Université de Montréal

Réalité augmentée en chirurgie : Développement d'un pointeur intelligent

par

Nicolas Lewis

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures

en vue de l'obtention du grade de

Maître ès sciences (M.Sc.)

en informatique

Décembre, 2003

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé :

Réalité augmentée en chirurgie : Développement d'un pointeur intelligent

présenté par

Nicolas Lewis

a été évalué par un jury composé des personnes suivantes:

Neil Stewart
(Président-rapporteur)

Jean Meunier
(Directeur de recherche)

Victor Ostromoukhov
(Membre du jury)

Mémoire accepté le 9 mars 2004

RÉSUMÉ

Ce mémoire porte sur la création d'un système de pointeur intelligent en salle d'opération. Notre approche se distingue des solutions existantes en évitant totalement de perturber le chirurgien dans sa tâche. En projetant l'information désirée directement sur le patient, nous évitons d'avoir recours aux supports visuels courramment utilisés comme les moniteurs ou les casques d'affichage ("Head-Mounted Display"). On permet ainsi au chirurgien de garder toute son attention aux endroits critiques, ce qui réduit les risques tout en accélérant la chirurgie.

Nous souhaitons donc, par exemple, projeter le contour d'une tumeur ou d'un organe directement sur la peau du patient. En ajoutant un effet de perspective à la projection, nous permettons au chirurgien d'accéder rapidement à toute l'information reccueillie avant l'opération, sans avoir à quitter le patient des yeux. Évidemment, cette projection demande de connaître précisément la forme du patient, la position des yeux du chirurgiens ainsi que la position relative de l'objet à projeter, par rapport au corps du patient. Toutes ces informations doivent être mises à jour en temps réel, puisque nous désirons permettre au chirurgien de bouger librement.

Nous avons étudié une solution n'utilisant que de l'équipement simple, évitant ainsi les coûts exorbitants des appareils trop spécialisés. Notre principale contribution dans ce mémoire est le développement d'un outil de simulation permettant de créer des modèles de projections pour une surface déformable avec plusieurs projecteurs. Cet outil nous a permis de présenter une preuve de concept pour un système complet de réalité augmentée, en situation chirurgicale.

Mots clés : réalité augmentée, chirurgie assistée par ordinateur (CAO), modélisation, vision par ordinateur.

ABSTRACT

This thesis deals with the creation of an intelligent pointer in computer-assisted surgery. We eliminate the use of any expensive and inconvenient equipment by designing a system that avoids any interference with the current tasks of the surgeon. By projecting pre-operatively selected information directly onto the patient's body, we reduce the need for traditional support devices such as monitors or Head-Mounted Displays (HMD). The surgeon's focus of attention can be kept where it is needed at all times, reducing risk while accelerating the overall surgery.

As an example, we could project the surrounding of a tumor or an organ directly over the patient's body. The projection being computed accordingly to the position of the surgeon's head, it is possible to enhance it with accurate depth value. The surgeon would then have access to all the information's needed, without taking his gaze away from the patient. Those information will have to be updated in real-time since we want the surgeon to be allowed free movement during the surgery.

Our system has been designed to use off-the-shelf projectors and cameras, which are far less expensive than specialized equipment such as HMD. Our main contribution in this Master's thesis is the development of a simulation tool generating a projection model to map a deformable surface, using multiple projectors. This tool has enabled a feasibility study of a complete augmented reality system in a surgical environment.

Keywords : augmented reality, computer assisted surgery (CAS), modeling, computer vision.

TABLE DES MATIÈRES

Liste des Figures iii

Glossaire v

Chapitre 1: Introduction 1

1.1 Design Proposé 4

1.2 Structure du Mémoire 6

Chapitre 2: Suivi 3D du Regard 7

2.1 Suivi Actif (Intrusif) 8

2.2 Suivi Passif - Segmentation 12

2.3 Solution Proposée 17

Chapitre 3: Estimation de profondeur 21

3.1 Esimation à Vue Unique 22

3.2 Stéréo-Vision 26

3.3 Résumé 39

Chapitre 4: Modélisation de surface 41

4.1 Subdivision Spatiale 42

4.2 Fonctions de Distances 44

4.3 Reconstruction Incrémentale 45

4.4 Solution Proposée 48

Chapitre 5: Recalage Élastique 3D 50

5.1 Complexité du Problème 51
5.2 Solution Hybride Parallélisée 53
5.3 Piste de Solution 56
5.4 Solution Proposée 57

Chapitre 6: Modèle de Projection et Simulation **59**
6.1 Image Chirurgien 59
6.2 Image Projecteur 61
6.3 Simulation 64

Chapitre 7: Conclusion et Discussion **69**
7.1 Notre Solution 69
7.2 Discussion 72
7.3 Futur vs Réalité 73

Références **75**

Annexe A: Intelligent Pointer in Computer Assisted Surgery - Design and Feasibility **90**
A.1 Intelligent Pointer in Computer Assisted Surgery–Design and Feasibility 90

LISTE DES FIGURES

1.1 Système de réalité augmentée en salle d'opération avec HMD 2
1.2 Résultat d'un pointeur intelligent simulé. 4
1.3 Pointeur intelligent 5

2.1 Erreur induite par estimation de postion 7
2.2 Différent délais inhérents au suivi 3D d'un objet. 11
2.3 Suivi passif par segmentation d'images 13
2.4 Super-quadriques étendus (ESQ) 16
2.5 Suivie par solution multiple 20

