formé par la pyramide de projection sur le support. Il est alors possible d'utiliser un carré bien plus grand que le marqueur imprimé nécessaire dans la méthode précédente. La taille relative de ce marqueur par rapport à la projection de la pyramide de vision est le principal

facteur de la précision de la matrice de positionnement calculée par *ARToolkitPlus*. Plus ce rapport est faible, meilleur est le positionnement du dispositif. Dans notre cas, cela revêt encore plus d'importance. En effet, pour que l'image projetée soit nette, un vidéo-projecteur doit être placé à une distance minimale relativement importante ($\approx 1m$ dans le cas général). A l'inverse, la caméra doit être relativement proche du *marqueur origine* où sera placé par la suite l'échantillon à déformer. Plus la caméra est proche de ce repère, plus les détails de la déformation filmée seront importants. Avoir deux marqueurs de tailles différentes est donc primordial.

- **Souplesse** : La taille du marqueur projeté est programmable. Il n'est donc pas nécessaire de déplacer physiquement la caméra pour que l'image capturée de celui-ci soit entièrement visible : il suffit de réduire ou d'augmenter la taille du marqueur projeté jusqu'à ce qu'il soit complètement visible par la caméra. Cette procédure peut être entièrement automatisée. Cela permet également d'optimiser le rapport entre le marqueur visible par la caméra et la résolution de celle-ci (cf **Précision**).
- **Modularité** : On pourra projeter ce motif sur une feuille de couleur unie. Il sera lors plus simple par la suite de détourer la déformation filmée de l'échantillon et cela évite une manipulation supplémentaire qui consiste à recouvrir le *marqueur origine* imprimé. Une fois la caméra positionnée, le motif projeté peut également être modifié. Par exemple, il est possible de projeter la silhouette de l'échantillon, voire le résultat de l'ombre portée, pour aider à le positionner physiquement.
- **Robustesse** : Le marqueur projeté peut être modifié « à la volée » si le positionnement de la caméra n'a pas fonctionné. On peut également envisager de projeter directement un grand nombre de marqueurs [Fiala 05] pour maximiser les chances de réussite et ainsi encore accroître la robustesse d'*ARToolkitPlus*.

4.3 Dispositif de mesure mécanique

Pour recréer virtuellement la déformation filmée, il est nécessaire d'« enregistrer » les actions et les forces qui caractérisent cette déformation. Nous avons vu en 3.3 une liste des différents types de dispositifs permettant de faire ce genre de capture pour les objets déformables. Nous avons également déterminé qu'il est possible de les diviser en deux grands groupes : les dispositifs automatiques, motorisés et contrôlés par ordinateur, qui allient précision et répétabilité d'une déformation et les dispositifs interactifs, souvent moins encombrants et apportant une plus grande liberté de mouvements.

Nous avons choisis d'utiliser un dispositif interactif. En effet, l'une des grandes lignes directrices de ces travaux est de concevoir une méthode nécessitant un dispositif simple d'utilisation et de conception pour permettre un meilleur accès et un coût réduit. Les dispositifs manuels nécessitent généralement moins de pièces mécaniques que les dispositifs motorisés et sont donc moins onéreux et plus faciles à fabriquer.

Il nous faut donc choisir un outil interactif permettant de réaliser la capture d'une information sur la force appliquée à l'objet par l'utilisateur ainsi que l'enregistrement de la position et du déplacement de l'outil. Nous allons voir ici les différents choix que nous avons effectués, le calibrage de l'outil en termes de mesure d'effort, de position et d'orientation, ainsi que les évolutions de ce dispositif.

4.3.1 Capteur de force

Le cœur de l'outil est un *capteur de forces*. Il en existe un très grand nombre suivant les caractéristiques principales : taille, degré de mesure (1D, 2D, . . . 6D), charge nominale, fixation, etc. . . . Selon ces critères, nous avons choisi d'utiliser un capteur unidirectionnel « K11107 », © SCAIME (cf fig. 4.10) :

- *Taille* : Cylindrique, avec un diamètre de 17mm, il est particulièrement adapté à l'insertion dans un outil de type « stylet ».
- *Degré de mesure* : Il permet une mesure uni-axiale en traction/compression. Il peut donc être utilisé pour mesurer l'effort appliqué dans l'axe de l'outil. Le coût de ce type de capteur est de plus bien inférieur aux capteurs multi-directionnels (environ 80% moins cher qu'un capteur 6D de taille équivalente).
- *Charge nominale* : La plage de mesure ce capteur est $[-50 \dots 50]N$ (traction/compression) ce qui est suffisant pour mesurer un effort manuel.
- *Fixation* : Ce capteur est composé d'un corps cylindrique muni de deux tiges filetées pour faciliter la fixation au bâti.

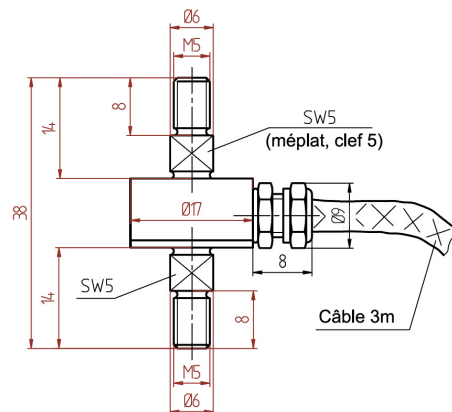


Figure 4.10 - Schéma du capteur de forces « K11107 », © SCAIME⁹

Calibrage Le calibrage de ce capteur unidirectionnel est classique et s'apparente au calibrage d'un système de mesure de poids. Il s'agit de charger le capteur avec des ajouts de poids successifs connus et d'effectuer les mesures correspondantes (cf fig 4.11, à gauche). On peut ainsi tracer la courbe de réponse du capteur en associant la force réelle appliquée (déduite du poids de charge) et la mesure en incréments, unité spécifique à chaque capteur. Cette courbe permet de déduire les caractéristiques du capteur comme la sensibilité (ici 0.00065N par incrément) ou la linéarité.

4.3.2 Conditionnement et évolution

Ce capteur est donc adapté à la conception d'une sonde manuelle de mesure d'effort. Il est composé de deux parties : l'une doit être fixe par rapport au référentiel et l'autre « mobile »¹⁰.

9. Image tirée de <http://www.scaime.com/Acrobat3/force/FT-K1107-FE-0999.pdf>

10. Le capteur étant constitué de *jauges de contraintes*, les déplacements en jeux ici sont minimes et invisibles à l'œil nu.

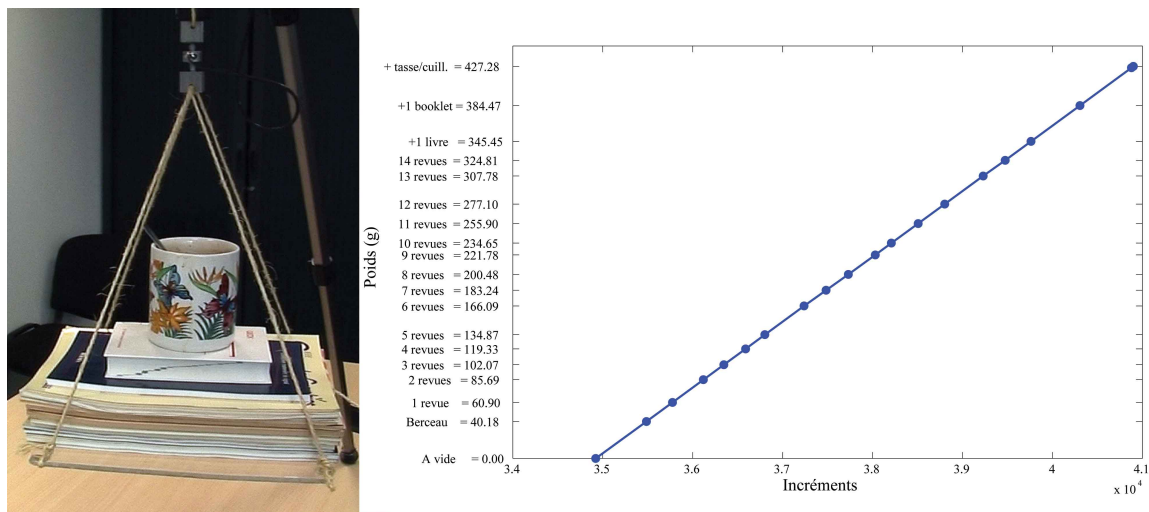


Figure 4.11 - Calibrage du capteur de forces : La photo de gauche montre l'ajout de poids successifs. Les revues sont tout d'abord pesées puis ajoutées unes-à-unes. D'autres poids comme la tasse ou la cuillère sont également utilisées. En notant la mesure associée à chaque ajout de charge, on peut tracer la courbe de réponse du capteur (à droite). La pente définit la sensibilité du capteur (ici, 0.00065N par incrément).

Ici, le référentiel est la main de l'utilisateur. Pour rendre son utilisation ergonomique, il est nécessaire de réaliser un montage autour de ce capteur (*conditionnement*). Nous décrivons ici les étapes de ce conditionnement et les réflexions qui ont amené les différentes évolutions de ce dispositif.

4.3.2.1 Prototype

Dans un premier temps, pour valider la faisabilité d'un tel dispositif, nous avons réalisé un prototype. Nous l'avons conçu pour qu'il puisse être manipulé comme un gros stylet. Il peut s'adapter autour d'une base cylindrique qui dépend du support utilisé (cf 4.3.3). Sur cette base, un trou taraudé permet de visser l'une des deux tiges filetées du capteur. Nous avons ensuite modifié une pince pour multimètre électronique afin de pouvoir la visser sur l'autre tige du capteur. Cette « prolongation » du capteur permet de lui transmettre des efforts depuis une distance plus importante. Pour protéger l'ensemble, nous avons fabriqué un bâti cylindrique à partir d'une seringue coupée à une extrémité. Cette coque de protection est fixée sur la base cylindrique. Une rainure est faite dans la seringue pour laisser passer le câble du capteur et pouvoir réaliser le montage. Cette coque n'est pas en contact avec le capteur. Pour limiter les efforts tangentiels sur le capteur, nous avons fabriqué une bague en résine à l'extrémité de la coque. Le diamètre interne de celle-ci est plus grand que celui de la pince de multimètre. Elle ne génère donc pas de frottements lors d'une utilisation normale. La figure 4.12 illustre ce prototype.

Les premiers tests de mesures réalisés à l'aide de ce prototype ne sont pas très concluants. En effet, les résultats sont très bruités et parmi les causes possibles on peut identifier les divers jeux induits par ce montage rudimentaire. D'autre part, la conception de la bague en résine s'est avérée être défectueuse car il est pratiquement impossible de réaliser un test sans frottement. Cependant, ces essais ont validé l'ergonomie générale d'un dispositif de type

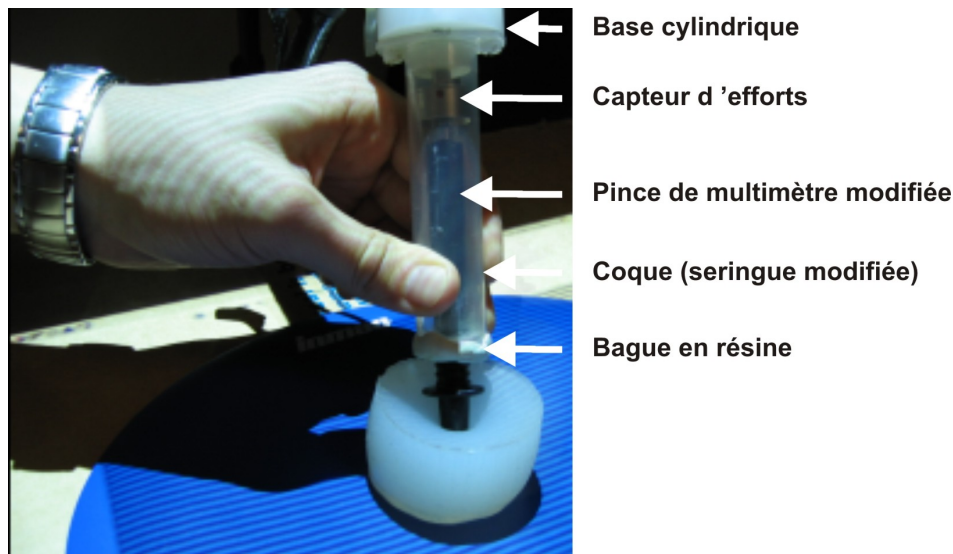


Figure 4.12 - Prototype du dispositif de mesure d'effort.

« stylet » pour des mesures sur des objets déformables.

4.3.2.2 Montage « vissé »

Nous avons donc conçu une évolution de ce prototype. Le but est de tester si un montage plus rigoureux permet de réduire le bruit observé sur les mesures. Pour cela, nous avons créé un bâti cylindrique dans lequel est inséré le capteur. Le capteur est vissé à ce bâti à l'aide d'une des deux tiges filetées. La partie « mobile », constituée d'une tige cylindrique en métal est ensuite vissée sur la seconde tige filetée. La touche qui appliquera la déformation à l'objet est alors fixée sur cette partie mobile. Pour assurer une mesure uni-axiale et garantir l'intégrité du capteur, nous ajoutons une bague de protection qui guide la partie mobile tout en limitant les efforts tangentiels. Cette bague est un palier en bronze effrité pour limiter au maximum les frottements qui pourraient perturber la mesure. Le schéma décrit sur la figure 4.13 détaille l'évolution du prototype, les éléments qui le constituent et le montage final de cette première version du dispositif.

4.3.2.3 Montage « segmenté »

Après avoir effectué de nombreuses expérimentations à l'aide de la première version du dispositif, nous nous sommes aperçu que les mesures étaient souvent incohérentes et toujours très bruitées. De plus, l'amplitude de ces erreurs rend la mesure totalement fautive.

Pour repérer l'origine de ces erreurs, nous avons réalisé un dispositif simple. Il s'agit de créer un pendule, composé d'un poids accroché à une cordelette, et de fixer l'ensemble à l'extrémité de la partie mobile (cf fig. 4.14 gauche). Ce pendule permet de vérifier la cohérence des mesures en les mettant en rapport avec l'angle formé par le dispositif. En toute logique, lorsque l'angle formé par le pendule et le dispositif est de 90° , la force axiale mesurée doit être nulle. Lors de nos expérimentations, ce n'est pas le cas : des efforts tangentiels sont donc transmis au capteur malgré les précautions prises et celui-ci n'est pas conçu pour ce type de

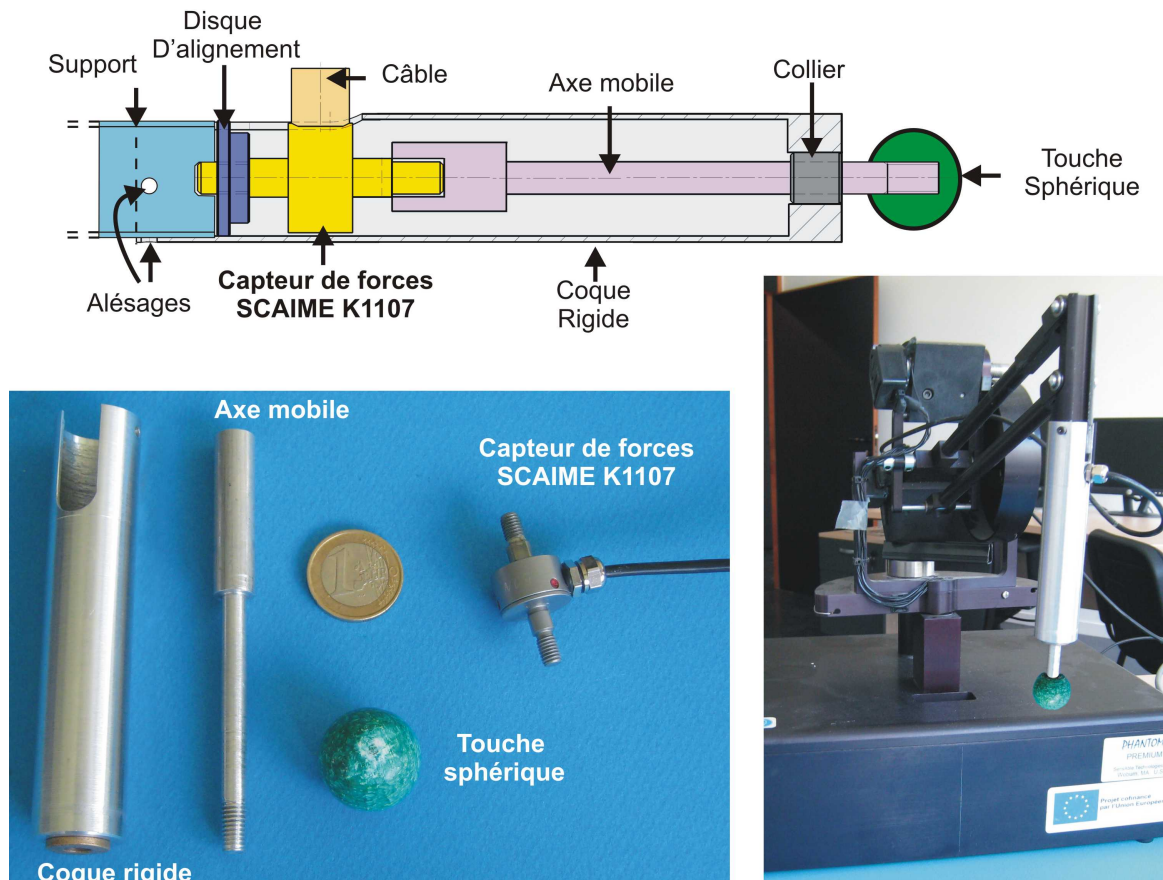


Figure 4.13 - Première version du dispositif de mesure d'efforts (En haut, le schéma, en bas à gauche les différents composants et en bas à droite le montage final sur le « Phantom 1.5 », © Sensable technologies).

mesure (cf fig. 4.14 droite).

Plusieurs phénomènes peuvent expliquer cette transmission d'effort. Tout d'abord, le jeu entre la partie mobile et la bague de protection génère un déplacement tangentiel de la tige mobile. Même si ce jeu est infime, l'effort tangentiel au niveau du capteur est proportionnel à la distance qui le sépare de la bague (cf fig. 4.15 gauche). Un autre phénomène peut aussi apparaître. Il s'agit de la flexion de la tige mobile lorsque les efforts en jeu sont importants. Dans ce cas, la taille de la tige ainsi que le matériau choisi est critique (cf fig. 4.15 droite).

Pour remédier à ces problèmes, nous avons conçu une seconde version du dispositif. Le principe est de segmenter l'axe de transmission et d'ajouter une touche sphérique entre la tige filetée « mobile » du capteur et la partie mobile. Cette bille assure la transmission des efforts axiaux uniquement (cf fig. 4.16), contrairement à un montage « vissé ». Cependant, il devient alors impossible d'utiliser le capteur en traction. Nous limitons donc nos expérimentations à la compression des matériaux que nous estimons suffisante pour caractériser les propriétés voulues. Dans une moindre mesure, il est également nécessaire d'une part de « bloquer » la sortie de la tige mobile et d'autre part, de détecter le moment exact où la tige, la bille et le capteur sont en contact (ce qui est trivial puisqu'il suffit de détecter que la mesure sur le capteur est non-nulle). Pour minimiser les risques de flexion de la tige, celle-ci est réalisée en

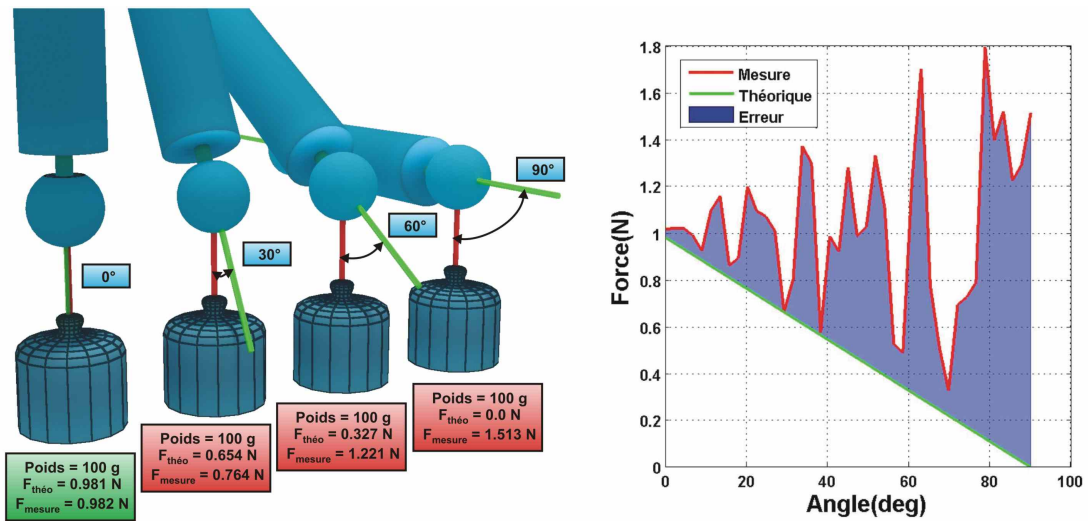


Figure 4.14 - Dépistage de l'erreur de mesure : Un pendule de 100g fixé à l'extrémité de la partie mobile permet de quantifier l'influence de l'angle du dispositif. L'incohérence de certaines mesures, notamment à angle droit ou l'effort mesuré doit être nul, permet d'identifier la source de l'erreur : Une partie des efforts tangentiels est malgré tout transmise au capteur qui n'est pas conçu pour ce type de mesures. La courbe à droite montre l'incohérence des mesures obtenues par rapport à la mesure des efforts axiaux attendue.

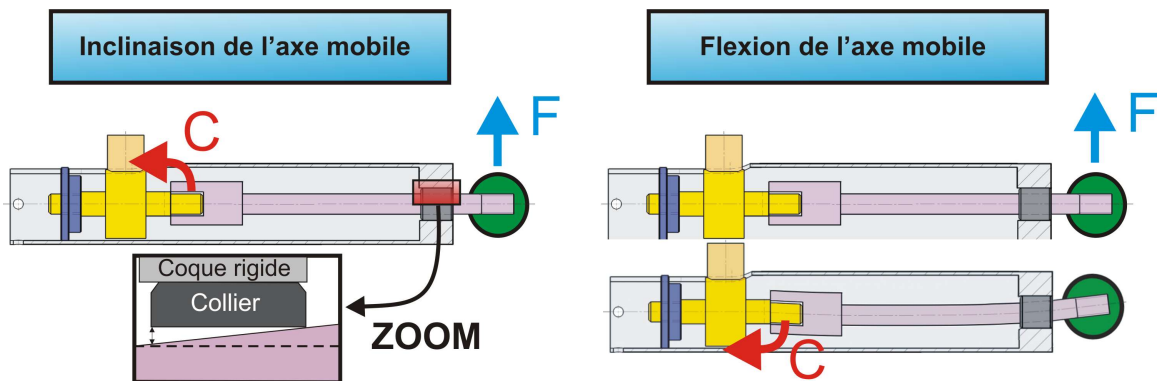


Figure 4.15 - Transmission d'une partie des efforts tangentiels : (À gauche) Le jeu infime entre la bague et la tige mobile induit un couple sur le capteur. (À droite) La flexion de la tige mobile peut également induire un couple sur le capteur. Ces deux phénomènes expliquent les erreurs de mesure de la composante axiale de la force car le capteur n'est pas conçu pour gérer ce couple.

aluminium.

Touche La touche est la partie du dispositif qui est en contact avec le matériau déformable. Celle-ci sera donc modélisée informatiquement et utilisée dans le calcul de la déformation par le modèle mécanique par élément finis (cf 5.1). Sa forme et sa taille sont donc capitales pour une simulation rapide et précise. En effet, l'objet déformable est décomposé en éléments volumiques. La taille et le nombre de ces éléments sont des facteurs importants pour la précision de la simulation, tout particulièrement à l'endroit du contact avec la touche virtuelle : si celle-ci a une forme angulaire, le nombre d'éléments à cet endroit devra être très important

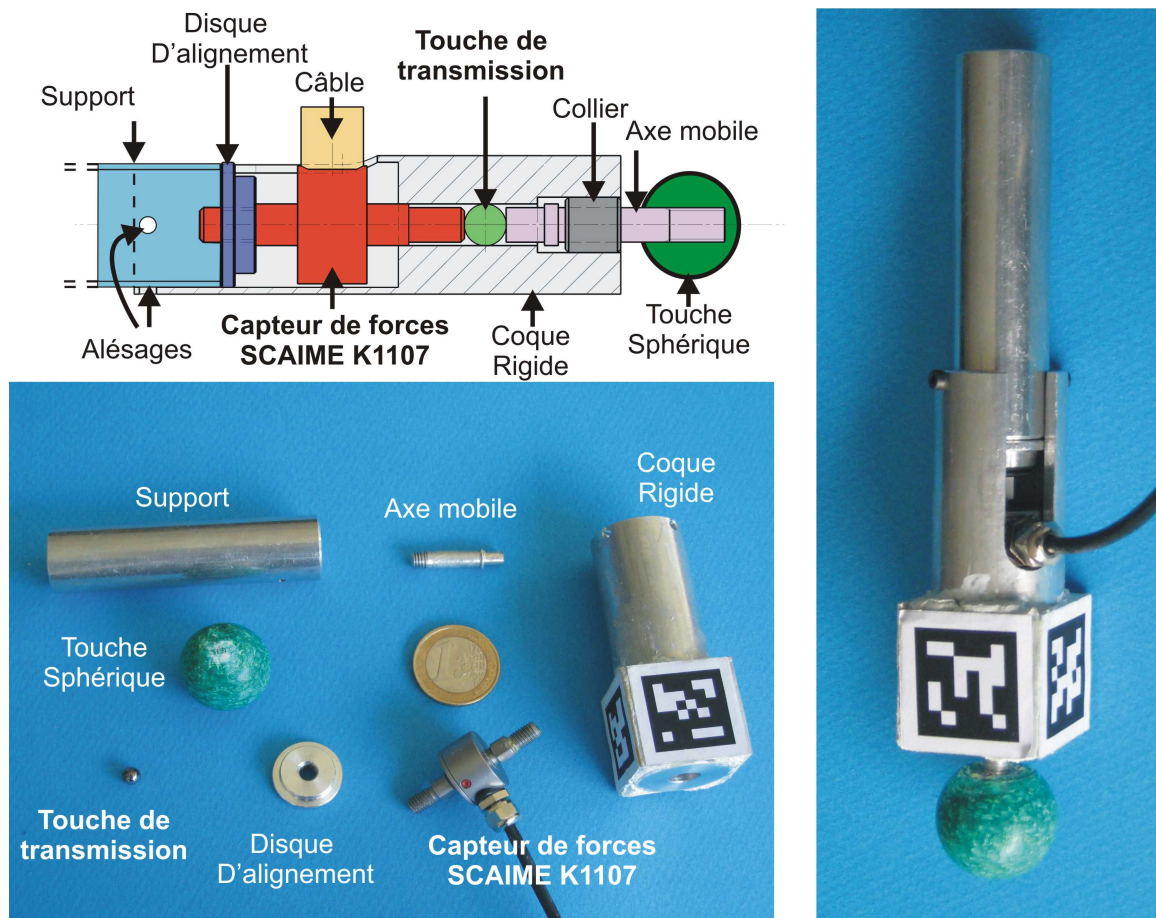


Figure 4.16 - Seconde version du dispositif de mesure d'effort : la touche sphérique garantit la transmission des efforts axiaux seuls. (En haut, le schéma, en bas à gauche les différents composants et à droite le montage final).

pour garantir une simulation précise du phénomène (cf fig. 4.17). Or, augmenter ce nombre d'éléments influe directement sur le temps de calcul de l'algorithme.

La forme tri-dimensionnelle la moins angulaire possible est une sphère. Nous avons donc opté pour une touche de ce type. Elle offre un autre avantage : il est relativement peu complexe de calculer les intersections entre une sphère et un maillage tétraédrique. Cela permet de réduire les temps de calcul nécessaires à la simulation.

4.3.3 Positionnement virtuel

Une fois le dispositif de mesures créé, il est nécessaire de le positionner dans l'espace. En effet, pour pouvoir recréer virtuellement l'action de l'utilisateur, on doit être en mesure de retrouver la position et l'orientation de cet outil à chaque instant de la simulation. Cela nous permettra de connaître à tout moment la position de la touche sphérique. Il faut donc être capable de capturer ces informations lors de l'enregistrement des mouvements de l'utilisateur. La capture de ces éléments est appelée *tracking*. Dans un premier temps, nous avons développé une méthode de capture utilisant un outil de *tracking magnétique*. Dans un second temps, nous

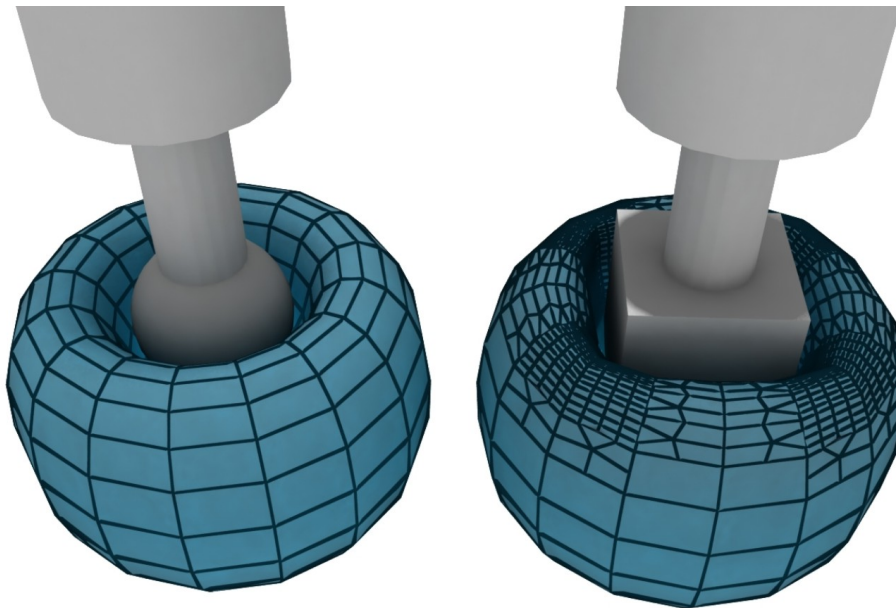


Figure 4.17 - Influence de la forme de la touche sur la définition du modèle par éléments finis.

utilisons une méthode mécanique en adaptant le dispositif sur un « Phantom 1.5 », © *Sensable technologies* (cf Fig. 4.13, à droite) pour s'affranchir des phénomènes électro-magnétiques qui perturbent les mesures. Enfin, pour des raisons pratiques et économiques, nous utilisons une méthode de capture visuelle pour réduire l'encombrement global du système.

4.3.3.1 Capture magnétique

Le principe de base du *tracking magnétique* est relativement simple. Il s'agit de générer un champ magnétique à l'aide d'un émetteur et d'y immerger un capteur mobile qui perturbe ce champ. En étudiant les perturbations, on peut en déduire plusieurs informations. Le « Flock of birds », © *Ascension* est un système qui permet de capturer en temps réel les informations en six dimensions (positions et orientations) de plusieurs capteurs mobiles (jusqu'à quatre simultanément). Ce matériel est disponible au sein de l'équipe Graphix du LIFL et nous l'avons utilisé pour nos premiers tests de tracking.

Pour adapter notre outil de mesure, nous avons réalisé une base cylindrique en nylon sur laquelle on peut fixer d'une part notre dispositif (décrit en 4.3.2), et d'autre part un capteur du « Flock of birds », © *Ascension* (cf Fig. 4.18). Les informations capturées sont données dans le repère de l'émetteur magnétique. Il suffit donc de placer cet émetteur à une position connue par rapport au repère origine de la scène. En appliquant cette transformation aux informations mesurées, on peut retrouver la position et l'orientation du stylet dans l'espace et donc de la touche sphérique.

Certains problèmes sont apparus lorsque nous avons utilisé ce dispositif. Étant magnétique, il est sensible aux perturbations du champ créé. Il est donc préférable d'avoir un environnement qui contient peu de métal, particulièrement entre l'émetteur et le capteur.

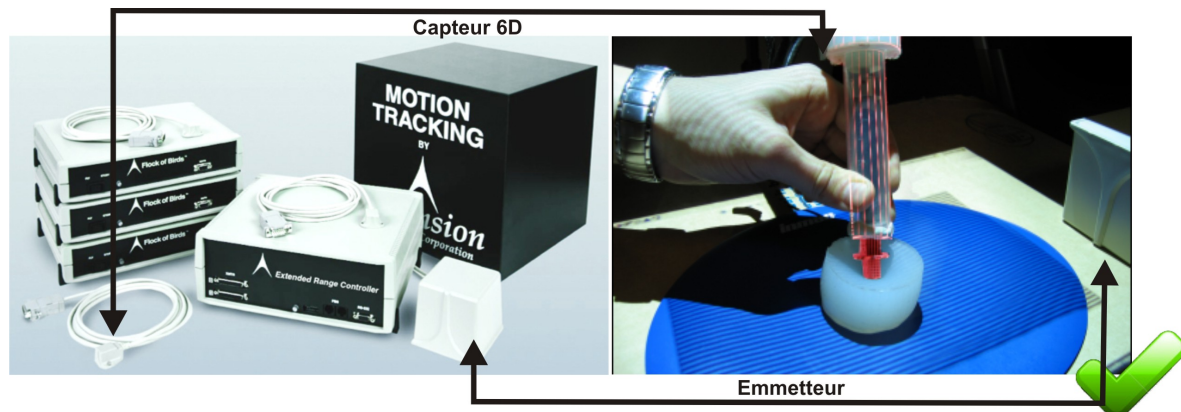


Figure 4.18 - Positionnement virtuel magnétique à l'aide du système « Flock of birds », © Ascension . À gauche, l'ensemble des éléments du système de tracking magnétique. À droite, le montage sur le prototype de l'outil et le résultat du recalage en surimpression (en rouge).

C'est pourquoi nous avons choisi de créer la base cylindrique en nylon. Les vis utilisées pour assembler le dispositif sont également en nylon. Cependant, le capteur d'efforts est lui-même en partie métallique. De plus, certains autres facteurs sont difficiles à contrôler, comme par exemple la quantité de métal présent sur l'utilisateur (gourmettes, lunettes, ...) ¹¹.

4.3.3.2 Capture mécanique

Pour remédier à ces problèmes, nous avons décidé d'utiliser une méthode de capture mécanique. Il s'agit de fixer notre dispositif sur un bras articulé muni de capteurs. Le « Phantom 1.5 », © Sensable technologies est un dispositif à retour d'effort. Ce type de machine est composé de couples moteurs/capteurs qui permettent de mesurer les angles de chaque partie du mécanisme ainsi que de fournir un effort pour simuler un environnement virtuel par exemple. L'équipe Graphix possède un tel dispositif. Nous avons donc opté pour l'adaptation de notre outil de capture sur cette machine en n'utilisant que la capacité de mesure des angles (cf Fig 4.13, droite). Les moteurs n'étant pas indispensables à cette utilisation, nous les avons désactivés.

Une librairie est fournie avec ce dispositif. Elle permet, entre autres, de mesurer directement la position et l'orientation d'un point situé au bout de la chaîne mécanique du bras. Cependant, pour pouvoir fixer notre dispositif, nous avons démonté une partie de cette chaîne. Nous devons donc utiliser les mesures bas niveau de chaque capteur d'angle puis recréer la chaîne cinématique du bras en y intégrant la géométrie de notre dispositif.

Cela est possible si l'on connaît les dimensions exactes de chaque partie du bras. De plus, une autre information est nécessaire. En effet, les mesures des différents angles sont relatives à leurs valeurs lors de l'initialisation du *Phantom*. Une étape préliminaire est donc nécessaire. Il s'agit de demander à l'utilisateur de mettre le dispositif dans une position de référence ¹².

11. D'autres dispositifs de tracking magnétique existent. À la suite de ces travaux, nous avons pu tester un matériel plus récent : le « Patriot », © Pohlemus . Celui-ci semble beaucoup plus robuste aux environnements métalliques.

12. Cette méthode est connue sous le nom de *docking*.

Cette manipulation, à pratiquer une seule fois par mise sous tension de l'appareil, permet de calculer un décalage fixe pour les valeurs de chaque capteur (simple différence entre l'angle connu dans la position de référence et la valeur mesurée à cet instant, cf Fig.4.19).

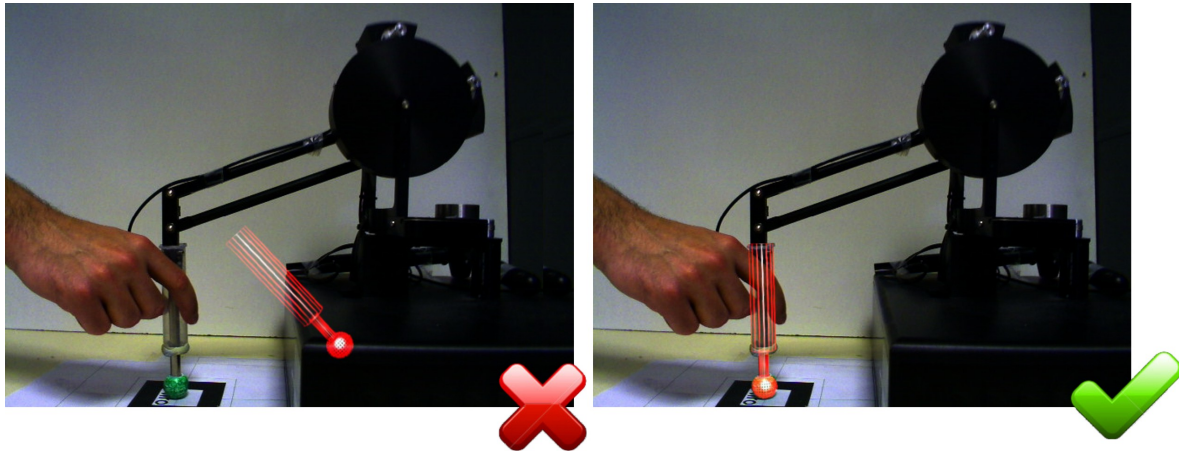


Figure 4.19 - *Positionnement virtuel mécanique* : À gauche, sans position de référence (décalages angulaires). À droite, après la position de référence (calibrage de la mesure des angles, ou docking).

Ce montage offre plusieurs avantages. Tout d'abord, ce type de mesure est très peu perturbé par les champs magnétiques. La quantité de métal environnant n'est donc plus un problème. De plus, il est possible d'utiliser les moteurs pour répéter une action enregistrée. Enfin, la précision de la mesure par rapport à la base du dispositif est relativement bonne ($0.03mm$ en translation et $0.0023deg$). Cependant, la précision finale dépend de la phase de calibrage. L'erreur induite lors de cette phase crée un décalage pour toutes les mesures suivantes¹³. De plus, l'espace de travail est assez restreint et le mode de fixation choisi limite les mouvements possibles car il entraîne une perte de trois degrés de liberté. Cela induit un manque de transparence pour l'utilisateur et peut même perturber l'action voulue.

4.3.3.3 Capture visuelle

Une autre approche consiste à utiliser les informations visuelles pour retrouver la position en six dimensions du dispositif. On peut séparer les travaux sur ce sujet en deux parties : les approches multi-vues, précises et robustes mais relativement complexes à mettre en œuvre, et les approches mono-vue, plus simples et plus rapides. Les premières nécessitent plusieurs points de vue pour évaluer précisément la position dans l'espace de points spécifiques. Si on connaît la géométrie de l'objet à localiser et la position réelle de ces points d'intérêt, on peut déduire, par recoupement des informations de chaque vue, la position et l'orientation du dispositif. L'investissement matériel que suppose ces méthodes, ainsi que la complexité à les mettre en œuvre, n'entrent pas dans le cadre d'une solution simple et peu onéreuse d'estimation de paramètres à partir de séquences d'images.

En revanche, les méthodes mono-vue ne nécessitent qu'une seule caméra et sont donc mieux adaptées à cette problématique. Si les paramètres intrinsèques de celle-ci sont connus,

13. Il serait préférable de fabriquer un *dock* mécanique comme sur le « Phantom Omni », © *Sensable technologies* pour limiter les erreurs dues au facteur humain.

il est possible d'estimer la distance relative des points d'intérêts par rapport à la caméra. Ces informations, combinées aux paramètres extrinsèques, permettent d'estimer la position et l'orientation de l'objet dans l'espace. Les points d'intérêts en question peuvent être de plusieurs natures, par exemple des motifs, des textures ou des formes géométriques facilement identifiables dans l'image capturée.

Nous utilisons dans un premier temps une méthode simple d'extraction de cercle (ou *circle fitting*) pour tenter de déterminer la position de la touche sphérique. Voici le principe de l'algorithme que nous utilisons :

- (a) Isoler la zone de l'image à traiter ¹⁴
- (b) Faire une détection de contour (*canny* ou *sobel* par exemple)
- (c) Tracer les droites perpendiculaires à la tangente de chaque pixel de ce contour dans une image « accumulateur » (transformée de *Hough*)
- (d) Choisir le pixel d'intensité maximale dans cet espace : c'est le centre du cercle.
- (e) Utiliser la méthode des moindres carrés pour déterminer le rayon à partir de ce centre
- (f) Coupler les données de ce cercle avec la taille réelle de la sphère
- (g) En déduire la distance avec la caméra puis la position dans l'espace

Malheureusement, cette méthode n'est pas très robuste aux problèmes d'occlusion qui surviennent dans notre cas lorsque la touche sphérique est en partie cachée par l'objet déformé (cf fig. 4.20).

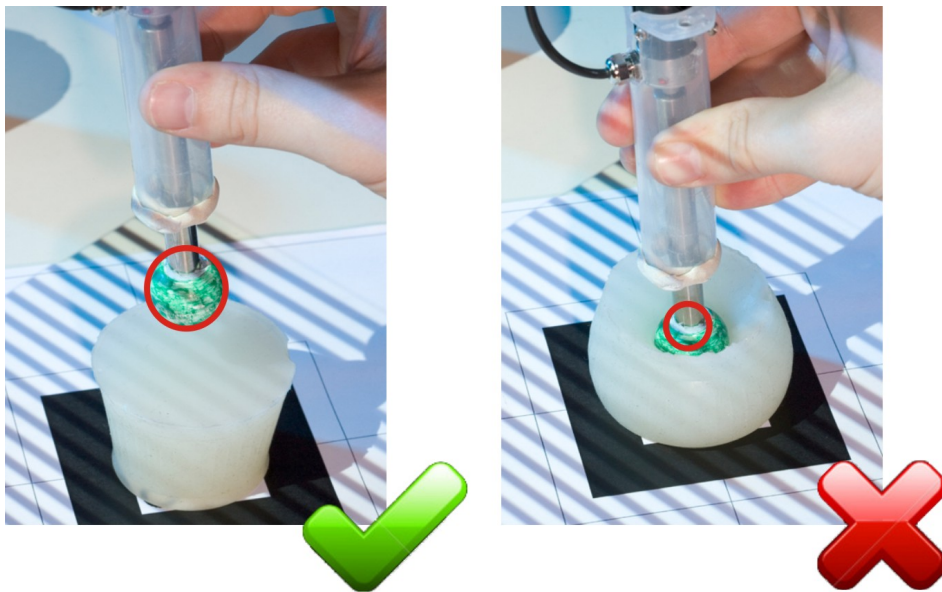


Figure 4.20 - Positionnement virtuel par vision de la position de l'outil à l'aide d'une extraction de contour. À gauche, le résultat d'une extraction sans occlusion. À droite, une illustration du problème d'occlusion à la suite de la déformation d'un objet.