3.1 Différents indices de profondeur dans une image 23
3.2 Profondeur par défocus 24
3.3 Reconstruction par ombrage 26
3.4 Géométrie épipolaire 28
3.5 Correspondance épipolaire 28
3.6 Rectification planaire 29
3.7 Stéréo Global 31
3.8 Stéréo par fenêtre de corrélation 33
3.9 Stéréo avec N-Caméras 34
3.10 Triangularisation par lumière structurée 35
3.11 Voisinage Spatiotemporel et Recherche Épipolaire 38

4.1 Reconstruction par Voronoi 43
4.2 Reconstruction par réseau de neurones 44

4.3 Graphe de Delaunay, de Gabriel et interpolant régulier 47
4.4 Modélisation de base 49

5.1 Exemple de recalage entre différentes modalités 50
5.2 Recalage rigide versus recalage élastique 52
5.3 Recalage hybride 55

6.1 Image Projecteur 61
6.2 Gestion des Obstructions 64
6.3 Simulation du système de projection. 65
6.4 État initiale de la simulation, avant la projection. 65
6.5 Simulation avec projection simple. 65
6.6 Image déformée par les projecteurs. 66
6.7 Modèle mal couvert par les projecteurs. 67
6.8 Simulation avec projection d'information supplémentaire. 67

A.1 Construction of the projector images. 95
A.2 Setup of the simulated operating room. 96
A.3 Projector images 99
A.4 Projected images in the scene, projectors' point of view. 99
A.5 Projected images in the scene, surgeon's point of view. 100
A.6 Wireframe rendering of the surgeon's point of view 100

GLOSSAIRE

ALBÉDO: Fraction de l'énergie (lumière) qui est réflétée à la surface.

DIAPHONIE: Superposition parasite ou inopportune d'un signal d'une voie de transmission sur celui d'une autre occasionnant un mélange des signaux. [59]

DISPARITÉ: Profondeur relative d'un point dans une scène, calculée grâce au décalage enregistré entre deux images d'un système stéréoscopique.

ÉTIQUETTE: Élément de l'ensemble $\mathcal{L}$, contenant les différentes disparités possibles de la scène, associé à chaque pixel de l'image.

HMD: *Head Mounted Display.* Casque d'affichage permettant à l'utilisateur de se deplacer librement dans un environnement partiellement ou complètement synthétique. Dépendamment du modèle, le casque affiche à l'utilisateur l'environnement souhaité sur des écrans opaques ou transparents.

ICP: *Iterative Closest Point.* Algorithme générale permettant d'aligner des nuages de points représentant une même surface selon des points de vues différents.

IRM: *Image à Raisonnance Magnétique* Méthode d'imagerie médicale basée sur le phénomène de la résonance magnétique qui permet d'obtenir des images tomographiques de la distribution d'éléments atomiques tels que l'hydrogène [59].

RMS: *Root-Mean-Square.* Racine carrée de la moyenne des valeurs au carré de x. Plus précisément, pour des valeurs discrètes : $\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}$, formule tirée de [86].

SAD: *Sum of Absolute Differences.* Sommes des différences absolues, semblables à la SSD. Pour éliminer les données négatives, on va prendre la valeur absolue des entrées. Les différences plus grandes seront moins accentuées que lorsque mises au carré.

SSD: *Sum of Squared Differences.* Sommes des différences au carré, cette fonction est très souvent utilisée lors de diverses mises en correspondance.

REMERCIEMENTS

Je veux tout d'abord remercier tous les membres du laboratoire de vision et d'imagerie de l'Université de Montréal qui m'ont si gentillement soutenu durant le développement de ce mémoire. J'aimerais également remercier mes parents, sans qui toute cette belle aventure n'aurait pu aboutir. Merci spécialement à ma mère pour sa maîtrise hors pair de la langue française. Également merci au Laboratoire d'Informatique Graphique de l'Université de Montréal (LIGUM) pour m'avoir permis d'utiliser leur logiciel de simulation de caméra développé sous la direction de Pierre Poulin. Finalement, un merci tout spécial à mon directeur de recherche, Jean Meunier, pour son support moral et financier.

Chapitre 1

INTRODUCTION

La recherche en médecine aura, au cours des temps, toujours gardée le même objectif, soigner ceux qui souffrent. Certains patients, laissés pour mort il n'y a encore pas si longtemps, peuvent maintenant être sauvés et ce, avec des taux de succès très élevés. Toutefois, certains problèmes viennent obscurcir les promesses apportées par ces nouvelles découvertes. Par exemple, les nouvelles techniques de chirurgie, si miraculeuses qu'elles soient, sont souvent longues et très coûteuses. En effet, la technologie actuellement utilisée pour certaines opérations, ainsi que la main-d'oeuvre qualifiée requise pour l'utiliser correctement, sont à la fois rares et dispendieuses. La complexité intrinsèque de ces opérations demandant des médecins de plus en plus spécialisés (neurologie, cardiologie, ...) et leur préparation pouvant être très longue (multiple tests préalables, analyse des données), il devient très difficile, voire impossible de traiter tout ceux qui le nécessiteraient. L'apport de ces avancements s'assombrit quelque peu à la lumière de cette situation, forçant les personnes responsables à faire des choix difficiles, pouvant causer des injustices parfois fatales. Ainsi, plusieurs recherches récentes se portent maintenant sur des moyens d'accélérer les traitements, tout en minimisant les coûts des opérations. Pour assister le chirurgien dans sa tâche, et ainsi augmenter son efficacité, certains chercheurs ont étudié la possibilité de créer un système de réalité augmentée directement dans la salle d'opération. Leurs solutions sont variées, mais comportent toutes différents problèmes qui les rendent difficilement applicable pour une situation réelle.

Par exemple, l'équipe de recherche en vision de l'Université de Caroline du Nord

Figure 1.1. Solution proposée par les chercheurs de l'UNC pour développer un système permettant au chirurgien de "voir" au travers du patient pendant une laparoscopie.

(UNC) a développé un système utilisant un casque d'affichage (HMD) [1], qui permet au chirurgien de "voir" au travers de la peau de ses patients pendant une opération, une laparoscopie plus précisément, voir la figure 1.1 . Le principal problème de leur solution provient justement de l'utilisation de ces HMDs. Ces casques sont encore trop encombrant pour pouvoir être utilisés pendant une longue période de temps sans perturber l'utilisateur. De plus, la précision minimale de ces systèmes nuit à leur mise en application, la résolution des HMDs étant encore trop basse. Leur idée est toutefois fort intéressante, de permettre au chirurgien d'accéder durant l'opération à l'information acquise pré-opératoirement et ce, directement en regardant le patient. Cette possibilité permet à l'utilisateur de conserver son attention exactement où elle est le plus nécessaire, c'est-à-dire sur le patient.

La solution élaborée par Bajura *et al.* [9] est beaucoup plus intéressante au niveau de son applicabilité. En effet, elle ne vient pratiquement pas déranger le chirurgien durant son travail et sont coût reste relativement bas. En projetant directement sur le patient l'information que l'on souhaite faire parvenir au médecin, le système devient beaucoup plus intuitif et intéressant. Bien évidement, la technologie qu'ils utilisent

ne leur permet pas d'obtenir des résultats suffisamment intéressants pour parler d'un système portable et général, mais les promesses sont intéressantes. Leur système projette ainsi des images recueillies par ultrasons avant l'opération pour assister le chirurgien. Les résultats, souffrant grandement au niveau visuel, donnent davantage l'impression qu'une image a été peinte sur le patient, plutôt que le sentiment de "vision X" qu'on souhaiterait donner. Ce problème est causé en bonne partie parce qu'il n'y a pas de mise-à-jour du système durant l'opération. Ainsi, l'image n'est jamais modifiée, ni lorsque le médecin se déplace, ce qui induirait une certaine perspective, ni lorsque le patient se déforme. Ces constatations réduisent grandement les utilisations que l'on peut faire de ce système.

C'est Hoppe et Wörn [89] qui en 2001 ont proposé la solution la plus intéressante à notre point de vue. C'est également celle qui ressemble le plus à la solution que nous allons proposer plus loin. Les informations collectées pré-opérativement sont projetées directement sur le patient pendant l'opération. Le système utilise un seul projecteur dont la position est déterminée avant l'opération pour éviter les occlusions au maximum. Le modèle 3D du patient est recueilli grâce aux techniques de stéréo-vision par lumière structurée et associé au patient à l'aide de marqueur visible déposé sur le crâne de ce dernier. Leur recherche s'étant portée exclusivement sur des opérations de type crano-maxillo-facial, il est attendu que le patient (et le modèle) devront supporter un certain déplacement, mais sans toutefois permettre de déformation (transformation rigide seulement). Les marqueurs serviront donc, tout le long de l'opération, à rajuster la projection en fonction de la position de la tête du patient. L'information qui sera projetée sur le patient permettra au chirurgien de se concentrer davantage sur la zone d'opération. Toutefois, en ne connaissant pas le point de vue exact de l'utilisateur, le système ne pourra pas ajouter d'information en perspective ni de caractères alpha-numériques ou schémas graphiques. Le premier permettrait d'appliquer cette technologie à plusieurs autres types de chirurgies, plutôt que seulement les opérations crâniennes. Tandis qu'ajouter des informations secondaires sur le

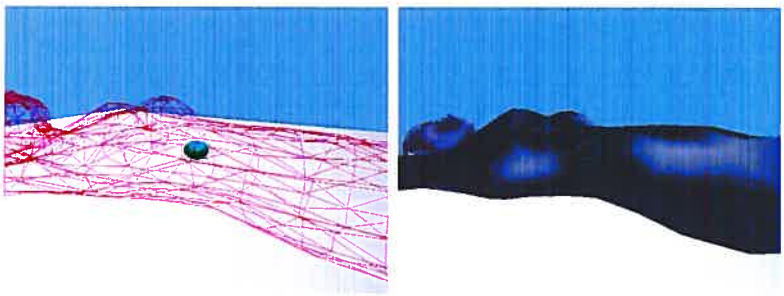

Figure 1.2. On peut voire, grâce à cette simulation, la représentation de notre "pointeur intelligent". La figure de droite représente la scène en fil de fer. Tandis que celle de gauche représente ce que le chirurgien voit, une fois la projection activée.

corps du patient, lisible par le chirurgien, permettrait d'éliminer toutes autres sources d'information (par exemple moniteur de pression et/ou électrocardiogramme) et de garder l'attention de ce dernier en tout temps fixée sur son patient.