Pour remédier à ce problème, nous utilisons une nouvelle fois la librairie *ARToolkitPlus*. En effet, celle-ci permet de déduire la position de la caméra par rapport à un marqueur visible dans l'image (cf 4.2.5). Si l'on place un second marqueur sur le dispositif, on peut en déduire, par la matrice de transformation inverse, la position de celui-ci par rapport à la caméra, et

14. Ici, nous utilisons un masque avec la couleur verte, notre touche sphérique étant peinte en vert.

donc par rapport au centre de la scène. Si l'on connaît de plus la position de ce marqueur dans le repère local de l'outil, on peut en déduire la position de la touche sphérique dans l'espace :

Soient les matrices de transformations 4×4 :

- $[M_{WC}]$, transformation du repère global vers la caméra (calculée par *ARToolkitPlus*)
- $[M_{DC}]$, transformation du marqueur dispositif vers la caméra (calculée par *ARToolkitPlus*)
- $[M_{DT}]$, transformation du marqueur dispositif vers la touche sphérique (donnée géométrique)

La position P_t et l'orientation D_t de la touche sphérique dans l'espace sont calculées par :

$$P_t = [M_{WC}][M_{DC}]^{-1}[M_{DT}] \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (4.1)$$

$$D_t = [M_{WC}][M_{DC}]^{-1}[M_{DT}] \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} \quad (4.2)$$

Cette méthode donne de bons résultats car elle profite de la robustesse d'*ARToolkitPlus*. Cependant un pourcentage minimum du marqueur placé sur le dispositif doit être visible dans l'image. On utilisera alors différentes tailles de marqueurs : plus petits lorsque le zoom utilisé pour l'expérimentation est important, plus grand dans le cas contraire pour une meilleure précision (cf fig. 4.21).

Enfin, certains problèmes de positionnement virtuel peuvent se produire lorsque l'angle du dispositif par rapport à la caméra est trop important (cf fig. 4.22, à gauche). De plus, diverses occlusions visuelles comme les doigts de l'utilisateur ou le câble du capteur d'effort peuvent altérer la précision du recalage (cf fig. 4.22, au centre). Pour en améliorer la robustesse, nous disposons quatre marqueurs différents à 360° autour de l'outil de mesure. Nous utilisons ensuite le marqueur ayant reçu d'*ARToolkitPlus* la meilleure note de fiabilité pour positionner la touche sphérique à l'aide des équations 4.1 et 4.2 (cf fig. 4.22, à droite).

4.4 Mise en œuvre

Le matériel nécessaire à la mesure des données du monde réel est décrit dans les sections précédentes. Il nous est donc possible de réaliser des acquisitions lors d'expérimentations sur divers matériaux. Cependant, il est nécessaire de créer un logiciel permettant de capturer toutes ces informations en *temps réel*, de les enregistrer et de pouvoir les réutiliser lors de l'estimation de paramètres. De plus, ces informations proviennent de systèmes d'enregistrement différents (caméra vidéo, module d'acquisition du capteur d'effort, position et orientation de l'outil, ...) et sont cadencées **à des fréquences différentes**. Il est donc nécessaire d'enregistrer également des informations sur le *timing* de chaque mesure pour être capable de les restituer de manière *synchronisée*. Nous avons donc conçu un logiciel spécifique (cf Fig. 4.23) permettant de mesurer et d'enregistrer de manière asynchrone les informations depuis ces

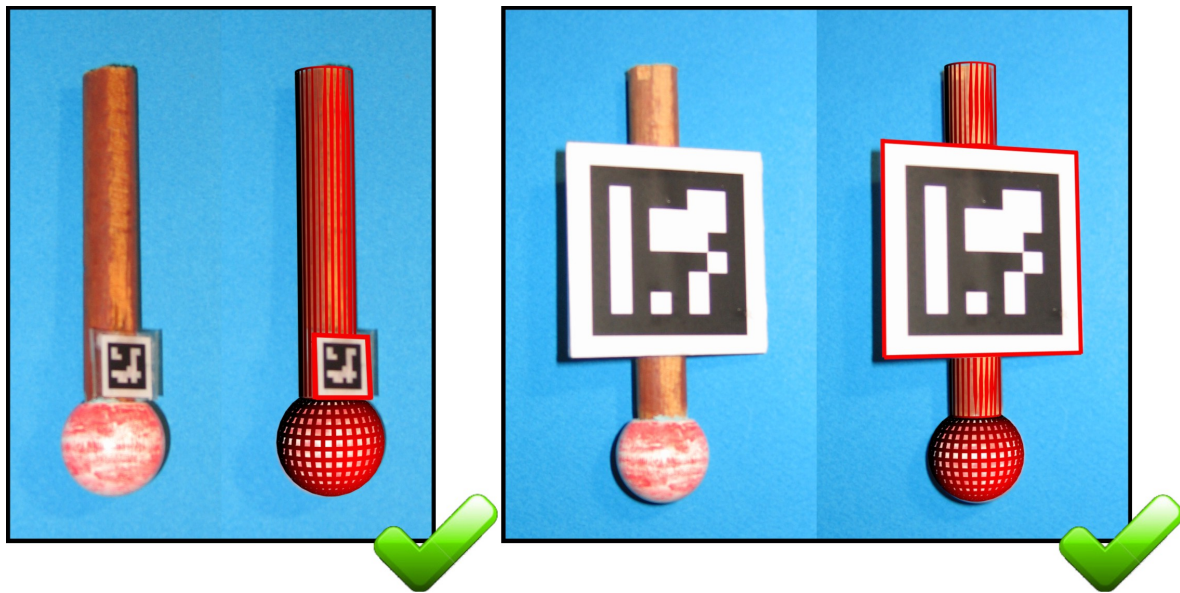


Figure 4.21 - Positionnement virtuel par vision à l'aide d'un marqueur ARToolkitPlus. À gauche, utilisation d'un marqueur de 1cm sur 1cm pour un zoom important. À droite, utilisation d'un marqueur de 4cm sur 4cm pour une meilleure précision sur une vue plus globale.

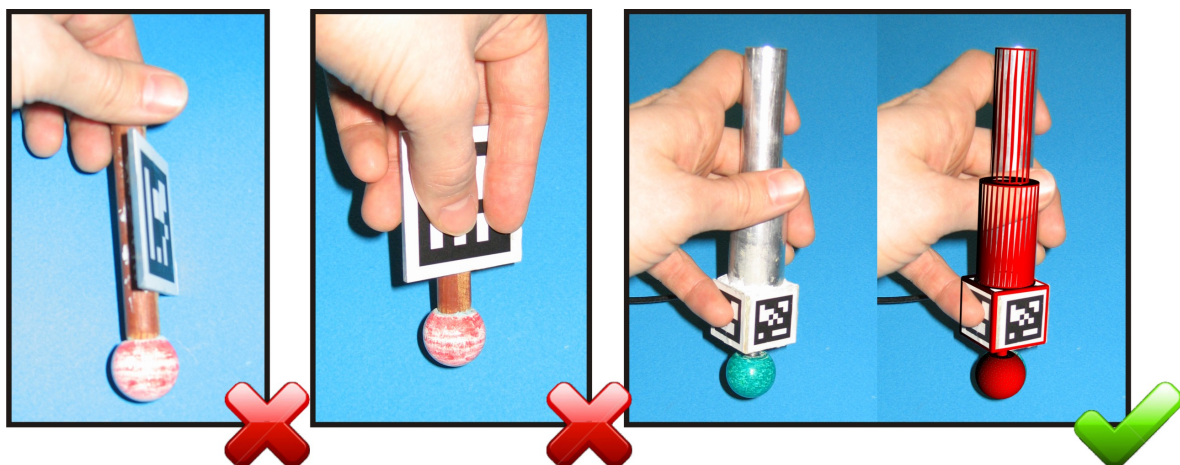


Figure 4.22 - Multiplication des marqueurs ARToolkitPlus sur le dispositif pour améliorer la robustesse. À gauche, problème d'angle de vue. Au centre, problème d'occlusion. À droite, utilisation de quatre marqueurs.

différents modules d'acquisition et de les restituer de manière synchrone (par interpolations le cas échéant).

La dernière version du positionnement virtuel de l'outil (par capture visuelle) nécessite une étape supplémentaire. En effet, bien que *ARToolkitPlus* permette un calcul de la position des marqueurs en temps interactif ($\approx 25Hz$), nous désirons maintenir une fréquence minimale des informations à 60Hz, soit la fréquence de capture vidéo. Nous verrons par la suite qu'en

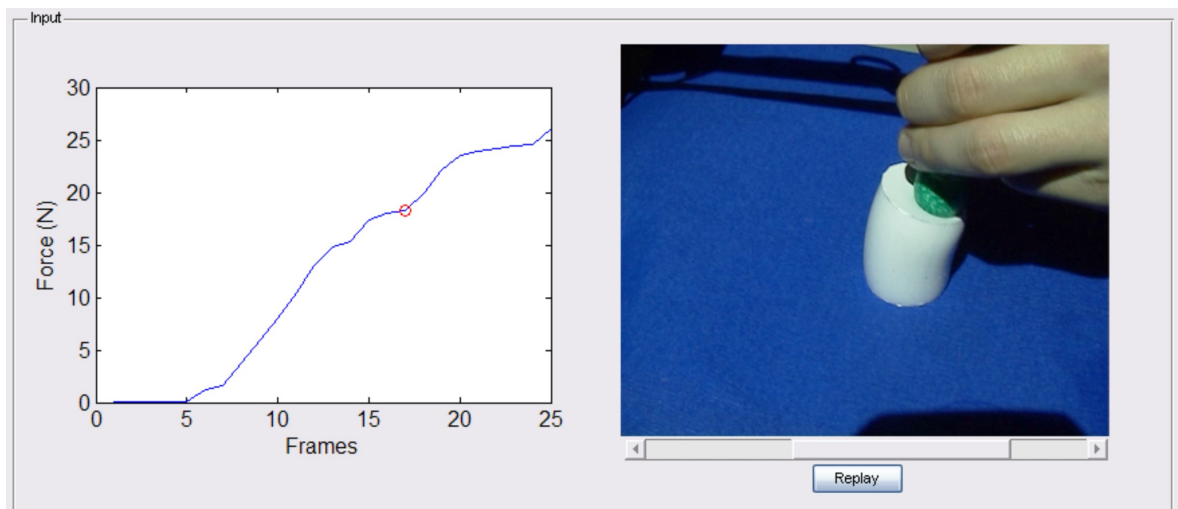


Figure 4.23 - Illustration du logiciel d'acquisition de données. À gauche, les forces mesurées lors de l'acquisition et à droite, l'image courante. Un marqueur sur le graphique des forces permet de synchroniser la force mesurée à l'image courante dans la vidéo correspondante.

effet, plus cette fréquence est élevée, plus les déplacements entre les itérations de l'algorithme de simulation mécanique (par éléments finis, cf 2.3.2) sont petits et donc moins l'erreur induite par l'hypothèse de petits déplacements est importante. Cet objectif ne peut donc être atteint que par un calcul du positionnement de l'outil *post-acquisition*, et donc non temps-réel. Une fois l'acquisition terminée, le logiciel propose donc un module de calcul *off-line* de la position et de l'orientation de l'outil qui sera réalisé pour chaque image de la séquence vidéo (cf Fig. 4.24).

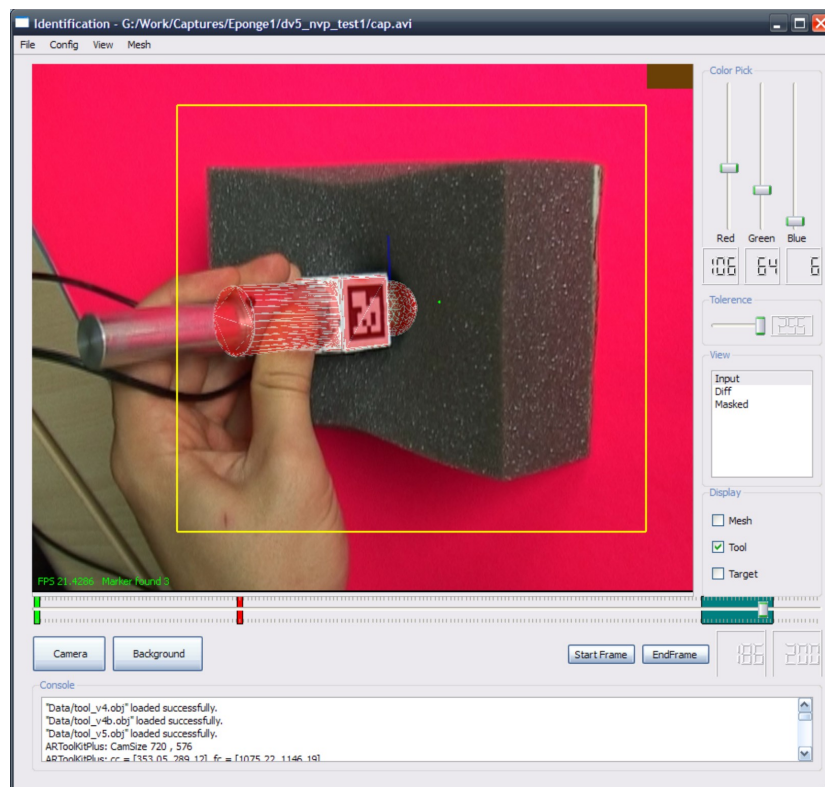


Figure 4.24 - Calcul offline du positionnement de l'outil par capture visuelle. Pour chaque image de la plage sélectionnée pour l'estimation de paramètre (zone verte sur le slider), la position de l'outil est précalculée à partir des marqueurs ARToolkitPlus (L'outil est affiché en transparence sur l'image initiale pour vérifier cette détection automatique).

Cinquième Chapitre

MÉTHODES ET ALGORITHMES D'ESTIMATION

Dans ce chapitre, nous exposons notre méthode d'estimation du coefficient de Poisson et du module de Young à partir de séquences d'images et de capture de forces. Nous détaillons tout d'abord notre algorithme de simulation mécanique d'objets déformables ainsi que notre méthode de rendu graphique puis nous montrons qu'il est possible d'estimer ces deux paramètres séparément pour minimiser l'espace de recherche. Nous expliquons ensuite notre méthode d'estimation inverse du coefficient de Poisson, basée sur des métriques d'erreur permettant de quantifier les différences entre deux séquences d'images. Enfin, nous proposons deux techniques pour estimer le module de Young, l'une directe, rapide, et l'autre, plus précise, basée sur un schéma d'estimation inverse. Nous proposons également deux algorithmes d'optimisation permettant d'accélérer l'évaluation de ces paramètres mécaniques.

5.1 Simulation mécanique

Notre méthode d'estimation de propriétés mécaniques à partir d'une séquence d'images nécessite un algorithme de simulation mécanique permettant de recréer le plus fidèlement possible les phénomènes physiques présents dans la vidéo réelle. Le pseudo-code décrit dans l'algorithme 2 illustre les grandes étapes de notre méthode d'estimation mécanique par éléments finis. Il fait référence aux équations décrites en 2.3.2. Cet algorithme est illustré sur la figure 5.1.

Algorithm 2 Algorithme général de notre simulation mécanique par éléments finis

```

for Chaque pas de temps de la vidéo réelle do
  // Initialisation
  Mise à jour de la position de la touche sphérique
  // Simulation mécanique par éléments finis contrainte en position
  (1) Recherche des sommets pénétrant la sphère
  (2) Résolution des contacts en repoussant ces sommets en dehors de la sphère
  (prise en compte de la friction [Duriez 04])
  (3) Blocage de ces sommets
  // Conditions de Dirichlet (cf 2.3.2.4)
  (4) Recherche des sommets en contact avec le support
  (5) Blocage de ces sommets
  // Résolution du système
  (6) Formulation linéaire en mode quasi-statique (cf equation 2.16)
  (7) Prise en compte du modèle corotationnel pour les grands déplacements (cf
  équation 2.18)
  (8) Calcul des déplacements des autres sommets et des efforts internes (cf
  équation 2.23)
  // Extraction de la force de contact
  (9) Projection des efforts internes des sommets en contact avec la touche
  sphérique sur la direction de l'action utilisateur
  (10) Somme et calcul de la norme du vecteur résultant
  // Enregistrement des résultats pour ce pas de temps
  (11) Sauvegarde des nouvelles positions des sommets
  (12) Sauvegarde de la force de contact calculée
end for

```

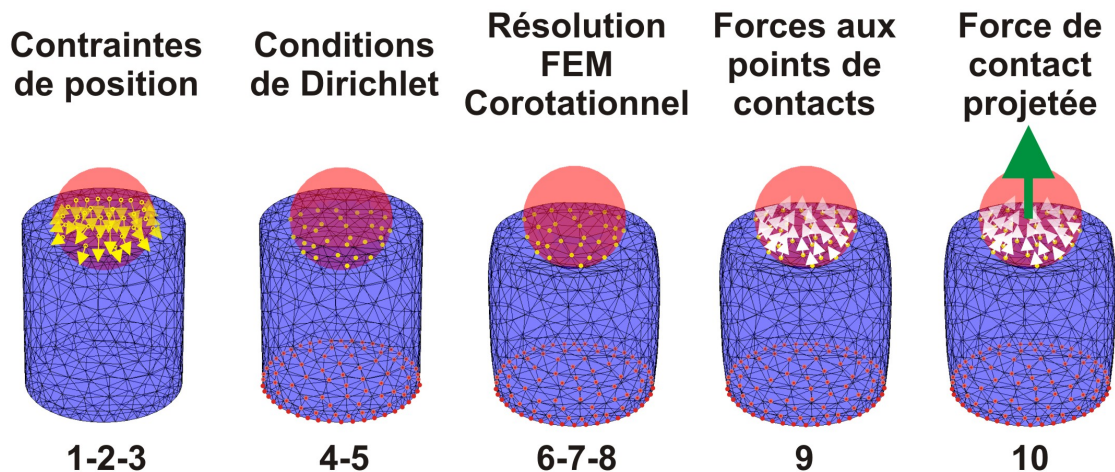


Figure 5.1 - Illustration des étapes de notre algorithme de simulation par éléments finis pour un pas de temps de simulation (décrites dans l'algorithme 2). Pour plus de clarté, seuls les sommets de la surface sont affichés. De plus, pour une meilleure compréhension, le déplacement de la sphère pour ce pas de temps est volontairement augmenté ($\times 15$).

5.2 Rendu graphique

Nous verrons dans la suite de ce document que notre méthode nécessite une étape de rendu graphique (cf 5.4). Celui-ci est destiné à transformer une animation en trois dimensions issue de notre algorithme de simulation mécanique (cf 5.1) en une *séquence vidéo synthétique*. Celle-ci sera ensuite comparée à la séquence vidéo réelle acquise avec le logiciel de capture.

Il existe de nombreuses méthodes pour calculer une image synthétique à partir de données en trois dimensions. Pour choisir l'algorithme adéquat, il faut évaluer différents critères. Nous considérons ici les critères de *réalisme* et de *temps de calcul* car bien qu'il existe une multitude de techniques différentes, il s'agit dans tous les cas de faire un compromis entre ces deux critères.

Nous classons les méthodes de rendu graphique 3D en deux catégories : les méthodes de rendu en *temps-réel* et les méthodes dites « *off-line* ». Les méthodes en temps-réel sont assez rapides pour afficher des images à une cadence telle qu'il est impossible de distinguer la mise à jour à l'œil nu. Elles sont utilisées par exemple dans les jeux vidéos ou dans les logiciels de visualisation en temps interactif et se basent principalement, à l'heure actuelle, sur deux bibliothèques : *Direct3D*¹ et *OpenGL*². Bien que ces méthodes offrent des résultats de plus en plus réalistes, notamment depuis l'adjonction d'unités programmables directement sur les cartes graphiques à l'aide des *shaders*, elles sont encore loin d'égaliser le *photo-réalisme* qu'il est possible d'obtenir avec les algorithmes plus complexes, basés par exemple sur les interactions des photons avec l'environnement, comme le *ray-tracing*, le *photon-mapping*[Jensen 01] ou encore le *photon-tracing*. Il existe une multitude d'autres algorithmes, ayant chacun leurs spécificités, utilisés dans les cas où le *réalisme* prime sur le *temps de calcul*, comme par exemple dans l'industrie cinématographique avec le rendu de films d'animations ou l'ajout d'effets spéciaux.

Dans notre cas, il semble dans un premier temps plus judicieux de privilégier le réalisme à la rapidité. En effet, plus l'algorithme de rendu pourra s'approcher de la réalité et donc de l'animation réelle filmée, plus les chances de retrouver les bons paramètres mécaniques seront importantes. Cependant, plusieurs observations viennent contredire cette hypothèse. Tout d'abord, les méthodes de comparaison de vidéos n'ont pas toujours besoin d'un grand réalisme. En effet, dans la plupart des cas, il s'agira plutôt de dégrader l'animation réelle, par exemple par une extraction de silhouette. Dans ce cas, inutile donc par exemple de calculer pendant des heures l'interaction de la lumière avec un milieu participatif comme un matériau translucide si le but est uniquement de détecter le contour de l'objet déformé avant de le comparer à la silhouette extraite de la vidéo réelle.

D'autre part, même si nous ne nous imposons pas de contrainte de temps-réel (les estimations de propriétés mécaniques peuvent être majoritairement effectuées « *off-line* »), il faut garder à l'esprit que cette étape de rendu graphique est effectuée à l'intérieur d'une boucle itérative (cf fig. 3.4). Cette étape est donc amenée à être appelée un grand nombre de fois³ et ne doit donc pas être un goulot d'étranglement algorithmique.

De plus, nous limitons nos expérimentations à des matériaux simples et un éclairage relativement basique. L'algorithme de rendu doit être capable de calculer un éclairage un

1. <http://fr.wikipedia.org/wiki/Direct3D>

2. <http://fr.wikipedia.org/wiki/OpenGL>

3. Le nombre d'itérations dépendra de la méthode d'optimisation choisie, du nombre de paramètres à estimer et de la précision finale désirée.

point avec ombres projetées, de simuler des matériaux simples comme le silicone blanc opaque et de projeter une texture depuis la source de lumière pour simuler le vidéo-projecteur. La librairie *OpenGL* remplit toutes ces conditions et nous avons donc opté pour celle-ci. La figure 5.2 donne un aperçu des résultats qu'il est possible d'obtenir en quelques fractions de secondes à l'aide d'une carte graphique standard (« GeForce 6600 », © *NVIDIA*).

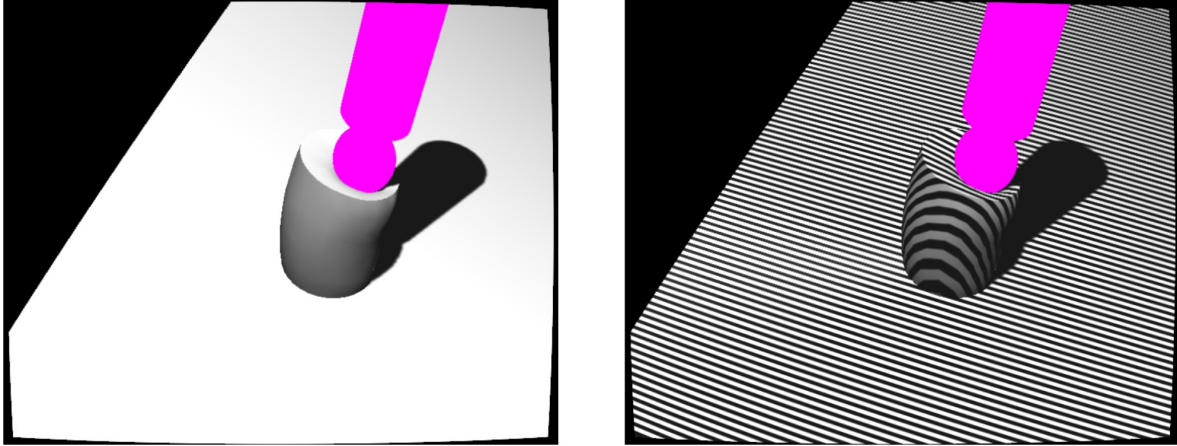


Figure 5.2 - Exemple de rendu graphique en temps-réel avec la librairie *OpenGL*. À gauche, avec un éclairage classique un point et projection d'ombre, à droite avec une projection de texture.

L'algorithme 3 décrit les différentes étapes permettant de générer une animation synthétique depuis le résultat de notre simulation mécanique à l'aide de la librairie *OpenGL*.

Algorithm 3 Algorithme de génération d'une séquence vidéo synthétique

```
// Initialisation
(1) Positionner et orienter la caméra à l'aide des informations de calibrage
(2) Positionner et orienter la source lumineuse
(3) Charger la texture de projection (blanc si aucune)
(4) Charger l'animation en trois dimensions issue de la simulation mécanique
(5) Initialiser le fichier vidéo
(6) Charger les propriétés des matériaux
for Chaque pas de temps do
  (a) Retrouver les facettes des tétraèdres externes
  (b) Positionner l'outil depuis les informations utilisateur mesurées
  (c) Effectuer le rendu OpenGL au format de la vidéo réelle
  (d) Simuler la distortion optique
  (e) Capturer l'image résultat
  (f) Enregistrer cette image dans le fichier vidéo
end for
```

Distortion optique Les images qui seront calculées en temps réel par *OpenGL* seront par la suite comparées aux images de la séquence vidéo réelle. Or, la grande majorité des caméras réelles ont un défaut optique. En effet, l'image filmée est un peu déformée. Ce phénomène est connu sous le nom de *distortion optique*. On peut noter par ailleurs que plus la qualité de la caméra vidéo est importante, moins ce phénomène est visible. Notre choix de caméra

étant relativement modeste, la distortion qu'elle génère devient évidente lorsque l'on fait une comparaison avec une source synthétique « parfaite ». Pour que la comparaison d'images soit cohérente, il faut donc prendre en compte ce phénomène.

Malheureusement, le modèle de caméra pyramidal utilisé par OpenGL (aussi appelé *pin-hole* ne permet pas de simuler cet effet d'optique. Deux options sont alors possibles. Supposons que nous disposions d'une fonction **Distortion** qui, pour un pixel ϕ_{orig} donné dans l'espace non déformé et un certain nombre de *paramètres intrinsèques* $P_{intrinsèques}$ de la caméra retourne l'emplacement du pixel correspondant dans l'espace déformé :

$$\phi_{disto} = \mathbf{Distortion}(\phi_{orig}, P_{intrinsèques}) \quad (5.1)$$

S'il est possible de calculer l'inverse de cette fonction $\mathbf{Distortion}^{-1}$, il est alors possible de modifier l'image initiale pour en éliminer la distortion (généralement avec une petite perte d'information sur les bords de l'image). Dans le cas contraire, il est possible de modifier l'image synthétique non déformée et ainsi simuler, après le rendu OpenGL, la distortion optique de la caméra. Nous avons vu en 4.2.3 que le *Camera Calibration Toolbox* permet de retrouver les paramètres intrinsèques de caméras vidéos tels que la distortion radiale. La librairie *ARToolkitPlus* fournit quand à elle les fonction **Distortion** et $\mathbf{Distortion}^{-1}$. Nous utilisons ces éléments pour simuler la distortion optique en modifiant le rendu OpenGL par une étape supplémentaire, illustrée par la figure 5.3.

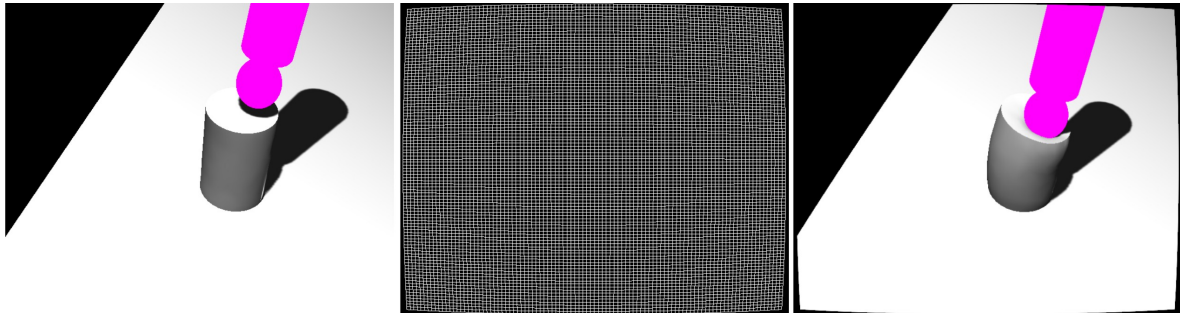


Figure 5.3 - Simulation de la distortion optique : le rendu OpenGL est copié dans une texture (ou *render to texture*, à gauche). Ensuite, l'image est morcelée en utilisant la fonction **Distortion** (au centre) puis réaffichée en prenant cette distortion en compte (à droite). Plus la définition de la grille est élevée, moins l'erreur induite par cette méthode est importante.

5.3 Estimations séparées

Nous avons vu en section 3.2 les grandes familles d'algorithmes d'optimisation. De manière générale, la vitesse de convergence de ces algorithmes dépend de la taille de l'espace de recherche. Il est évident que plus le nombre de paramètres à estimer est important, plus cet espace de recherche des paramètres optimaux est vaste. Effectuer cette recherche pour np paramètres à la fois revient à rechercher dans un espace de taille x^{np} , où x est le nombre de valeurs possibles pour chaque paramètre, soit la taille de l'intervalle de validité du paramètre divisé par la précision voulue (soit $\frac{[min...max]}{précision}$).

Quel que soit l'algorithme de recherche choisi, il serait intéressant pouvoir estimer les paramètres les uns après les autres et ainsi réduire cet espace de recherche à $x \times np$. Pour y parvenir, il est tout d'abord nécessaire de s'assurer de la faisabilité d'une telle opération, autrement dit que les paramètres ne soient pas inter-dépendants (ou *corrélés*). Ensuite, il faut pouvoir déterminer l'ordre dans lequel ces estimations doivent être réalisées.

5.3.1 Étude de corrélation/décorrélation

Pour étudier le comportement induit par les paramètres ainsi que leurs éventuelles corrélations, il suffit de générer, à l'aide de l'algorithme de simulation mécanique à tester, plusieurs animations synthétiques initiales en utilisant des paramètres connus. Le but de l'étude est alors de retrouver ces paramètres de référence en balayant l'espace de recherche, ce qui permet également d'étudier le comportement des *fonctions objectives* choisies. Dans notre cas, une fonction objective doit quantifier la différence entre deux animations synthétiques. Nous définissons pour cette étude trois fonctions de comparaison. La première compare les différences géométriques (cf equation 5.2) et la seconde compare les efforts internes (cf equation 5.3). Ces deux fonctions servent à étudier les *domaines d'influence* des paramètres. La dernière fonction de comparaison les combine (cf equation 5.4). Voici le détail de la procédure que nous avons réalisée pour une animation initiale générée avec un coefficient de Poisson $\nu = 0.33$ et un module de Young $E = 50000Pa$.

Procédure Nous générons l'animation synthétique de référence en trois dimensions. Cette animation est une séquence d'un même modèle géométrique de base déformé à chaque pas de temps. Nous prenons dans cet exemple un cylindre volumique composé de tétraèdres et nous appliquons une simulation mécanique par éléments finis en contraignant la base de ce cylindre (fixe). Nous imposons une déformation à l'aide d'une sphère mobile qui contraint l'objet à se déformer. Cette sphère simule une action de l'utilisateur sur un cylindre fait d'un matériau élastique. La figure 5.4 (en haut) illustre quelques pas de temps de cette animation géométrique. Un jeu de paramètres P_{init}^{np} est utilisé pour la générer : $P_{init}^1 = \nu = 0.33$ et $P_{init}^2 = E = 50000Pa$. Ces valeurs sont les paramètres à identifier⁴. Outre la partie géométrique, les efforts internes obtenus par la simulation mécanique sont stockés pour chaque pas de temps.

Ensuite, nous procédons au balayage de l'espace de recherche. Pour chaque paire de paramètres, nous générons $n \times m$ jeux de paramètres, où n est le nombre d'échantillons que nous désirons tester pour le coefficient de Poisson (soit $\frac{P_{max}^1 - P_{min}^1}{précision}$) et m le nombre d'échantillons pour le module de Young. À l'aide de ces $n \times m$ jeux de paramètres, nous simulons mécaniquement $n \times m$ animations géométriques en trois dimensions (cf 5.1) en utilisant les mêmes contraintes et le même algorithme que pour l'animation initiale (seul le jeu de paramètres utilisé change). Les efforts internes sont également stockés à chaque pas de temps et pour chaque animation synthétique.

Ces animations sont ensuite comparées à l'animation de référence. Nous définissons une *métrique d'erreur* (ou *fonction objective*) pour chaque domaine d'influence. Ces fonctions permettent de quantifier les différences existantes entre deux animations. La première, Δ_{Anim3D} ,

4. Les autres paramètres nécessaires à la simulation mécanique, comme le coefficient de friction, sont fixés pour toute la procédure.

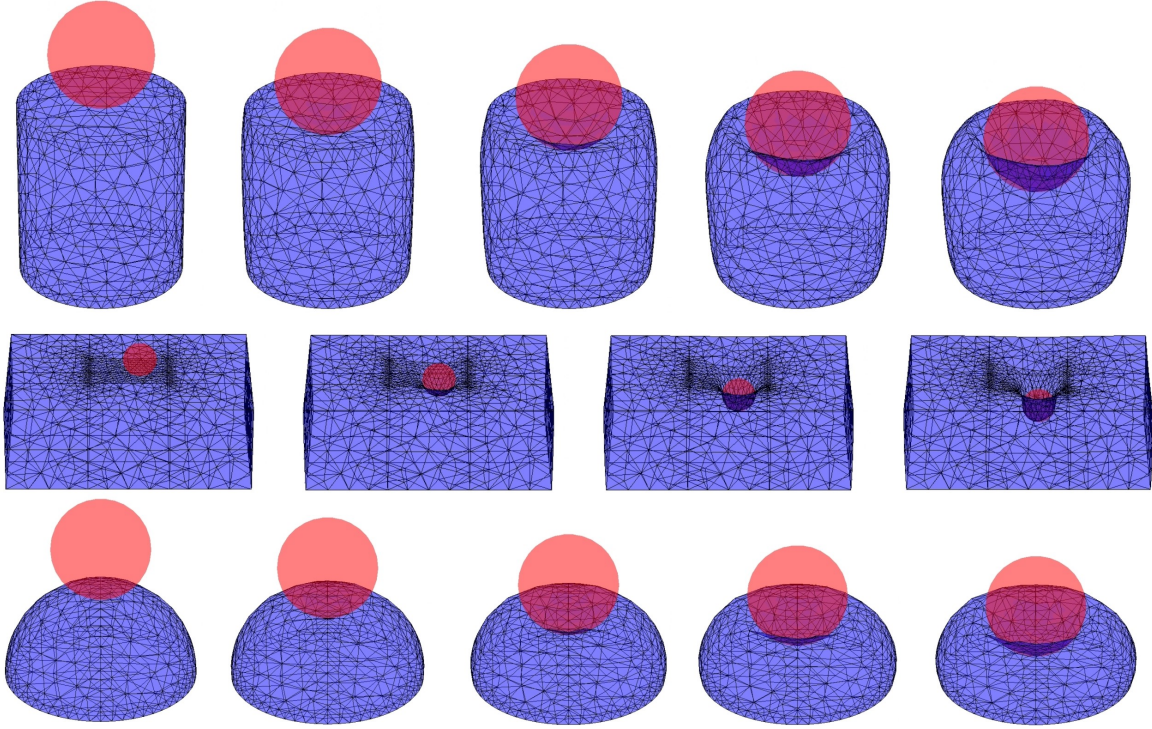


Figure 5.4 - Suite d'états géométriques obtenus avec une simulation par éléments finis formant une animation géométrique synthétique (module de Young $E = 50kPa$ et coefficient de Poisson $\nu = 0.33$). En haut, pour un cylindre volumique (3502 tétraèdres), au centre, pour un parallélépipède rectangle (4225 tétraèdres), et en bas pour une hémisphère (2910 tétraèdres).

permet de comparer deux animations géométriques en trois dimensions. La seconde, Δ_{Forces} permet quant à elle de quantifier les différences entre les efforts internes de deux simulations mécaniques.

Soit deux animations géométriques \mathbf{A} et \mathbf{B} de t pas de temps mettant en scène un objet déformable de s sommets, chaque sommet étant un vecteur en trois dimensions (x, y, z) , ainsi qu'une fonction $\text{length}(\mathbf{v}) = \sqrt{\mathbf{v}.x^2 + \mathbf{v}.y^2 + \mathbf{v}.z^2}$ qui retourne la taille d'un vecteur. On peut définir alors Δ_{Anim3D} par les sommes des carrés des distances pour chaque sommet sur chaque image de la séquence (moyenne globale) :

$$\varepsilon_{3D} = \Delta_{\text{Anim3D}}(\mathbf{A}, \mathbf{B}) = \frac{1}{t} \sum_{k=1}^t \sum_{i=1}^s \text{length}(\mathbf{A}_i^k - \mathbf{B}_i^k) \quad (5.2)$$

où \mathbf{A}_i^k est le $i^{\text{ème}}$ sommet du pas de temps k pour l'animation \mathbf{A} , où \mathbf{B}_i^k est le $i^{\text{ème}}$ sommet du pas de temps k pour l'animation \mathbf{B} , et ε_{3D} est la somme des distances inter-sommets quantifiant la différence entre ces deux animations géométriques en trois dimensions.

De la même manière, si l'on pose $\mathbf{F}\mathbf{a}_i^k$ l'effort simulé sur le $i^{\text{ème}}$ sommet du pas de temps k pour l'animation \mathbf{A} définie ci-dessus et $\mathbf{F}\mathbf{b}_i^k$ l'effort simulé sur le $i^{\text{ème}}$ sommet de l'image k pour l'animation \mathbf{B} , on peut définir Δ_{Forces} :

$$\varepsilon_{forces} = \Delta_{\text{Forces}}(\mathbf{Fa}, \mathbf{Fb}) = \frac{1}{t} \sum_{k=1}^t \sum_{i=1}^s (\mathbf{Fa}_i^k - \mathbf{Fb}_i^k)^2 \quad (5.3)$$

Pour pouvoir comparer la séparation des domaines à une erreur globale, nous définissons également une métrique générale :

$$\varepsilon_{glob} = \Delta_{\text{Glob}}(\mathbf{A}, \mathbf{BFa}, \mathbf{Fb}) = \Delta_{\text{Anim3D}}(\mathbf{A}, \mathbf{B}) \times \Delta_{\text{Forces}}(\mathbf{Fa}, \mathbf{Fb}) \quad (5.4)$$

L'étude de la fonction est alors faite en comparant, à l'aide de ces trois métriques d'erreur, l'animation initiale avec les animations qui balayent l'espace de validité des paramètres. Plusieurs grilles d'erreur par couple de paramètres nous permettent alors d'étudier l'interdépendance de ceux-ci, dans la globalité ou pour chaque domaine d'influence. La figure 5.5 illustre le résultat d'une étude de décorrélation des paramètres dans notre cas pour $\nu = 0.33$ et $E = 50000$ avec plusieurs géométries.

5.3.2 Analyse et ordre d'estimation

Trois remarques peuvent être faites en analysant la figure 5.5. Tout d'abord, avec la fonction Δ_{Glob} (1), l'algorithme converge bien vers les paramètres initiaux (minimums globaux de la fonction) puisque la paire de paramètres générant l'erreur la plus faible est celle qui s'approche le plus des valeurs initiales connues.

Ensuite, on peut noter qu'avec la fonction Δ_{Anim3D} (2), l'erreur est identique pour une valeur du coefficient de Poisson donnée, et ce quelle que soit la valeur du module de Young. Cela prouve que le module de Young n'influence pas la géométrie de l'objet et donc, par conséquent, qu'il n'influera pas sur la vidéo synthétique qui sera générée à partir de celle-ci. Ce résultat est logique puisque nous contraignons la simulation mécanique en position. Le module de Young n'affecte donc que les efforts internes (comportement d'un ressort). Par conséquent, on peut donc dire que les deux paramètres sont décorrélés pour le domaine d'influence « géométrie ».

En revanche, en analysant le comportement de la fonction Δ_{Forces} (3), on peut voir que bien que le module de Young aie la plus grande influence sur les efforts générés, le coefficient de Poisson participe dans une moindre mesure à la modification de ces efforts (principalement pour des valeurs proches de 0.5). Les deux paramètres ne sont donc pas totalement décorrélés sur le domaine d'influence « efforts internes ».

Cependant, ces observations nous permettent tout de même d'effectuer une estimation des paramètres séparément. En effet, puisque le module de Young *n'intervient pas dans la déformation géométrique* obtenue, il suffit d'estimer, **dans un premier temps, le coefficient de Poisson à partir des animations géométriques**, puis, **ensuite, d'utiliser le résultat obtenu et de le fixer pour estimer le module de Young à partir des forces**.