1.1 Design Proposé

La figure 1.2 démontre un exemple de ce que l'on désigne par "pointeur intelligent". Ces résultats seront expliqués plus en détails dans le chapitre sur les modèles de projection. Pour l'instant, il suffit de remarquer que la projection identifie la position d'une tumeur à l'intérieur du patient, d'où le terme "pointeur". Nous considérons ce pointeur comme "intelligent" puisque la projection est mise à jours continuellement, en fonction du point de vue du chirurgien.

En projetant l'information directement sur le patient, nous évitons l'utilisation d'un système d'affichage classique, comme un HMD ou un simple moniteur. Cependant, il devient impossible d'ajouter de l'information 3D dans notre projection. En effet, bien que les casques d'affichage aient plusieurs inconvénients, ils ont l'avantage de pouvoir afficher deux signaux distincts, un pour chaque oeil de l'utilisateur, afin de permettre une immersion 3D. Il serait cependant possible d'inférer une troisième

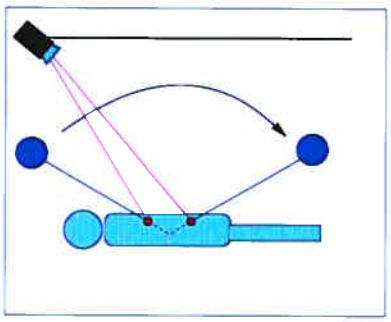

Figure 1.3. Schéma d'un pointeur intelligent avec suivi de la tête du chirurgien.

dimension dans le signal projeté si on arrivait à ajouter un effet de perspective à la projection. Pour que la perspective soit correcte, il sera absolument nécessaire de connaître la position exacte de notre "caméra" (l'oeil du chirurgien). Ainsi, en mesurant en temps réel la direction du regard du chirurgien, nous pourrons projeter une image donnant une impression efficace de 3D. Pour arriver à cette fin, nous devrons donc augmenter le système de Hoppe [89] d'une fonction permettant de retrouver la tête du médecin dans la salle d'opération. Il sera également nécessaire de recalculer le signal projeté à chaque instant afin qu'il représente bien ce que le chirurgien devrait voir. La figure 1.3 donne une intuition de ce qu'on entend ici par pointeur intelligent.

Afin de rendre notre solution encore plus générale et portable, nous devrons tolérer que le patient se déforme durant l'opération. Il serait en effet souhaitable de permettre à ce dernier de respirer pour garantir le succès de l'opération. Cette contrainte supplémentaire rajoute donc une autre tâche à notre système qui devra, encore en temps réel, connaître la forme du patient et juxtaposer les informations pré-opérativement recueillies à ce modèle, tout en suivant ses déformations. Cette dernière partie est sans doute la plus complexe. En effet, certaines solutions existent déjà pour trouver un visage dans une scène ou pour retrouver la forme d'un objet déformable, ces dernières étant même proche du temps réel. Toutefois, aligner deux

modèles entre eux, en assumant qu'ils peuvent bouger ou se déformer, reste encore un problème non négligeable.

1.2 Structure du Mémoire

Ce mémoire sera donc divisé en trois grandes parties distinctes. C'est-à-dire les trois grands défis à résoudre pour la réalisation de notre pointeur intelligent : (1) le suivie de la tête du chirurgien (chapitre 2), (2) la reconstruction du modèle du patient (chapitres 3 et 4) et (3) la création d'une image de synthèse à projeter (chapitres 5 et 6). Une bonne partie des solutions qui sont proposées dans ce mémoire n'ont pas été implantées. Les différents chapitres seront donc majoritairement une revue de littérature des multiples possibilités qui s'offrent à nous. Les critères d'évaluation de ces solutions ont été choisis dans le but de créer un système à la fois pratique et discret. Si on veut que notre système soit accepté dans les hôpitaux, il devra en effet être rapide, précis, non-intrusif et évidement applicable à un environnement chirurgical. Suivra ensuite une présentation des résultats que nous avons obtenus, entre autre pour la construction des images de synthèse en simulation.

Le lecteur pourra également trouver, en annexe, un article présenté lors de la conférence "25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society" résumant (en anglais) une partie du présent document.

Chapitre 2

SUIVI 3D DU REGARD

L'objectif final de ce projet est d'arriver à ajouter de la profondeur à une projection planaire. À l'aide d'un effet de perspective réussi il est possible de convaincre un usager que ce qu'il voit est en 3D. Il suffit de penser à toutes les images en 3 dimensions qui sont affichées sur un moniteur d'ordinateur. L'utilisation d'un projecteur nous empêche malheureusement de compter sur un seul et unique point de vue, comme ce peut être le cas lorsqu'on regarde un moniteur. Pour que la projection imite correctement la perspective, il sera essentiel de connaître le point de vue de l'utilisateur. Cette position se doit d'être précise puisque, comme on peut le voir dans la figure 2.1, même si l'erreur ne survient que lors de l'estimation de la position du point de vue, l'erreur finale reste élevée. Cette constation est importante car les sources d'erreurs sont multiples dans un tel système (erreur d'affichage du point Q, erreur de position E, erreur de recalage du point P, erreur de calcul de surface I). Plusieurs méthodes ont été développées pour arriver à retrouver soit la tête, soit le vis-

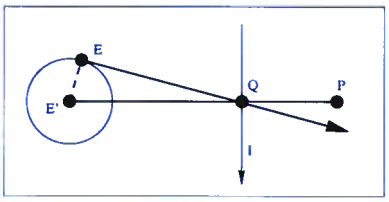

Figure 2.1. En supposant que toutes les informations connues sont exactes, à l'exception de la position même du point de vue, une erreur relativement faible peut induire un déplacement non-négligeable du point estimé. E' représente la position réel du point de vue et Q le point P tel que projeté sur un écran. Simplification d'un schéma tiré de [38].

age d'une personne. Ces méthodes peuvent être classées en deux catégories, passives et actives. Les solutions actives doivent utiliser des appareils spécialisés pour calculer la position recherchée[1]. Ces appareils sont souvent restreignants et encombrants, cependant ils ont la réputation d'être plus précis et rapides que leurs équivalences passives. Malgré tout, nous tenons absolument à éviter une quelconque perturbation du protocole de chirurgie. Nous devrons donc nous rabattre sur une méthode passive, utilisant la segmentation d'images afin de retrouver la tête du chirurgien dans un ensemble d'images. La position 3D dans la scène pourra ensuite être recalculée par triangularisation, selon une méthode développée en stéréo-vision. Nous verrons à la section 2.2 que, bien que moins précise et plus lente, la segmentation d'images comporte plusieurs avantages pour nous. En effet, les mêmes caméras pourront être utilisées pour le suivi de la tête et pour la modélisation du patient. Nous évitons ainsi d'avoir recours à du matériel spécialisé et dispendieux, tout en gardant notre solution générale au maximum et ce, sans venir perturber le travail de l'équipe chirurgicale.

2.1 Suivi Actif (Intrusif)

Les solutions actives sont très nombreuses et diversifiées. Chaque solution comporte des restrictions qui ne la rend vraiment applicable que dans certaines situations précises. Par exemple, les lecteurs mécaniques sont habituellement constitués d'un bras robotisé qui calcule la position d'un objet à l'aide des angles de ses différentes articulations. Évidement, on ne pourrait attacher un bras à la tête d'un chirurgien pour en calculer la position. Cependant, dans le cas où cette solution serait applicable, l'utilisateur jouirait d'un des systèmes les plus rapides et précis existant. Toutes les positions possibles sont calculées, par rapport à un point fixe, à l'aide de potentiomètres et de lecteurs de torsion transformant les différentes déformations en

[1] Le terme actif est utilisé pour les solutions venant interagir dans la scène. Contrairement aux solutions dites passives qui se contente plutôt d'analyser des images prises de l'extérieur.

courant électrique [2]. La position étant calculée instantanément et de façon directe, c'est sans doute le système le moins sujet aux erreurs de lecture. Il est également possible de calculer une position à l'aide d'accéléromètres et de gyroscopes. L'inertie et la force centrifuge permettent de calculer la position relative de l'objet étudié à un certain temps t, à partir de sa position au temps $t - \Delta t$. Ces appareils sont malheureusement sujets au glissement : il est difficile de faire la différence entre un objet arrêté et un autre se déplaçant très lentement. La position étant toujours relative à celle calculée à l'étape précédente, les erreurs qui apparaissent en début d'expérience seront ainsi trainées, et probablement exagérées, jusqu'à la fin. Donc, non seulement ces systèmes sont encombrants, mais en plus, ils ne sont vraiment efficaces que jumelés à un autre type de suivi, qui contre-vérifie les résultats, afin d'en assurer la précision.

2.1.1 Lecteurs d'Ondulations

Les lecteurs à ondes[2] peuvent être implantés de deux façons distinctes, "inside-looking-out" ou "outside-looking-in". La différence est simple. Ces lecteurs sont tous basés sur le même système, un émetteur d'ondes couplé à un récepteur qui calcule les distances en fonction des longueurs d'ondes et de l'affaiblissement du signal[3]. La direction de l'onde varie selon que l'émetteur est porté par le sujet ou fixé dans la scène. Ainsi, inside-looking-out représente un système ou les émetteurs sont fixes dans la scène. L'appellation vient du lecteur d'onde qui est fixé au sujet et donc "regarde" vers l'extérieur pour retrouver le signal (les ondes) qu'il recherche. Les solutions outside-looking-in s'avèrent très intéressantes pour un système de réalité augmentée. Les émetteurs sont habituellement légers et peu encombrants. Ils peuvent donc facilement être portés par un usager durant de longues périodes. On peut fixer les lecteurs

[2] Accoustique, Magnétique, Micro-Ondes ou Radio.