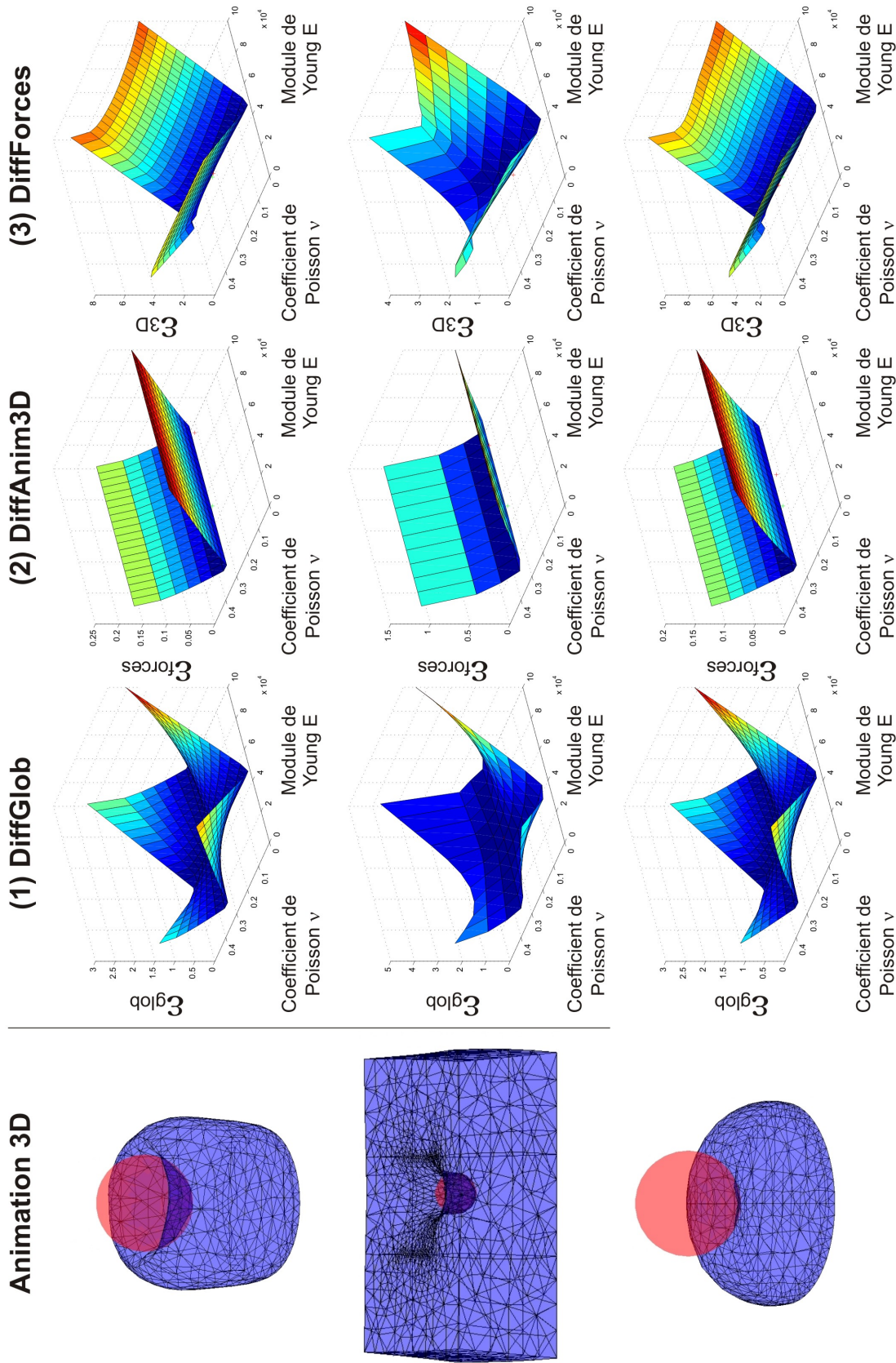


Figure 5.5 - Étude de décorrélation des paramètres pour les animations géométriques initiales de la figure 5.4 créées avec $\nu = 0.33$ et $E = 50000$ (en haut, cylindre volumique, au centre, parallélépipède, en bas, hémisphère). L'échelle de couleurs utilisée est thermique : le bleu marine représente une erreur nulle (meilleure estimation) et le rouge représente une erreur forte (moins bonne estimation).

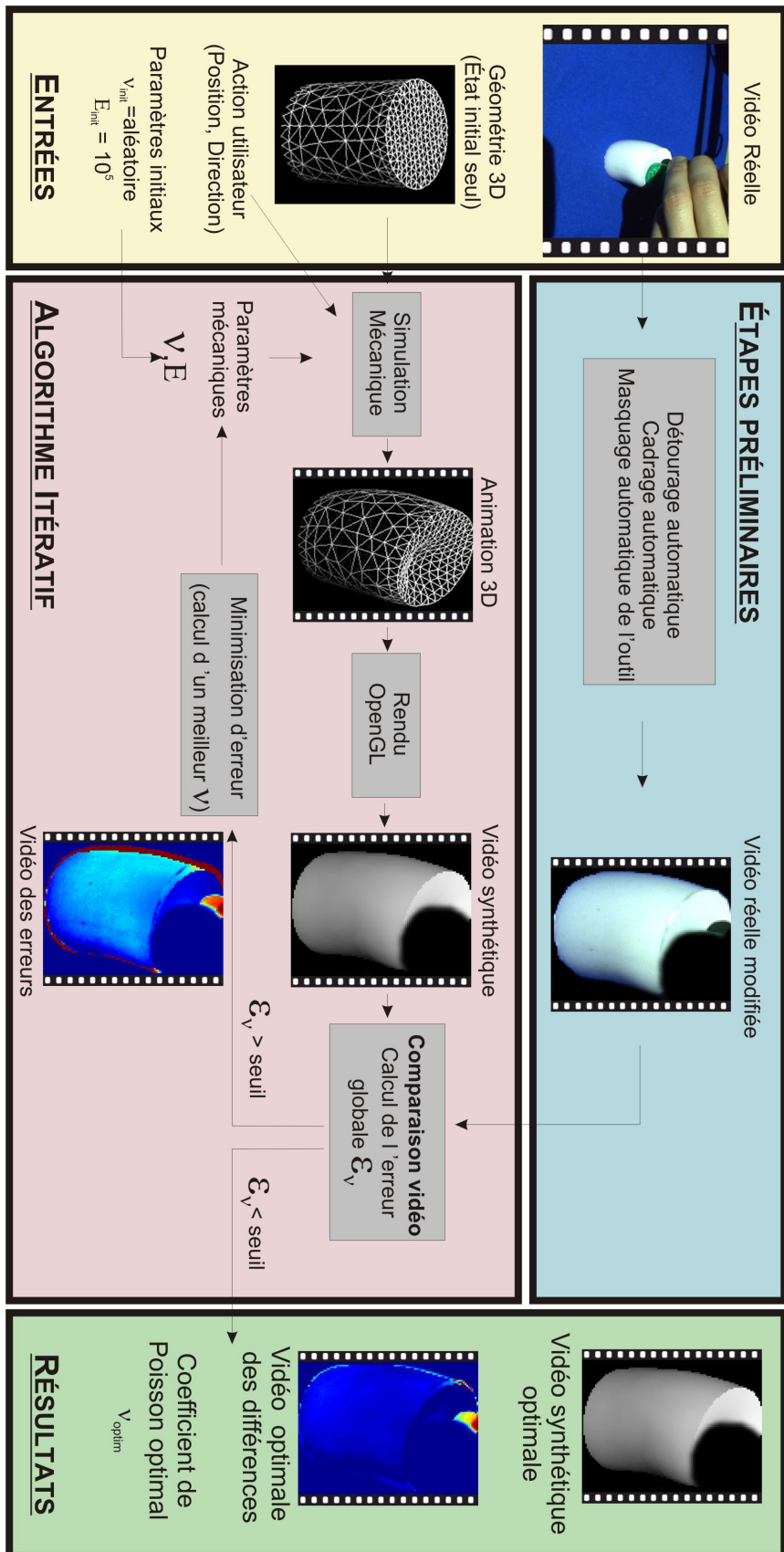


Figure 5.6 - Schéma global de notre algorithme d'estimation du coefficient de Poisson à partir de séquences d'images.

5.4 Estimation du coefficient de Poisson

5.4.1 Méthode

La figure 5.6 décrit notre algorithme pour estimer le coefficient de Poisson à partir d'une séquence d'images et d'une action utilisateur enregistrée. Tout d'abord, nous procédons à plusieurs étapes préliminaires destinées à réaliser les pré-calculs qui seront utilisés par l'algorithme itératif d'estimation de paramètres (cf 5.4.2). Ensuite, nous choisissons une valeur initiale pour le *coefficient de Poisson* (ν_{init}). La valeur initiale du *module de Young* n'est pas importante puisqu'elle n'influera pas sur l'animation géométrique (cf 5.3). Nous utilisons ce jeu de paramètres initial pour la première boucle de l'algorithme d'estimation du coefficient de Poisson. Tout d'abord, nous l'utilisons pour générer une animation géométrique à l'aide de l'algorithme de simulation mécanique par éléments finis décrit en 2.3.2. Ensuite, nous transformons cette animation 3D en une séquence d'images par un rendu graphique (cf 5.2). Celle-ci est alors comparée à la séquence vidéo réelle modifiée (cf 5.4.2 et 5.4.3). Si le résultat de ce calcul d'erreur est supérieur à un seuil prédéfini, une étape de minimisation d'erreur est alors enclenchée. Celle-ci permet de déterminer une nouvelle valeur du *coefficient de Poisson* ν qui sera utilisée pour la prochaine boucle de l'algorithme itératif. Lorsque l'erreur calculée est inférieure au seuil prédéfini, la dernière valeur de ν utilisée est alors considérée optimale. Le résultat est alors la valeur du coefficient de Poisson ν_{optim} .

5.4.2 Traitements d'images préliminaires

Nous définissons ici quelques étapes de traitement d'image destinées à réaliser les pré-calculs qui seront utilisés par l'algorithme itératif d'estimation de paramètres. En effet, les éléments traités ici ne varient pas en fonction des paramètres recherchés, il est inutile et contre-performant de les recalculer à chaque itération.

5.4.2.1 Détourage automatique

Pour extraire la déformation du solide à partir de la vidéo, nous utilisons tout d'abord un algorithme de *détourage par couleur* ou *color keying*. Le principe est simple : l'enregistrement de la vidéo se fait sur un fond de couleur unie. Cette couleur étant connue, il est facile de l'extraire de la vidéo résultante. Cette technique est très utilisée pour l'ajout d'effets spéciaux ou d'incrustations dans l'industrie cinématographique, par exemple avec l'utilisation d'un *blue screen* (sur fond bleu).

Soit une couleur \mathbf{C} définie par ses trois composantes rouge, vert et bleu (respectivement \mathbf{C}_r , \mathbf{C}_g , et \mathbf{C}_b) et une fonction $\Phi_{i,j}^{V^k}$ qui retourne les composantes couleur du pixel de coordonnées i, j de la $k^{\text{ième}}$ images d'une séquence \mathbf{V} , la vidéo \mathbf{V}_{Coul} contenant l'animation extraite, définie à partir de la vidéo réelle $\mathbf{V}_{Réelle}$ est définie pour chaque pixel de chaque image par :

$$\delta_{coul}(\mathbf{C}^A, \mathbf{C}^B) = (\mathbf{C}_r^A - \mathbf{C}_r^B)^2 + (\mathbf{C}_g^A - \mathbf{C}_g^B)^2 + (\mathbf{C}_b^A - \mathbf{C}_b^B)^2 \quad (5.5)$$

$$\Phi_{i,j}^{V_{Coul}^k} = \begin{cases} \Phi_{i,j}^{V_{Réelle}^k} & \text{si } \delta_{coul}(\Phi_{i,j}^{V_{Réelle}^k}, \mathbf{C}) < Tol_{coul} \\ (0, 0, 0) & \text{sinon} \end{cases} \quad (5.6)$$

où Tol_{coul} est un seuil de tolérance qui permet de déterminer si un pixel est masqué lorsque sa couleur est suffisamment proche de \mathbf{C} . Bien que cette méthode soit simple à mettre en œuvre,

on peut voir sur la figure 5.7 (a et c) que ce paramètre est relativement sensible. Comme toute méthode utilisant un seuillage, elle est difficile à régler.



Figure 5.7 - Exemple de détourage automatique sur fond bleu (avec $\mathbf{C} = (0, 0.01, 0.20)$) : à gauche, $Tol_{coul} = 0.10$, au centre avec $Tol_{coul} = 0.25$ et à droite avec $Tol_{coul} = 0.77$. Pour une meilleure visibilité, le masque calculé est affiché en rose soutenu.

Il existe d'autres méthodes de détourage. Nous avons également implémenté une méthode par « apprentissage d'arrière-plan » (*background learning*). Il s'agit de faire une capture de l'environnement sans les informations utiles. Dans notre cas, il s'agira de la table sous l'éclairage du vidéo-projecteur. Ensuite, chaque image de la vidéo est comparée à cette capture initiale et une « soustraction » permet d'isoler les éléments ajoutés depuis cette capture. Cette méthode fonctionne bien dans les cas où la caméra est fixe, ce qui est notre cas.

Soit l'image capturée *Capture*, le résultat d'un détourage par apprentissage d'arrière-plan est défini pour chaque pixel de chaque image de l'animation par :

$$|\mathbf{C}| = (|\mathbf{C}_r|, |\mathbf{C}_g|, |\mathbf{C}_b|) \quad (5.7)$$

$$\Phi_{i,j}^{VBack^k} = \left| \Phi_{i,j}^{VReelle^k} - \Phi_{i,j}^{Capture} \right| \quad (5.8)$$

Pour améliorer la robustesse du détourage automatique, nous utilisons une combinaison de ces deux méthodes. Dans un premier temps, nous faisons une extraction par apprentissage d'arrière-plan puis nous améliorons le résultat par une extraction de couleur (cf Fig. 5.8). La vidéo résultat V_{Detour} est défini par :

$$\Phi_{i,j}^{V_{Detour}^k} = \begin{cases} \Phi_{i,j}^{VBack^k} & \text{si } \delta_{coul}(\Phi_{i,j}^{VBack^k}, \mathbf{C}) < Tol_{coul} \\ (0, 0, 0) & \text{sinon} \end{cases} \quad (5.9)$$

Cette technique hybride, bien qu'utilisant toujours un seuil, à montré dans nos expérimentations qu'elle était moins sujette au bruit que l'une des deux méthodes seules. Des travaux plus évolués sur ce sujet peuvent être trouvés dans la littérature (par exemple [Li 04]), mais notre solution s'est révélée suffisante pour nos séquences vidéos et présente l'avantage d'être rapide à mettre en œuvre et très peu coûteuse en temps de calcul.

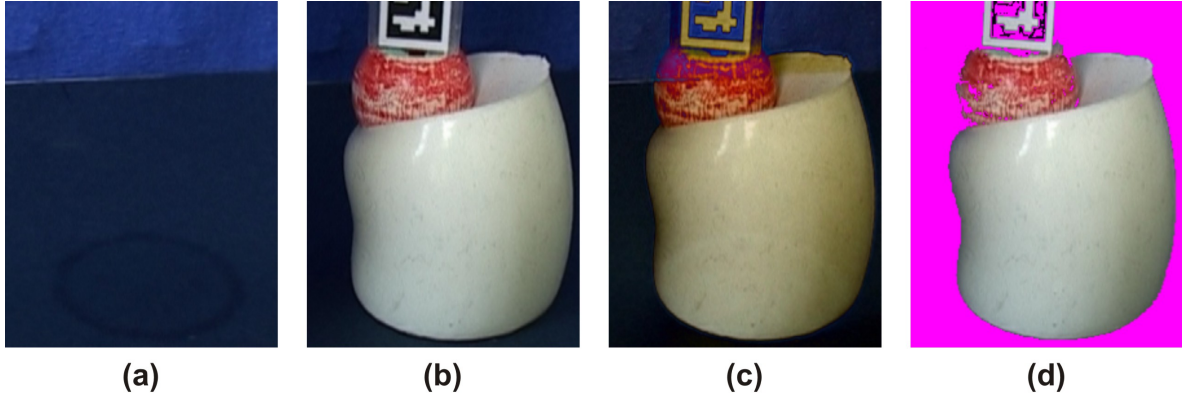


Figure 5.8 - Méthode hybride de détourage : (a) Image d'arrière-plan capturée (**Capture**). (b) Image de l'animation réelle. (c) Résultat de l'extraction par apprentissage d'arrière-plan (cf équation 5.8). (d) Résultat de la méthode hybride après l'application d'un détourage par couleur (avec $\mathbf{C} = (0.60, 0.58, 0.41)$ et $Tol_{coul} = 0.57$. Pour une meilleure visibilité, le masque calculé est affiché en rose soutenu.)

5.4.2.2 Masquage automatique de l'outil

Pour extraire les informations utiles d'une image, il est nécessaire d'en masquer certaines zones. C'est le cas de toutes les occlusions visuelles, comme la main de l'utilisateur ou le câble du capteur d'effort mais également de l'outil en lui-même. Pour masquer cet outil, nous effectuons une étape préliminaire. Nous utilisons les données utilisateur en entrée ainsi que tous les paramètres de la caméra et la géométrie de l'outil pour générer une animation synthétique $\mathbf{V}_{\text{Outil}}$. Cette vidéo ne contient que l'outil en mouvement. De plus, le rendu de cette vidéo n'utilise pas de calcul d'éclairage. C'est donc une animation binaire (cf Fig. 5.9, au centre) qui est définie, pour chaque pixel i, j de chaque image k , par :

$$\Phi_{i,j}^{\mathbf{V}_{\text{Outil}}^k} = \begin{cases} 1 & \text{si le pixel est sur l'outil,} \\ 0 & \text{sinon} \end{cases} \quad (5.10)$$

Nous ajoutons une étape supplémentaire. En effet, pour minimiser les erreurs de positionnement de l'outil et le phénomène de *flou de bougé* (également connu sous le nom de *motion blur*), nous réalisons un flou de cette vidéo. Soit une matrice carrée $[\mathbf{H}]$, de taille tf^2 , contenant les coefficients qui déterminent une Gaussienne. L'équation 5.11 illustre une telle matrice pour $tf = 3$.

$$[\mathbf{H}]_{33} = \begin{bmatrix} 0.0625 & 0.125 & 0.0625 \\ 0.125 & 0.25 & 0.125 \\ 0.0625 & 0.125 & 0.0625 \end{bmatrix} \quad (5.11)$$

La *convolution* d'une image \mathbf{I} par une telle matrice résulte en un flou gaussien de celle-ci, défini pour chaque pixel $\Phi_{i,j}^{\mathbf{I}}$:

$$\mathbf{Flou} \left(\Phi_{i,j}^I, tf \right) = \frac{1}{tf^2} \sum_{dx=-\frac{tf}{2}}^{\frac{tf}{2}} \sum_{dy=-\frac{tf}{2}}^{\frac{tf}{2}} \Phi_{i+dx,j+dy}^I [\mathbf{H}]_{tf \times tf} \left(dx + \frac{tf}{2}, dy + \frac{tf}{2} \right) \quad (5.12)$$

Nous définissons alors le masque automatique de l'outil (cf Fig. 5.9, à droite) par :

$$\Phi_{i,j}^{VMasqueOutil^k} = \mathbf{Flou} \left(\Phi_{i,j}^{VOutil^k}, 5 \right) \quad (5.13)$$

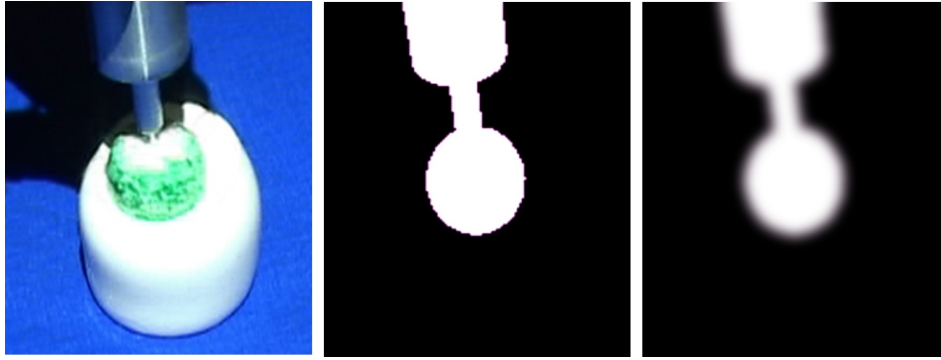


Figure 5.9 - Masquage automatique de l'outil : À gauche, l'image initiale d'une animation. Au centre, la première étape du masquage de l'outil (cf équation 5.10). À droite, le masque de l'outil final (cf équation 5.13).

Le lecteur notera que cette méthode entraîne une petite perte d'information. En effet, selon l'angle de vue et quand le matériau est très déformable, un phénomène de recouvrement de l'outil apparaît. Notre masquage automatique de l'outil supprime cette information. Cependant, celle-ci est également supprimée sur les vidéos synthétiques et n'entraîne donc pas de biais dans l'estimation. Nous faisons néanmoins l'hypothèse que les expérimentations ne contiennent pas l'information utile uniquement sur ce recouvrement, c'est à dire qu'elle doit être visible dans une autre zone de la vidéo.

5.4.2.3 Recadrage semi-automatique

Dans notre cas, l'information utile est limitée au matériau déformé. Selon le cadrage utilisé pour filmer la déformation, la place dans l'image que celui-ci occupe peut varier. Il est possible qu'il ne soit visible que dans une petite zone de l'image. Lorsque c'est le cas, il est inutile de faire tous les calculs sur les autres zones. Nous procédons donc dans un premier temps à un cadrage automatique. L'algorithme 4 décrit le processus utilisé pour le recadrage automatique sur une séquence vidéo de résolution *largeur* \times *hauteur*.

5.4.2.4 Extraction de la position de l'outil

Le schéma global d'estimation du coefficient de Poisson (cf Fig. 5.6) est légèrement modifié lorsque l'une des dernières versions du dispositif de mesure est utilisée. En effet, celles-ci disposant de marqueurs ARToolKit, le positionnement virtuel de la touche sphérique est

Algorithm 4 Racadrage automatique

```

 $x_{min} = largeur - 1, x_{max} = 0, y_{min} = hauteur - 1, y_{max} = 0$ 
for Chaque image  $k$  de la vidéo do
  for Chaque pixel  $i, j$  de l'image  $k$  do
    Calculer  $Pixel = \Phi_{i,j}^{V_{Detour}^k} \times (1 - \Phi_{i,j}^{V_{MasqueOutil}^k})$ 
    if  $Pixel \neq 0$  then
       $x_{min} = MIN(i, x_{min})$ 
       $x_{max} = MAX(i, x_{max})$ 
       $y_{min} = MIN(j, y_{min})$ 
       $y_{max} = MAX(j, y_{max})$ 
    end if
  end for
end for
 $(x_{min}, x_{max}, y_{min}, y_{max})$  est le nouveau cadrage de la vidéo

```

pré-calculé à partir de la séquence vidéo initiale (cf équations 4.1, 4.2 et Fig. 4.24). Cette modification est illustrée sur le nouveau schéma d'estimation global du coefficient de Poisson de la figure 5.10 par l'ajout d'une étape préliminaire.

Ce mode de positionnement semble donner de bons résultats. Cependant, une validation future plus rigoureuse de l'erreur induite sera effectuée. Pour cela, il sera nécessaire d'avoir une base de référence, par exemple en effectuant l'acquisition vidéo avec les dernières versions de l'outil (comprenant les marqueurs *ARToolKitPlus*) monté sur un *Phantom*. Il suffira alors de comparer les résultats des deux techniques de positionnement virtuel pour mesurer le biais induit par cette méthode visuelle.

5.4.3 Métriques d'erreur

Les métriques d'erreurs Δ sont les fonctions qui permettent de quantifier les différences entre deux séquences vidéos. Elles doivent donc prendre comme entrée deux vidéos V_A et V_B et calculer une valeur ε (cf equation 5.14).

$$\varepsilon = \Delta(V_A, V_B) \quad (5.14)$$

Dans le cas où V_A et V_B sont identiques, les métriques doivent retourner $\varepsilon = 0$ et $\varepsilon \neq 0$ dans le cas contraire. D'autre part, elle doivent produire une notion de *quantification des différences*. Ainsi, si une vidéo V_C montre plus de différences avec V_A qu'avec V_B , alors $\Delta(V_A, V_C) > \Delta(V_A, V_B)$.

Les métriques d'erreur que nous proposons dans ces travaux sont pour la plupart basées sur un même schéma. Chaque comparaison est rapportée au niveau des pixels de chaque image. Ainsi, si une fonction Φ^{V^k} retourne la k ^{ième} image d'une vidéo V et qu'une fonction $\Phi_{i,j}^{V^k}$ renvoie les composantes couleur du pixel de coordonnées i, j de cette même image, l'équation 5.14 peut être réécrite au niveau de chaque image (cf equation 5.15) ou de chaque pixel (cf equation 5.16) :

$$\varepsilon = \frac{1}{k} \sum_{k=1}^s \delta(\Phi^{V_A^k}, \Phi^{V_B^k}) \quad (5.15)$$

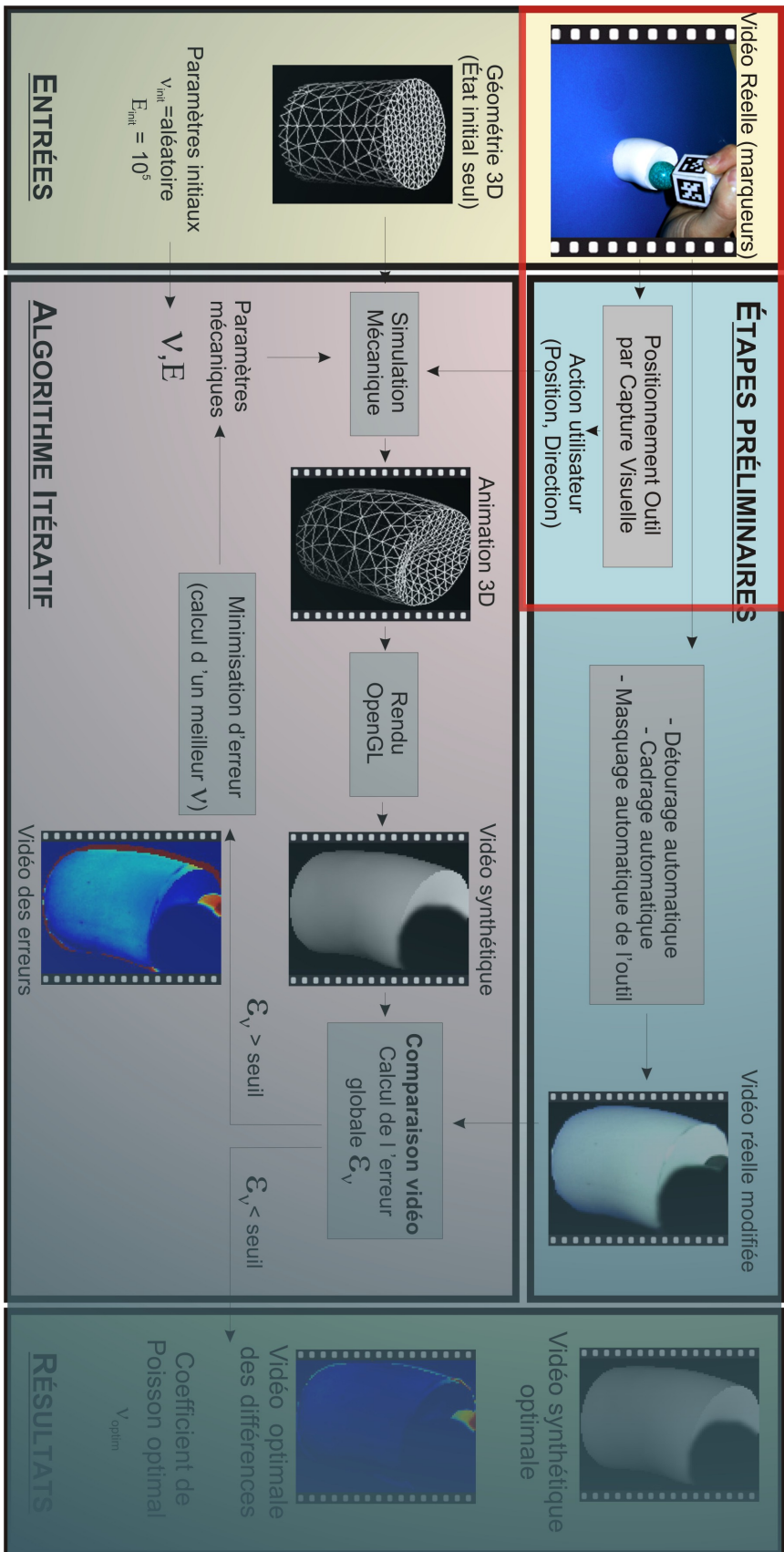


Figure 5.10 - Modification du schéma global de notre algorithme d'estimation du coefficient de Poisson à partir de séquences d'images avec les dernières versions du dispositif de mesure (positionnement virtuel par capture visuelle).

$$\varepsilon = \frac{1}{k} \sum_{k=1}^s \sum_{j=y_{min}}^{y_{max}} \sum_{i=x_{min}}^{x_{max}} \delta \left(\Phi_{i,j}^{V_A^k}, \Phi_{i,j}^{V_B^k} \right) \quad (5.16)$$

où la fonction δ , quantifiant les différences entre deux images ou deux pixels, permet de définir la métrique d'erreur. Ce schéma global de calcul est illustré sur la figure 5.11. On notera que les métriques peuvent être définies pour chaque canal de couleur des pixels (r, g, b) à l'aide de la fonction $\Phi_{i,j}^{V^k}$ ⁵, soit au niveau de l'intensité du pixel ϕ définie comme suit :

$$\phi_{i,j}^{V^k} = \frac{1}{3} \sum_{c=(r,g,b)} \Phi_{i,j}^{V^k}(c) \quad (5.17)$$

En outre, chaque illustration de métrique est accompagnée d'une « courbe de réponse ». Celle-ci montre le résultat obtenu en faisant varier le coefficient de Poisson ν pour l'animation illustrée sur la figure 5.11. Le détail de l'algorithme utilisé pour obtenir ces courbes est donné en section 5.4.4.

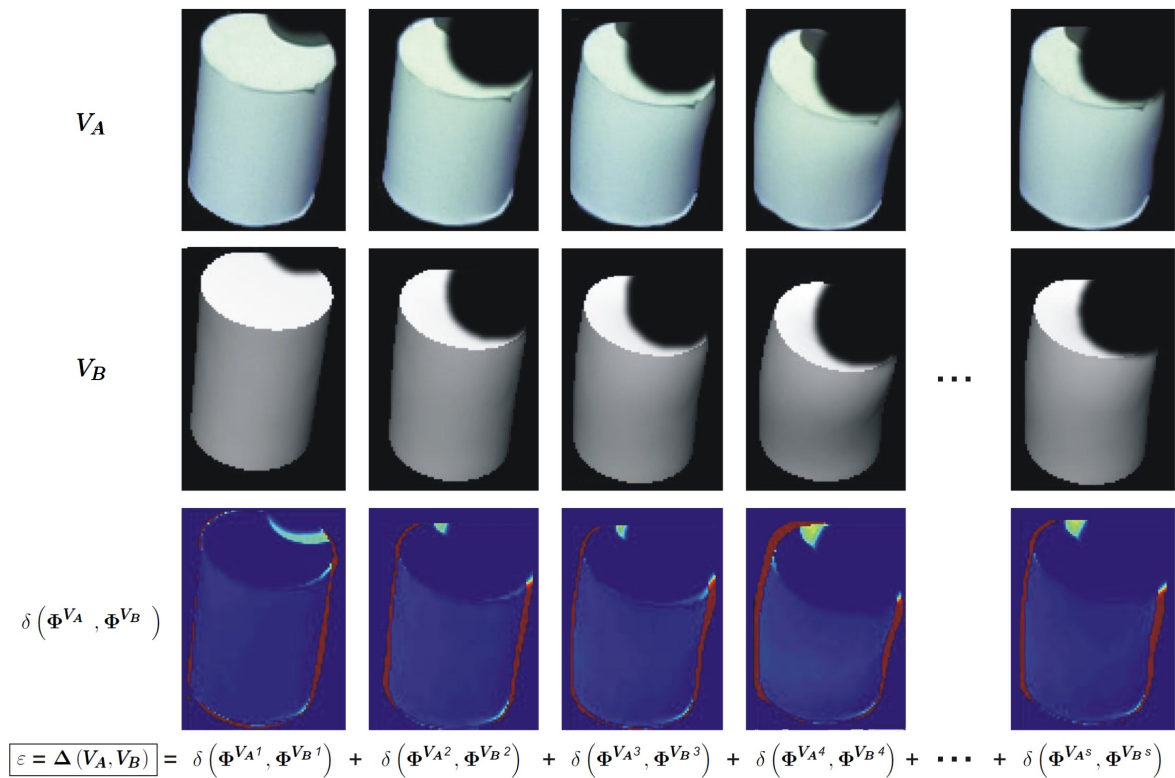


Figure 5.11 - Schéma global des métriques d'erreur avec $V_A = V_{Detour} \times (1 - V_{MasqueOutil})$ et $V_B = V_{Synthétique} \times (1 - V_{MasqueOutil})$.

5. Dans ce cas, toutes les opérations élémentaires telles que l'addition ou la soustraction sont effectuées sur chaque composante du pixel couleur.

5.4.3.1 Silhouette

L'un des principaux effets du coefficient de Poisson sur les objets déformés est la modification significative de leurs silhouettes lors d'une compression ou d'une traction. Cela s'explique très simplement par la définition même de ce paramètre (cf 2.4.2). Une métrique permettant de calculer la différence entre deux silhouettes semble donc très adaptée à cette estimation.

La différence de silhouette, définie par l'équation 5.18, utilise l'opération booléenne \oplus ou *exclusif*. Elle se base sur l'équation 5.5 qui permet de quantifier la différence entre deux pixels. Un seuil $seuilSil$ est utilisé. Le résultat, illustré sur la figure 5.12 est une carte binaire des endroits où les silhouettes des objets diffèrent.

$$\varepsilon_{sil} = \delta_{sil} \left(\Phi_{i,j}^{V_A^k}, \Phi_{i,j}^{V_B^k} \right) = \left(\phi_{i,j}^{V_A^k} > seuilSil \right) \oplus \left(\phi_{i,j}^{V_B^k} > seuilSil \right) \quad (5.18)$$

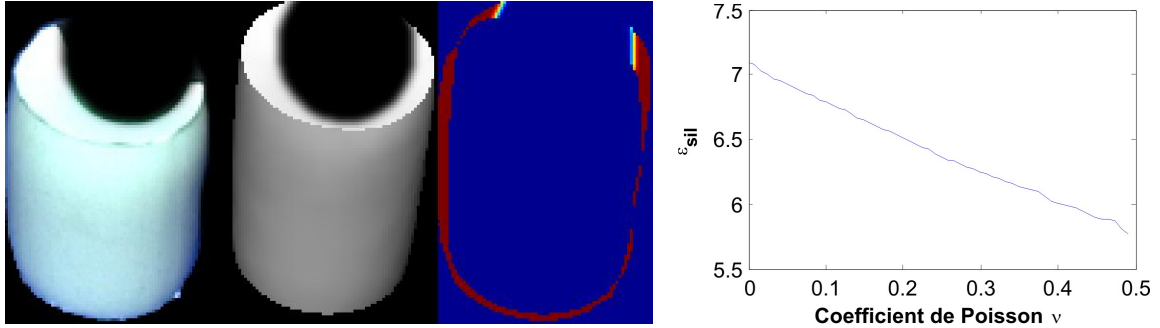


Figure 5.12 - Métrique d'erreur : Silhouette. À gauche, $V_A = V_{Detour} \times (1 - V_{MasqueOutil})$. Au centre, $V_B = V_{Synthétique} \times (1 - V_{MasqueOutil})$. À droite, le résultat de l'équation 5.18 (échelle thermique).

5.4.3.2 Différence absolue

Bien évidemment, il n'est possible d'extraire des informations de la silhouette d'un objet que lorsque celle-ci est visible. Dans le cas où elle n'apparaît pas sur la séquence vidéo, par exemple si l'objet est de taille importante ou que le cadrage est très serré, il est nécessaire de déterminer d'autres métriques. Par exemple, la formation de « bourrelets » qu'occasionne un fort coefficient de poisson sur la silhouette peut également modifier la surface de l'objet sur l'image. Lorsque l'objet est texturé, cette déformation est d'autant plus évidente.

Nous définissons deux métriques simples de comparaison de textures. La première, basique, est la différence absolue (cf équation 5.19 et Fig. 5.13).

$$\varepsilon_{abs} = \delta_{abs} \left(\Phi_{i,j}^{V_A^k}, \Phi_{i,j}^{V_B^k} \right) = \left| \phi_{i,j}^{V_A^k} - \phi_{i,j}^{V_B^k} \right| \quad (5.19)$$

5.4.3.3 Différence au carré

La seconde est destinée à améliorer la convergence des algorithmes de minimisation d'erreur. La méthode des moindres carrés [Johann Carl Friedrich Gauß] est couramment utilisée pour des problèmes de minimisation d'erreur. L'équation 5.20 définit une métrique de différence au carré, illustrée sur la figure 5.14.

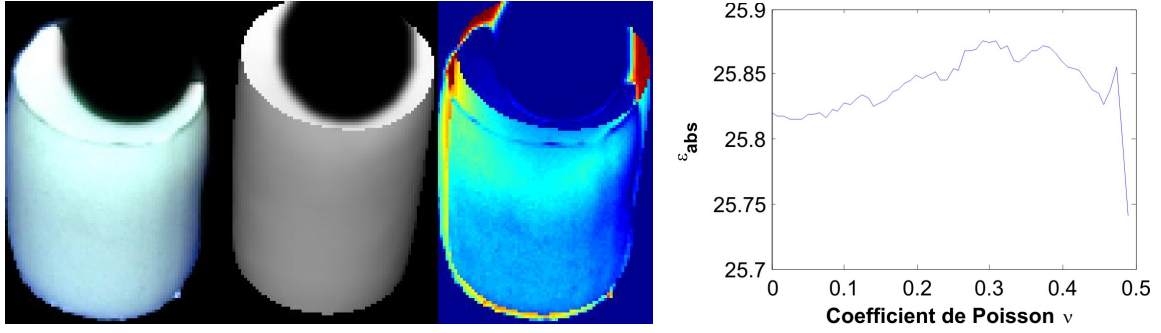


Figure 5.13 - Métrique d'erreur : Différence absolue. À gauche, $\mathbf{V}_A = \mathbf{V}_{\text{Detour}} \times (1 - \mathbf{V}_{\text{MasqueOutil}})$. Au centre, $\mathbf{V}_B = \mathbf{V}_{\text{Synthétique}} \times (1 - \mathbf{V}_{\text{MasqueOutil}})$. À droite, le résultat de l'équation 5.19 (échelle thermique).

$$\varepsilon_{sq} = \delta_{sq} \left(\Phi_{i,j}^{V_A^k}, \Phi_{i,j}^{V_B^k} \right) = \left(\phi_{i,j}^{V_A^k} - \phi_{i,j}^{V_B^k} \right)^2 \quad (5.20)$$

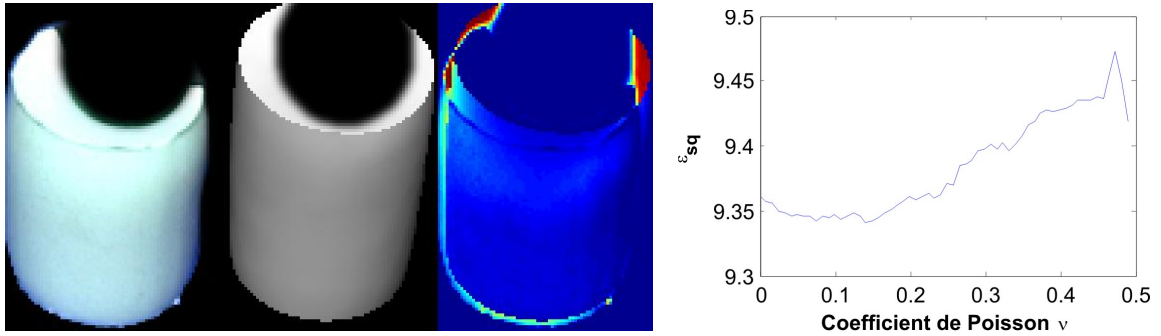


Figure 5.14 - Métrique d'erreur : Différence au carré. À gauche, $\mathbf{V}_A = \mathbf{V}_{\text{Detour}} \times (1 - \mathbf{V}_{\text{MasqueOutil}})$. Au centre, $\mathbf{V}_B = \mathbf{V}_{\text{Synthétique}} \times (1 - \mathbf{V}_{\text{MasqueOutil}})$. À droite, le résultat de l'équation 5.20 (échelle thermique).

5.4.3.4 Flou

Pour atténuer l'effet d'éventuelles erreurs de recalage dues à un mauvais positionnement de la caméra, nous définissons une métrique utilisant un flou gaussien (cf equation 5.21). Celle-ci se base sur la fonction de flou gaussien décrite par l'équation 5.12. La figure 5.15 illustre cette nouvelle métrique d'erreur.

$$\varepsilon_{flou} = \delta_{flou} \left(\Phi_{i,j}^{V_A^k}, \Phi_{i,j}^{V_B^k} \right) = \left(\text{Flou} \left(\phi_{i,j}^{V_A^k}, tf \right) - \text{Flou} \left(\phi_{i,j}^{V_B^k}, tf \right) \right)^2 \quad (5.21)$$

5.4.3.5 Corrélation inverse

Une autre métrique se base sur le coefficient de corrélation entre deux matrices de même taille, défini par l'équation 5.22 pour des matrices de taille $M \times N$. Ce coefficient donne

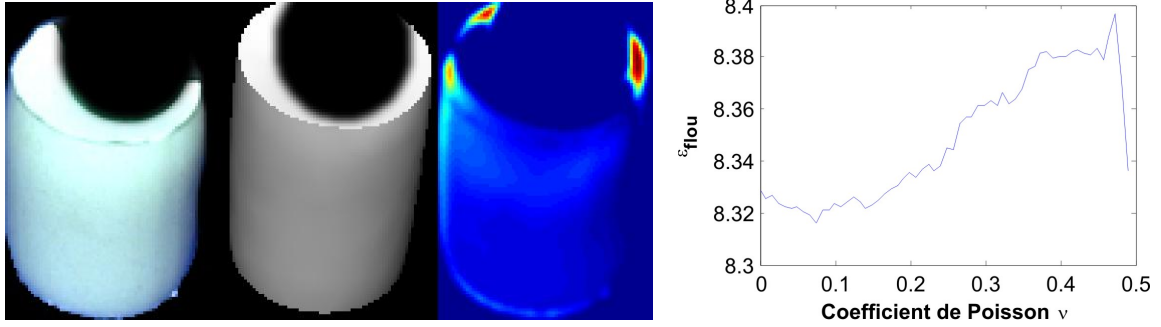


Figure 5.15 - Métrique d'erreur : Flou. À gauche, $V_A = V_{Detour} \times (1 - V_{MasqueOutil})$. Au centre, $V_B = V_{Synthétique} \times (1 - V_{MasqueOutil})$. À droite, le résultat de l'équation 5.21 (échelle thermique).

une indication de correspondance entre les deux matrices. Il vaut 1 quand les matrices sont identiques et 0 quand elles sont totalement différentes.

$$\text{Corr}([A], [B]) = \frac{\sum_{m=0}^M \sum_{n=0}^N ([A]_{mn} - \overline{[A]}) ([B]_{mn} - \overline{[B]})}{\sqrt{\left(\sum_{m=0}^M \sum_{n=0}^N ([A]_{mn} - \overline{[A]})^2 \right) \left(\sum_{m=0}^M \sum_{n=0}^N ([B]_{mn} - \overline{[B]})^2 \right)}} \quad (5.22)$$

Nous utilisons cette équation pour créer une métrique plus robuste aux différences de luminosité sur les images à comparer (cf Fig. 5.16), définie au niveau image par :

$$\varepsilon_{corr} = \delta_{corr} (\Phi^{V_A^k}, \Phi^{V_B^k}) = 1 - \text{Corr} (\Phi^{V_A^k}, \Phi^{V_B^k}) \quad (5.23)$$

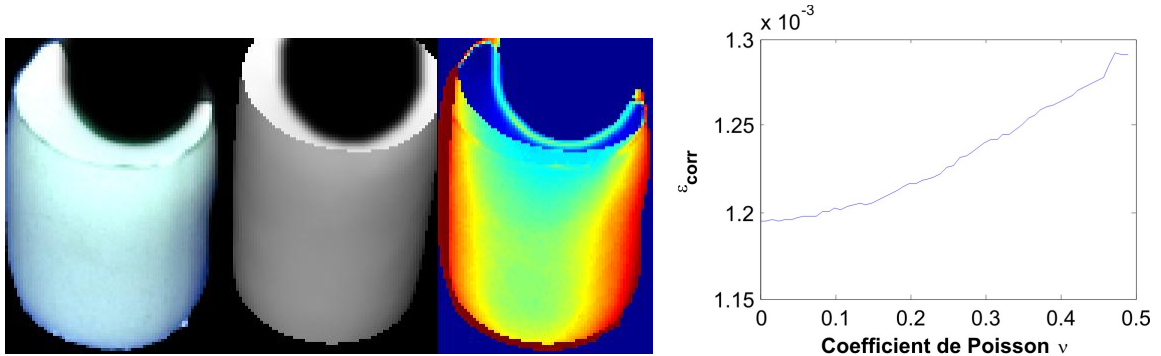


Figure 5.16 - Métrique d'erreur : Corrélation inverse. À gauche, $V_A = V_{Detour} \times (1 - V_{MasqueOutil})$. Au centre, $V_B = V_{Synthétique} \times (1 - V_{MasqueOutil})$. À droite, le résultat de l'équation 5.23 (échelle thermique).

5.4.3.6 Métriques composées

Toutes ces métriques peuvent être combinées entre elles pour définir des métriques plus robustes et tirant partie des avantages de chaque métrique de base (cf Fig. 5.17 et Fig. 5.18). Voici quelques combinaisons possibles :

$$\delta_{\text{absSil}}(\Phi_{i,j}^{V_A^k}, \Phi_{i,j}^{V_B^k}) = \text{MAX}(\delta_{\text{abs}}(\Phi_{i,j}^{V_A^k}, \Phi_{i,j}^{V_B^k}) + \delta_{\text{sil}}(\Phi_{i,j}^{V_A^k}, \Phi_{i,j}^{V_B^k}), 1) \quad (5.24)$$

$$\delta_{\text{sqSil}}(\Phi_{i,j}^{V_A^k}, \Phi_{i,j}^{V_B^k}) = \text{MAX}(\delta_{\text{sq}}(\Phi_{i,j}^{V_A^k}, \Phi_{i,j}^{V_B^k}) + \delta_{\text{sil}}(\Phi_{i,j}^{V_A^k}, \Phi_{i,j}^{V_B^k}), 1) \quad (5.25)$$

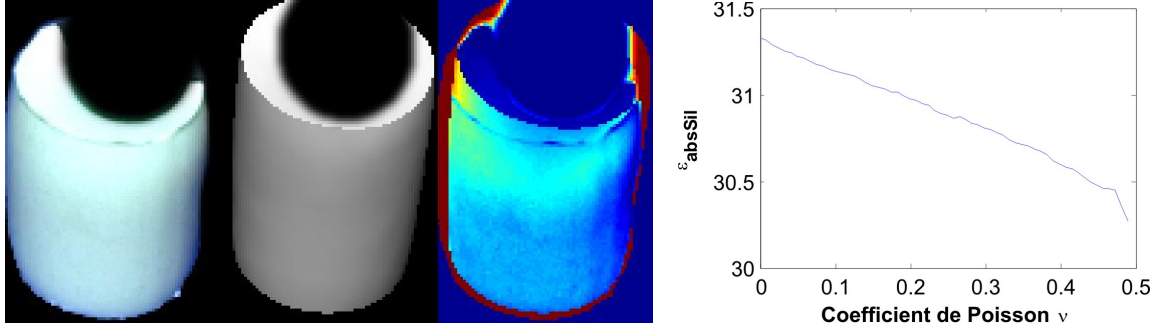


Figure 5.17 - Métrique d'erreur : Différence absolue et silhouette. À gauche, $V_A = V_{\text{Detour}} \times (1 - V_{\text{MasqueOutil}})$. Au centre, $V_B = V_{\text{Synthétique}} \times (1 - V_{\text{MasqueOutil}})$. À droite, le résultat de l'équation 5.25 (échelle thermique).

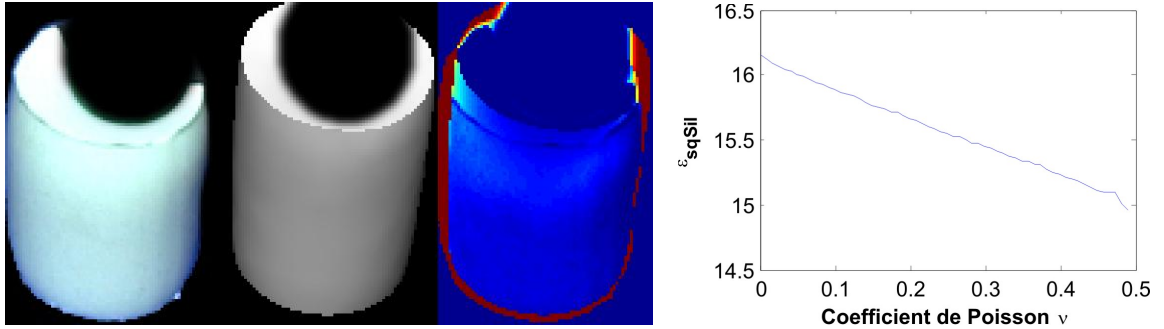


Figure 5.18 - Métrique d'erreur : Différence au carré et silhouette. À gauche, $V_A = V_{\text{Detour}} \times (1 - V_{\text{MasqueOutil}})$. Au centre, $V_B = V_{\text{Synthétique}} \times (1 - V_{\text{MasqueOutil}})$. À droite, le résultat de l'équation 5.25 (échelle thermique).

5.4.3.7 Angular Map

Certains effets de lumière, comme les « éclats spéculaires » peuvent être gênants pour les comparaisons. En effet, bien que nous pensons pouvoir créer une métrique future tirant partie de ce phénomène (cf 7.2), il est très difficile, à la main, de retrouver les paramètres exacts qui permettent de les recréer à l'identique, ce qui pourrait introduire un biais dans l'estimation. Nous décidons donc dans un premier temps d'éliminer ces effets optiques.

Pour cela, nous définissons une métrique issue des travaux de Bhat (*et al.*) [Bhat 03]. Les auteurs l'utilisent initialement pour extraire les informations de plis et de creux sur une séquence vidéo mettant en scène un morceau de tissu en mouvement sur lequel est projeté un motif structuré. Elle tire partie des *steerable filters* [Freeman 91]. Ces filtres ont deux propriétés : ils sont séparables en x et y (c'est à dire que la convolution peut s'effectuer sur

une dimension puis sur l'autre ensuite) et sont rangés par séries. Chaque série propose le filtre sous un certain nombre d'angles. Il est alors possible de synthétiser uniquement par combinaison linéaire des ces filtres de base le même filtre sous un angle quelconque. La même combinaison linéaire peut être appliquée directement sur le résultat des convolutions des filtres de base sur une image.

Comme Bhat (*et al.*) [Bhat 03], nous utilisons une combinaison G2/H2 (gaussienne d'ordre 2 et Hessian d'ordre 2) afin de déterminer, pour chaque pixel $\Phi_{i,j}^{VP^k}$, l'orientation dominante **SteerAngle** ($\Phi_{i,j}^{VP^k}$) et la force du signal d'entrée **SteerForce** ($\Phi_{i,j}^{VP^k}$). Ces deux fonctions sont créés à partir des trois premiers coefficients de la série de Fourier correspondante⁶.

Soit la fonction **AngularMap** prenant en paramètre un pixel ϕ :

$$\mathbf{AngularMap}(\phi) = \begin{cases} \mathbf{SteerAngle}(\phi) & \text{si } \mathbf{SteerForce}(\phi) > \mathit{seuilForce}, \\ 0 & \text{sinon} \end{cases} \quad (5.26)$$

où *seuilForce* est un seuil prédéfini par l'utilisateur, la métrique *Angular Map*, illustrée sur la figure 5.19, est alors définie par :

$$\begin{aligned} \varepsilon_{amap} &= \left(\mathbf{Flou} \left(\mathbf{AngularMap} \left(\Phi_{i,j}^{VA^k} \right), 5 \right) - \mathbf{Flou} \left(\mathbf{AngularMap} \left(\Phi_{i,j}^{VB^k} \right), 5 \right) \right)^2 \\ \varepsilon_{angle} &= \delta_{angle} \left(\Phi_{i,j}^{VA^k}, \Phi_{i,j}^{VB^k} \right) = \mathbf{MAX} \left(\varepsilon_{amap} + \delta_{sil} \left(\Phi_{i,j}^{VA^k}, \Phi_{i,j}^{VB^k} \right), 1 \right) \end{aligned} \quad (5.27)$$

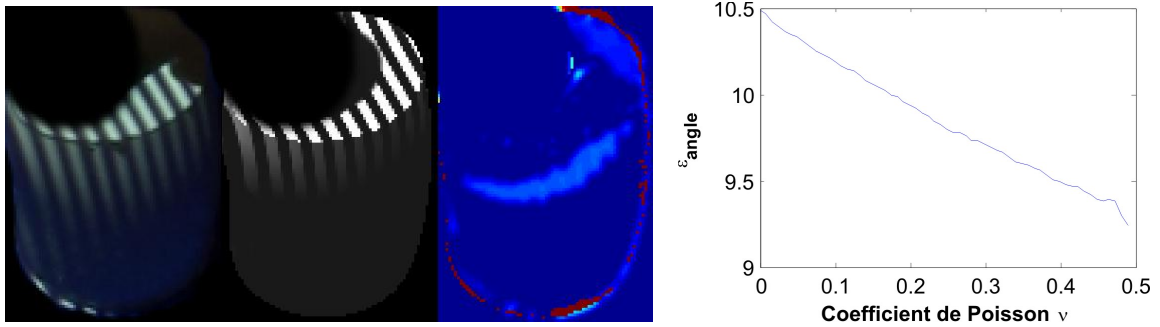


Figure 5.19 - Métrique d'erreur : Angular Map. À gauche, $\mathbf{V}_A = \mathbf{V}_{Detour} \times (1 - \mathbf{V}_{MasqueOutil})$. Au centre, $\mathbf{V}_B = \mathbf{V}_{Synthétique} \times (1 - \mathbf{V}_{MasqueOutil})$. À droite, le résultat de l'équation 5.27 (échelle thermique).

5.4.3.8 Classes de métriques

Les métriques peuvent être séparées en deux groupes. Nous qualifions le premier de groupe des « métriques de silhouettes ». Les métriques de ce groupe mettent l'accent sur la différence entre les silhouettes de l'objet pour chaque image des animations (« Silhouette », « Métriques

6. Pour plus de détails sur l'utilisation des *steerable filters* pour créer ces deux fonctions, le lecteur est invité à consulter l'appendice I des travaux de Freeman et Adelson [Freeman 91] ainsi que la thèse de Freeman [Freeman 92]

composées »). L'autre est composé des « métriques de surface » (« Différence absolue », « Différence au carré », « Flou », « Corrélation inverse »). Bien que ces métriques incluent implicitement une différence entre les silhouettes, le poids de celle-ci est moins important que celui attribué aux différences « visibles » sur la surface de l'objet. La dernière métrique, « Angular-Map », peut elle aussi être classée dans ce groupe mais nécessite un élément supplémentaire : la projection d'un motif.

Nous utilisons ces dénominations lors de la présentation de nos résultats d'estimation du coefficient de Poisson (cf 6.2).

5.4.4 Optimisation

La section 5.4.1 décrit, en s'appuyant sur le schéma de la figure 5.6, la procédure pour déterminer le coefficient de Poisson ν_{optim} à partir d'une valeur initiale ν_{init} et d'une fonction de variation **Var**. Plusieurs méthodes, dont les principes sont donnés dans la section 3.2, permettent d'y parvenir plus ou moins rapidement. Nous décrivons ici une méthode brute dite « exhaustive » qui permet d'avoir une idée générale du comportement des métriques et une méthode plus intelligente qui permet d'obtenir le résultat plus rapidement.

5.4.4.1 Méthode « exhaustive »

La méthode « exhaustive » est un algorithme basique. Le but est de parcourir intégralement l'espace de travail. Quand celui-ci est continu, un paramètre de précision doit être donné. Ainsi, dans le cas de la recherche de coefficient de Poisson optimal ν_{optim} , on utilisera $\nu_{next} = \mathbf{Var}(\nu) = \nu + precision$ et ν_{init} , initialisé à la borne inférieure de l'intervalle. Dans ces conditions, pour un coefficient de Poisson pouvant varier dans l'intervalle $[0 \dots 0,5[$, il s'agira d'évaluer $\frac{0,5}{precision} - 1$ fois la fonction $\varepsilon = \mathbf{Fun}(\nu)$ et de retenir la valeur ν_{optim} lorsque ε est minimale. Il est ainsi possible d'affirmer que ν_{optim} est la valeur optimale du coefficient de Poisson avec une précision de $precision$.

La figure 5.20 illustre le calcul de la courbe obtenue par cette méthode. Bien, entendu, plus la précision demandée est importante, plus le temps de calcul est long. Cette technique est donc très utile pour étudier le comportement des métriques par exemple, mais ne peut pas être utilisable pour l'application finale car les temps de calcul pour l'évaluation de la fonction $\varepsilon = \mathbf{Fun}(\nu)$ peuvent vite être très importants (simulation mécanique + traitement d'image + rendu + comparaison vidéo à l'aide d'une métrique). C'est pourquoi un algorithme plus « intelligent » doit être adopté. Cependant, à notre connaissance, c'est la seule méthode qui permette d'affirmer que la valeur trouvée est le minimum global d'une fonction lorsque aucune information n'est à priori connue sur celle-ci.

5.4.4.2 Choix de l'algorithme d'optimisation

La figure 5.21 montre le cheminement qui a conduit notre choix de la méthode d'optimisation sur l'organigramme de classification générale des méthodes d'optimisation mono-objectif. Pour commencer, nous sommes face à un problème d'optimisation continue bornée. En effet, quel que soit le coefficient de Poisson appartenant à l'intervalle de validité, il est possible de calculer une animation et donc de comparer la séquence d'images synthétiques résultante à la vidéo initiale. Ensuite, la fonction objective n'est pas connue analytiquement. Le calcul

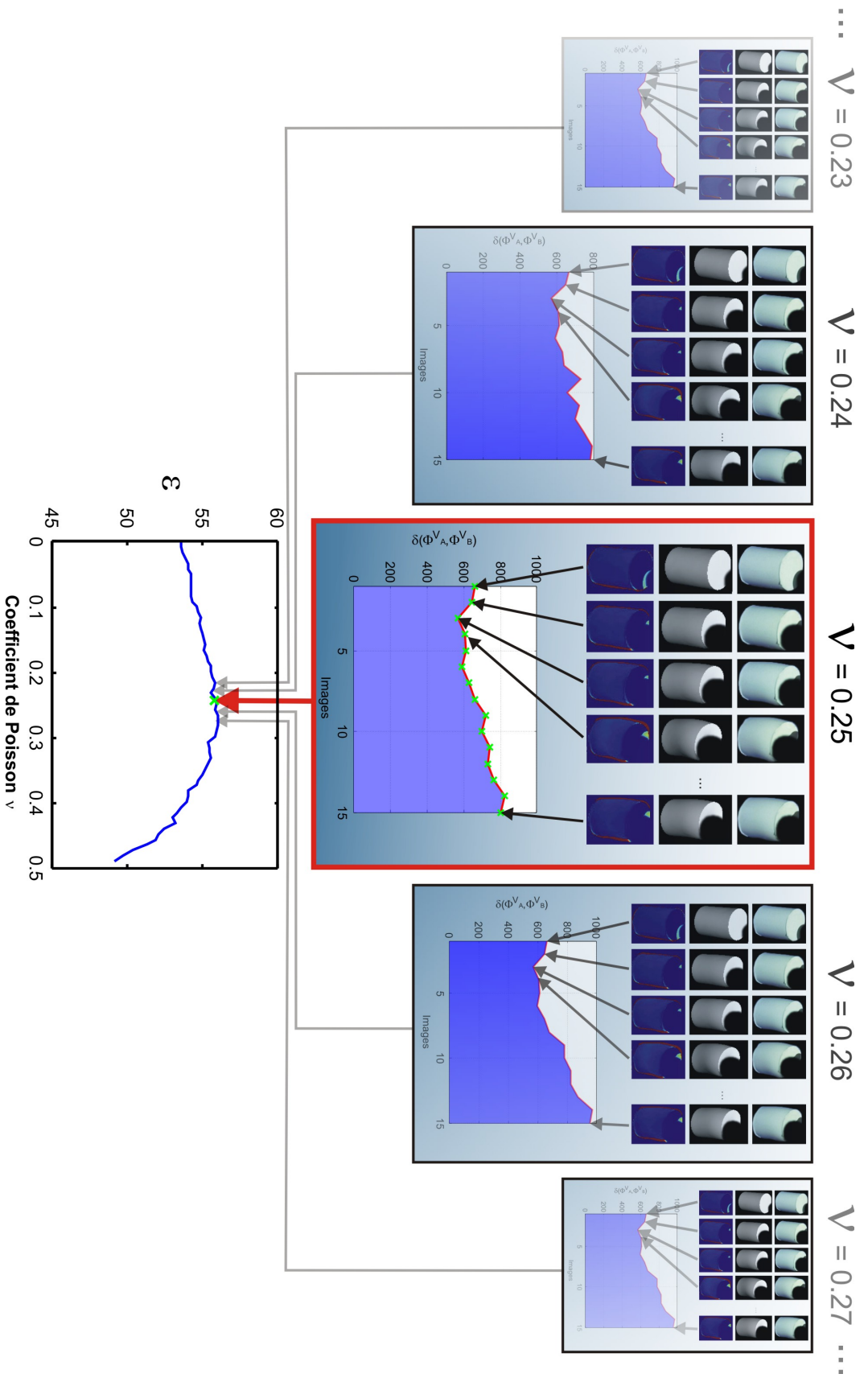


Figure 5.20 - Illustration du calcul d'une courbe de « réponse » complète d'une métrique d'erreur pour une expérimentation donnée à l'aide de la méthode « exhaustive ».

des courbes de réponse illustrant le comportement des métrique en 5.4.3 montre en outre que le problème n'est pas linéaire (cf Fig. 5.12 à 5.19). Ces mêmes images montrent également certains minimums locaux. Puisque ces courbes sont extraites à partir d'un objet relativement simple (un cylindre de silicone), on peut raisonnablement penser qu'il en sera de même pour des objets plus complexes géométriquement. Il est donc nécessaire de choisir une méthode d'optimisation globale.

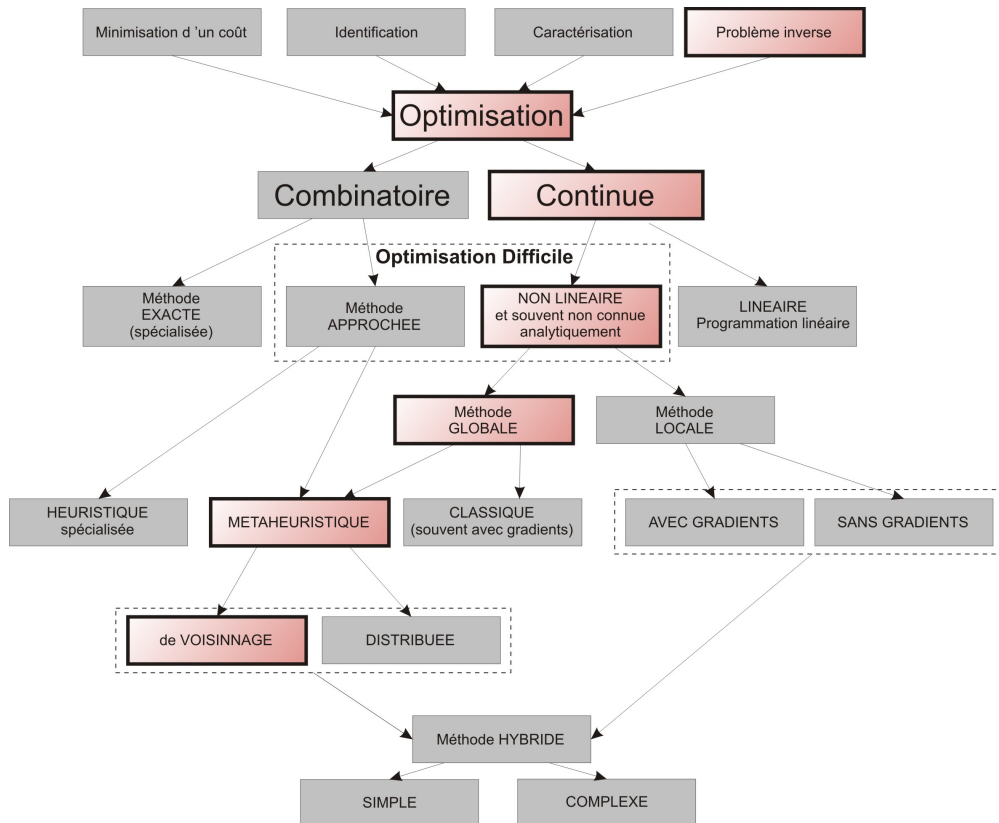


Figure 5.21 - Choix de la méthode d'optimisation pour l'estimation du coefficient de Poisson à partir de l'organigramme de classification générale des méthodes d'optimisation mono-objectif.

Or, puisque les fonctions d'erreur n'ont pas de formulation analytique, il est impossible d'en calculer les gradients efficacement. Il est alors nécessaire de choisir une métaheuristique. Le tableau de classification montre que les métaheuristicques sont divisées en deux parties : les métaheuristicques dites « de voisinage » et les métaheuristicques « distribuées ». Ces dernières, souvent basées sur des populations, nécessitent un grand nombre d'évaluations de la fonction objectif à chaque itération. Or, dans notre cas, chaque évaluation de cette fonction est très coûteuse (simulation mécanique, rendu graphique, comparaison vidéo). Nous optons donc pour une métaheuristique de voisinage : le *recuit simulé* [Kirkpatrick 83]. Le fonctionnement de celle-ci est détaillé en 3.2.1.2.

La figure 5.22 illustre le déroulement de cet algorithme dans le cas de l'estimation décrite sur la figure 5.20. Dans cet exemple, le coefficient de Poisson initial est $\nu_{init} = 0.3$. L'algorithme évalue donc la fonction objective (simulation mécanique, rendu graphique, comparaison vidéo) pour cette valeur puis « essaie » d'autres valeurs dans le voisinage de celle-ci. Il choisit ensuite la meilleure et recommence depuis ce point. La particularité du recuit simulé

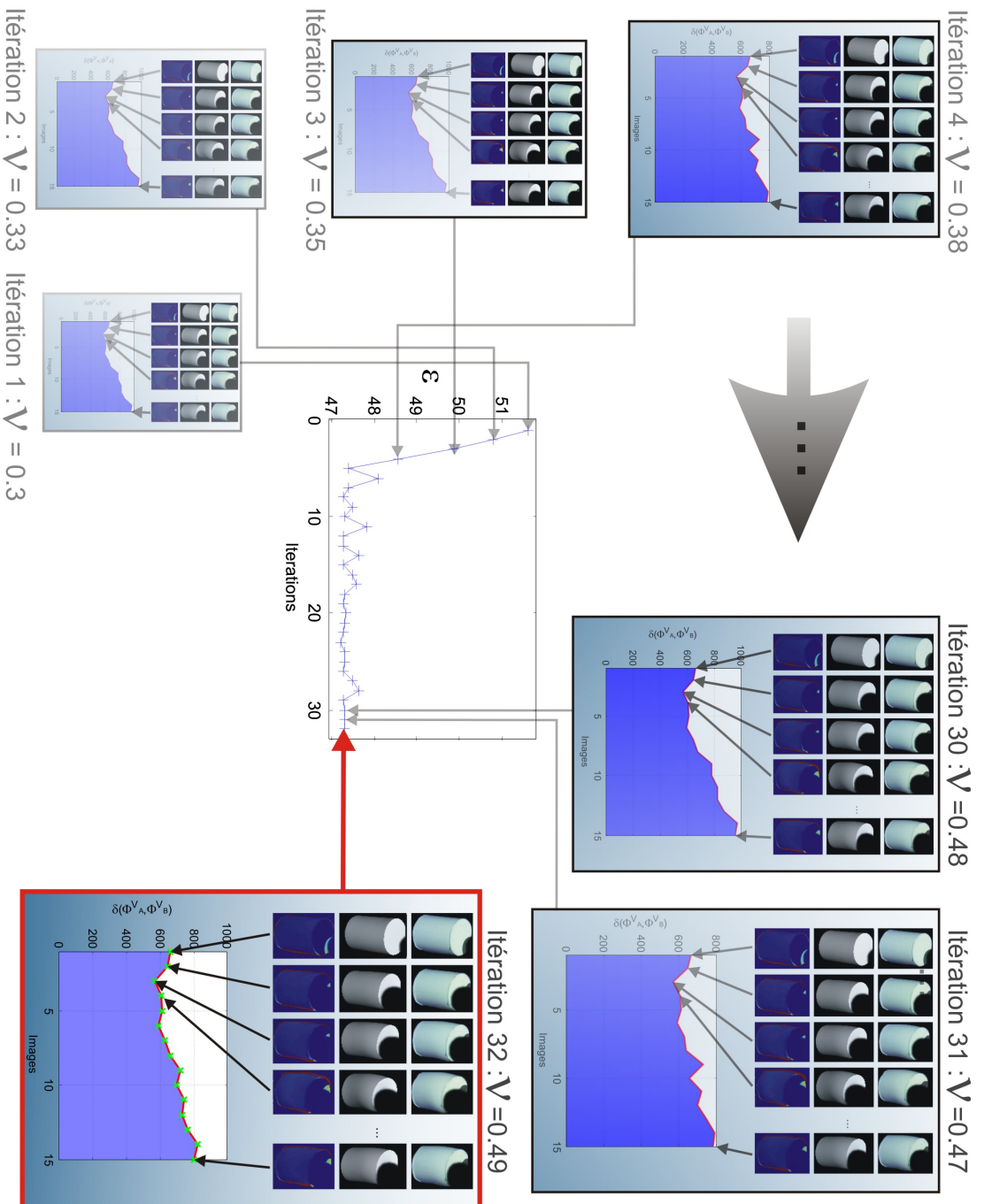


Figure 5.22 - Illustration du déroulement de l'algorithme lors d'une estimation optimisée par recuit simulé. Le cas est identique à celui traité sur la figure 5.20.

est qu'il « accepte » parfois une valeur moins « bonne » que la valeur courante selon un critère qui décroît avec le temps. Cela évite de rester « piégé » dans les minima locaux.

Cependant, l'un des problèmes de cet algorithme est la difficulté de paramétrage. Nous avons donc effectué une petite étude statistique sur différents exemples pour effectuer les réglages (température initiale, taux du décroissement de la température par itération, critère d'arrêt). Pour renforcer les chances de convergence, nous faisons pour chaque estimation une évaluation grossière de la fonction objective au préalable (méthode « exhaustive » avec une précision de 0.1, soit cinq tests initiaux pour l'intervalle de validité du coefficient de Poisson) et choisissons la valeur donnant le meilleur résultat pour l'initialisation pour ν_{init} . Sur cet exemple, on peut voir que seulement 32 évaluations (initialisation incluse) de la fonction sont nécessaires pour retrouver la valeur finale, contre 100 avec la méthode exhaustive pour la même précision.

5.5 Estimation du module de Young

Nous avons développé deux méthodes pour estimer le module de Young à partir de la capture de l'action de l'utilisateur. La première tente de retrouver directement ce paramètre en se basant sur l'équation 2.24. Cette méthode faisant des approximations relativement grossières, nous avons également conçu une méthode d'estimation inverse, utilisant la simulation mécanique.

5.5.1 Méthode directe

La représentation mathématique la plus simple d'un corps élastique est celle du ressort. Le module de Young représentant la raideur de cette élasticité, nous pouvons nous baser sur l'équation 2.24 pour tenter d'approximer ce paramètre. Pour plus de clarté, nous la réécrivons ici :

$$\frac{f}{s} = E \cdot \frac{l - l_0}{l_0}$$

où, pour rappel, f est la force appliquée ou mesurée, s est l'aire de la surface de la section, E est le module de Young, l_0 est la longueur de l'objet élastique dans le sens de la déformation, et l est cette longueur observée après la déformation ($\frac{l-l_0}{l_0}$ est l'allongement relatif). Estimer le module de Young directement d'après cette formule semble trivial :

$$E = \frac{f}{s \cdot \frac{l-l_0}{l_0}}$$

Cependant, cette formule est destinée aux objets de forme « extrudée » (un cylindre est un objet créé par un disque extrudé sur une certaine longueur par exemple) et où la **force est appliquée uniformément sur toute la section et dans la direction de la longueur de l'objet**. C'est également à partir de cette formule que nous extrayons le module de Young de référence à partir de mesures mécaniques effectuées par la machine Zwick [Zwick] sur des objets cylindriques (cf 6.1.1). Or, dans notre cas, l'objet appliquant la force est une sphère, généralement de diamètre plus petit que l'objet déformé, à l'aide de laquelle l'effort est appliqué dans une direction quelconque et sur des objets à-priori de forme non-définie.

Toutefois, nous pouvons tenter d'approximer les diverses erreurs induites par la violation de ces hypothèses pour tenter d'estimer le module de Young directement à partir des mesures de forces \mathbf{F} , de positionnement \mathbf{P} et de direction \mathbf{D} de notre outil de capture à chaque temps de mesure t :

$$E \approx E_{directMoy} = \frac{1}{N} \sum_{t=1}^N N \frac{\mathbf{F}^t}{\alpha(\mathbf{P}^t, r, h) \cdot \eta(\mathbf{P}^t, \mathbf{D}^t, h)} \quad (5.28)$$

où N est le nombre total de pas de temps, $E_{directMoy}$ est l'estimation du module de Young, α est une fonction d'estimation de la surface de contact entre un objet de hauteur h et une sphère de rayon r en position \mathbf{P}^t , et η est une fonction calculant une estimation de l'allongement relatif de l'objet dans la direction \mathbf{D}^t :

$$\alpha(\mathbf{P}^t, h, r) = 2\pi r \begin{pmatrix} 0 & \text{si } \mathbf{P}_y^t - h < 0 \\ r & \text{si } \mathbf{P}_y^t - h > r \\ \mathbf{P}_y^t - h & \text{sinon} \end{pmatrix} \quad (5.29)$$

$$\eta(\mathbf{P}^t, \mathbf{D}^t, h) = \frac{(\mathbf{P}^t - \mathbf{P}^{contact}) \cdot \mathbf{D}^t}{h} \quad (5.30)$$

où $\mathbf{P}^{contact}$ est la position de la sphère à l'instant du dernier contact et « \cdot » est le produit scalaire.

L'équation 5.29 réalise donc une approximation la surface de contact par la surface d'une sphère coupée avec un plan à la hauteur h . Il semble d'ores et déjà évident que cette estimation sera faussée pour les objets qui ne seront pas planaires à cette hauteur (comme une hémisphère par exemple) ainsi que par le fait qu'il semble exister également une corrélation entre cette surface au contact en compression et le coefficient de Poisson (un coefficient de Poisson élevé aura tendance à créer un effet de « recouvrement » qui augmentera la surface de contact)⁷. De plus, l'équation 5.30 effectue un rapport entre le déplacement projeté sur la direction et la hauteur totale. Pour plus de rigueur, il faut par exemple effectuer un lancer de rayon au moment du contact pour mesurer l'épaisseur de l'objet dans cette direction (cf 7.2).

Cependant, cette évaluation permet d'avoir une première représentation des échelles en jeu dans l'estimation de ce paramètre. En outre, nous effectuons une estimation un peu plus robuste. En effet, le calcul de la moyenne dans l'équation 5.28 est relativement sensible au bruit qui peut être induit par les mesures expérimentales. C'est pourquoi nous adoptons l'approche classique *stress/strain*. Il s'agit, pour chaque échantillon, de calculer la contrainte σ_{direct}^t (*stress*, cf équation 5.31) et la déformation ε_{direct}^t (*strain*, cf équation 5.32). Dans le cas idéal, les échantillons sont répartis sur une droite dont la valeur de la pente est le module de Young. En pratique cependant, plusieurs facteurs viennent altérer cette linéarité, comme par exemple le bruit induit par des mesures expérimentales. La méthode consiste donc à trouver l'équation de la droite qui minimise les distances entre les points et cette droite. Cette technique s'appelle le *fitting par moindres carrés*. Le graphique de gauche sur la figure 5.23 illustre ce procédé.

7. Une solution pourrait être alors de réaliser une simulation mécanique avec le coefficient de Poisson retrouvé précédemment pour mieux estimer cette surface de contact (cf 7.2)

$$\sigma_{direct}^t = \frac{\mathbf{F}^t}{\alpha(\mathbf{P}^t, r, h)} \quad (5.31)$$

$$\varepsilon_{direct}^t = \eta(\mathbf{P}^t, \mathbf{D}^t, h) \quad (5.32)$$

Cependant, il peut s'avérer que la non-linéarité réelle de l'élasticité du matériau (voir le chapitre 2) perturbe également ce *fitting*, et donc par conséquent l'estimation du module de Young. En effet, la méthode par moindres carrés considère tous les échantillons. Dans notre cas, nous désirons réaliser le *fitting* sur la partie linéaire uniquement. Pour cela, nous utilisons la méthode du *RanSaC* [Fischler 81]. Dans sa forme la plus courante, cet algorithme stochastique détermine le nombre N de tirages aléatoires de n points (2 dans notre cas pour une droite) nécessaires pour avoir 99% de chances d'obtenir au moins un tirage où tous les points « proches » du modèle (ici la droite) ne sont pas considérés comme du bruit. Le paramètre nécessaire est donc un seuil t au delà duquel le point est marqué comme étant du bruit. Ensuite, un *fitting par moindres carrés* est appliqué sur les « bons » points.

Le graphique de droite sur la figure 5.23 comporte une partie non-linéaire. La ligne rouge montre le résultat obtenu par un *fitting par moindres carrés* classique et la droite verte montre le résultat de ce même *fitting* après l'application d'un *RANSAC* pour éliminer les « mauvais » points (avec $t = 0.001$). On peut voir que la méthode classique ne fait pas la différence entre une partie linéaire et une partie non-linéaire, tous les points étant pris en compte. En revanche, avec l'application du *RanSaC*, seule la partie linéaire est utilisée pour le *fitting* du modèle. On utilise donc cette pente pour calculer E_{direct} .

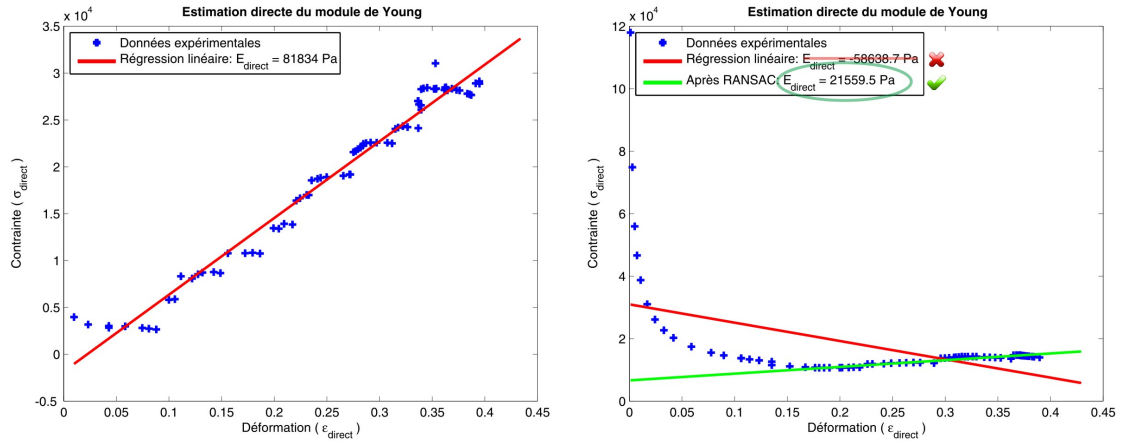


Figure 5.23 - Estimation directe du module de Young par fitting sur un nuage de points contrainte/déformation (en bleu). À gauche, illustration d'une régression linéaire (fitting par moindres carrés, en rouge). À droite, illustration du problème de la régression linéaire lorsqu'une partie des données expérimentales est non-linéaire et résolution par filtre *RanSaC* (en vert).

5.5.2 Méthode inverse

Nous avons vu que la méthode directe fait certaines approximations qui entraînent par conséquent une erreur dans l'estimation du module de Young. Elle a l'avantage d'être très

rapide (quasiment instantanée par régression linéaire et quelques secondes avec le filtre *Ran-SaC*). Cependant, il n'est possible de faire cette estimation directe que lorsque l'on connaît la signification physique du paramètre et sa formulation mathématique. De plus nous faisons plusieurs hypothèses quant à la géométrie de l'objet déformé. Il faut donc s'affranchir de ces hypothèses. Nous développons ici une méthode d'estimation inverse permettant d'évaluer le module de Young à partir des forces mesurées par notre dispositif de capture d'efforts. Une méthode inverse est également plus générique : elle permet de pouvoir estimer d'autres paramètres influant sur les forces mesurées. D'autre part, une telle méthode est la seule technique d'estimation possible lorsque l'algorithme de simulation mécanique est inconnu et est utilisé comme une *boîte noire*, dont on ne connaît que les paramètres d'entrée et le résultat. Cela peut également parfois aider à mieux comprendre le fonctionnement d'un algorithme inconnu en *reverse engineering*.

Notre méthode inverse se base sur le même schéma que celui utilisé pour l'estimation de coefficient de Poisson (cf Fig. 5.24). À partir de la géométrie de l'objet déformable, des positions de l'outil capturées et de paramètres initiaux, l'algorithme de simulation mécanique génère une animation synthétique en trois dimensions ainsi que les forces internes à l'objet (cf 5.1). On extrait de ces forces les efforts aux points de contacts que l'on projette sur l'axe de l'outil, donnant, pour chaque pas de temps, une force de contact (cf Fig. 5.1, 10). Ces forces de contact sont comparées aux forces mesurées par notre outil pendant l'enregistrement de l'expérimentation (cf 5.5.2.2). Si ces forces ne sont pas assez proches (la notion de proximité est simulée par un seuil prédéfini par l'utilisateur), une fonction de minimisation d'erreur est ensuite utilisée pour tenter de déterminer une meilleure valeur du module de Young (cf 5.5.2.3) et l'algorithme recrée une nouvelle animation synthétique à partir de ce nouveau paramètre. Ce processus itératif est répété jusqu'à ce que la fonction de comparaison des forces détermine que les forces simulées sont suffisamment proches des valeurs mesurées. Dans ce cas, le module de Young utilisé pour la dernière itération est considéré comme le représentant optimal de la réalité.

L'avantage principal de cette méthode par rapport à l'estimation directe du module de Young est que la valeur trouvée reproduira exactement la sensation haptique voulue, codée par ce paramètre, en l'utilisant avec l'algorithme de simulation mécanique choisi pour l'estimation inverse. En revanche, si ce même simulateur n'est pas assez représentatif de la réalité, l'estimation inverse donnera un résultat d'autant biaisé par rapport à la valeur physique réelle du paramètre (lorsque celui-ci a une signification physique réelle).

5.5.2.1 Paramètres initiaux

Puisque nous réalisons cette estimation du module de Young après avoir effectué une estimation du coefficient de Poisson (cf 5.3.2), nous possédons une valeur optimale ν_{optim} pour ce paramètre. Celle-ci est utilisée pour toutes les simulations mécaniques du processus itératif. Seul le paramètre du module de Young E varie. L'estimation se fait donc sur une seule variable, ce qui accélère considérablement le processus.

5.5.2.2 Comparaison des forces

Soit $\mathbf{F}_{réelles}$ un vecteur contenant pour chaque pas de temps (k) de la vidéo initiale (contenant t pas de temps) les forces mesurées par notre dispositif de capture d'efforts, décrit en

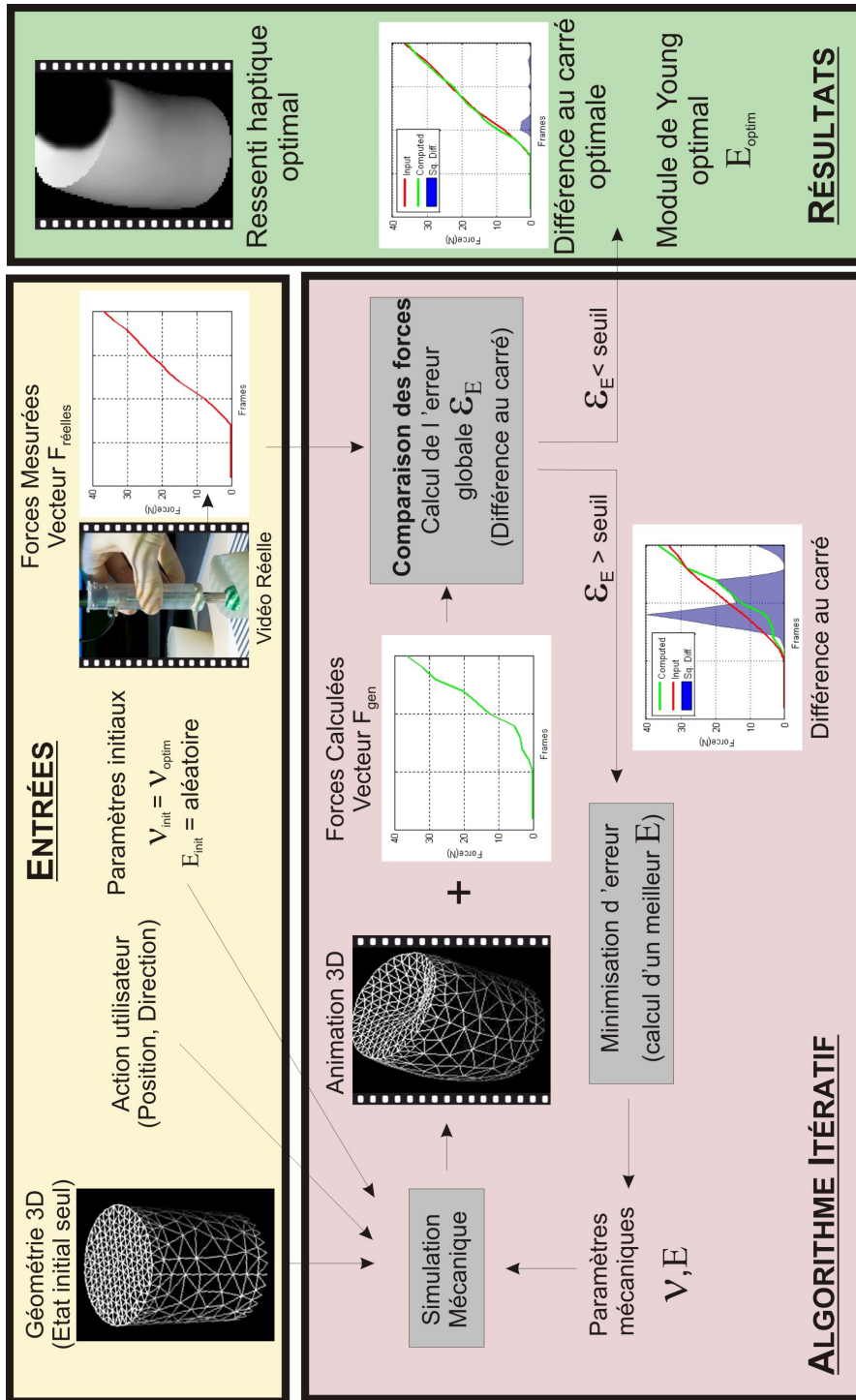


Figure 5.24 - Schéma global de notre algorithme d'estimation du module de Young à partir des forces mesurées.