[3] Les détails techniques sur les différentes implantations ainsi que les principes sous-jacents à chacun des modèles de lecteurs d'ondulation peuvent être trouvés dans le cours donné par Allen, Bishop et Welch pour SIGGRAPH 2001 [2].

d'ondes, qui sont habituellement plus encombrants, tout autour de la scène, et ainsi obtenir un système efficace sans trop déranger l'utilisateur. Toutefois, ils sont sujets aux erreurs de façon significative. Dans sa thèse, Holloway [38] développe un modèle d'erreurs pour les systèmes de réalité augmentée. Bien que relativement vieille, son étude reste encore tout à fait applicable aux systèmes actuels de réalité augmentée. Les erreurs de lecture sont classées en trois catégories d'importance variable. La première source d'erreurs apparaît lors du positionement du capteur dans l'espace. La référence utilisée comme origine du volume de travail est rarement la même que celle du capteur. Il est donc important de connaître la position exacte de l'origine du capteur dans la salle avant même de commencer le suivi. Les pires erreurs viennent cependant de la tranformation entre l'émetteur et le capteur. Holloway [38] sépare ces erreurs selon qu'elles soient statiques, aléatoires ou dynamiques. Les erreurs statiques désignent les erreurs intrinsèques à l'appareil. Ces erreurs sont faciles à annuler grâce à un bon calibrage puisqu'elles sont répétées à chaque lecture. Ensuite, une certaine imprécision aléatoire apparaît en fonction de la distance émetteur-capteur. Il est impossible de prédire et d'éliminer ces erreurs puisqu'elles sont directement reliées aux capacités de l'appareil. Il faut donc faire très attention à garder le sujet étudié à l'intérieur du volume de travail du récepteur. Finalement les erreurs dynamiques sont causées par le délai de lecture de l'appareil. Plus le sujet se déplacera rapidement, plus cette erreur sera élevée. Les délais à l'intérieur même du système doivent également être pris en compte minutieusement. Lorsque c'est applicable, il est possible de retarder la sortie vidéo pour arriver à synchroniser l'image réelle avec l'image synthétique [9]. Cependant, si le médium d'affichage n'est pas opaque (projection, casque d'affichage semi-transparent) cette solution n'est pas possible. Azuma, dans un article de 1994 [7], proposait de prédire la position de la tête, quelques millisecondes à l'avance, à l'aide de la vitesse et de l'accélération. En combinant à une bonne optimisation des algorithmes et du matériel, il serait ainsi possible d'éliminer une très bonne partie des délais [82]. La figure 2.2 représente l'ensemble des délais accumulés

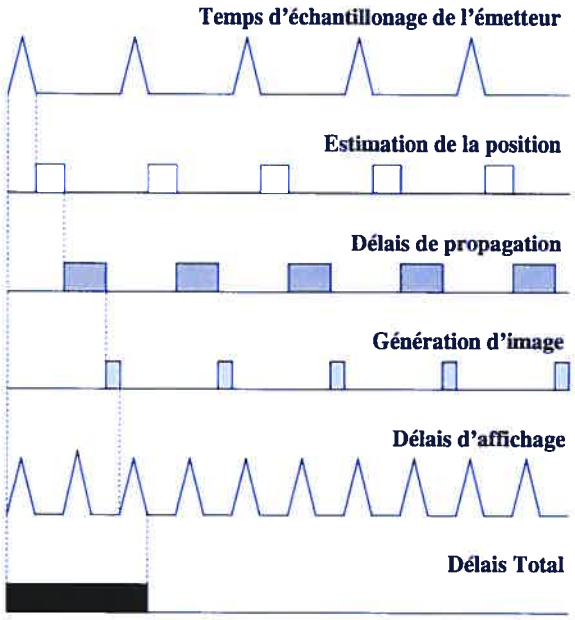

Figure 2.2. Liste des différents délais générés lors du suivi d'un objet dans l'espace par un système émetteur-récepteur quelconque. Inspiré des modèles de [2, 38]

entre la première estimation de pose et la fin de l'affichage de l'image qui lui est associée. Les distances ne sont pas proportionnelles et les temps de lectures/écritures sur les différents ports des appareils ne sont pas affichés. Il est simplement intéressant de noter l'importance d'utiliser des appareils rapides et des algorithmes optimisés.

Les lecteurs magnétiques sont souvent préférés aux autres types de lecteurs. Ils sont moins dispendieux et plus précis que la plupart des autres solutions du même genre [2]. Toutefois, plusieurs des appareils utilisés en situation chirurgicale sont sensibles à la présence de champs magnétiques trop puissants et l'utilisation d'émetteurs/récepteurs nécessaire au bon fonctionnement des lecteurs magnétiques sont autant de raisons qui rendent ce choix inapproprié.

2.2 Suivi Passif - Segmentation

La recherche passive de visage dans une ou plusieurs images aura motivé plusieurs publications au cours des dernières années. Plusieurs conférences portent même exclusivement sur ce sujet. La raison en est très simple, la complexité du problème à résoudre ainsi que les débouchés possibles. Par exemple, au niveau de la sécurité dans les aéroports, il serait intéressant de pouvoir identifier une personne rapidement, sans avoir à l'immobiliser pendant des heures. De façon plus intéressante au niveau technologique, des interfaces humain-machine basées sur le regard de l'utilisateur pourraient permettre de faciliter l'accès à certains systèmes aux personnes ayant des déficiances physiques. Les possibilités offertes ici sont tout à fait à la hauteur du problème à résoudre. Les visages sont non seulement très différents d'une personne à l'autre, mais ont aussi tendances à varier énormément en fonction des expressions d'un même sujet. De plus, dans un scénario général, il faudra tenir compte de l'occlusion, des conditions d'éclairage ansi que du positionnement du visage par rapport aux caméras. La figure 2.3 représente une idée de l'objectif recherché.