4.3. Nous définissons également un vecteur analogue \mathbf{F}_{gen} , de même taille, contenant les forces de contact générées à partir de l'animation synthétique et des points de contact pour cette itération. La fonction de comparaison des forces peut alors s'écrire :

$$\varepsilon_E = \Delta_E \{ \mathbf{F}_{réelles}, \mathbf{F}_{gen} \} = \frac{1}{t} \sum_{k=1}^t \left(\mathbf{F}_{réelles}^k - \mathbf{F}_{gen}^k \right)^2 \quad (5.33)$$

Le graphique de la figure 5.25 illustre ce calcul d'erreur.

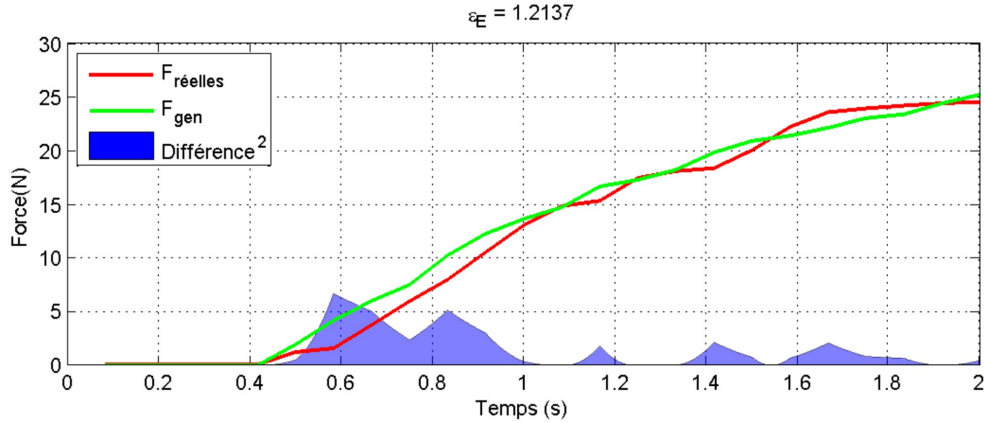


Figure 5.25 - Illustration de la comparaison des forces : La courbe rouge représente les forces réelles mesurées. La courbe verte représente les forces de contact simulées. La différence entre ces deux courbes au carré est représentée par la surface bleue. La moyenne de cette différence donne ε_E pour cette comparaison.

5.5.2.3 Optimisation

Nous avons vu qu'une méthode d'optimisation est nécessaire à l'estimation inverse du coefficient de Poisson car une itération du processus est relativement coûteuse en temps d'exécution (cf 5.4.4). Bien que certaines étapes comme le rendu ou la comparaison vidéo ne soient pas nécessaires pour le module de Young, une itération de l'algorithme inverse illustré par le schéma de la figure 5.24 demande une simulation mécanique complète ainsi que la comparaison des forces simulées avec les forces mesurées. Or ces calculs, en particulier la simulation mécanique, peuvent également être longs à exécuter. De plus, l'espace de validité du module de Young est beaucoup plus vaste que celui du coefficient de Poisson, comme le montre le tableau 2.1. On peut par exemple noter que les caoutchoucs peuvent atteindre des valeurs allant de $700MPa$ à près de $5000MPa$. La précision lors d'une étude exhaustive est alors très faible si l'on souhaite, par exemple, ne pas dépasser 100 itérations. La précision d'une telle évaluation serait de $50MPa$.

Nous utilisons donc tout d'abord une évaluation directe pour améliorer cette précision. En effet, le résultat d'une estimation directe obtenue par la méthode décrite en 5.5.1 nous permet de réduire l'intervalle de recherche. Ainsi, pour une approximation du module de Young $E_{direct} \approx 0.1MPa$ on utilisera l'intervalle $[0 - 0.2] MPa$. La précision de l'estimation pour 100 évaluations de la fonction objectif est alors de $2000Pa$. Cet exemple est illustré sur le graphique gauche de la figure 5.27.

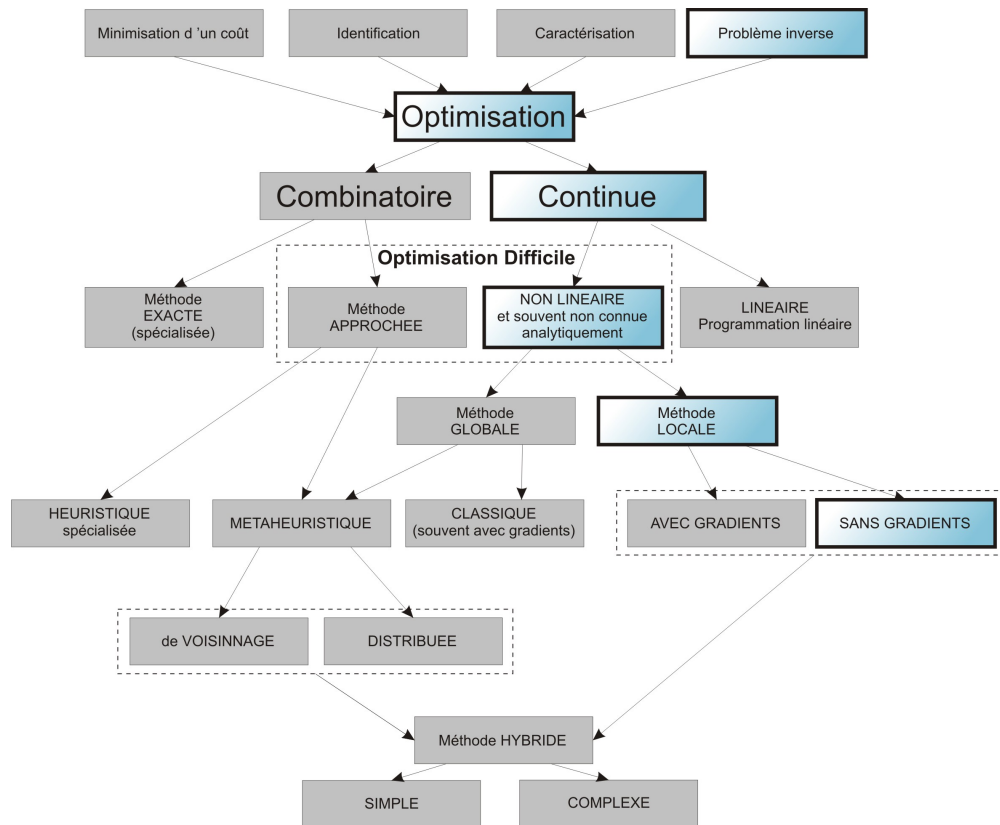


Figure 5.26 - Choix de la méthode d'optimisation pour l'estimation du module de Young à partir de l'organigramme de classification générale des méthodes d'optimisation mono-objectif.

Nous adoptons ensuite une méthode d'optimisation pour réduire le nombre d'évaluations de la fonction. Pour la choisir, nous nous basons également sur l'organigramme de classification générale des méthodes d'optimisation mono-objectif (cf Fig. 5.26). Là encore, nous sommes face à problème d'optimisation continue puisque le module de Young peut prendre une infinité de valeurs. La fonction objectif est également non linéaire (cf Fig. 5.27 gauche). En revanche, grâce à la conception de notre dispositif de mesure utilisant une « bille de transmission », le bruit de mesure très faible. La fonction objectif présente donc peut de minima locaux. Il est donc possible d'utiliser une méthode d'optimisation locale. Nous avons opté pour l'algorithme *Golden Section* (cf 3.2.1.1). Celui-ci n'a pas recours au calcul des gradients de la fonction qui sont inconnus dans notre cas (pas d'expression analytique). De plus, nous utilisons également une amélioration de cet algorithme, l'interpolation parabolique, qui permet de réduire le nombre d'évaluations de la fonction à chaque itération.

Grâce à cette technique, l'estimation inverse du module de Young est réalisée en quelques itérations. Le graphique de droite sur la figure 5.27 montre le déroulement d'une telle évaluation pour le même cas que celui traité sur le graphique de gauche. Seules six évaluations de la fonction objectif (six simulations mécaniques et six comparaisons de forces) sont nécessaires pour évaluer le module de Young avec une précision de $1Pa$.

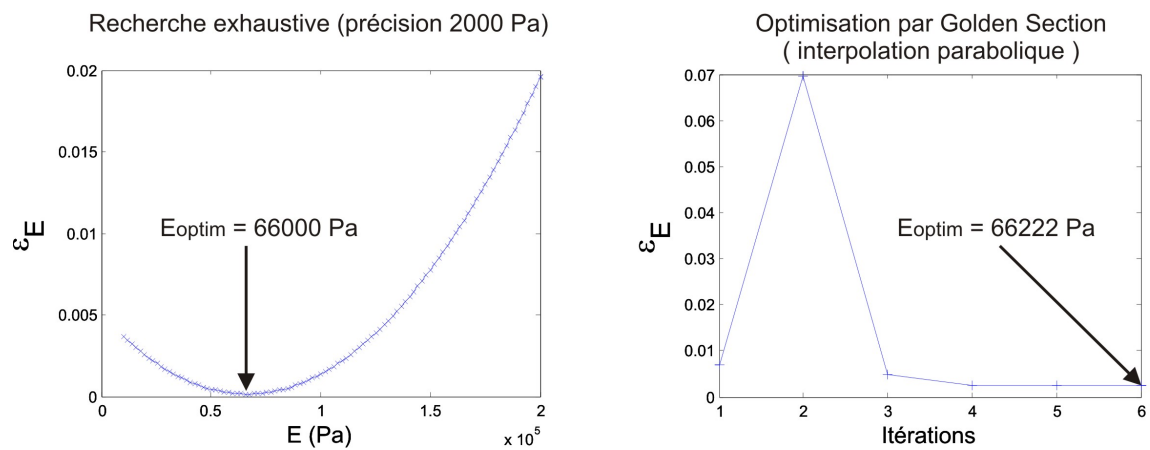


Figure 5.27 - Exemple d'estimation inverse du module de Young. À droite, déroulement d'une estimation inverse « exhaustive » après réduction de l'espace de recherche par estimation directe, pour une précision de 2000 Pa en 100 itérations. À droite, déroulement d'une estimation inverse pour le même cas avec une optimisation par Golden Section, interpolation parabolique.

Sixième Chapitre

RÉSULTATS

Nous exposons dans ce chapitre les résultats que nous avons obtenus lors des expérimentations. Nous montrons tout d'abord les différents matériaux testés en expliquant ces choix. Nous détaillons ensuite les expérimentations destinées à estimer le coefficient de Poisson à partir de séquences d'images. Enfin, nous reportons les résultats de nos estimations du module de Young puis les comparons à ceux obtenus lors de mesures sur une machine de test en traction/compression universelle.

6.1 Matériaux testés

6.1.1 Silicones

Le silicone est un matériau ayant des propriétés mécaniques relativement « pures » : il peut être simulé mécaniquement par un modèle d'élasticité non-linéaire comme le *Neo-Hookean* par exemple (cf 2.2.4). Le comportement de la plupart des silicones est semblable à celui du caoutchouc, c'est à dire quasiment incompressible (coefficient de Poisson $\nu \rightarrow 0.5$). Ces propriétés ont fait de ce matériau une référence dans le domaine de la mesure de déformations mécaniques. Plusieurs travaux l'utilisent notamment comme base pour valider leur techniques d'estimation de propriétés mécaniques [Kauer 01a] [Kerdok 01] [Samani 01] [Lang 02].

Nous choisissons également le silicone pour tester nos méthodes d'estimation du coefficient de Poisson et du module de Young. Le silicone « Ecoflex » a été par exemple utilisé par Leskovsky (*et al.*) [Leskovsky 06] pour une étude de la « fidélité » d'un rendu haptique par tests sur différents sujets devant répondre à la question « réel ou virtuel ». Les auteurs ont créé pour cela plusieurs densités de silicone, ce qui permet de faire varier le module de Young des matériaux obtenus.

Ce silicone est fabriqué à l'aide de deux composants liquides. Une fois mélangés, on obtient un gel pâteux qui durcit après une période de repos. Pour créer quatre densités (et donc ici quatre « duretés ») différentes, nous ajoutons au mélange, avant repos, un diluant dans des proportions croissantes. Chaque mixture est préparée à partir d'un même mélange initial des deux produits de base avant d'être coulée dans des moules. Ainsi, le silicone A ne contient pas de diluant, le silicone D est celui qui en contient le plus, et les silicones B et C en contiennent dans des proportions moyennes relativement proches, ce qui nous permettra d'évaluer la précision de notre estimation du module de Young. Le silicone obtenu est translucide. Pour simplifier le rendu et minimiser les erreurs lors des étapes de traitement d'image, nous ajoutons un pigment blanc qui opacifie le matériau. Nous moulons donc ici des objets cylindriques et hémisphériques en silicone blanc pour chaque densité (cf Fig. 6.1).

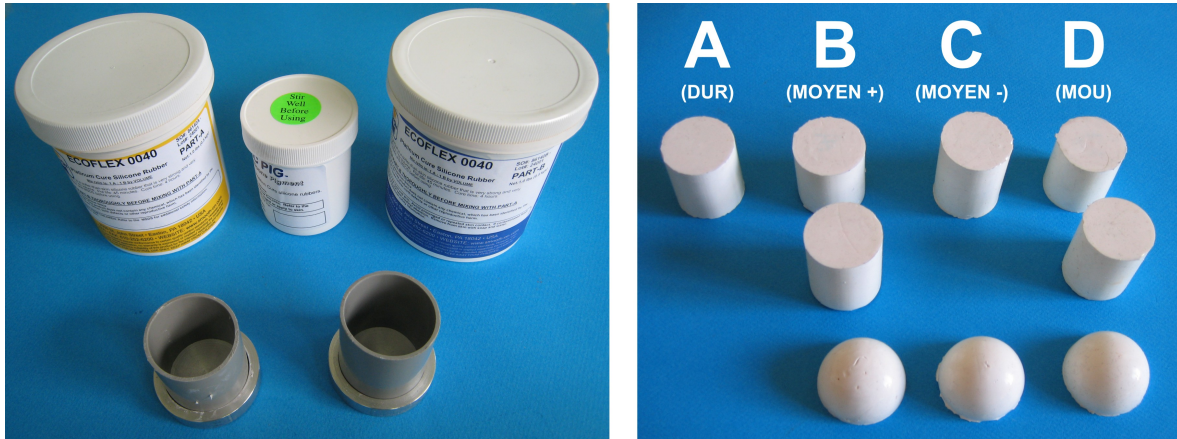


Figure 6.1 - Objets en silicone. À gauche, les produits utilisés pour créer le mélange du silicone (© Ecoflex A et B), un pigment pour opacifier le silicone (blanc) et deux moules pour créer des cylindres. À droite, les objets en silicone après moulage sur lesquels nous effectuons nos expérimentations.

Mesures mécaniques du module de Young Pour évaluer la précision de nos algorithmes d'estimation du module de Young, nous comparons nos résultats avec des valeurs mesurées mécaniquement sur nos échantillons de silicone. La machine « Zwick » est une machine de test universelle en traction/compression. Le fonctionnement est simple. En traction, il s'agit d'attacher une extrémité de l'échantillon à la partie fixe de la machine et l'autre à l'extrémité mobile. Une force connue est alors appliquée sur cette partie mobile et le déplacement obtenu est mesuré avec précision. Le fonctionnement en compression est identique, à ceci près que la force est inversée. Le résultat de ces mesures est une liste de couples forces/déplacements et il est nécessaire d'extraire les informations utiles de ces données.

Pour ce faire, nous utilisons une méthode semblable à celle que nous avons définie pour l'estimation directe du module de Young (cf 5.5.1). Nous transformons les données pour les placer dans l'espace « contrainte/allongement relatif » ($\sigma_{Zwick}/\varepsilon_{Zwick}$). Les échantillons que nous avons envoyés étant cylindriques et la compression axiale, nous avons :

$$\sigma_{Zwick} = \frac{force}{surface} = \frac{F}{\pi r^2} \quad (6.1)$$

$$\varepsilon_{Zwick} = \frac{d}{L_0} \quad (6.2)$$

où F et d forment respectivement le couple force/déplacement renvoyé par la machine, et r et L_0 respectivement le rayon et la longueur initiale du cylindre. Nous procédons ensuite à une extraction par régression linéaire et par *RanSaC*, de la même manière que pour notre algorithme d'estimation directe, à ceci près que nous retirons au préalable manuellement la partie non linéaire de la courbe obtenue. Le module de Young de référence est alors obtenu en prenant la pente de la droite ainsi extraite. La figure 6.2 illustre l'extraction du module de Young de référence pour les quatre mélanges de silicone. Nous obtenons à l'aide de cette méthode les valeurs de référence $E_{siliconeA} = 150505Pa$, $E_{siliconeB} = 84317Pa$, $E_{siliconeC} = 81162Pa$, $E_{siliconeD} = 54876Pa$.

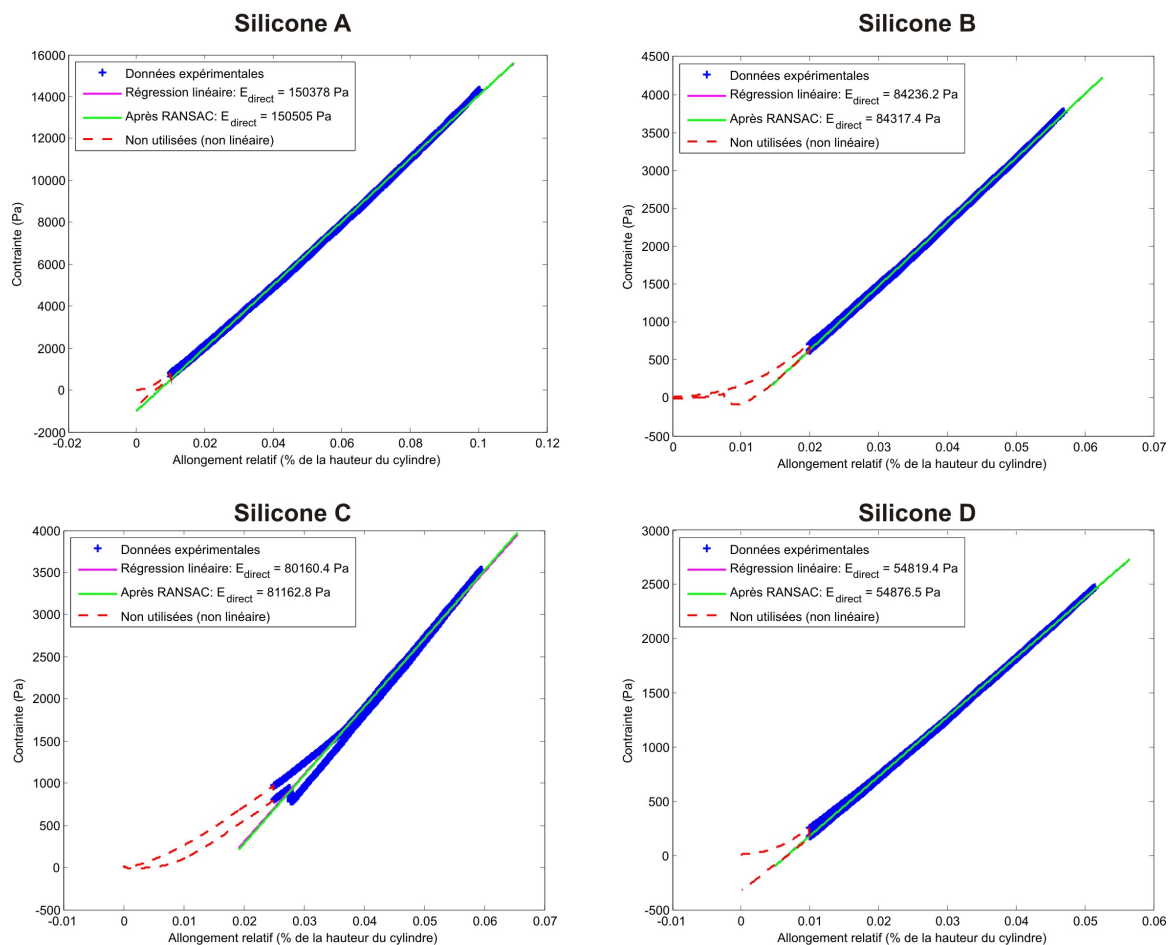


Figure 6.2 - Extraction du module de Young d'après les mesures sur machine universelle Zwick. Les données acquises sont transposées dans l'espace contrainte/déformation (stress/strain, en bleu). La partie non-linéaire de la déformation n'est pas utilisée pour l'extraction du module de Young (en pointillés rouges). Deux méthodes d'extraction sont utilisées : par régression linéaire (en magenta), et par RanSaC (en vert). La pente de ces droites (très proches) donne le module de Young pour chaque mélange de silicone.

6.1.2 Éponges

Tester nos estimations sur des éponges nous semble un bon moyen d'évaluer les capacités de notre méthode. En effet, le matériau a un comportement totalement différent du silicone. Nous espérons ainsi retrouver la compressibilité de ceux-ci (coefficient de Poisson $\nu \rightarrow 0$). De plus, les formes géométriques de celles-ci sont très simples (parallélépipèdes rectangles, cf Fig. 6.3), ce qui évite des approximations trop importantes lors de la modélisation virtuelle des objets et limite les occlusions lors des déformations.

Nous réalisons des expérimentations sur plusieurs types d'éponges : « cimentier », « gros travaux », et « végétale » (cf Fig. 6.3). Il est important de noter ici que la « dureté » des matériaux est altérée avec les temps d'exposition à l'air et l'humidité ambiante, ce qui laisse présager quelques dérives lors de l'évaluation du module de Young. De plus, les éponges sont par nature non-homogènes. Il y a donc de fortes chances qu'elles montrent un comportement *anisotrope* (cf 2.1.4).

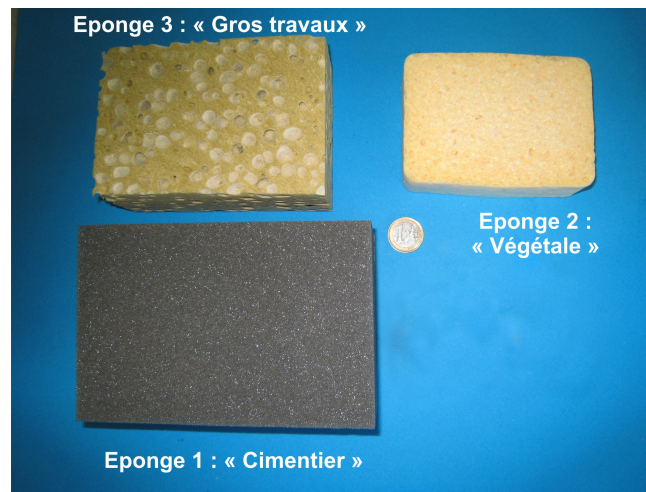


Figure 6.3 - Différentes éponges utilisées dans nos expérimentations.

6.1.3 Peluches

Les peluches sont quant à elles composées de plusieurs matériaux. Elles devraient donc également montrer un comportement anisotrope, mais elles ont des formes géométriques plus complexes. Nous pouvons donc tester la robustesse des métriques aux probables occlusions visuelles. Leur manque de rigidité nous permettra également de vérifier que la précision de notre dispositif de mesure de force est suffisamment discriminante.

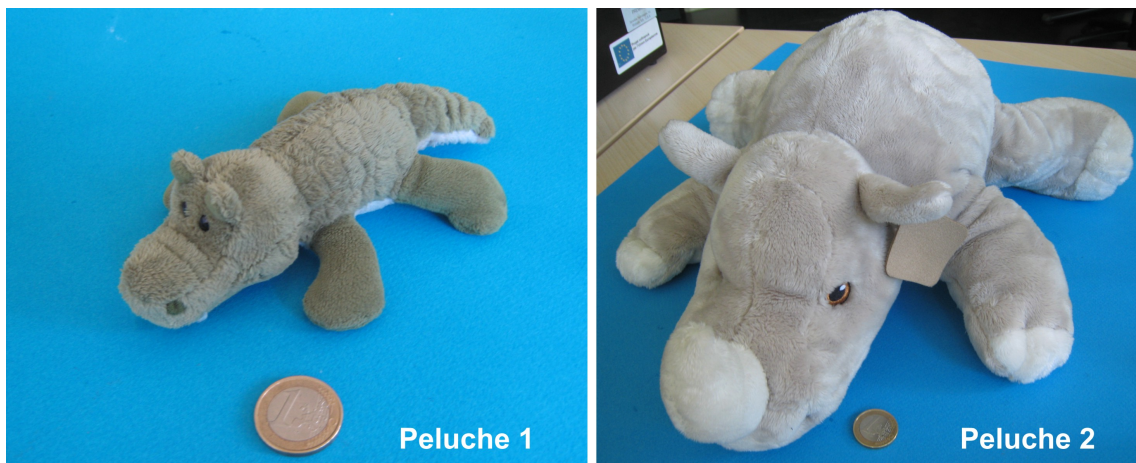


Figure 6.4 - Les deux peluches utilisées dans nos expérimentations.

Nous testons ici deux peluches différentes (cf Fig 6.4). La première est relativement petite. Elle a l'avantage d'avoir un aimant dans chaque patte, ce qui permet de bien connaître les contraintes appliquées au système lors de la simulation mécanique (cf 2.3.2.4). La seconde, plus grande et un peu plus « molle » au toucher, nous permettra de tester notre évaluation du module de Young.

6.2 Estimations du coefficient de Poisson

6.2.1 Silicones

Nous montrons ici les résultats que nous avons obtenus pour l'estimation du coefficient de Poisson à partir d'objets en silicone. Dans la première partie, nous détaillons les premières expérimentations sur un cylindre réalisé avec le mélange A. Celles-ci ont permis de trouver la bonne configuration de l'ensemble des éléments de la scène lors de la capture.

La seconde partie est dédiée aux résultats obtenus sur tous les autres objets en silicone, cylindres et hémisphères, issus des trois autres mélanges (B, C et D). Nous donnons un tableau récapitulatif de ces résultats ainsi qu'une évaluation de chaque métrique. En effet, le silicone présente généralement un coefficient de Poisson proche 0.5 (quasi-incompressible). Pour valider nos résultats, nous calculons donc une erreur basée sur la valeur 0.49, valeur maximale requise pour la stabilité numérique de la simulation informatique. Une moyenne des erreurs pour chaque métrique est alors calculée et les résultats sont commentés.

Enfin, nous illustrons par un autre tableau les gains obtenus par l'utilisation de l'optimisation par recuit simulé.

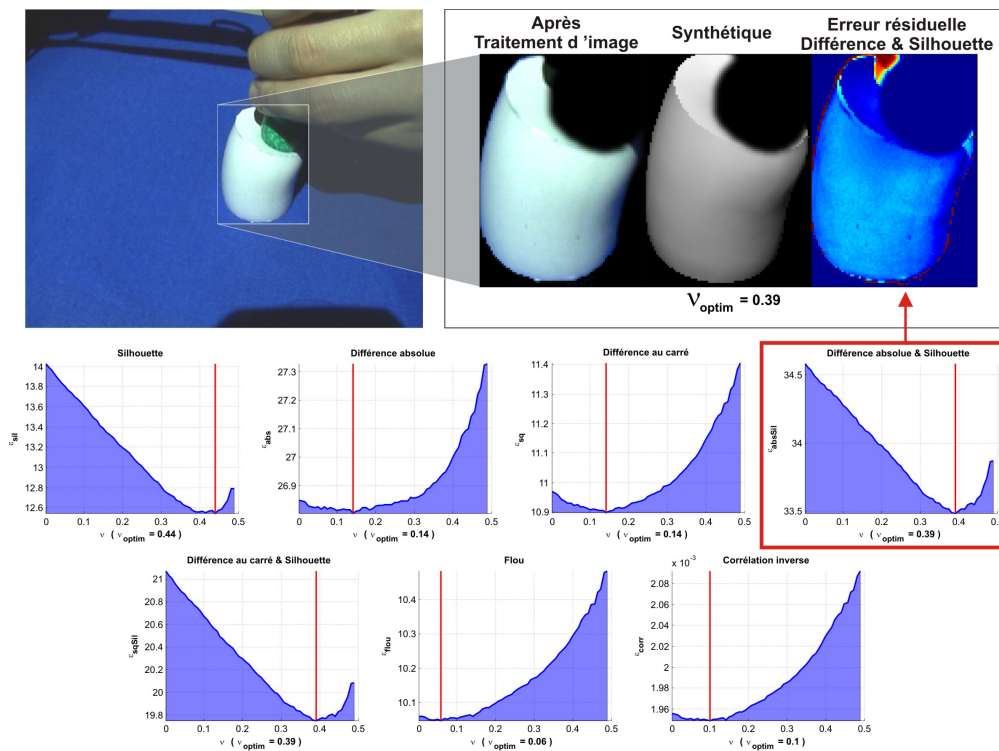


Figure 6.5 - Estimations du coefficient de Poisson obtenues pour le premier test sur le silicone A.

Le premier test sur le cylindre A est illustré sur la figure 6.5. La photo en haut à gauche est une capture de la vidéo filmée. On peut y voir que la déformation est un peu excentrée par rapport à l'axe principal du cylindre. Le but est d'accentuer le « bourrelet » généré par la déformation qui, à notre sens, caractérise bien le coefficient de Poisson. Le cadre en haut à droite montre le calcul d'une métrique composée (« Différence absolue et silhouette ») pour la même image. L'image de gauche dans ce cadre montre l'état de la vidéo initiale après

les pré-traitements (détourage et masquage automatique de l'outil). L'image centrale montre l'image correspondante de la vidéo synthétique. L'erreur résiduelle de la métrique pour cette image est montrée à droite (échelle thermique, bleu = faible différence, rouge = différence importante). On peut y voir un fin liseré rouge autour de l'objet, caractéristique d'une petite erreur de calage entre l'objet réel et l'objet virtuel. Un autre artéfact est également visible. Il s'agit de l'ombre de la main de l'utilisateur qui n'est pas simulée sur l'animation synthétique.

Les graphiques situés en bas de la figure montrent les courbes de réponse de chaque métrique. Les lignes rouges indiquent le coefficient de Poisson retrouvé pour chacune d'elles. On peut tout d'abord noter un échec général des « métriques de surface ». Cela peut s'expliquer par le manque de réalisme du rendu (par exemple la composante « réflexion ») mais également par un décalage dans le positionnement de la source de lumière. Les « métriques de silhouette » fonctionnent relativement bien (pour rappel, le coefficient de Poisson du silicone tend vers 0.5). Cependant, des instabilités numériques dues à des contraintes relativement importantes dans la simulation mécanique apparaissent au delà d'un coefficient de Poisson de 0.44, ce qui explique l'échec partiel des métriques de silhouette. De plus, sur cette expérimentation, la partie au contact avec le support (la table) se décolle légèrement à la fin de l'animation ce qui introduit un biais dans l'estimation, la simulation numérique ne gérant pas ce cas de figure.

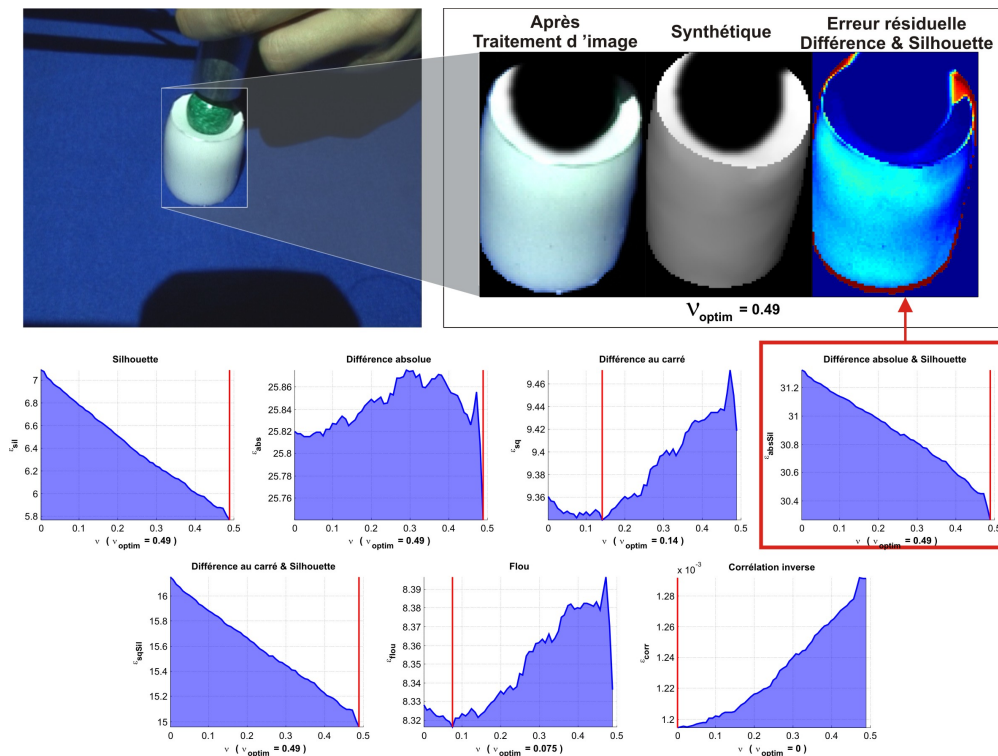


Figure 6.6 - Résultats des estimations du coefficient de Poisson obtenus pour le second test sur le silicone A (pression au centre du cylindre).

Pour éviter ce phénomène, nous réalisons une seconde expérimentation en appuyant cette fois-ci au centre du cylindre. La figure 6.6 illustre ce test. Elle est basée sur le même schéma que la figure 6.5. On peut faire les mêmes remarques pour l'erreur résiduelle (artéfact, liseré rouge). Les métriques de surface ne convergent toujours pas. En effet, cette expérimentation

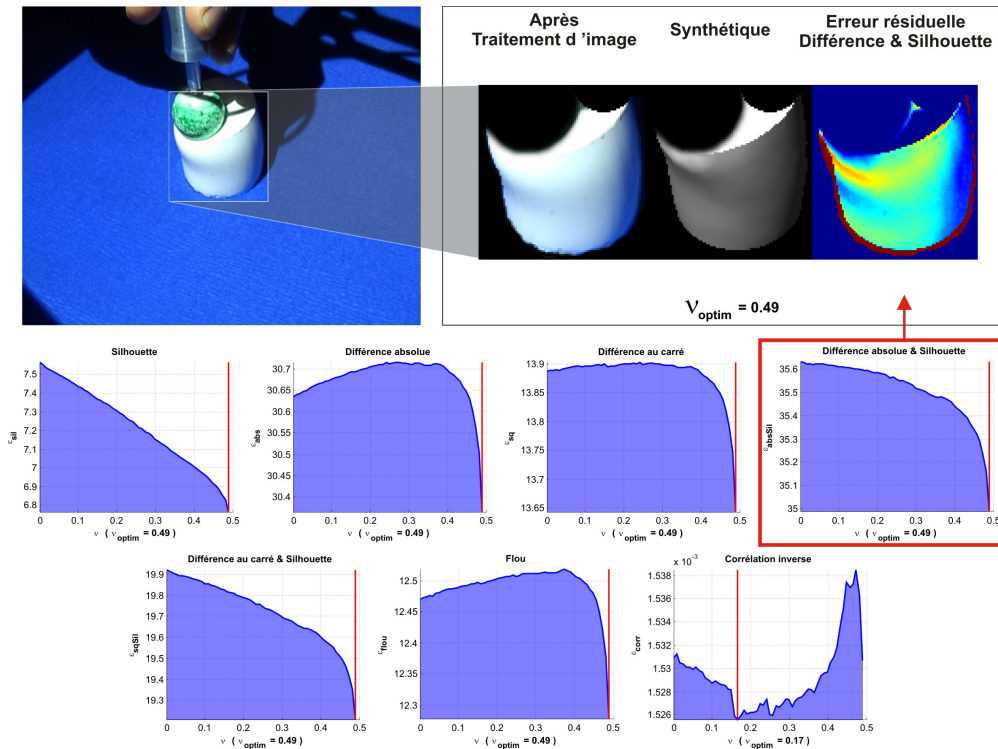


Figure 6.7 - Résultats des estimations du coefficient de Poisson obtenus pour le troisième test sur le silicone A (augmentation du contraste et de l'émission lumineuse).

n'est toujours pas discriminante pour la surface de l'objet (l'écart de l'erreur pour la différence absolue, ε_{abs} n'est que de 0.1 niveaux de gris de moyenne). En revanche, les métriques de silhouette convergent toutes vers le résultat attendu (la simulation numérique est plus stable pour les hautes valeurs de ν).

Lors du troisième test, nous avons augmenté le contraste par un pré-traitement additionnel de la vidéo capturée ainsi qu'une émission accrue de la source lumineuse (cf Fig. 6.7). Cela a triplé la précision des métriques de surface qui convergent désormais, exceptée la « corrélation inverse » qui n'est pas affectée par ce changement puisqu'elle est constituée d'une étape de « normalisation » des images.

Sur ces trois premiers tests, la caméra est relativement loin de l'objet. Pour améliorer la résolution de la zone d'intérêt, nous la rapprochons pour le quatrième test (cf Fig. 6.8). Les différences sont alors plus « visibles » et les réponses des métriques de silhouette sont moins perturbées. Toutes les métriques convergent exceptée la corrélation inverse, peu discriminante.

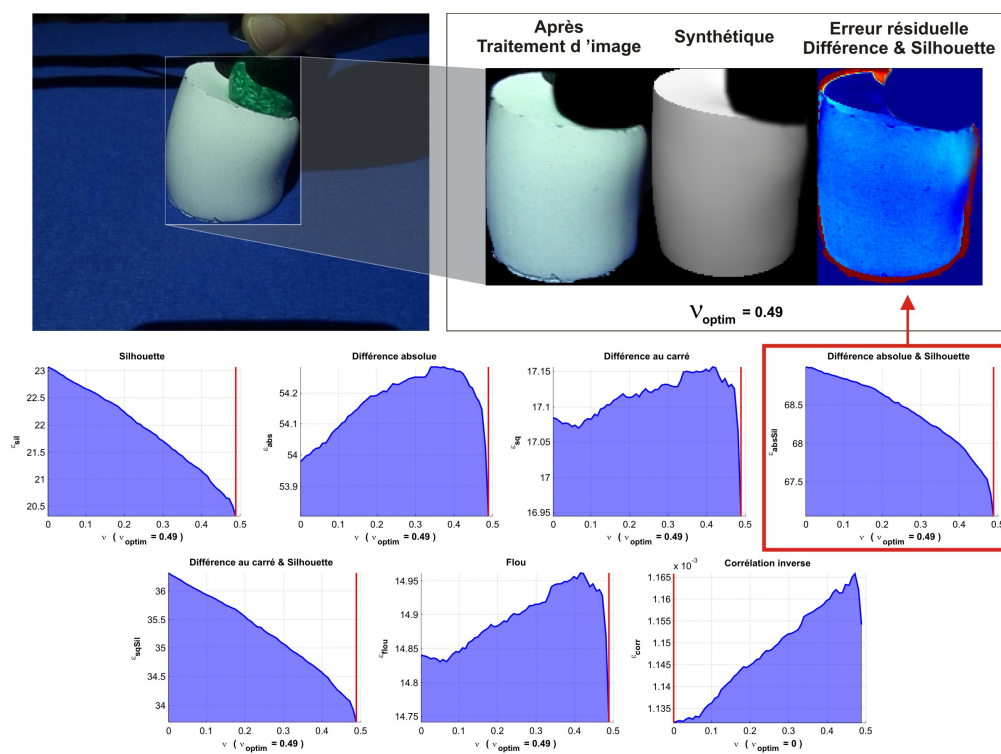


Figure 6.8 - Résultats des estimations du coefficient de Poisson obtenus pour le quatrième test sur le silicone A (caméra rapprochée).

6.2.1.1 Autres objets silicones

| Exp. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Illus. |
|-------------------------|---------------|--------|--------|--------|-------|--------|--------|-----------|
| Cylindres | | | | | | | | |
| A - 1 | 0.44 | 0.14 | 0.14 | 0.39 | 0.39 | 0.06 | 0.10 | Fig. 6.5 |
| A - 2 | 0.49 | 0.49 | 0.14 | 0.49 | 0.49 | 0.08 | 0.00 | Fig. 6.6 |
| A - 3 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.17 | Fig. 6.7 |
| A - 4 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.00 | Fig. 6.8 |
| B - 1 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.07 | Fig. 6.9 |
| B - 2 | 0.48 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.00 | |
| C - 1 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | |
| C - 2 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | |
| D - 1 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | Fig. 6.10 |
| D - 2 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | |
| Hémisphères | | | | | | | | |
| B - 1 | 0.49 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | Fig. 6.11 |
| B - 2 | 0.49 | 0.00 | 0.49 | 0.01 | 0.49 | 0.00 | 0.49 | |
| C - 1 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | |
| C - 2 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | Fig. 6.12 |
| D - 1 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | 0.49 | |
| Erreurs moyennes | | | | | | | | |
| | 0.94 % | 20.88% | 18.68% | 16.80% | 9.26% | 28.57% | 48.51% | |

Table 6.1 - Tableau récapitulatif des résultats d'estimation du coefficient de Poisson pour les objets en silicone (sans méthode d'optimisation, par recherche exhaustive avec une précision de 0.01, 100 itérations) obtenus pour chaque métrique (1 = Silhouette, 2=Différence absolue, 3= Différence au carré, 4=Différence absolue & Silhouette, 5=Différence au carré & Silhouette, 6=Flou, 7= Corrélacion inverse). Le calcul de l'erreur moyenne est effectué pour une valeur vérité de 0.49 (incompressibilité quasi complète). Certaines expérimentations sont commentées et illustrées (liens dans la dernière colonne).

Cylindres Nous avons effectué deux tests semblables sur les cylindres fabriqués avec le mélange B. Les résultats sont identiques. Toutes les métriques convergent, exceptée la corrélation inverse (cf Fig. 6.9).

Pour les mélanges plus « mous », la déformation est plus importante et permet également à la métrique de « corrélation inverse » de converger malgré un artéfact d'ombre (cf Fig. 6.10). Cela prouve que le biais introduit par ce genre de perturbation ne perturbe pas la convergence des métriques.

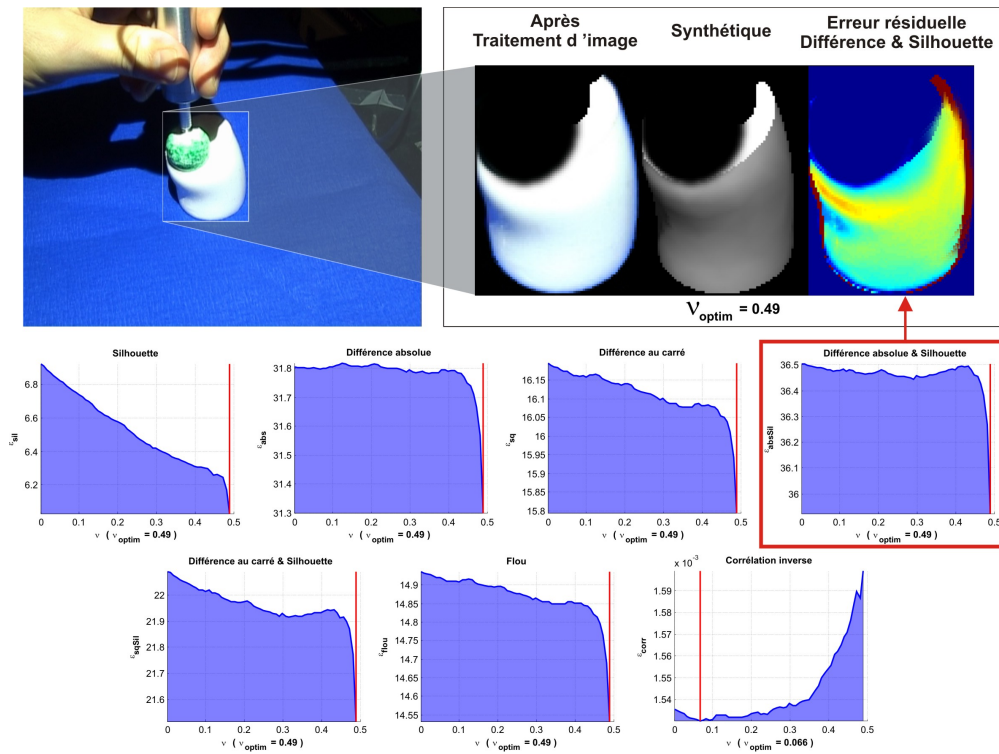


Figure 6.9 - Résultats des estimations du coefficient de Poisson obtenus pour le premier test sur un cylindre en silicone B.

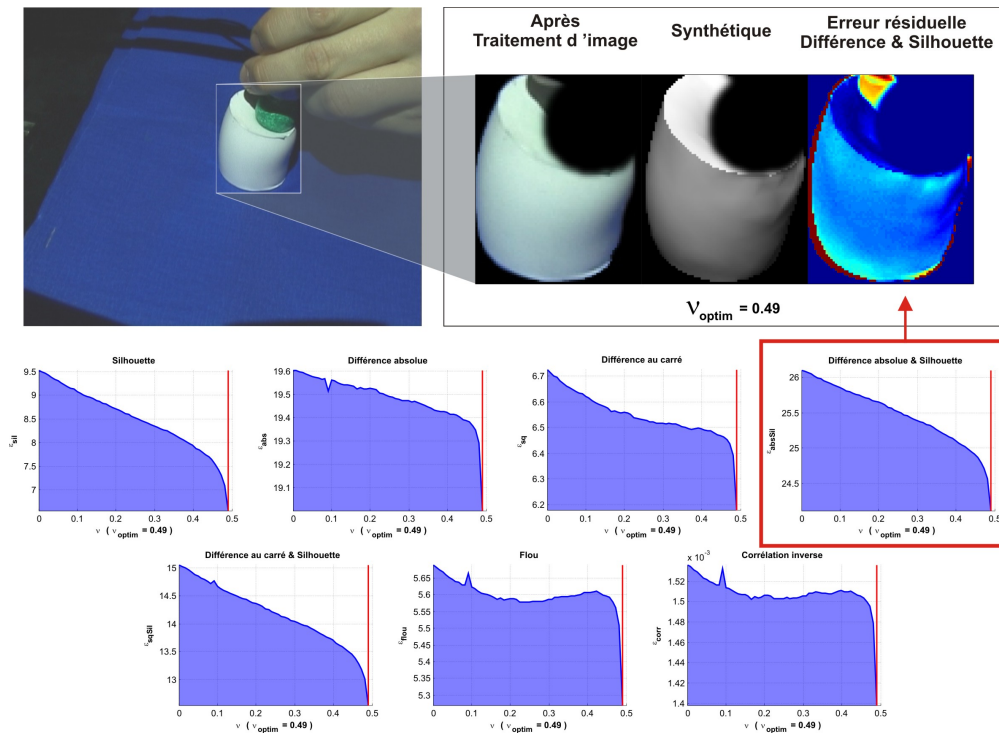


Figure 6.10 - Résultats des estimations du coefficient de Poisson obtenus pour le premier test sur un cylindre en silicone D.

Hémisphères Les tests sur les hémisphères en silicone donnent des résultats similaires. Cependant les premières expérimentations (sur l'hémisphère de silicone B) ont été réalisées avec des conditions d'éclairage non maîtrisées (illumination ambiante). Par conséquent, seule la métrique « Silhouette » converge car la différence de l'état de surface avec la vidéo synthétique est trop importante. Ce problème est illustré sur la figure 6.11. Même sur les métriques composées, on peut voir que l'erreur résiduelle sur la surface est bien plus importante que celle de la silhouette pure (en haut à droite pour les deux valeurs extrêmes du coefficient de Poisson, 0.0 et 0.49).

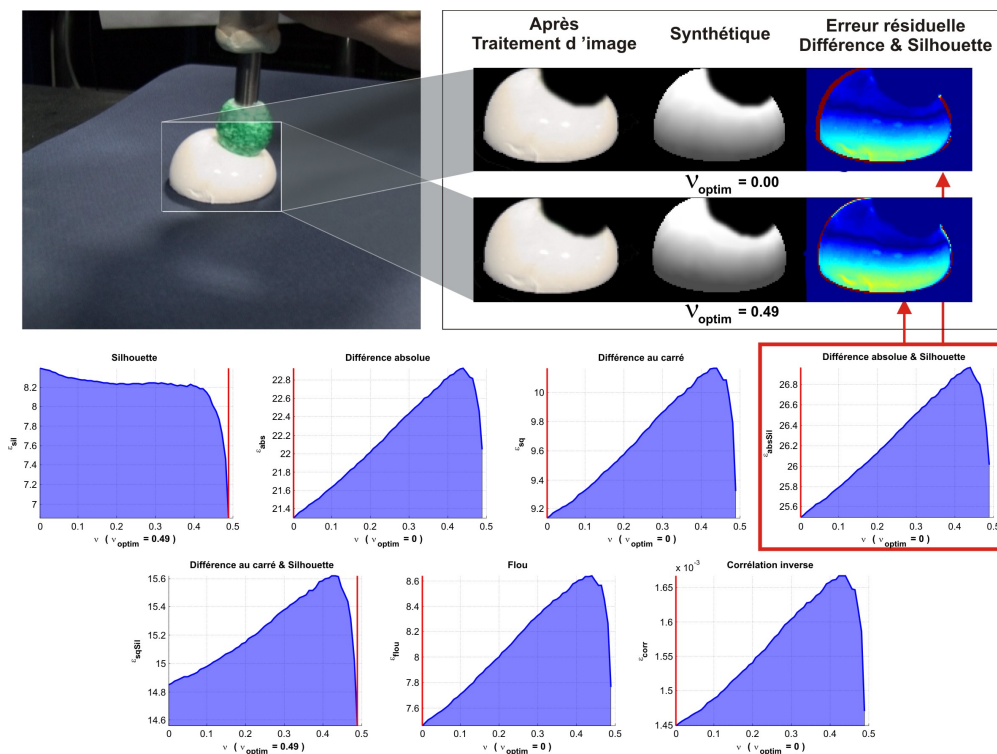


Figure 6.11 - Résultats des estimations du coefficient de Poisson obtenus pour le premier test sur un hémisphère en silicone B (éclairage ambiant).

Ce problème est résolu lorsque l'on utilise une source lumineuse connue. La figure 6.12 montre la convergence de toutes les métriques pour l'hémisphère en silicone C.

Le tableau 6.1 récapitule tous les résultats obtenus pour chaque métrique d'erreur sans optimisation (méthode « exhaustive »). Les liens de la dernière colonne pointent vers la figure correspondante. Ces résultats sont séparés par type d'objets, cylindriques ou hémisphériques. La dernière ligne donne l'erreur d'estimation de chaque métrique (moyenne calculée sur toutes les expérimentations en se basant sur la valeur vérité $\nu = 0.49$ pour le silicone).

La méthode générant les meilleures estimations est la métrique « Silhouette » avec une erreur d'estimation inférieure à 1%. Les métriques composées utilisant également la silhouette donnent de meilleurs résultats que les métriques de surface pures mais avec un taux de réussite insuffisant. Avec près de 50% d'erreur, la métrique « Corrélation inverse » donne les moins bons résultats. Il est important de noter ici que de nombreuses approximations quant à l'éclairage ont été faites sur un bon nombre d'expérimentations ce qui dégrade fortement les performances des métriques de surface. Cependant, cela montre la sensibilité de ces métriques « basiques »

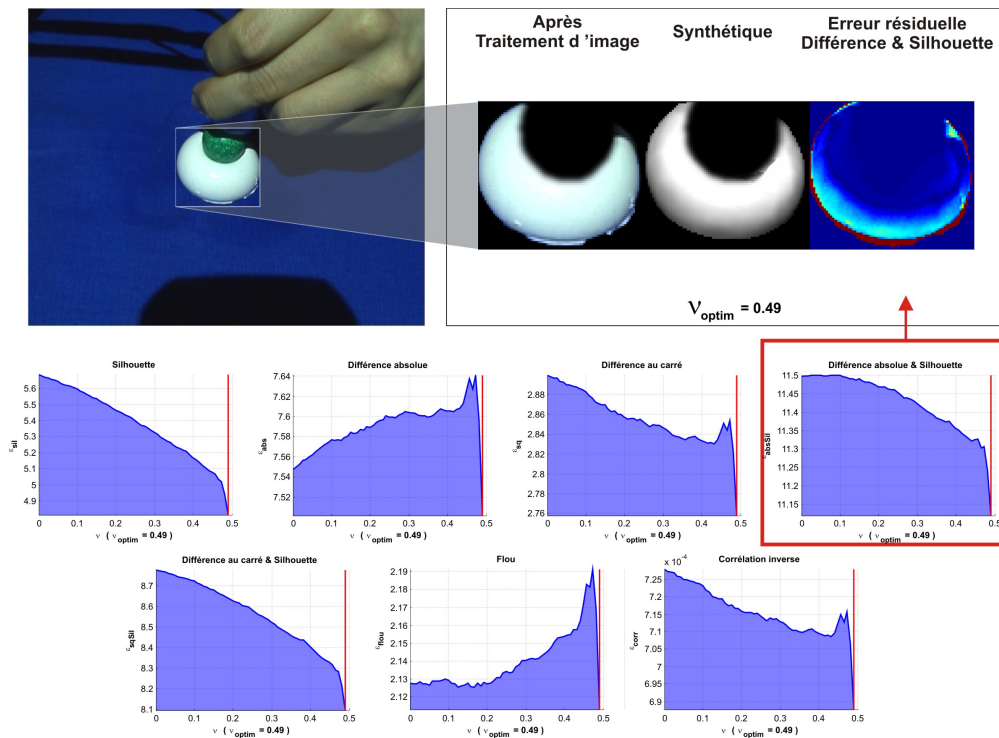


Figure 6.12 - Résultats des estimations du coefficient de Poisson obtenus pour le second test sur une hémisphère en silicone *B* (source lumineuse calibrée).

aux conditions d'éclairage. Il serait donc intéressant de développer les métriques utilisant des motifs projetés différents pour limiter cet effet (cf 7.2).

Les bons résultats obtenus par la métrique « Silhouette » sont également à attribuer au fait que dès les premières estimations nous avons porté un soin particulier à faire apparaître la déformation induite sur l'objet dans l'axe de la caméra (création d'un « bourrelet » bien visible). De plus, chaque test a été effectué avec un calage relativement précis de l'objet (erreur < 1mm). Une étude de sensibilité de cette erreur de placement de l'objet virtuel pourrait permettre d'évaluer la robustesse de cette métrique.

6.2.1.2 Métrique de projection

La figure 6.13 présente les résultats obtenus lors d'estimation du coefficient de Poisson à l'aide d'une métrique utilisant un motif par lignes projetées, nommée « Angular Map » (cf 5.4.3.7). Ces expérimentations ont été réalisées sur plusieurs objets en silicone et avec plusieurs angles de projection (cf colonne gauche). Malheureusement, dans aucun des cas l'algorithme converge ne vers $\nu = 0.49$ (cf colonne du centre). On peut voir sur l'erreur résiduelle (cf colonne droite) que des artefacts apparaissent sur la surface de l'objet. Ces artefacts sont probablement dus à l'erreur de calibrage du vidéo projecteur. Ces tests ont montré que cette métrique y était très sensible. D'autres métriques basées sur des motifs projetés différents pourront être développées pour tenter de réduire ces problèmes.

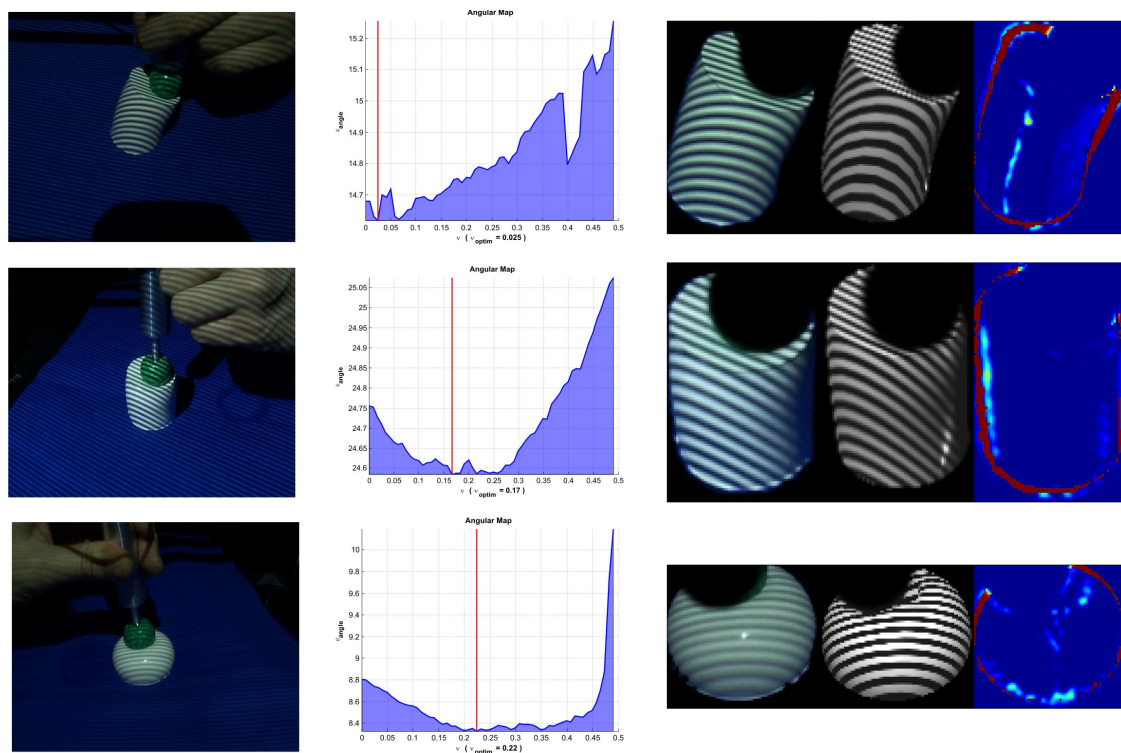


Figure 6.13 - Résultats des estimations du coefficient de Poisson obtenus pour plusieurs objets de silicone avec la métrique « Angular Map ».

6.2.1.3 Optimisation par recuit simulé

Le tableau 6.2 donne les résultats obtenus avec la méthode d'optimisation par recuit simulé (cf 5.4.4.2). Le nombre d'évaluations de la fonction d'erreur (simulation mécanique + rendu vidéo + comparaison vidéo) est reporté pour chaque expérimentation et chaque métrique (1 = Silhouette, 2 = Différence absolue, 3 = Différence au carré, 4 = Différence absolue & Silhouette, 5 = Différence au carré & Silhouette, 6 = Flou, 7 = Corrélation inverse). Les gains moyens obtenus par comparaison avec une étude exhaustive de 100 évaluations sont calculés pour chaque expérimentation et chaque métrique. Pour chaque estimation optimisée, la valeur estimée est très proche de la valeur trouvée par recherche exhaustive (erreur < 0.005).

| Exp. | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Gain |
|---------------------------|-------|-------|-------|-------|-------|-------|-------|--------------------|
| Cylindres | | | | | | | | |
| A - 1 | 22 | 22 | 22 | 22 | 37 | 71 | 69 | 339 % |
| A - 2 | 17 | 22 | 32 | 17 | 22 | 32 | 70 | 408 % |
| A - 3 | 17 | 17 | 17 | 17 | 22 | 27 | 69 | 475 % |
| A - 4 | 27 | 22 | 27 | 17 | 17 | 17 | 69 | 444 % |
| B - 1 | 17 | 22 | 22 | 42 | 27 | 22 | 68 | 387 % |
| B - 2 | 22 | 32 | 22 | 17 | 22 | 17 | 72 | 427 % |
| C - 1 | 32 | 27 | 17 | 27 | 22 | 22 | 74 | 384 % |
| C - 2 | 22 | 27 | 17 | 17 | 17 | 17 | 67 | 475 % |
| D - 1 | 17 | 17 | 22 | 17 | 17 | 22 | 77 | 485 % |
| D - 2 | 22 | 27 | 17 | 17 | 32 | 22 | 32 | 440 % |
| Hémisphères | | | | | | | | |
| B - 1 | 22 | 27 | 22 | 22 | 22 | 17 | 75 | 416 % |
| B - 2 | 17 | 17 | 17 | 37 | 27 | 22 | 86 | 425 % |
| C - 1 | 32 | 17 | 17 | 27 | 17 | 22 | 70 | 435 % |
| C - 2 | 17 | 17 | 17 | 22 | 22 | 37 | 76 | 439 % |
| D - 1 | 37 | 17 | 27 | 27 | 17 | 27 | 44 | 398 % |
| Gains par métrique | | | | | | | | Gain global |
| | 471 % | 476 % | 496 % | 473 % | 466 % | 436 % | 156 % | 425 % |

Table 6.2 - Illustration du gain obtenu pour l'estimation du coefficient de Poisson à l'aide d'une optimisation par recuit simulé. Le nombre d'évaluations de la fonction d'erreur (simulation mécanique + rendu vidéo + comparaison vidéo) et reporté pour chaque expérimentation et chaque métrique (1 = Silhouette, 2=Différence absolue, 3= Différence au carré, 4=Différence absolue & Silhouette, 5=Différence au carré & Silhouette, 6=Flou, 7= Corrélation inverse). Les gains moyens obtenus par comparaison avec une étude exhaustive de 100 évaluations sont calculés par expérimentation et par métrique. Le gain moyen global est de 425 %.

En moyenne l'estimation optimisée par recuit simulé est plus de quatre fois plus rapide que la recherche exhaustive (100 itérations) pour un résultat de précision similaire. On peut noter également que globalement, le gain obtenu pour la métrique « corrélation inverse » est plus faible que pour les autres métriques. Cela est dû au fait que les courbes de réponse pour cette métrique sont beaucoup plus perturbées que celles des autres métriques, ce qui induit une convergence moins rapide du recuit simulé.

6.2.2 Éponges

Nous lançons tout d'abord une recherche exhaustive sur l'éponge 1 (cf Fig. 6.14). Nous définissons « à la main » une zone dans laquelle, à notre avis, se trouve l'information importante (rétractation de la matière). Cette zone est affichée par surbrillance en jaune. Le calcul des métriques n'est effectué que sur cette zone, comme le montrent les deux images dans le cadre en haut à droite. Ces erreurs résiduelles sont obtenues pour les valeurs extrêmes de l'intervalle de validité ($\nu = 0.0$ et $\nu = 0.49$) avec la métrique « différence absolue & silhouette » (qui nous semble être la meilleure pour déterminer les différences à l'œil nu). On peut voir sur ces deux images que la différence reste relativement importante dans les deux cas (bande orangée épaisse). Or, puisque les métriques convergent pour la plupart vers 0.0, nous pensons qu'il est possible d'être dans un cas de matériau « auxétique » ($\nu < 0.0$, cf 2.4.2.2). Nous refaisons donc l'estimation pour ce même cas mais en élargissant l'intervalle de recherche aux matériaux *auxétiques* ($-1.0 \leq \nu < 0.5$).

Le résultat obtenu est alors toujours négatif (matériau auxétique) et on peut voir sur l'image de l'erreur résiduelle de la figure 6.15 que la « bande orangée » a disparu. Nous retenons ici la valeur produite par la métrique ayant obtenu les meilleurs résultats lors de

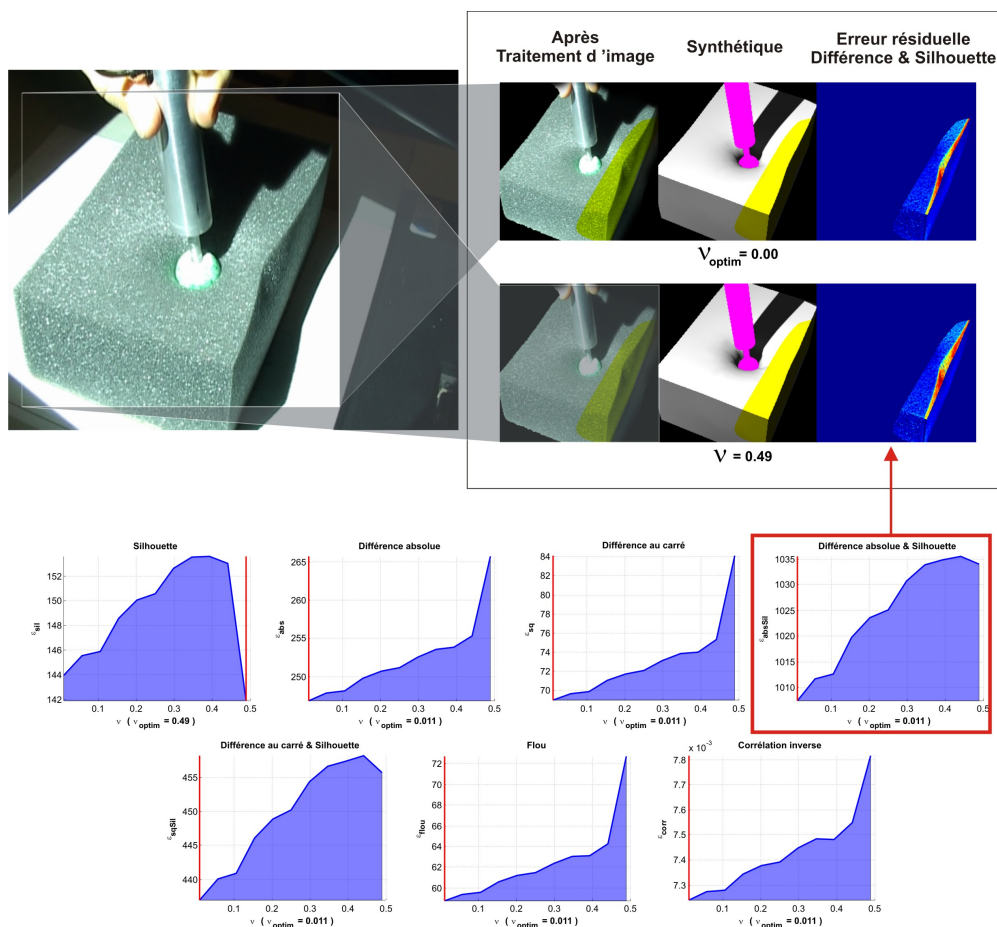


Figure 6.14 - Résultats des estimations du coefficient de Poisson obtenus sur le premier test de l'éponge 1 (recherche classique). Seule la partie en surbrillance jaune est utilisée pour réaliser les estimations.

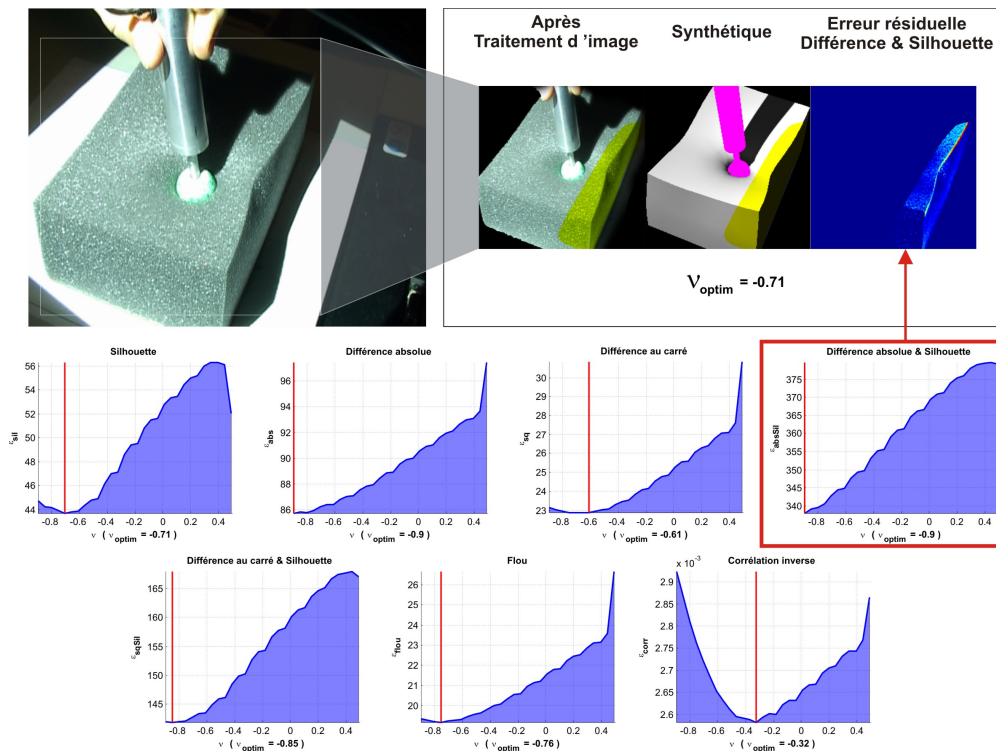


Figure 6.15 - Résultats des estimations du coefficient de Poisson obtenus sur le second test de l'éponge 1 (recherche auxétisme). Seule la partie en surbrillance jaune est utilisée pour réaliser les estimations.

l'estimation du coefficient de Poisson sur les objets en silicone (« Silhouette »), soit $\nu_{\text{éponge1}} = -0.71$. D'autre part, nous n'utilisons actuellement pas de textures lors du rendu des vidéos synthétiques, ce qui introduit un biais lors de l'évaluation des métriques de surface dans le cas où les objets montrent beaucoup d'informations visuelles comme les trous des éponges. Ce résultat demandera cependant validation (cf 7.2). En effet, le phénomène de rétraction semble nettement plus visible lors des grandes déformations. Bien que nous simulions ces grandes déformations à l'aide d'une formulation corotationnelle, il est possible qu'au delà d'une certaine limite, l'effet induit soit insuffisant lorsque $\nu > 0$ et qu'un phénomène similaire apparaisse lorsque $\nu < 0$. Cependant, bien que les matériaux auxétiques soit rares dans la nature, les matériaux synthétiques ainsi que certaines éponges naturelles peuvent très bien reproduire ce genre de comportement. Les estimations suivantes sont donc toutes effectuées sur l'intervalle complet des valeurs possibles pour le coefficient de Poisson.

Nous réalisons un test semblable sur l'éponge 2 (cf Fig. 6.16). Cette fois, la métrique « Silhouette » donne $\nu_{\text{éponge2}} = 0.36$. Sur cet exemple, il est difficile de juger de la pertinence du résultat car la déformation induite n'est pas très visible. Cependant, hormis au niveau des angles de l'objet, la différence de silhouettes semble relativement faible sur l'erreur résiduelle.

Comme nous le faisons pour le test sur l'éponge 1, nous définissons manuellement une zone d'intérêt pour le test effectué sur l'éponge 3 (cf Fig. 6.17). Cette fois, toutes les métriques tendent vers un résultat proche de $\nu_{\text{éponge3}} \approx -0.3$ (auxétique). Nous validons donc là aussi le résultat de la métrique « Silhouette » avec $\nu_{\text{éponge3}} = -0.37$.

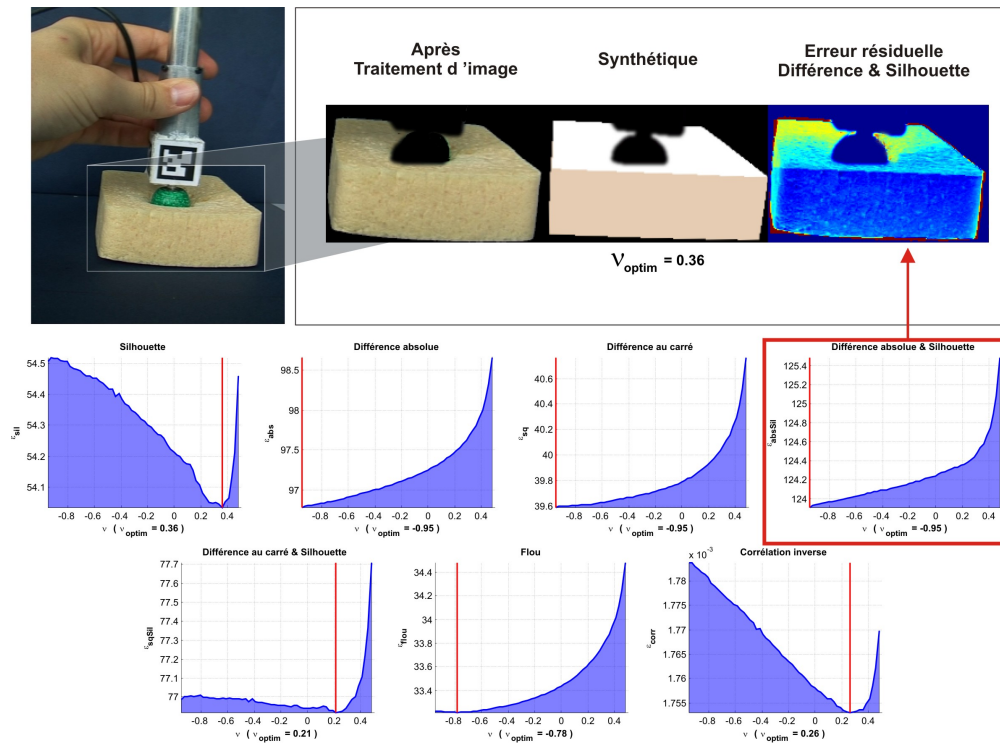


Figure 6.16 - Résultats des estimations du coefficient de Poisson obtenus sur l'éponge 2.

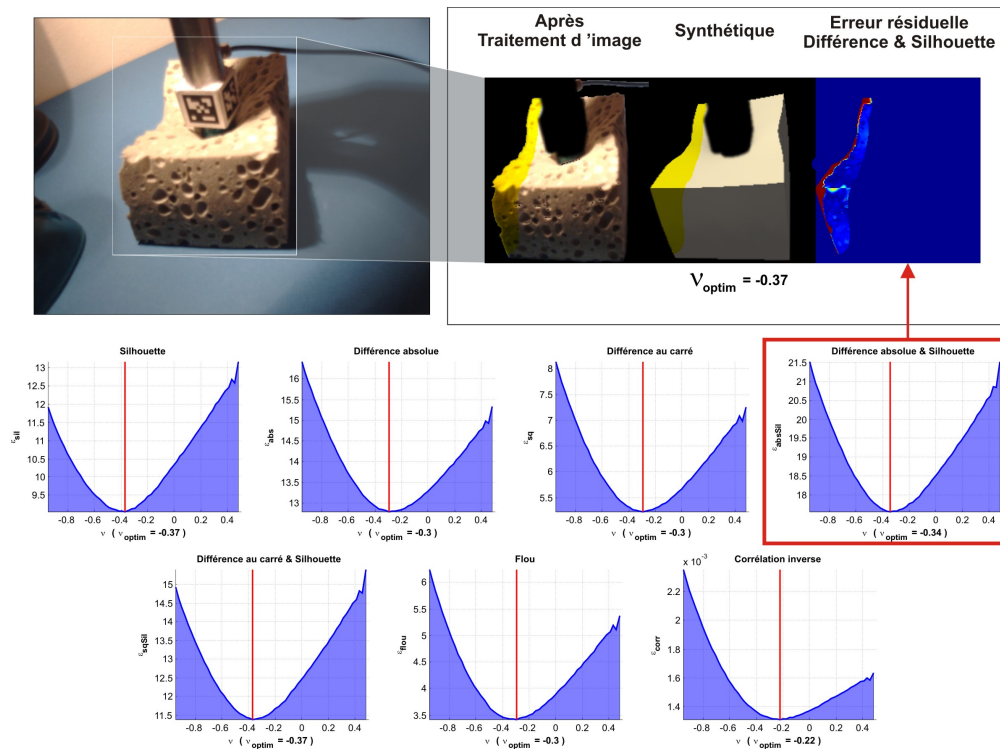


Figure 6.17 - Résultats des estimations du coefficient de Poisson obtenus sur l'éponge 3. Seule la partie en surbrillance jaune est utilisée pour réaliser les estimations.

6.2.3 Peluches

Problème de densité Réaliser des expérimentations sur un objet ayant une géométrie plus complexe s'avère difficile. Le problème majeur que nous rencontrons ici est la densité de l'objet virtuel (ou résolution). En effet, pour obtenir un objet volumique composé de tétraèdres, nous utilisons un logiciel qui se base sur la surface de l'objet (version d'évaluation de « GiD v9.0.4 », © CIMNE ¹). Il faut donc modéliser l'objet surfacique puis lancer le maillage tétraédrique. Or, il s'avère que plus la surface de l'objet est complexe, plus il est nécessaire d'utiliser de tétraèdres ². Or, au delà de quelques milliers de tétraèdres (≈ 4000), la simulation mécanique n'est plus en mesure d'allouer l'espace mémoire nécessaire sur le système 32 bits que nous utilisons. D'autre part, ce maillage doit tout de même être relativement dense pour permettre à la simulation mécanique de calculer convenablement les interactions entre l'outil et le maillage (éviter que la touche sphérique passe « au travers » de l'objet par exemple). Il est donc nécessaire de faire un compromis entre la taille de l'objet à tester et la résolution du modèle géométrique.

Pour résoudre ces problèmes lors des expérimentations sur les peluches, nous faisons le choix de ne modéliser que certaines parties du corps. Ainsi, pour la première peluche, nous faisons l'expérimentation sur la queue uniquement, en faisant l'hypothèse que l'influence de la jonction avec le reste du corps reste relativement faible. Cette méthode offre également un autre avantage : il est alors possible de considérer chaque partie comme homogène car l'assemblage et le rembourrage des peluches semble, au toucher, être différent pour chacune de ces parties.

L'expérimentation sur la peluche 1 s'effectue donc sur la queue du crocodile (cf Fig. 6.18). Les métriques de silhouette ainsi que la corrélation inverse tendent à montrer que le coefficient de Poisson de cette partie de la peluche est $\nu_{peluche1} = 0.33$. La faible erreur résiduelle et l'écart relativement important entre les erreurs maximales et minimales renforce la confiance dans cette estimation (> 4 pour la différence de silhouettes).

L'expérimentation sur la peluche 2 s'effectue quant à elle sur la patte du rhinocéros (cf Fig. 6.19). Les métriques de silhouette convergent vers un coefficient de Poisson aux alentours de $\nu_{peluche2} \approx 0.4$ alors que les métriques de surfaces tendent vers $\nu_{peluche2} \approx 0.2$. Là encore, nous faisons confiance à la métrique de silhouette car on peut voir sur l'erreur résiduelle que l'erreur de surface est relativement forte, notamment à cause du manque de texture de l'objet simulé. Nous validons donc $\nu_{peluche2} = 0.41$.

1. <http://gid.cimne.upc.es/>

2. Ce problème est rencontré avec d'autres logiciels. Nous avons également testé « GMesh » et « Netgen ».

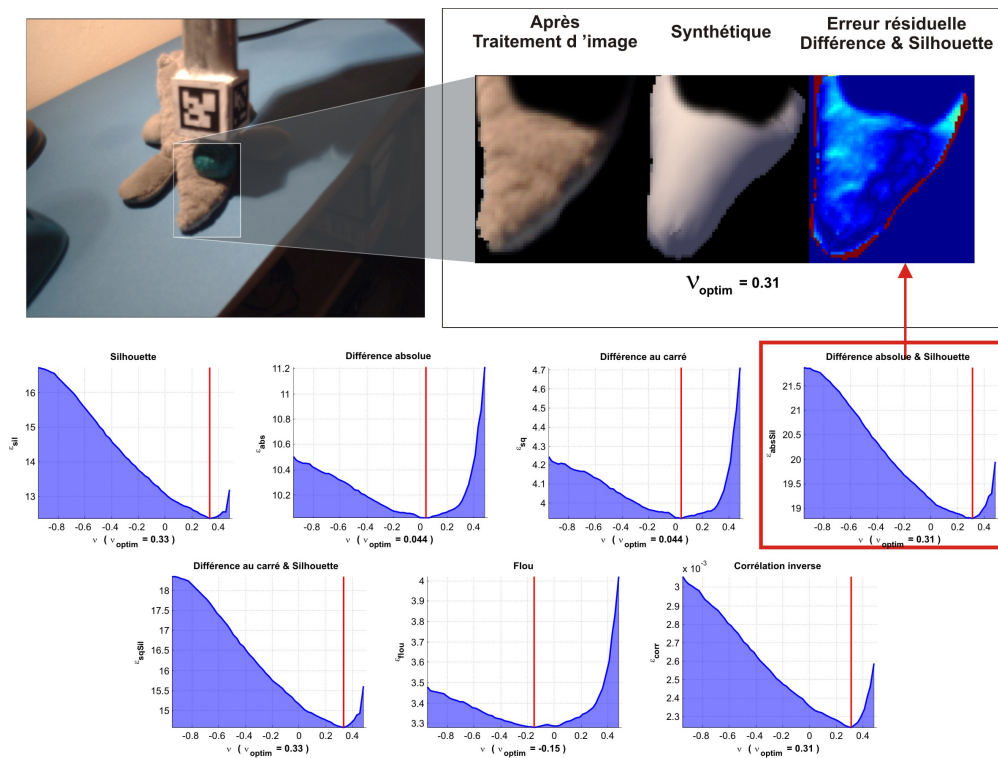


Figure 6.18 - Résultats des estimations du coefficient de Poisson obtenus sur la queue de la première peluche.

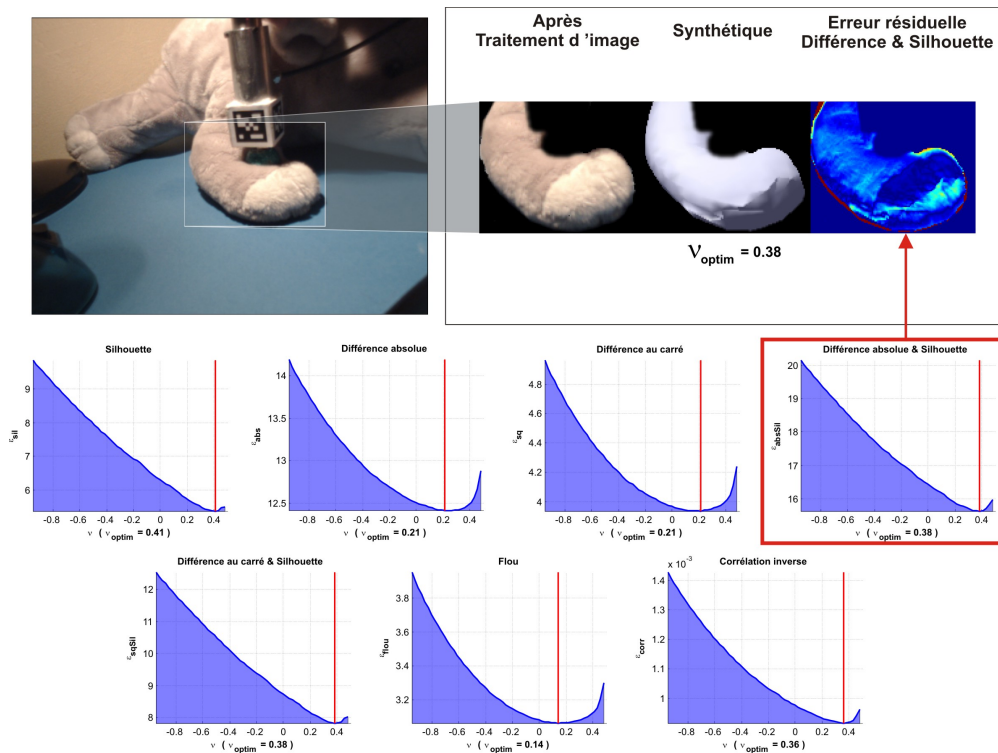


Figure 6.19 - Résultats des estimations du coefficient de Poisson obtenus sur la patte de la seconde peluche.