Les méthodes existantes pour retrouver un visage peuvent être classées en qua-

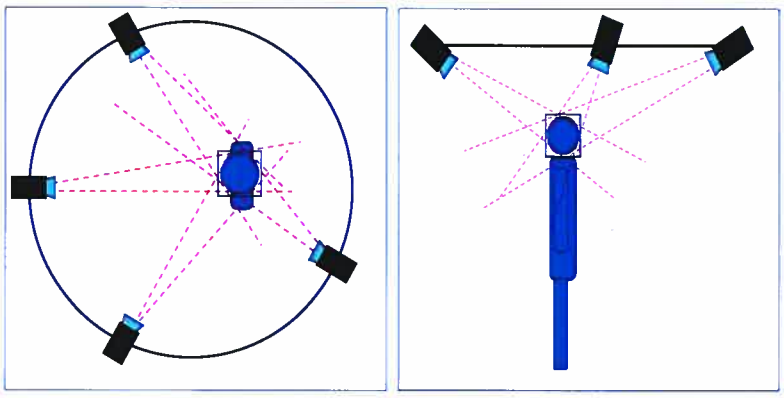

Figure 2.3. En utilisant plusieurs caméras, il est possible d'appliquer les principes de stéréo-vision pour retrouver la position précise d'une tête dans une scène 3D.

tre catégories [93]. Les systèmes experts reposent sur des règles prédéfinies entrées avant la simulation. Un "expert" décidera donc d'un ensemble de relations propres à tous les visages et à ses sous-parties qu'il donnera en entrée à l'algorithme. Lorsque le programme rencontre une région pouvant être un visage, ce dernier vérifiera l'applicabilité des différentes règles pour trouver le taux d'appartenance aux différentes catégories possibles (soit un visage ou un non-visage). Han *et al.* [35] par exemple, commence donc par retrouver l'ensemble des régions d'une image pouvant appartenir à des yeux. Ces régions sont ensuite sélectionnées selon qu'elles peuvent ou non faire partie d'un visage. Les différentes règles utilisées comparent les angles ainsi que les distances relatives entre les différentes régions sélectionnées pour décider de leur élligibilité. La recherche peut également se faire au niveau des éléments invariants du visage, la couleur ou la texture par exemple. Évidement, ces éléments auront tendance à évoluer au cours du temps, mais il reste possible d'établir des modèles dynamiques pouvant s'adapter au changement d'inclinaison ou d'illumination de la scène. Dans l'article de Yang *et al.* [91], un modèle de couleur de peau est établi au début de la recherche. Ce modèle s'adapte ensuite en fonction de la réponse par la caméra aux couleurs de la scène ainsi que les différentes variations au niveau de

l'illumination. La recherche par couleur peut se faire très rapidement. Les auteurs de [91] projettent simplement les trois coordonnées d'une couleur rouge-vert-bleu dans un domaine 2D de couleurs pures. Il semble que les visages humains se regroupent dans une même région du domaine chromatique, les différences de couleurs étant d'avantages liées à l'intensité, même entre différentes races. Le modèle se modifie ensuite en fonction d'une fenêtre temporelle de N images selon la formule :

$$\hat{c_k} = \Sigma_{i=0}^{N-1} \alpha_{k-i} c_{k-i} \tag{2.1}$$

où $\hat{c_k}$ représente la nouvelle couleur après adaptation, α_{k-i} est un facteur de poids et c_{k-i} est la moyenne estimée de la couleur c au temps k. Pour accélérer la recherche un modèle de prédiction de pose est également implanté. Ce qui permet d'atteindre des vitesses satisfaisantes pour une application en temps réel. Il est également possible de mener la recherche en fonction de modèles de comparaison. La recherche simple de caractéristiques modélisées s'avère efficace uniquement si le visage est exactement à l'endroit voulu dans l'image sans aucune occlusion. Il est possible d'utiliser des modèles variables pouvant être mis à différentes échelles et modifiés selon l'angle. Toutefois ces solutions sont très lentes et ne supportent pas du tout les occlusions. Si on tourne la tête jusqu'à ce qu'un oeil disparaisse, le programme aura beau modifier ses modèles, il n'arrivera pas à retrouver le visage. La recherche de caractéristiques étant faite de façon locale, les résultats restent plus précis qu'une recherche globale, comme dans le cas de la couleur. Il est donc intéressant de jumeler les deux pour améliorer les résultats. Spors *et al.* [78] recherchent premièrement les visages à l'aide d'un modèle de couleurs. L'image de fond est ensuite retirée grâce à un masque et les zones restantes sont validées par analyse en composantes principales (ACP) pour détecter les yeux. L'effet "yeux rouges", créé par la réflection sur la rétine de la lumière, a été utilisé pour faciliter la détection et ainsi le suivi du visage de quelqu'un. Quelques chercheurs [36, 43] ont travaillé à développer un système de caméras permettant d'éclairer les yeux de quelqu'un en infrarouge (IR) et d'en suivre la réflexion. En