6.3 Estimations du module de Young

6.3.1 Silicones

Le tableau 6.3 présente les résultats obtenus lors des estimations du module de Young pour les objets en silicone, séparés en deux catégories : cylindres et hémisphères. La première colonne donne le code de l'expérimentation. « A - 1 » signifie le premier test sur le silicone A. Le résultat de l'estimation directe du module de Young, décrite en 5.5.1, se trouve dans la deuxième colonne ($E_{directe}$). L'unité est le Pascal. La troisième colonne donne le résultat de l'estimation inverse ($E_{inverse}$, cf 5.5.2). Enfin, le nombre d'itérations nécessaires pour réaliser cette dernière estimation se trouve dans la dernière colonne.

| Exp. | $E_{direct}(Pa)$ | $E_{inverse}(Pa)$ | Itérations |
|--------------------|------------------|-------------------|------------|
| Cylindres | | | |
| A - 1 | 134 295 | 150 000 | 6 |
| A - 2 | 107 305 | 132 981 | 6 |
| A - 3 | 97 433 | 141 836 | 8 |
| A - 4 | 77 243 | 146 052 | 8 |
| B - 1 | 64 147 | 87 841 | 9 |
| B - 2 | 95 653 | 81 357 | 8 |
| C - 1 | 31 371 | 82 919 | 6 |
| C - 2 | 62 842 | 83 061 | 6 |
| D - 1 | 41 255 | 63 614 | 6 |
| D - 2 | 33 284 | 55 046 | 8 |
| Hémisphères | | | |
| B - 1 | 62 792 | 74 446 | 7 |
| B - 2 | 34 659 | 81 386 | 7 |
| C - 1 | 71 715 | 83 645 | 7 |
| C - 2 | 73 208 | 86 719 | 7 |
| D - 1 | 39 945 | 57 156 | 6 |

Table 6.3 - Tableau récapitulatif des résultats d'estimation du module de Young pour les objets en silicone. La seconde colonne donne le résultat de l'estimation du module de Young par méthode directe (cf 5.5.1) pour chaque expérimentation (première colonne). La troisième colonne indique le résultat obtenu par estimation inverse (cf 5.5.2, le nombre d'itérations est donné dans la dernière colonne).

L'estimation directe donne visiblement des résultats peu convaincants et très bruités (grosses variations entre deux estimations pour un même échantillon). Comme nous le pensions, les diverses hypothèses sur la forme de l'objet et le point d'application (cf 5.5.1) biaisent ces estimations. Néanmoins, elle permettent de donner rapidement une idée globale de l'échelle de la valeur, par exemple pour les cylindres A-1 et A-2 où elle est nettement plus forte que pour le cylindre D-2 ou l'hémiphère D1.

L'estimation inverse donne quant à elle des résultats qui semblent bien meilleurs et plus stables. Pour un même mélange, les estimations sont relativement proches. En revanche, elle n'est pas assez précise pour distinguer des valeurs proches comme les mélanges B et C. On peut cependant noter que la géométrie de l'objet n'altère pas la qualité de l'estimation réalisée.

La figure 6.20 présente l'erreur résiduelle pour chaque cas après l'estimation inverse. Sur ces schémas, nous tentons d'exprimer la qualité du rendu haptique qui serait ressenti par

l'utilisateur si la simulation mécanique était couplée à un bras à retour d'effort. Les courbes rouges représentent les forces mesurées lors de la capture de l'expérimentation. Les courbes vertes représentent les efforts synthétiques générés lors d'une simulation mécanique utilisant le module de Young estimé par la méthode inverse. La surface bleue représente l'erreur résiduelle (différence au carré). La valeur inscrite au dessus de chaque schéma est la moyenne de cette erreur sur toute l'animation. Ainsi, l'erreur maximale ressentie en moyenne sur une expérimentation n'excède pas $\sqrt{5}N$. En règle générale, celle-ci est d'environ $\sqrt{1.5}N$. Elle semble donc très difficile à repérer haptiquement pour la plupart des utilisateurs. Une étude statistique sur un ensemble d'individus pourrait être réalisée pour évaluer cette sensibilité.

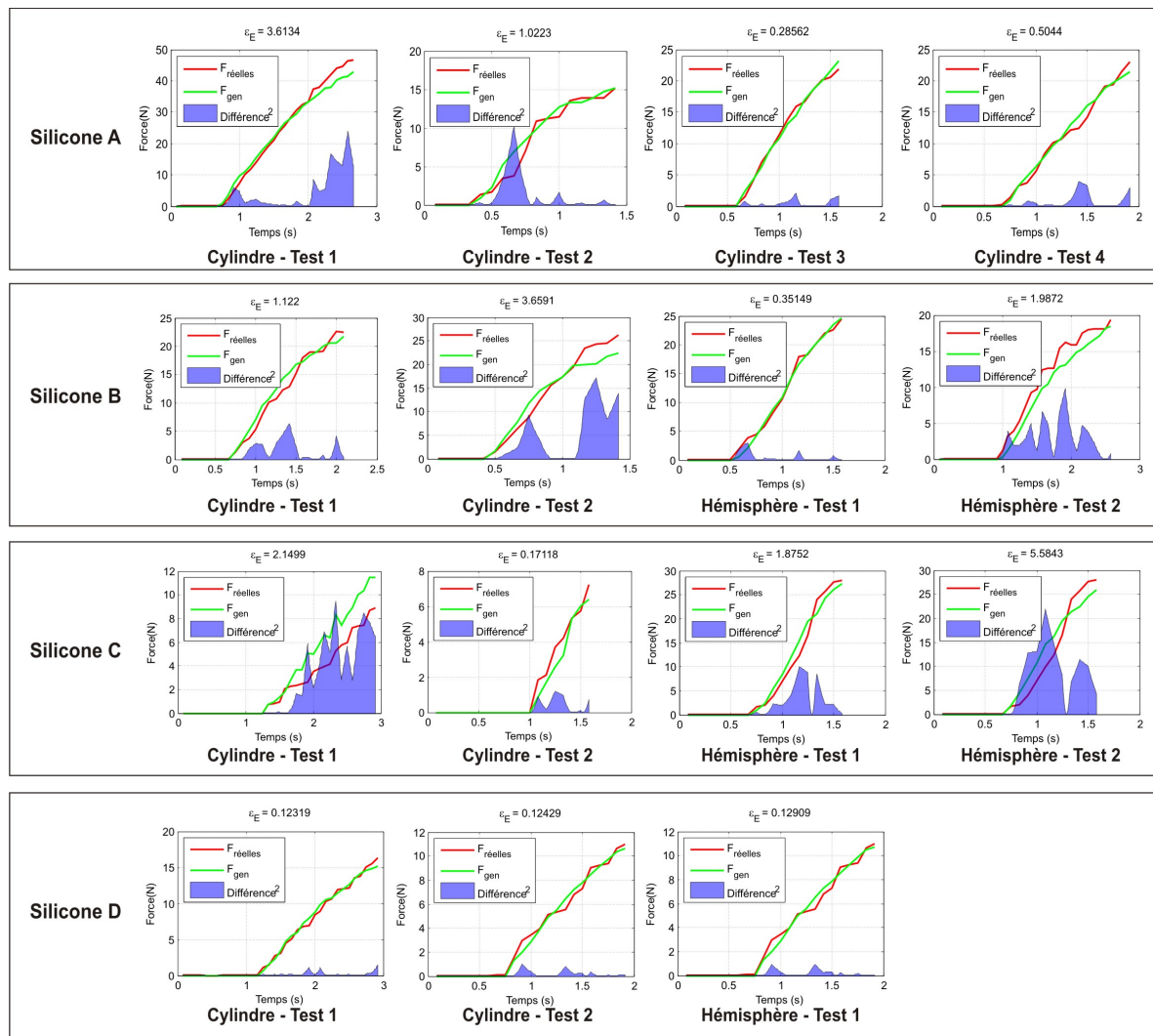


Figure 6.20 - Erreurs résiduelles après l'estimation du module de Young pour les tests sur les objets en silicone. La courbe rouge représente les forces réelles mesurées. La courbe verte représente les forces générées lors de la simulation mécanique optimale. La surface bleue représente l'erreur résiduelle au carré par pas de temps. La valeur inscrite au dessus de chaque schéma est la moyenne de cette erreur sur toute l'animation.

6.3.1.1 Validation par mesures mécaniques

Pour valider les résultats obtenus, nous faisons une comparaison avec les valeurs extraites des données mesurées par la machine « Zwick » (cf 6.1.1). Les résultats du calcul de la précision des estimations réalisées par la méthode directe et la méthode inverse se trouvent dans le tableau 6.4. L'erreur moyenne est calculée par méthode (directe ou inverse) et par type d'objet (cylindre ou hémisphère). Comme nous le supposons, la méthode directe présente une erreur moyenne d'estimation de 31,21% ce qui est suffisant pour avoir rapidement une idée de l'échelle du module de Young. En revanche, l'estimation par méthode inverse évalue le module de Young avec une erreur moyenne de seulement 5,12%.

| Erreur moyenne | Estimation directe | Estimation inverse |
|--------------------|--------------------|--------------------|
| Cylindres | 33,30 % | 4,79 % |
| Hémisphères | 26,61 % | 5,85 % |
| Globale | 31,21 % | 5,12 % |

Table 6.4 - Évaluation de la précision des estimations. Le calcul de l'erreur est réalisé par comparaison avec les modules de Young mesurés par la machine « Zwick » ($E_{siliconeA} = 150505Pa$, $E_{siliconeB} = 84317Pa$, $E_{siliconeC} = 81162Pa$, $E_{siliconeD} = 54876Pa$, cf 6.1.1).

Plusieurs facteurs peuvent être à l'origine de ces erreurs. Tout d'abord, il est probable que les grandes déformations en jeu dans les mesures réalisées faussent la simulation mécanique. En effet, la formulation corotationnelle est purement géométrique et n'intervient pas dans les efforts générés. L'hypothèse de petits déplacements de la théorie d'élasticité linéaire est alors violée dans ces expérimentations. Une étude future consisterait à n'utiliser que les petits déplacements en jeu pour l'évaluation des différences de forces. D'autre part, il est probable que le capteur d'effort dérive dans le temps et que le calibrage effectué au début des expérimentations ne soit plus valide aux cours des différentes manipulations. Il en résulterait donc une mesure erronée qui conduirait à une sous-évaluation ou sur-évaluation du module de Young. Cependant, ce résultat, bien qu'insuffisant pour discriminer les deux échantillons proches (mélanges B et C) reste selon nous suffisant pour reproduire fidèlement, en utilisant un dispositif haptique, les comportements d'objets déformables relativement mous comme les éponges, les peluches ou encore les organes.

6.3.2 Éponges

La figure 6.21 présente les résultats obtenus pour les estimations du module de Young par méthode directe et par méthode inverse pour les trois éponges. La première méthode, par estimation directe (cf 5.5.1), est illustrée dans la colonne de gauche. La seconde, par estimation inverse (cf 5.5.2), est illustrée dans les deux colonnes de droite : l'erreur évaluée par itération du processus inverse puis l'erreur résiduelle lors de la comparaison des forces avec le résultat trouvé, $E_{inverse}$. Les résultats de l'estimation inverse semblent cohérents avec une évaluation manuelle par pression au toucher : les éponges 1 et 3 semblent avoir un comportement semblable alors que l'éponge 2 paraît plus « dure ». En revanche, sur ces exemples, l'estimation directe semble donner des résultats incohérents. Une des raisons probables est que pour les éponges 1 et 3, la déformation imposée est relativement importante, ce qui peut conduire à des problèmes car l'hypothèse de petits déplacements est également violée. Par conséquent, la partie non linéaire dans l'espace contrainte/déplacement fausse ces estimations directes.

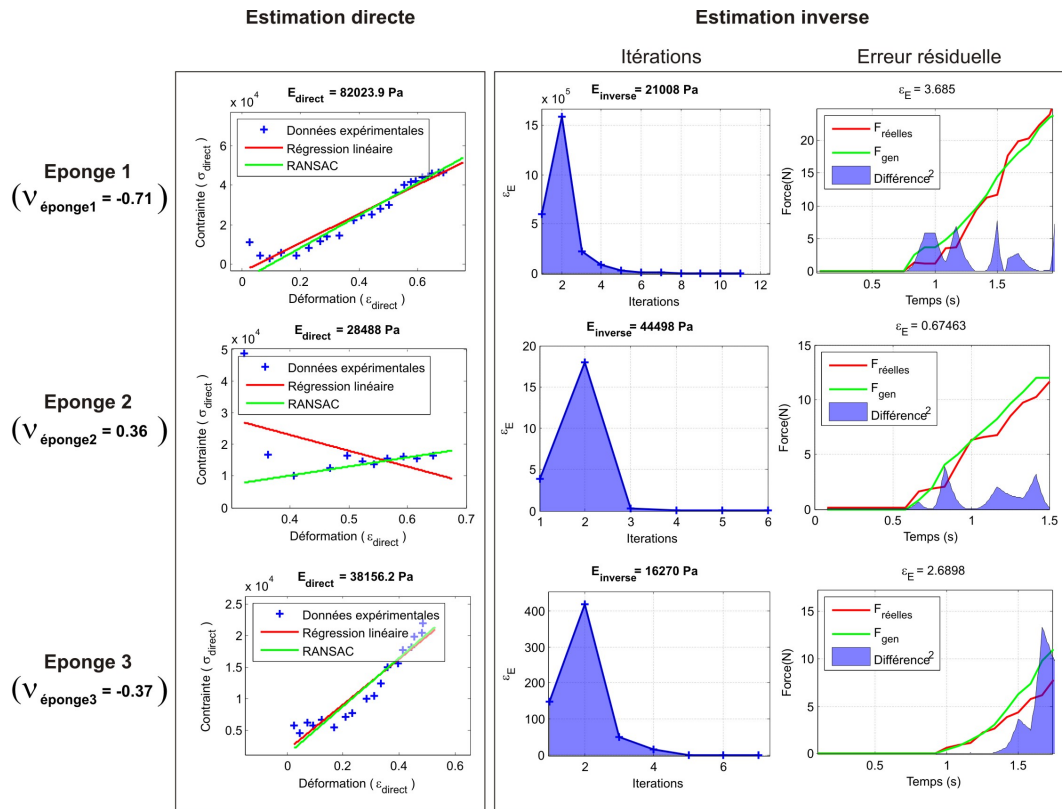


Figure 6.21 - Résultats des estimations directe et inverse du module de Young pour les trois éponges. La première colonne montre l'extraction du module de Young par la méthode directe (E_{directe} , cf 5.5.1). La seconde montre les itérations nécessaires à l'estimation inverse et l'erreur à chaque itération (E_{inverse} , cf 5.5.2). La dernière colonne montre l'erreur résiduelle lors de la comparaison de forces obtenue avec E_{inverse} . Les estimations sont réalisées avec la valeur du coefficient de Poisson évaluée en 6.2 pour chaque éponge.

Nous nous basons donc sur les résultats obtenus pour le silicone pour accorder notre confiance dans le résultat des estimations inverses, d'autant que les erreurs résiduelles sont relativement faibles ($< \sqrt{3.6N}$) et que l'algorithme converge très rapidement (respectivement en 11, 6 et 7 itérations seulement). Nous validons donc $E_{\text{éponge1}} = 21008Pa$, $E_{\text{éponge2}} = 44498Pa$, et $E_{\text{éponge3}} = 16270Pa$. Nous notons ici que des tests ultérieurs effectués avec la même méthode donneraient des résultats probablement différents car l'exposition à l'air et l'humidité ambiante modifie les propriétés mécaniques des éponges.

6.3.3 Peluches

Les mêmes expérimentations sont réalisées pour les peluches (cf Fig. 6.22). Là encore, l'estimation inverse converge rapidement vers une solution à erreur résiduelle faible (8 itérations pour une erreur $< \sqrt{3.3N}$, $E_{\text{peluche1}} = 28471Pa$, et $E_{\text{peluche2}} = 14868Pa$).

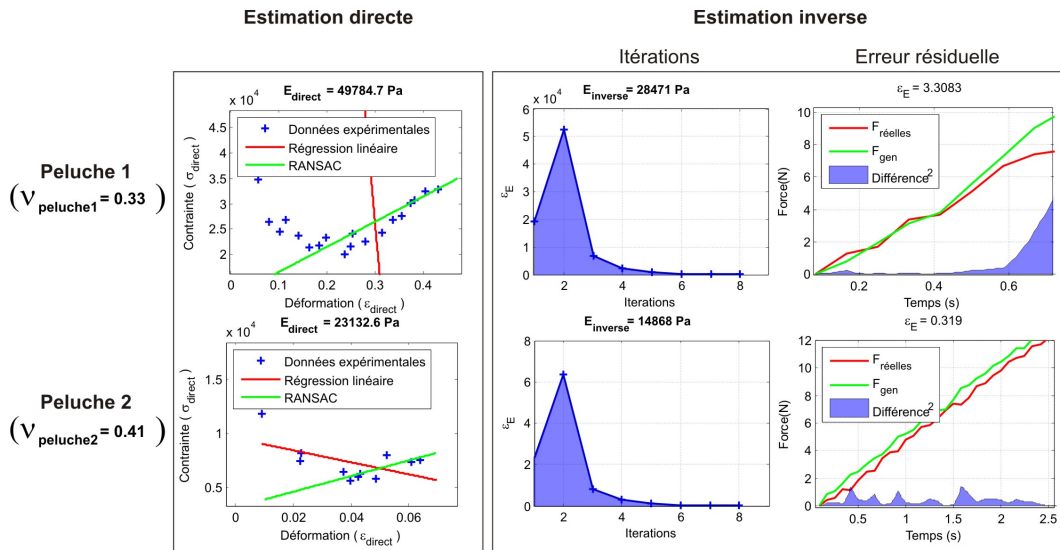


Figure 6.22 - Résultats des estimations directe et inverse du module de Young pour les deux peluches. La première colonne montre l'extraction du module de Young par la méthode directe ($E_{directe}$, cf 5.5.1). La seconde montre les itérations nécessaires à l'estimation inverse et l'erreur à chaque itération ($E_{inverse}$, cf 5.5.2). La dernière colonne montre l'erreur résiduelle lors de la comparaison de forces obtenue avec $E_{inverse}$. Les estimations sont réalisées avec la valeur du coefficient de Poisson évaluée en 6.2 pour chaque peluche.

6.4 Bilan et réutilisation

Le tableau 6.5 présente un résumé des estimations du coefficient de Poisson et du module de Young réalisées dans ces travaux.

| Matériau | Coefficient de Poisson | Module de Young |
|----------------------------------|------------------------|-----------------|
| Silicone, mélange A | 0.49 | 150 505 Pa |
| Silicone, mélange B | 0.49 | 84 317 Pa |
| Silicone, mélange C | 0.49 | 81 163 Pa |
| Silicone, mélange D | 0.49 | 54 876 Pa |
| Eponge 1, « Cimentier » | -0.71 | 21 008 Pa |
| Eponge 2, « Végétale » | 0.36 | 44 498 Pa |
| Eponge 3, « Gros travaux » | -0.37 | 16 270 Pa |
| Peluche 1, « Crocodile » (Queue) | 0.33 | 28 471 Pa |
| Peluche 2, « Rhino » (Patte) | 0.41 | 14 868 Pa |

Table 6.5 - Tableau récapitulatif des estimations du coefficient de Poisson et du module de Young.

Ces deux paramètres étant très utilisés en simulation mécanique, nous montrons, sur la figure 6.23 un exemple de réutilisation dans framework de simulation physique dynamique, *SOFA* (*Simulation Open Framework Architecture*)[Allard 07].

Puisqu'ils ne sont pas dépendants de la géométrie de l'objet à partir duquel ils ont été estimés, il est très simple de les appliquer à d'autres objets. La figure 6.24 montre leur utilisation dans *SOFA* mais appliqués à d'autres géométries.

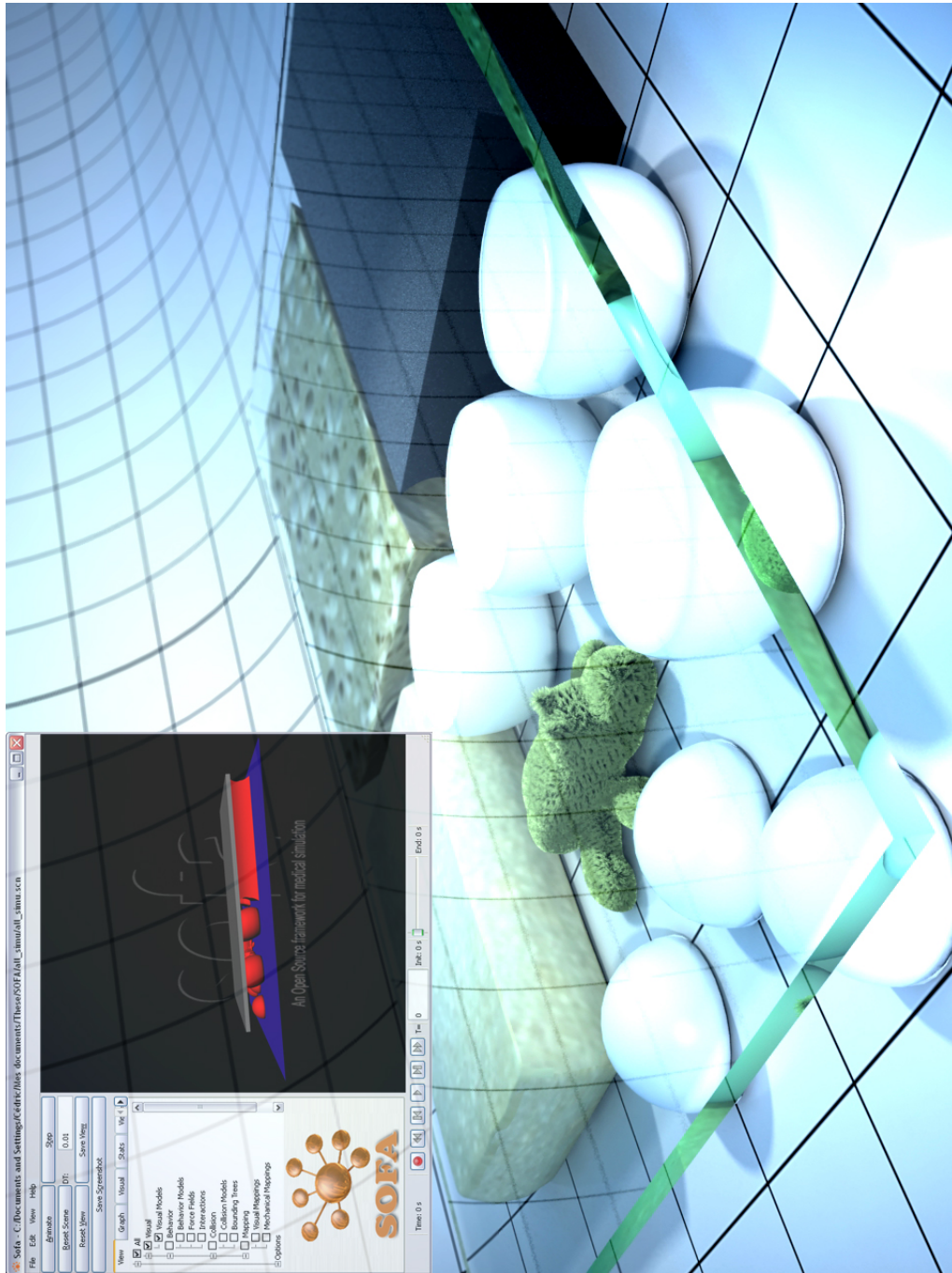


Figure 6.23 - Utilisation des résultats dans un autre moteur de simulation physique dynamique, SOFA (Simulation Open Framework Architecture)[Allard 07]. Dans cette simulation, tous les objets utilisés dans ces travaux sont modélisés (excepté la peluche 2). Les paramètres mécaniques estimés correspondant leur sont attribués (cf tableau 6.5). Une plaque de verre descend et les comprime (Sur cette capture, les hémisphères et la peluche 1 ne sont pas encore comprimées). En haut à gauche se trouve la capture d'écran de la scène dans SOFA.

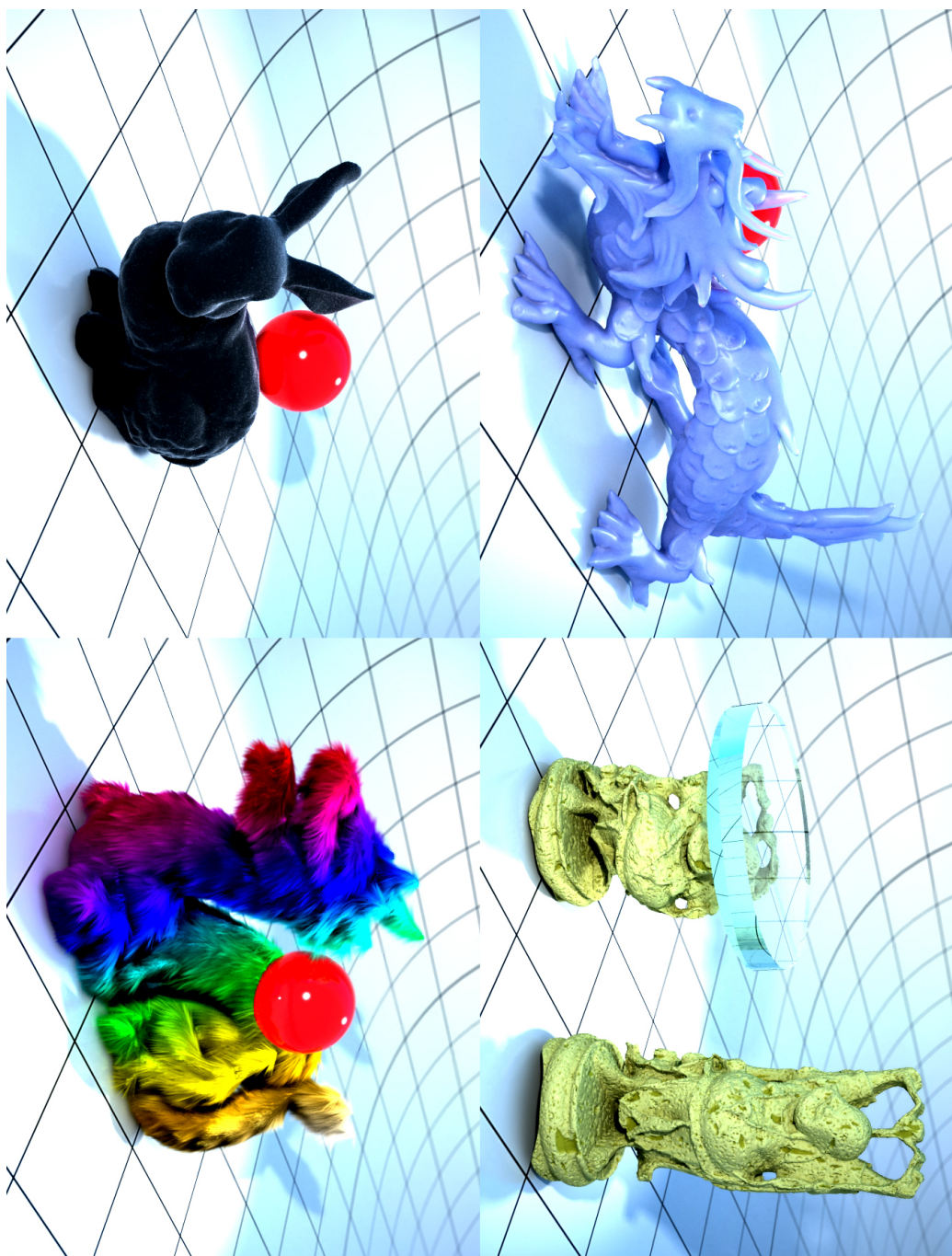


Figure 6.24 - Extrapolation à d'autres modèles géométriques dans SOFA (issus du Stanford Scanning Repository, <http://graphics.stanford.edu/data/3Dscanrep/>). En haut à gauche, utilisation des paramètres du silicone A sur le « Asian Dragon ». En haut à droite, utilisation des paramètres de l'éponge 2 sur le « Happy Buddha ». En bas à gauche, utilisation des paramètres de l'éponge 1 sur le « Stanford Bunny ». En bas à droite, utilisation des paramètres de la peluche 2 sur le « Dragon ».

Septième Chapitre

CONCLUSION ET PERSPECTIVES

7.1 Conclusion

L'objectif de cette thèse était de développer une solution matérielle et logicielle permettant d'estimer des propriétés mécaniques d'objets complexes à partir de séquences d'images. L'ensemble matériel devait être peu onéreux et simple à mettre en œuvre. L'expérimentation devait consister à filmer et mesurer une déformation appliquée sur un objet « mou » par un utilisateur. Nous avons déterminé une solution basée sur un vidéo projecteur standard, une caméra numérique classique et un dispositif de capture de forces. Après une étude des solutions existantes, nous avons conclu qu'aucun dispositif actuel ne remplissait les conditions nécessaires à notre vision du projet. Nous avons donc conçu et fait évoluer un dispositif original de type « stylet », ergonomique et simple d'utilisation, pour mesurer les forces en jeu dans les déformations appliquées par l'utilisateur. De plus, nous avons créé une méthode de calibrage des paramètres intrinsèques et extrinsèques du vidéo projecteur. Grâce à cette méthode, nous avons développé une technique interactive visuelle de positionnement virtuel de la caméra et du vidéo-projecteur dans la scène. Enfin, nous avons développé trois méthodes (magnétique, mécanique et visuelle) pour le positionnement de l'outil interactif de capture de force.

Après avoir dressé un inventaire des propriétés mécaniques des objets déformables, nous avons déterminé que le coefficient de Poisson et le module de Young étaient deux paramètres très largement utilisés dans la simulation mécanique. Parfois difficiles à évaluer « à la main », l'estimation automatique des valeurs de ces paramètres est un enjeu important pour le réalisme des simulations numériques, notamment pour les simulateurs en temps réel. Après avoir prouvé qu'il était possible d'estimer séquentiellement ces paramètres à partir de la séquence vidéo de la déformation et de la mesure des forces en jeu, nous avons développé plusieurs techniques d'estimation.

Tout d'abord, nous avons défini un schéma global d'estimation inverse pour retrouver le coefficient de Poisson à partir de la vidéo de l'objet déformé. Au cœur de notre algorithme se trouve la « métrique d'erreur ». Cette fonction permet de quantifier la différence entre deux séquences vidéo. Nous avons défini huit métriques d'erreur différentes que nous avons testées pour l'évaluation du coefficient de Poisson d'objets déformables en silicone. Le silicone étant un matériau connu, nous avons pu évaluer la précision de chacune de ces métriques dans nos expérimentations.

Nous avons ensuite développé deux techniques d'estimation du module de Young à partir des forces capturées. La première est une méthode d'estimation directe. Tout d'abord basée sur une régression linéaire appliquée sur l'ensemble des échantillons mesurés (*data fitting*), elle a ensuite évolué vers un *RanSaC* qui permet une évaluation plus robuste du module de Young. La seconde méthode est basée sur un schéma itératif d'estimation inverse. Pour les tester, nous avons tout d'abord créé des objets de silicone avec quatre mélanges différents. La « dureté » de ces mélanges était contrôlée grâce à un diluant qui nous a permis d'obtenir, à des concentrations différentes, des objets relativement durs, d'autres relativement mous, et enfin deux autres mélanges assez proches. Coulés dans des moules différents, nous avons obtenu des cylindres et des hémisphères sur lesquels nous avons évalué le module de Young avec les méthodes directe et inverse. Pour valider nos résultats, nous les avons comparés aux mesures réalisées sur les mêmes échantillons par une machine de tests en traction/compression universelle. L'erreur obtenue par la méthode directe, 33%, permet simplement et rapidement d'avoir une idée de l'échelle du module de Young. En revanche, la méthode inverse permet l'évaluation du module de Young avec une erreur d'environ 5% seulement. Bien que celle-ci ne

puisse pas distinguer les différences entre les deux mélanges relativement proches, elle permet de générer des forces synthétiques proches des forces réelles mesurées. Les forces ressenties lors d'un couplage haptique vers un bras à retour d'effort, par exemple dans un simulateur en temps réel, devraient donc être relativement réalistes.

Nous avons ensuite appliqué ces évaluations du coefficient de Poisson et du module de Young à deux autres types de matériaux, les éponges synthétiques et les peluches. Des expérimentations effectuées sur trois éponges différentes permettent de mettre en évidence un comportement auxétique de deux d'entre elles et de discriminer les « raideurs » de celles-ci. Les évaluations que nous avons réalisées sur certaines parties des deux peluches montrent un comportement cohérent des diverses méthodes d'estimation. Nous avons ensuite réutilisé les différents résultats dans un simulateur mécanique différent, temps-réel, pour prouver que notre méthode permet d'extrapoler vers des modèles mécaniques et géométriques différents de ceux que nous avons développés.

7.2 Travaux futurs

Se situant aux confluent de nombreux domaines de la recherche informatique, comme la vision par ordinateur, la simulation mécanique, la synthèse d'image ou encore l'optimisation, ce travail apporte à notre sens une vision globale des méthodes possibles pour faire évoluer l'estimation de propriétés mécaniques d'objets complexes à partir de séquences d'images. Nous n'avons donc aucunement la prétention d'avoir résolu tous les problèmes liés à ce sujet, et soulevons de nombreuses questions, en donnant parfois les pistes que nous suivrions pour les extensions futures.

7.2.1 Dispositif matériel

Plusieurs pistes sont envisageables pour faire évoluer la solution matérielle. Tout d'abord, un des enjeux est de retrouver ces deux paramètres uniquement à partir de la vidéo. Or, comme nous l'avons expliqué, le module de Young n'a d'effet, en simulation quasi-statique, que sur les forces de réponse du matériel. Par conséquent, son influence n'est pas visible et donc impossible à déterminer depuis la vidéo uniquement. Une solution pourrait donc être de transformer visuellement cette information, par exemple en utilisant un ressort à la place du capteur de force. En plaçant deux marqueurs visuels à chaque extrémité du ressort, il serait possible de déterminer le facteur de compression de celui-ci et, en choisissant un ressort de raideur connu, d'en extraire la force appliquée à chaque image de la séquence vidéo. L'avantage principal serait le coût car un ressort est beaucoup moins onéreux qu'un capteur de forces. Cependant, la précision d'un tel dispositif ne serait peut-être pas suffisante et plusieurs ressorts seraient nécessaires. De plus, cela impliquerait que le ressort lui-même soit visible et donc, implicitement, que moins de pixels seraient dédiés à la déformation de l'objet.

Une autre piste pour le dispositif de capture de force serait de s'interfacer sur le dispositif de Vuskovic. En effet, celui-ci fonctionne par succion et définit des conditions limites propres et connues, mais au prix d'une hypothèse sur le pré-stress induit par l'appui sur l'organe. Rajouter un capteur de force permettrait de mieux caractériser ce pré-stress. Un tel outil pourrait alors être plus précis pour la mesure de paramètres mécaniques sur des organes.

Il serait également intéressant de déterminer l'erreur de positionnement de l'outil de mesure. En effet, les trois méthodes, magnétique, mécanique et visuelle demandent une validation. Une procédure possible serait de créer un dispositif hybride, comprenant les trois outils de positionnement plus un ensemble de mesure de référence précis, par exemple la capture laser. Il suffirait alors de comparer les positions obtenues par nos trois méthodes avec celle du dispositif de référence.

Certains effets visuels sont parfois gênants lors des étapes de traitement d'image. Par exemple, les ombres gênent le détourage automatique. Rajouter d'autres sources de lumière pourrait résoudre ce problème et améliorer la qualité générale de la vidéo capturée. En revanche, cela nécessiterait une étape de calibrage supplémentaire.

7.2.2 Estimation du coefficient de Poisson

Lors de nos estimations, nous avons remarqué que nos métriques sont relativement sensibles. Par exemple, une erreur de positionnement de l'objet est souvent source de mauvaise estimation des métriques de silhouette. Nous avons donc porté un soin particulier à posi-

tionner correctement l'objet avant d'appliquer la déformation. Une étude de sensibilité à ce problème serait par ailleurs intéressante. D'autre part, une limitation de notre simulation mécanique perturbe également, dans certaines expérimentations, les métriques de silhouette. Il s'agit du « décollement » observé sur les objets déformés : la base en contact avec le support se soulève parfois, surtout lors de grandes déformations. Or, nous simulons ce contact par des conditions de Dirichlet où les points sont fixes et ne peuvent se déplacer. Nous avons dans un premier temps résolu ce problème en collant les objets réels sur un support planaire. Mais il serait préférable de rajouter une contrainte « plan » dans la simulation mécanique.

Plus de facteurs viennent perturber le calcul des métriques de surface, comme la position de la source lumineuse, de l'outil, ou encore les propriétés de reflectance propres de l'objet. Un problème pour ces métriques est notamment la texture des objets. Actuellement, nous ne texturons pas les objets virtuels ce qui introduit également un biais lors du calcul de ces métriques. Rajouter une texture sur ces objets, par exemple de manière semi-automatique en projetant la première image capturée (sans déformation) sur l'objet en trois dimensions, permettrait sans doute d'améliorer le résultat de ces métriques.

Un autre phénomène peut perturber ces métriques. Sur les objets en silicones par exemple, nous avons pu constater de nombreux éclats spéculaires. Si ceux-ci ne sont pas correctement reproduits lors du rendu, un biais est introduit dans le calcul de la différence absolue ou encore la différence au carré. Une approche serait donc de tenter d'éliminer ces effets « néfastes » en projetant des motifs spéciaux. Nous pourrions nous baser sur des travaux de *shape from shading* qui consistent à estimer une surface en trois dimensions à partir de son éclairage [Horn 89] [Haddon 98]. Une autre approche par projection de motifs de couleurs structurés pourrait être encore plus robuste [Zhang 02]. L'approche inverse pourrait également être adoptée. En effet, ce type de phénomène pourrait nous permettre, s'ils étaient maîtrisés, de déterminer une métrique d'erreur précise car les éclats spéculaires semblent contenir beaucoup d'informations sur la déformation. Il serait alors nécessaire de bien connaître les fonctions de reflectance de l'objet, par exemple par une passe de rendu inverse [Yu 99] [Boivin 01]. Cela ne fonctionnerait cependant que sur des objets ayant un comportement spéculaire important, comme par exemple les organes.

7.2.3 Estimation du module de Young

Plusieurs pistes permettent d'améliorer nos méthodes d'estimation du module de Young. Tout d'abord, il serait utile de lancer une étude en refaisant toutes les estimations réalisées dans ces travaux mais en se limitant aux petites déformations. D'autre part, nous pensons qu'il est judicieux d'améliorer la qualité de la méthode d'estimation directe. Nous proposons pour cela quelques pistes simples. La première serait de faire une meilleure estimation de la surface de contact avec l'objet, par opération booléenne par exemple (intersection sphere/objet initial). Une autre piste pour évaluer cette surface de contact serait de faire une seule simulation mécanique en utilisant le coefficient de Poisson retrouvé au préalable. Toujours pour améliorer une connaissance à-priori de la géométrie de l'objet, il serait possible de faire un lancer de rayon pour estimer la longueur initiale de l'objet dans la direction de l'outil. Enfin, la prise en compte de la friction améliorerait sans doute l'estimation finale [Zhang 97].

Comme nous l'avons suggéré, une validation des estimations actuelles pourrait être effectuée par une étude statistique sur un groupe d'individus. Il s'agirait de se baser sur l'expérience de Leskovsky (*et al.*) [Leskovsky 06]. L'utilisateur tient un dispositif à retour d'effort.

Un mécanisme rigide le relie à une partie cachée où on positionne les échantillons. Il est ainsi possible soit d'utiliser une « palpation » directe mais cachée des échantillons, soit de renvoyer des efforts par le biais du bras à retour d'effort. L'utilisateur doit alors répondre à la question « vrai ou synthétique », ce qui permet d'évaluer la qualité du ressenti haptique et donc de également de l'estimation du module de Young dans le cas de simulateurs en temps-réel.

Enfin, nous proposons une approche totalement différente pour estimer le module de Young directement depuis la vidéo, sans dispositif de capture de force. Il s'agirait d'utiliser une simulation mécanique dynamique. L'expérimentation pourrait alors être par exemple de laisser tomber une bille de poids connu sur l'échantillon et de mesurer, grâce à la vidéo, la hauteur du rebond. Il serait alors possible de calculer la force de la contrainte (gravité et poids de la bille) ainsi que la force de réponse (gravité, poids, hauteur). Plusieurs billes permettraient de tester des échelles différentes et, pour éviter des rebonds intempestifs sur des objets complexes géométriquement, de placer ces billes dans des tubes transparents. Il serait cependant nécessaire de bien synchroniser le moment du lâcher de la bille (dans l'expérimentation réelle et dans la simulation mécanique), par exemple en le commandant informatiquement à l'aide d'un électro-aimant.

Bibliographie personnelle

Journaux internationaux

- [Syllebranque 08] Cédric Syllebranque & Samuel Boivin. *Estimation of mechanical parameters of deformable solids from videos*. The Visual Computer : International Journal of Computer Graphics, vol. 24, no. 11, pages 963-972, october 2008.

Conférences internationales

- [Syllebranque 07a] Cédric Syllebranque & Samuel Boivin. *Estimation of Mechanical Parameters of Deformable Solids from Videos*. 4th International INTUITION Conference, Athens, October 2007.
- [Syllebranque 07b] Cédric Syllebranque, Samuel Boivin, Christian Duriez & Christophe Chaillou. *Estimation of Hookean Parameters of Deformable Bodies from Real Videos*. Cyberworlds International Conference - Haptex Workshop, pages 330-337, Hannover, October 2007.
- [Syllebranque] Cédric Syllebranque & Christian Duriez *A training simulator for dental implantology* . 5th International Symposium on Biomedical Simulation, Phoenix, Arizona, USA January 23 - 24, 2010 (à paraître).

Autres

- [Syllebranque 04b] Cédric Syllebranque & Samuel Boivin. *Étude d'un simulateur mécanique de tissus à partir de séquences d'images synthétiques*. AFIG'04 - Actes des 17èmes journées de l'Association Française d'Informatique Graphique, pages 123-132, 2004.
- [Syllebranque 04a] Cédric Syllebranque. *Estimation de propriétés mécaniques d'un modèle de tissus à partir de séquences d'images*. Master's thesis, Université des Sciences et Technologies de Lille, 2004.

Bibliographie

- [Allard 07] Jérémie Allard, Stéphane Cotin, François Faure, Pierre-Jean Bensoussan, François Poyer, Christian Duriez, Hervé Delinquette & Laurent Grisoni. *SOFA, an Open Source Framework for Medical Simulation*. In *Medicine Meets Virtual Reality (MMVR)*, 2007. (Cité pages [158](#) et [159](#).)
- [Aluffi-Pentini 85] Filippo Aluffi-Pentini, Valerio Parisi & Francesco Zirilli. *Global optimization and stochastic differential equations*. *J. of optimization theory and applications*, vol. 47, no. 1, pages 1–16, 1985. (Cité page [55](#).)
- [Ashdown 05] Mark Ashdown & Yoichi Sato. *Steerable Projector Calibration*. In *CVPR '05 : Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 98, Washington, DC, USA, 2005. IEEE Computer Society. (Cité pages [81](#) et [82](#).)
- [Baraff 98] D. Baraff & A. Witkin. *Large steps in cloth simulation*. In *Proceedings of ACM SIGGRAPH 98*, pages 43–54. ACM Press, 1998. (Cité pages [49](#), [50](#) et [68](#).)
- [Baraff 03] D. Baraff, A. Witkin & M. Kass. *Untangling cloth*. *ACM Transactions on Graphics (ACM SIGGRAPH 2003)*, vol. 22, no. 3, pages 862–870, July 2003. (Cité page [73](#).)
- [Benech 05] Nicolas Benech & Carlos A. Negreira. *Longitudinal and lateral low frequency head wave analysis in soft media, tissue*. *The Journal of the Acoustical Society of America*, vol. 117, no. 6, pages 3424–3431, 2005. (Cité pages [62](#), [63](#) et [69](#).)
- [Bertocchi 91] M. Bertocchi & C. D. Odoado. *A stochastic algorithm for global optimization based on threshold accepting techniques*. In *11th European Congress on Operational Research (EURO XI)*, pages 128–136, Aachen, Germany, 1991. (Cité page [55](#).)
- [Bhat 03] K. Bhat, C. Twigg, J. Hodgins, P. Khosla, Z. Popović & S. Seitz. *Estimating cloth simulation parameters from video*. In *Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA 2003)*, pages 37–51. ACM Press, 2003. (Cité pages [49](#), [50](#), [68](#), [69](#), [73](#), [74](#), [75](#), [121](#) et [122](#).)
- [Bianchi 04] Gérald Bianchi, Barbara Solenthaler, Gábor Székely & Matthias Harders. *Simultaneous Topology and Stiffness Identifica-*

- tion for Mass-Spring Models Based on FEM Reference Deformations*. In MICCAI (2), pages 293–301, 2004. (Cité page 28.)
- [Bickel 09] Bernd Bickel, Moritz Bächer, Miguel A. Otaduy, Wojciech Matusik, Hanspeter Pfister & Markus Gross. *Capture and modeling of non-linear heterogeneous soft tissue*. ACM Trans. Graph., vol. 28, no. 3, 2009. (Cité pages 72 et 74.)
- [Boivin 01] Samuel Boivin & Andre Gagalowicz. *Image-based rendering of diffuse, specular and glossy surfaces from a single image*. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pages 107–116. ACM Press, 2001. (Cité pages 14, 45, 46 et 165.)
- [Boudou 06] T. Boudou, J. Ohayon, Y. Arntz, G. Finet, C. Picart & P. Tracqui. *An extended modeling of the micropipette aspiration experiment for the characterization of the Young's modulus and Poisson's ratio of adherent thin biological samples : Numerical and experimental studies*. Journal of Biomechanics, vol. 39, pages 1677–1685, 2006. (Cité page 71.)
- [Bouguet 04] Jean-Yves Bouguet. *Camera Calibration Toolbox for MATLAB*, 2004. (Cité page 80.)
- [Bradley 08] Derek Bradley, Tiberiu Popa, Alla Sheffer, Wolfgang Heidrich & Tamy Boubekur. *Markerless garment capture*. In SIGGRAPH '08 : ACM SIGGRAPH 2008 papers, pages 1–9, New York, NY, USA, 2008. ACM. (Cité page 73.)
- [Bro-Nielsen 96] Morten Bro-Nielsen & Stephane Cotin. *Real-time Volumetric Deformable Models for Surgery Simulation using Finite Elements and Condensation*. Computer Graphics Forum, vol. 15, no. 3, pages 57–66, 1996. (Cité pages 32 et 74.)
- [Brouwer 01] I. Brouwer, J. Ustin, L. Bentley, A. Sherman, N. Dhruv & F. Tendick. *Measuring In Vivo Animal Soft Tissue Properties for Haptic Modeling in Surgical Simulation*. In Proc. Medicine Meets Virtual Reality, pages 69–74, 2001. (Cité pages 59, 60, 64 et 69.)
- [Brown 03] J. Brown, J. Rosen & Y. Kim. *Vivo and In-Situ Compressive Properties of Porcine Abdominal Soft Tissues*. In Proc. Medicine Meets Virtual Reality, pages 26–32, 2003. (Cité pages 64 et 69.)
- [Bruyns 02] Cynthia Bruyns & Mark P. Ottensmeyer. *Measurements of Soft-Tissue Mechanical Properties to Support Development of a Physically Based Virtual Animal Model*. In MICCAI '02 : Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention-Part I, pages 282–289, London, UK, 2002. Springer-Verlag. (Cité pages 60 et 69.)
- [Burion 08] Steve Burion, François Conti, Anna Petrovskaya, Charles Baur & Oussama Khatib. *Identifying physical properties of*

- deformable objects by using particle filters*. In ICRA, pages 1112–1117, 2008. (Cité page 28.)
- [Chen 96] E.J. Chen, J. Novakofski, W.K. Jenkins & W.D. Jr. O’Brien. *Young’s modulus measurements of soft tissues with application to elasticity imaging*. Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions, vol. 43, no. 1, pages 191–194, 1996. (Cité page 61.)
- [Choi 04] Kup-Sze Choi, Hanquiu Sun, Pheng-Ann Heng & Jun Zou. *Deformable Simulation Using Force Propagation Model with Finite Element Optimization*. Computer & Graphics, vol. 28, no. 4, pages 559–568, August 2004. (Cité page 28.)
- [Collette 02] Y. Collette & P. Siarry. *Optimisation multiobjectif*. Eyrolles, octobre 2002. (Cité pages 51 et 52.)
- [Cook 81] Robert L. Cook & Kenneth E. Torrance. *A Reflectance Model for COmputer Graphics*. SIGGRAPH, Computer Graphics Proceedings, vol. 15, pages 301–316, July 1981. (Cité page 45.)
- [Corporation] Ascension Technology Corporation. *Flock of birds*. <http://www.ascension-tech.com/products/flockofbirds.php>. (Cité page 25.)
- [Cotin 97] Stéphane Cotin. *Modèles anatomiques déformables en temps-réel*. PhD thesis, Université de Nice Sophia Antipolis, 1997. (Cité pages 31 et 74.)
- [Courbon 80] Jean Courbon. *Théorie des poutres*. Techniques de l’Ingénieur, traité Construction, no. C 2 010, 1980. (Cité page 66.)
- [Creutz 83] M. Creutz. *Microcanonical Monte Carlo simulation*. Physical Review Letters, vol. 50, no. 19, pages 1411–1414, 1983. (Cité page 55.)
- [Darwin 59] Charles Darwin. *On the origin of species by means of natural selection or the preservation of favored races in the struggle for life*. Murray, 1859. (Cité page 55.)
- [Delalleau 05] A. Delalleau, G. Josse, JM. Lagarde, H. Zahouani & JM. Berghéau. *Characterization of the mechanical properties of skin by inverse analysis combined with the indentation test*. In Journal of Biomechanics, volume 39, pages 1603–1610, 2005. (Cité pages 61, 62 et 69.)
- [Dréo 03] Johann Dréo, Alain Pétrowsky, Eric Taillard & Patrick Siarry. *Métaheuristiques pour l’optimisation difficile*. Eyrolles, 2003. (Cité pages 51, 52 et 55.)
- [Dueck 90] Gunter Dueck & Tobias Scheuer. *Threshold accepting : a general purpose optimization algorithm appearing superior to simulated annealing*. J. Comput. Phys., vol. 90, no. 1, pages 161–175, 1990. (Cité page 55.)
- [Dueck 93] Gunter Dueck. *New optimisation heuristics : The great deluge algorithm and the record-to-record travel*. J. Comput. Phys., vol. 104, pages 86–92, 1993. (Cité page 55.)

- [Duriez 04] Christian Duriez. *Contact frottant entre objets déformables dans des simulations temps-réel avec retour haptique*. PhD thesis, Université d'Evry, Robotique - Réalité Virtuelle, 2004. (Cité pages 37, 74 et 102.)
- [Felippa 00] C.A. Felippa. *A systematic approach to the element-independent corotational dynamics of finite elements*. Technical Report CU-CAS-00-03, Center for Aerospace Structures, 2000. (Cité pages 33, 34 et 74.)
- [Fiala 05] Mark Fiala. *Automatic Projector Calibration Using Self-Identifying Patterns*. In CVPR '05 : Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops, page 113, Washington, DC, USA, 2005. IEEE Computer Society. (Cité pages 85 et 86.)
- [Fiene 06] Jonathan Fiene, Katherine J. Kuchenbecker & Gunter Niemeyer. *Event-Based Haptic Tapping with Grip Force Compensation, tissue*. In VR '06 : Proceedings of the IEEE Virtual Reality Conference (VR 2006), page 87, Washington, DC, USA, 2006. IEEE Computer Society. (Cité page 65.)
- [Fischler 81] Martin A. Fischler & Robert C. Bolles. *Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography*. Commun. ACM, vol. 24, no. 6, pages 381–395, 1981. (Cité pages 75 et 129.)
- [Fraser 57] A. S. Fraser. *Simulation of genetic systems by automatic digital computers*. Australian Journal of Biological Sciences, vol. 10, pages 484–491, 1957. (Cité page 55.)
- [Freeman 91] William T. Freeman & Edward H. Adelson. *The Design and Use of Steerable Filters*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 0, pages 891–906, 1991. (Cité pages 121 et 122.)
- [Freeman 92] William Tafel Freeman. *Steerable filters and local analysis of image structure*. PhD thesis, Cambridge, MA, USA, 1992. (Cité page 122.)
- [Geman 86] S. Geman & C. Hwang. *Diffusions for global optimization*. SIAM J. on Control and Optimization, vol. 24, no. 5, pages 1031–1043, 1986. (Cité page 55.)
- [Goldberg 89] D.E. Goldberg. *Genetic algorithm in search, optimization and machine learning*. Addison-Wesley, 1989. (Cité page 55.)
- [Goldberg 94] D.E. Goldberg. *Algorithmes génétiques. exploration, optimisation et apprentissage automatique*. Addison-Wesley, 1994. (Cité page 55.)
- [Greenspan 03] M. Greenspan & I. Fraser. *Tracking a Sphere Dipole*. Vision Interface, pages 154–161, 2003. (Cité pages 68 et 69.)

- [Haddon 98] John Haddon & David Forsyth. *Shape Representations from Shading Primitives*. Lecture Notes in Computer Science, vol. 1407, pages 415–??, 1998. (Cité page 165.)
- [Hall 97] T.J. Hall, M. Bilgen, M.F. Insana & T.A. Krouskop. *Phantom materials for elastography*. Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions, vol. 44, no. 6, pages 1355–1365, 1997. (Cité pages 62 et 63.)
- [Hasler 06] Nils Hasler, Mark Asbach, Bodo Rosenhahn, Jens-Rainer Ohm & Hans-Peter Seidel. *Physically Based Tracking of Cloth*. In Leif Kobbelt, Torsten Kuhlen, Til Aach & Rüdiger Westermann, éditeurs, 11th International Fall Workshop on Vision, Modeling, and Visualization 2006 (VMV 2006), pages 49–56, Aachen, Germany, November 2006. European Association for Computer Graphics (Eurographics), IOS. (Cité pages 73 et 74.)
- [He 91] Xiao D. He, Kenneth E. Torrance, Francois X. Sillion & Donald P. Greenberg. *A Comprehensive Physical Model for Light Reflection*. SIGGRAPH, Computer Graphics Proceedings, vol. 25, pages 175–186, July 1991. (Cité page 45.)
- [Hochmuth 93] R.M. Hochmuth. *Measuring the Mechanical Properties of Individual Human Blood Cells*. Journal of Biomechanics Engineering, vol. 115, pages 515–519, 1993. (Cité page 66.)
- [Horn 89] Berthold K. P. Horn & Michael J. Brooks. *Shape from shading*. MIT Press, 1989. (Cité page 165.)
- [Immersion] Immersion. *MicroScribe Systems*. <http://www.immersion.com/digitizer/>. (Cité page 25.)
- [Instron 07] Instron. *Instron Universal Materials Testing Machines*. http://www.instron.us/wa/products/universal_material, 2007. (Cité page 56.)
- [Jensen 01] H.W. Jensen. *Realistic image synthesis using photon mapping*. AK, Peters, 2001. (Cité page 103.)
- [Johann Carl Friedrich Gauß] Adrien-Marie Legendre Johann Carl Friedrich Gauß. *Méthode des moindres carrés*. http://fr.wikipedia.org/wiki/M%C3%A9thode_des_moindres_carr%C3%A9s. (Cité page 118.)
- [Joukhadar 97] A. Joukhadar, F. Garat & Ch. Laugier. *Parameter Identification for Dynamic Simulation*. IEEE Int. Conf. on Robotics and Automation, page 1928, 1997. (Cité page 28.)
- [Kato 99] H. Kato & Mark Billinghurst. *Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System*. In Proceedings of the 2nd International Workshop on Augmented Reality (IWAR 99), San Francisco, USA, 1999. (Cité page 78.)
- [Kauer 01a] M. Kauer, V. Vuskovic, J. Dual, Gabor Szekely & Michael Bajka. *Inverse Finite Element Characterization of Soft Tis-*

- sues*. In MICCAI '01 : Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 128–136, London, UK, 2001. Springer-Verlag. (Cité pages 66, 67, 69, 70, 73, 74, 75 et 135.)
- [Kauer 01b] Martin Kauer. *Inverse Finite Element Characterization of Soft Tissues with Aspiration Experiments*. PhD thesis, Swiss Federal Institute of Technologies, 2001. (Cité pages 66 et 70.)
- [Kawabata 80] S. Kawabata. The standardization and analysis of hand evaluation. The Textile Machinery Society of Japan, 1980. (Cité pages 28, 49, 57 et 73.)
- [Kerdok 01] Amy E. Kerdok, Stephane M. Cotin, Mark P. Ottensmeyer, Anna M. Galea, Robert D. Howe & Steven L. Dawson. *Truth cube : Establishing physical standards for soft tissue simulation*. Medical Image Analysis, Volume 7, Issue 3, Functional Imaging and Modeling of the Heart, pages 283–291, September 2001. (Cité pages 66, 67, 75 et 135.)
- [Kiefer 53] J. Kiefer. *Sequential minimax search for a maximum*. In Proceedings of the American Mathematical Society, volume 4, pages 502–506, 1953. (Cité page 54.)
- [Kirkpatrick 83] Kirkpatrick, Gelatt & Vecchi. *Optimization by simulated annealing*. Science, pages 220 :671–680, 1983. (Cité pages 54 et 125.)
- [Koenderink 99] Jan J. Koenderink, Andrea J. van Doorn, Kristin J. Dana & Shree K. Nayar. *Bidirectional Reflection Distribution Function of Thoroughly Pitted Surfaces*. International Journal of Computer Vision, vol. 31, no. 2/3, pages 129–144, July 1999. (Cité page 45.)
- [Kry 06] Paul Kry & Dinesh Pai. *Interaction Capture and Synthesis*. In ACM Transactions on Graphics (Proceedings of the SIGGRAPH conference), volume 25, pages 872–880, 2006. (Cité page 65.)
- [Lang 02] Jochen Lang, Dinesh K. Pai & Robert J. Woodham. *Acquisition of Elastic Models for Interactive Simulation., tissue*. I. J. Robotic Res., vol. 21, no. 8, pages 713–734, 2002. (Cité pages 59, 66, 72, 74 et 135.)
- [Lenoir 02] Julien Lenoir, Philippe Meseure, Laurent Grisoni & Christophe Chaillou. *Surgical Thread Simulation*. In Marc Thiriet, editeur, Modelling and Simulation for Computer-aided Medicine and Surgery (MS4CMS), volume 12, pages 102–107, Rocquencourt (France), 12-15 nov 2002. INRIA, EDP Sciences. (Cité page 27.)
- [Leskovsky 06] P. Leskovsky, M. Harders & G. Székely. *Assessing the Fidelity of the Haptically Rendered Deformable Objects*. In 14th Symposium on Haptic Interfaces for Virtual Environment and Te-

- leoperator Systems, IEEE Virtual Reality, IEEE Virtual Reality, pages 19–25, March 2006. (Cité pages 135 et 165.)
- [Li 04] Yin Li, Jian Sun, Chi-Keung Tang & Heung-Yeung Shum. *Lazy snapping*. ACM Transactions on Graphics, vol. 23, no. 3, pages 303–308, August 2004. (Cité page 112.)
- [Liu 04] Huafeng Liu & Pengcheng Shi. *Robust Identification of Object Elasticity*. In ECCV Workshops CVAMIA and MMBIA, pages 423–436, 2004. A trouver absolument!!! (Cité pages 66, 69 et 71.)
- [Liu 05] C. Karen Liu, Aaron Hertzmann & Zoran Popović. *Learning physics-based motion style with nonlinear inverse optimization*. ACM Trans. Graph., vol. 24, no. 3, pages 1071–1081, 2005. (Cité pages 72 et 74.)
- [Louchet 95] Jean Louchet, Xavier Provot & David Crochemore. *Evolutionary identification of cloth animation models*. In Dimitri Terzopoulos & Daniel Thalmann, editeurs, Computer Animation and Simulation '95, pages 44–54. Springer-Verlag, 1995. (Cité page 28.)
- [Meseure 02] Philippe Meseure. *Animation basée sur la physique pour les environnements interactifs temps-réel*. Habilitation à Diriger des Recherches, 17 décembre 2002. (Cité page 27.)
- [Miller 97] K. Miller & K. Chinzei. *Constitutive Modeling of Brain Tissue : Experiment and Theory*. Journal of Biomechanics, vol. 30, pages 1115–1121, 1997. (Cité page 57.)
- [Miller 00] K. Miller, K. Chinzei, G. Orssengo & P. Bednarz. *Mechanical properties of brain tissue in-vivo : experiment and computer simulation*. Journal of Biomechanics, vol. 33, pages 1369–1376, 2000. (Cité pages 58 et 69.)
- [Nava 03] Alessandro Nava, Edoardo Mazza, Frederic Kleinermann, Nick J. Avis & John McClure. *Determination of the Mechanical Properties of Soft Human Tissues through Aspiration Experiments*. Medical Image Computing and Computer-Assisted Intervention - MICCAI 2003, vol. 2878/2003, pages 222–229, 2003. (Cité pages 66, 69, 70, 73, 74 et 75.)
- [NgoNgoc 04] C. NgoNgoc & S. Boivin. *Nonlinear Cloth Simulation*. Research report, INRIA Futurs, 2004. (Cité pages 27 et 58.)
- [Ostrander 92] L.E. Ostrander & Bok Y. Lee. *Testing Viscoelastic Properties Of Biological Soft Tissue*. Engineering in Medicine and Biology Society, vol. 14, pages 107–109, 1992. (Cité pages 63 et 69.)
- [Ottensmeyer 01] Mark P. Ottensmeyer & Jr. Kenneth J. Salisbury. *In Vivo Data Acquisition Instrument for Solid Organ Mechanical Property Measurement*. Medical Image Computing and

- Computer-Assisted Intervention - MICCAI 2001 : 4th International Conference Utrecht, The Netherlands, pages 975–982, 2001. (Cité pages 60 et 69.)
- [Ottensmeyer 02] Mark P Ottensmeyer. *In vivo measurement of solid organ visco-elastic properties*. Stud Health Technol Inform, vol. 85, pages 328–333, 2002. (Cité pages 60 et 69.)
- [Ottensmeyer 04] Mark P. Ottensmeyer, Amy E. Kerdok, Robert D. Howe & Steven L. Dawson. *The Effects of Testing Environment on the Viscoelastic Properties of Soft Tissues*. Medical Simulation, vol. Volume 3078/2004, pages 9–18, 2004. (Cité pages 60 et 69.)
- [Pai 00] D. Pai, J. Lang, J. Lloyd & R. Woodham. *ACME, a telero-botic active measurement facility*. Experimental Robotics VI, vol. 250, pages 391–400, 2000. (Cité pages 59, 66, 72, 73 et 74.)
- [Pritchard 03a] D. Pritchard. Cloth parameters and motion capture. Master's thesis, University of British Columbia, 2003. (Cité page 68.)
- [Pritchard 03b] David Pritchard & Wolfgang Heidrich. *Cloth motion capture*. In GRAPH '03 : Proceedings of the SIGGRAPH 2003 conference on Sketches & applications, pages 1–1. ACM Press, 2003. (Cité pages 68 et 74.)
- [Provot 95] X. Provot. *Deformation constraints in a mass-spring model to describe rigid cloth behavior*. In Proceedings of Graphics Interface (GI 1995), pages 147–154. Canadian Computer-Human Communications Society, 1995. (Cité page 27.)
- [Provot 97] X. Provot. *Collision and self-collision handling in cloth model dedicated to design garments*. In Proceedings of the Eurographics Workshop on Computer Animation and Simulation 1997 (CAS 1997), pages 177–189. Springer-Verlag, 1997. (Cité page 27.)
- [Revell 05] James Revell. *Computer Vision Elastography*. PhD thesis, University of Bristol, Department of Computer Science, April 2005. (Cité pages 20 et 62.)
- [Samani 01] Abbas Samani, Jonathan Bishop & Donald B. Plewes. *A Constrained Modulus Reconstruction Technique for Breast Cancer Assessment*. IEEE Trans. Med. Imaging, vol. 20, no. 9, pages 877–885, 2001. (Cité pages 66, 69, 71 et 135.)
- [Samani 07] Abbas Samani & Donald Plewes. *An inverse problem solution for measuring the elastic modulus of intact ex vivo breast tissue tumours*. Physics in Medicine and Biology, vol. 52, no. 5, pages 1247–1260, 2007. (Cité pages 68, 69 et 71.)
- [Samur 05] Evren Samur, Mert Sedef, Cagatay Basdogan, Levent Avtan & Oktay Duzgun. *A robotic indenter for minimally invasive characterization of soft tissues*. In CARS 2005 : Computer Assisted Radiology and Surgery, volume 1281, pages 713–718, 2005. (Cité pages 61 et 69.)

- [Samur 07] Evren Samur, Mert Sedef, Cagatay Basdogan, Levent Avtan & Oktay Duzgun. *A robotic indenter for minimally invasive measurement and characterization of soft tissue response*. *Med Image Anal*, vol. 11, no. 4, pages 361–373, Aug 2007. (Cité pages 61, 62 et 69.)
- [Scholz 05] Volker Scholz, Timo Stich, Marcus Magnor, Michael Keckeisen & Markus Wacker. *Garment motion capture using color-coded patterns*. In *SIGGRAPH '05 : ACM SIGGRAPH 2005 Sketches*, page 38, New York, NY, USA, 2005. ACM. (Cité page 73.)
- [Sedef 06] Mert Sedef, Evren Samur & Cagatay Basdogan. *Real-time finite-element simulation of linear viscoelastic tissue behavior based on experimental data*. *IEEE Comput Graph Appl*, vol. 26, no. 6, pages 58–68, 2006. (Cité pages 61 et 69.)
- [Shen 02] T.S. Shen & C.H. Menq. *Digital Projector Calibration for 3-D Active Vision Systems*. *Transactions- American Society Of Mechanical Engineers Journal Of Manufacturing Science And Engineering*, vol. 124, no. 1, pages 126–134, 2002. (Cité page 81.)
- [Shi 99] Pengcheng Shi, Albert J. Sinusas, R. Todd Constable & James S. Duncan. *Volumetric Deformation Analysis Using Mechanics-Based Data Fusion : Applications in Cardiac Motion Recovery*. *International Journal of Computer Vision*, vol. 35, no. 1, pages 87–107, 1999. (Cité pages 66 et 69.)
- [Shi 04] Pengcheng Shi, Huafeng Liu & Albert J. Sinusas. *Robust Filtering Strategies for Soft Tissue Young's Modulus Characterization*. In *ISBI*, pages 768–771, 2004. a trouver absolument!!! (Cité pages 66 et 69.)
- [Sussman 87] T. Sussman & K.J. Bathe. *A finite element formulation for nonlinear incompressible elastic and inelastic analysis*. *J. Comput Struct.*, vol. 26, pages 357–409, 1987. (Cité page 70.)
- [Syllebranque] Cedric Syllebranque & Christian Duriez. *A training simulator for dental implantology*. In *5th International Symposium on Biomedical Simulation*. (Cité page 167.)
- [Syllebranque 04a] C. Syllebranque. *Estimation de propriétés mécaniques d'un modèle de tissus à partir de séquences d'images*. Master's thesis, Université des Sciences et Technologies de Lille, 2004. (Cité page 167.)
- [Syllebranque 04b] Cédric Syllebranque & Samuel Boivin. *Étude d'un simulateur mécanique de tissus à partir de séquences d'images synthétiques*. In *AFIG'04 - Actes des 17èmes journées de l'Association Française d'Informatique Graphique*, pages 123–132, 2004. (Cité page 167.)
- [Syllebranque 07a] Cedric Syllebranque & Samuel Boivin. *Estimation of Mechanical Parameters of Deformable Solids from Videos*. In *4th*

- International INTUITION Conference, Athens, October 2007. (Cit  page 167.)
- [Syllebranque 07b] Cedric Syllebranque, Samuel Boivin, Christian Duriez & Christophe Chaillou. *Estimation of Hookean Parameters of Deformable Bodies from Real Videos*. In Cyberworlds International Conference - Haptex Workshop, pages 330–337, Hannover, October 2007. (Cit  page 167.)
- [Syllebranque 08] Cedric Syllebranque & Samuel Boivin. *Estimation of mechanical parameters of deformable solids from videos*. The Visual Computer : International Journal of Computer Graphics, vol. 24, no. 11, pages 963–972, october 2008. (Cit  page 167.)
- [Tao 02] Hai Tao & Thomas S. Huang. *Visual Estimation and Compression of Facial Motion Parameters Elements of a 3D Model-Based Video Coding System*. Int. J. Comput. Vision, vol. 50, no. 2, pages 111–125, 2002. (Cit  page 72.)
- [technologies] SensAble technologies. *Phantom series*. <http://www.sensable.com/products-haptic-devices.htm>. (Cit  page 26.)
- [Vacalebri 03] P. Vacalebri, R. Lazzarini, S. Roccella, F. Vecchi, M. Carrozza, M. Zecca & P. Dario. *A Robotic System for Soft Tissue Characterization*. International Journal of HWRS, vol. 4, no. 2, pages 1–4, 2003. (Cit  page 61.)
- [van den Doel 01] Kees van den Doel, Paul G. Kry & Dinesh K. Pai. *Foley Automatic : physically-based sound effects for interactive simulation and animation*. In SIGGRAPH, pages 537–544, 2001. (Cit  page 25.)
- [Vicon] Vicon. *Motion capture systems*. <http://www.vicon.com/>. (Cit  page 25.)
- [Vuskovic 01] Vladimir Vuskovic. *Device for In-Vivo Measurement of Mechanical Properties of Internal Human Soft Tissues*. http://e-collection.ethbib.ethz.ch/eserv/eth_24411/eth-24411-01.pdf, 2001. (Cit  pages 66, 69, 70, 73 et 74.)
- [Ward 92] Gregory J. Ward. *Measuring and modeling anisotropic reflection*. In SIGGRAPH '92 : Proceedings of the 19th annual conference on Computer graphics and interactive techniques, pages 265–272, 1992. (Cit  page 45.)
- [White 07] Ryan White, Keenan Crane & D. A. Forsyth. *Capturing and animating occluded cloth*. In SIGGRAPH '07 : ACM SIGGRAPH 2007 papers, page 34, New York, NY, USA, 2007. ACM. (Cit  page 73.)
- [Yu 99] Yizhou Yu, Paul Debevec, Jitendra Malik & Tim Hawkins. *Inverse global illumination : recovering reflectance models of real scenes from photographs*. In Proceedings of ACM SIGGRAPH, pages 215–224, 1999. (Cit  pages 14, 45 et 165.)

- [Zhang 97] M. Zhang, Y.P. Zheng & A.F.T. Mak. *Estimating the effective Young's modulus of soft tissues from indentation tests-nonlinear finite element analysis of effects of friction and large deformation*. Medical Engineering and Physics, vol. 19, no. 6, pages 512–517, 1997. (Cité page 165.)
- [Zhang 98] Zhengyou Zhang & Zhengyou Zhang. *A Flexible New Technique for Camera Calibration*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pages 1330–1334, 1998. (Cité page 78.)
- [Zhang 02] Li Zhang, Brian Curless & Steven M. Seitz. *Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming*. In The 1st IEEE International Symposium on 3D Data Processing, Visualization, and Transmission, pages 24–36, June 2002. (Cité page 165.)
- [Zwick] Zwick. *Universal hardness tester*. <http://www.globalspec.com/FeaturedProducts/Detail/ZwickUSA>. (Cité pages 56, 75 et 127.)

Index

Symbols

- élément de référence 30
 - élément finis 28
 - éléments finis 28, 29, 31, 33, 35
 - élasticité linéaire 21, 23, 24
 - élasticité non-linéaire 24
 - élastographie 62
 - énergie dissipée 18
 - équation aux dérivées partielles 29
 - équation de Volterra 23
 - équation dynamique 26
 - état d'équilibre 32
- ## A
- accélération 35
 - algorithmes évolutionnaires 55
 - algorithmes génétiques 55
 - amortissement global 26, 27
 - anisotrope 20, 137
 - ARTag 83–85
 - ARToolkit 83
 - ARToolkitPlus 85, 86
 - ARToolkit 78, 80, 83, 84
 - ARToolKitPlus 115
 - ARToolkitPlus 78, 80, 83–85, 96–98, 105, 184
 - auxétiques 40, 149
- ## B
- background learning 112
 - blue screen 111
 - boîte noire 130
 - Broyden 33
 - bulk modulus 37
- ## C
- caméra vidéo 78
 - Camera Calibration Toolbox 80, 105
 - capteur de forces 87
 - carte de profondeur 66
 - centre optique 79
 - chargement 18
 - circle fitting 96
 - coefficient de Poisson .. 15, 23, 25, 37, 39, 40, 48, 66, 74, 111
 - coefficients de Lamé 40
 - color keying 111
 - compressible 39
 - condition de Dirichlet 35
 - conditionnement 88
 - conditions d'inégalités 36
 - contrainte planaire 36
 - contraintes de points fixes 32
 - convolution 113
 - corrélés 106
- ## D
- Déformation élastique 18
 - déformation élastique 22
 - déformation géométrique 31
 - Déformation plastique 18
 - déformations élémentaires 21
 - déformations solides 17
 - détournage par couleur 111
 - data fitting 53, 162
 - data-driven models 74
 - degré de non-linéarité N 70
 - depth map 66
 - descente de gradient 47, 53
 - Direct3D 103
 - dispositif automatiques 56
 - dispositifs interactifs 56
 - dissipation d'énergie 18, 27
 - distortion optique 104
 - docking 94, 95
 - dynamique 31
 - dynamiques 25

- E**
- estimation de paramètres 34
 - estimation inverse 23
 - ex-vivo 57, 68
 - explicite 27
 - extrema locaux 53
- F**
- fitter 71
 - fitting 129
 - fitting par moindres carrés 128, 129
 - flou de bougé 113
 - fluage 23
 - focale 79
 - fonction de coût 51
 - fonction de fluage 23
 - fonction de relaxation 23
 - fonction de variation 44, 47
 - fonction objectif 51
 - fonction objective 44, 47, 106
 - fonctions objectives 106
 - force sensitive resistors(FSRs) 65
 - formulation corotationnelle 33, 34
 - formulation dynamique 35
 - formulation non-linéaire 33
 - formulation statique 32, 35
 - frottement 37
- G**
- Gauss-Seidel 32, 52
 - golden ratio 54
 - Golden Section 133, 134
 - golden section 54
 - gradient conjugué 52
 - grands déplacements 34
- H**
- hétérogénéité 20
 - hétérogène 20
 - haptiques 26
 - homogène 20
 - hystérésis 18
 - hystérectomie 66
- I**
- identification 33
 - imagerie par tomographie 66
 - implicite 27
 - in-situ 57–61
 - in-vitro 56–60, 62, 66
 - in-vivo 57–62, 64, 66, 73
 - incompressible 39
 - indenteur 56, 58–63
 - infographie 45
 - initial guess 54
 - intégration Euler explicite 35
 - intégration Euler implicite 35
 - interactions 35
 - invasif 57
 - inversion de la matrice 32
 - isotrope 20
- J**
- Jacobi 32
- L**
- la mécanique des fluides 17
 - laparoscopie 57
 - le modèle Kelvin-Voigt 23
 - le symbole de Kronecker 23
 - Levenberg-Marquard 54
 - loading 18
 - loi de comportement 30
 - loi de Coulomb 37
 - loi de Hooke 22, 24, 40, 46, 71
 - lois de déformations linéaires 25
 - lois de déformations non-linéaires 25
- M**
- mécanique des milieux continus 17
 - mécanique inverse 15, 45, 48
 - métaheuristique 54
 - métaheuristiques 51
 - méthode de Gauss-Newton 53
 - méthode de Jacobi 52
 - méthode de Newton 53
 - métrique d'erreur 44, 106
 - métriques d'erreur 15, 75
 - mammographie 20
 - marqueur origine 84–86
 - marqueurs visuels 83
 - masses-ressort 25
 - masses-ressorts 26
 - matériaux incompressibles 24
 - matrice élémentaire 29, 33
 - matrice creuse à bande 31

- matrice d'amortissement 35
matrice de masses 35
matrice de pression 24
matrice de rigidité 33
matrice de rigidité globale 30
Maxwell-Weichert 23
modèle de Maxwell 23
modèle de Maxwell généralisé 23
modèle de Mooney-Rivlin 24
modèle linéaire standard 23
modèle Neo-Hookean 24, 70
module d'élasticité 23, 37
module de cisaillement 24
module de Young .. 15, 23, 25, 37, 38, 40, 48,
70, 74, 111
moindres carrés 47
Mooney-Rivlin 74
motion blur 113
mouvement rigide 33
multiplicateur de Lagrange 36
- N**
- Neo-Hookean 135
Newton-Raphson 33
nombre d'or 54
non-invasif 57, 62
- O**
- off-line 99, 103
OpenGL 103, 104
optimisation 51
optimisation combinatoire 51
optimisation continue 51
optimisation difficile 52
optimisation mono-objectif 51
optimisation multi-objectif 51
ou exclusif 118
- P**
- paramètres extrinsèques 79, 80, 83
paramètres intrinsèques 79–81, 83, 105
Pascal 38
petites déformations 23, 32, 33
petits déplacements 33
peu invasif 57, 60
Phantom 115
phases comportementales 18
photo-réalisme 103
photo-réaliste 45
photon-mapping 45, 103
photon-tracing 103
pin-hole 105
pinhole 81
positionnement automatique à partir d'images
78
pré-conditionnement 19
problème de complémentarité linéaire 37
problème du voyageur de commerce 51
programmation linéaire 52
programmation quadratique 36
- Q**
- quantification des différences 115
- R**
- réalisme 103
résolution numérique 21
résolution par éléments finis 25
RANSAC 129
RanSaC 75, 129, 130, 136, 137, 162
ray-tracing 103
raytracing 44
recherche avec tabous 55
recuit simulé 51, 54, 125
relaxation 18, 23
render to texture 105
rendu 44
rendu inverse 45, 48
représentation Eulérienne 21
représentation Lagrangienne 21
représentations mathématiques 21
retour d'effort 26, 94
reverse engineering 130
Rupture 18
- S**
- séquence vidéo synthétique 103
shaders 103
shear modulus 37
simulated annealing 54
simulation interactive 26
simulations en temps-interactif 25
simulations interactives 25, 33
SOFA 158–160
sous-gradients 53
statique 31

| | |
|-----------------------------|----------|
| statiques | 25 |
| steepest descent | 53 |
| steerable filters | 121 |
| strain | 22, 128 |
| stress | 22, 128 |
| stress/strain | 128, 137 |
| Subsurface-scattering | 45 |

T

| | |
|------------------------------|-----------|
| temps de calcul | 103 |
| temps réel | 97 |
| temps-réel | 103 |
| tenseur de contrainte | 21–23, 33 |
| tenseur de déformation | 21–23, 33 |
| tenseur de Finger | 24 |
| tenseur de Green | 22 |
| trace du tenseur | 23 |
| tracking | 78, 92 |
| tracking 6D | 74 |
| tracking magnétique | 93 |
| Truthcube | 75 |

U

| | |
|-----------------|----|
| unloading | 18 |
|-----------------|----|

V

| | |
|--------------------------------|------------|
| vidéo-projecteur | 78 |
| viscoélasticité linéaire | 23 |
| viscoélastique | 18, 19, 23 |
| vitesse | 35 |

Quelques symboles

| | |
|---------------------|---|
| C_1 | Première constante du modèle de Mooney-Rivlin |
| C_2 | Seconde constante du modèle de Mooney-Rivlin |
| E | Module de Young (module d'élasticité) |
| Nb_{fixes} | Nombre de nœuds fixes |
| β | Amortissement global (masses-ressorts) |
| dim | Dimension du problème (2 en 2D, 3 en 3D, ...) |
| f_{max} | Force maximale mesurable en Newtons (N) ou l'équivalent en poids (sur terre, en grammes g) |
| k | Raideur d'un ressort |
| k^{ij} | Raideur du ressort reliant les particules i et j |
| l_0 | Longueur initiale d'un ressort |
| l_{0ij} | Longueur initiale du ressort reliant les particules i et j |
| m | Masse d'une particule |
| np | Nombre de paramètres |
| ν | Coefficient de Poisson |
| t_e | Taille d'un élément (ou nombre de noeuds, ex : 3 pour un triangle) |
| u_{max} | Déplacement maximal autorisé de l'indenteur ou de la plaque |
| v_{max} | Vitesse maximale du déploiement de l'indenteur ou de la plaque |
| \varnothing^{ind} | Diamètre de l'indenteur |
| Nb_{noeuds} | Nombre de nœuds du maillage |
| Distortion | Fonction de simulation de la distortion optique d'une caméra dont les paramètres intrinsèques sont connus ($P_{intrinsèques}$). Elle transforme un pixel ϕ_{orig} défini dans l'image non déformée en un pixel ϕ_{disto} défini dans l'image déformée par distortion optique. |
| Flu(t) | Fonction de fluage |
| Fun | Fonction quelconque dont les paramètres sont à estimer |
| Ind(i) | Indice du i^{eme} sommet de l'élément |
| Relax(t) | Fonction de relaxation |
| Var | Fonction de variation du processus d'optimisation |

| | |
|--|---|
| $\Delta_{\text{Anim3D}}(\mathbf{A}, \mathbf{B})$ | Métrique d'erreur ou fonction objective quantifiant la différence entre deux animations géométriques en trois dimensions \mathbf{A} et \mathbf{B} |
| $\Delta_{\text{Forces}}(\mathbf{Fa}, \mathbf{Fb})$ | Métrique d'erreur ou fonction objective quantifiant la différence entre deux vecteurs de forces \mathbf{Fa} et \mathbf{Fb} |
| $\Delta(\mathbf{VA}, \mathbf{VB})$ | Métrique d'erreur ou fonction objective quantifiant la différence entre deux vidéos \mathbf{VA} et \mathbf{VB} |
| $[\mathbf{A}_{eq}]$ | Matrice des contraintes d'égalité |
| $[\mathbf{A}_{ineq}]$ | Matrice des contraintes d'inégalité |
| $[\mathbf{B}]$ | Tenseur de Finger |
| $[\mathbf{C}]$ | Matrice de comportement du matériau |
| $[\mathbf{C}^{el}]$ | Propriétés élastiques du matériau (Tenseur d'ordre 4) |
| $[\mathbf{D}]$ | Matrice des amortissements |
| $[\mathbf{G}]$ | Module de cisaillement |
| $[\mathbf{I}]$ | Matrice identité |
| $[\mathbf{K}]$ | Matrice de rigidité globale |
| $[\mathbf{K}_e]$ | Matrice de rigidité élémentaire |
| $[\mathbf{M}]$ | Matrice des masses |
| $[\mathbf{P}]$ | Matrice de pression |
| $[\mathbf{R}_e]$ | Matrice de rotation élémentaire |
| $[\boldsymbol{\sigma}]$ | Tenseur de contraintes |
| $[\boldsymbol{\sigma}_{MR}]$ | Tenseur de contraintes du modèle Mooney-Rivlin |
| $[\boldsymbol{\sigma}_{NH}]$ | Tenseur de contraintes du modèle Néo-Hookean |
| $[\boldsymbol{\varepsilon}]$ | Tenseur de déformations |
| $[\mathbf{M}_{DC}]$ | Transformation du marqueur dispositif vers la caméra (calculée par <i>ARToolkitPlus</i>) |
| $[\mathbf{M}_{DT}]$ | Transformation du marqueur dispositif vers la touche sphérique (donnée géométrique) |
| $[\mathbf{M}_{WC}]$ | Transformation du repère global vers la caméra (calculée par <i>ARToolkitPlus</i>) |
| \mathbf{D}_t | Direction de l'outil de capture de force à l'instant t |
| \mathbf{D}_R | Résultat observé d'une fonction (<i>données réelles</i>) |
| \mathbf{D}_S | Résultat observé d'une fonction (<i>données synthétiques</i>) |
| \mathbf{F} | Forces sur tous les nœuds du maillage |
| \mathbf{F}_i | Forces sur les nœuds internes du maillage |
| \mathbf{F}_s | Forces sur les nœuds de surface du maillage |
| \mathbf{P}^{np} | Jeu de np paramètres |
| \mathbf{P}_S^{np} | Jeu de paramètres utilisé lors d'une simulation |
| \mathbf{P}_{optim}^{np} | Jeu de paramètres optimal trouvé par estimation inverse |

| | |
|-----------------|--|
| $P_{intrinsic}$ | Paramètres intrinsèques d'une caméra vidéo ou d'un vidéo projecteur |
| U | Déplacements de tous les nœuds du maillage |
| U_e | Déplacements de l'élément e |
| U_i | Déplacements des nœuds internes du maillage |
| U_s | Déplacements des nœuds de surface du maillage |
| V_i | Voisinage de la particule i |
| b_{ineq} | Vecteur des contraintes d'inégalité |
| f_i | Force sur le nœud i |
| p_i | Particule i |
| ϕ_{disto} | Pixel défini dans les coordonnées d'une image déformée par distortion optique (par la fonction Distortion) |
| ϕ_{orig} | Pixel défini dans les coordonnées d'une image non déformée |
| u | Déplacement d'un nœud |
| u_i | Déplacement du nœud i |
| Λ | Valeurs de réaction par multiplicateur de Lagrange |
| P_t | Position de l'outil de capture de force à l'instant t |
| λ | Premier coefficient de Lamé |
| μ | Second coefficient de Lamé |