alternant les images, illuminées, par des lampes IR, ou non, il est facile d'étudier la différence entre deux images consécutives pour retrouver les pupilles. Ce système est intéressant puisque simple et non-intrusif. La lumière IR ne dérange pas l'usager et le suivi peut se faire très rapidement. L'inconvénient, c'est que l'usager doit rester le plus souvent possible droit devant la caméra. Zhu *et al.* [101] ont combiné cette solution à un suivi par modèle d'apparence. Même si le système peut facilement retrouver les yeux lorsqu'il les perd, suite à un clignement ou par occlusion par exemple, la réponse de l'oeil aura tendance à varier beaucoup en fonction de l'éclairage ambiant. Les résultats sont intéressants, étant moins dépendants face à l'environnement, mais la recherche des pupilles reste restrictive quant à la taille du volume de travail et aux mouvements de l'utilisateur. Il serait intéressant de pouvoir profiter du grand nombre de caméras configurées en parallèle pour faciliter la recherche. Plutôt que d'utiliser une recherche 2D en parallèle sur chacune d'elles. Les recherches de Gorodnichy *et al.* [32] utilisent justement les correspondances épipolaires entre deux caméras pour suivre certains points en 3D. Les éléments à suivre, le nez, les arcades ou les yeux, doivent être sélectionnés manuellement au début de l'expérience. De plus, les recherches n'ont porté que sur un volume de travail très restreint. Ici, le chirurgien risque de passer le plus clair de son temps dans un endroit relativement compact, mais il serait dérangeant d'avoir à reinitialiser certains points à chaque fois que ce dernier sort de la zone d'intérêt. Dans notre cas, il serait probablement suffisant de ne connaître que la position de la tête du chirurgien. La position exacte du regard pourrait en être déduite, par généralisation ou encore en gardant un registre pré-calculé des dimensions de la tête de chacun des chirurgiens travaillant dans un hôpital donné. L'estimation d'un mouvement rigide, le déplacement de la tête, simplifie beaucoup les problèmes de recherche associés aux modèles déformables. En associant une modèle 3D à la tête, il est possible de suivre la position de la tête exactement grâce à l'étude du flux optique. Le modèle choisi influencera beaucoup l'efficacité de ce système. Plus le modèle est complexe et exact, plus le système sera stable, mais le travail

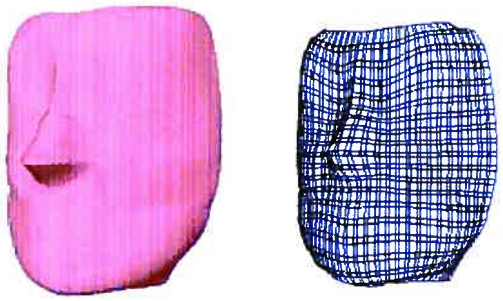

Figure 2.4. Modèle de recherche utilisé dans [97, 98] pour suivre la position d'une tête dans l'image. On peut facilement y reconnaître un visage, ce qui rend la mise en correspondance plus facile et résistante au bruit. Réimpression de [97].

demandé sera considérable. Il est donc essentiel de trouver un modèle suffisamment fiable pour ne pas perdre le lien avec la tête trop facilement, mais également simple pour pouvoir calculer la position le plus rapidement possible. Les modèles ont donc beaucoup évolué, en parallèle avec la puissance de calcul des ordinateurs. De simples surfaces planaires [15], les modèles utilisés ont évolué pour prendre des formes ellipsoïdales [11], plus représentatives de la forme d'une tête. Plus récemment, Zhang *et al.* [97, 98] ont utilisé des super-quadriques étendus (Extended super-quadrics, ESQ) pour simuler encore plus précisément la tête d'un sujet. Les ESQ sont des surfaces paramétriques en coordonnées polaires qui permettent de représenter facilement un modèle en 3 dimensions. L'ajout d'exposants permet d'éviter le problème de symétrie intrinsèque au super-quadriques. Ce modèle étant plus complexe, il réduit de beaucoup les ambiguïtés lors du suivi de la tête. On peut voir le modèle qu'ils ont utilisé dans la figure 2.4. Grâce à ce modèle, l'algorithme devient beaucoup plus tolérant face aux mouvements alloués au sujet, tout en restant relativement léger quant aux calculs à effectuer. Les occlusions étant un problème incontournable dans le cas du suivi d'une tête, une segmentation du mouvement est effectuée pour perme-

ttre de détecter les différentes régions visibles, ou non. L'algorithme de [98] supporte jusqu'à 50% d'occlusion en traitant ces zones cachées comme du bruit et en suivant les points de contour à l'aide d'un calcul de flux de segments. Leur objectif principal, ainsi qu'à leurs prédécesseurs était la fiabilité de la mise en correspondance. Les résultats qu'ils obtiennent sont donc très intéressants au niveau de la précision, mais le calcul de flux optique et du flux de segments rend le positionnement trop lent pour un travail en temps réel. Les travaux de Li *et al.* [48, 49] ont réussi à atteindre des vitesses d'environ 4 images par seconde en combinant deux algorithmes de recherche statistique. L'image est premièrement segmentée par recherche de visages propres (eigenfaces). En réduisant la dimension du problème cette recherche permet d'atteindre des vitesses impressionantes. Cette étude reste toutefois imprécise, la zone d'indécision étant relativement grande. En établissant deux seuils, le premier pour une décision positive, t_a (la zone est un visage à 99% sûr) et le second pour la négative, t_b (la zone n'est pas un visage à 99% sûr), il est possible d'exécuter une recherche plus précise dans le cas où la région étudiée tombe exactement entre les deux. Les auteurs ont donc implanté un système basé sur les machines à vecteurs de support (SVM), système qui est plus efficace pour séparer une image en différentes classes. La zone d'incertitude étant plus étroite pour cet algorithme, les points considérés ambigus par la recherche de visages propres réussissent généralement à être classés avec plus de certitude grâce à la SVM. Il serait cependant sous-optimal de traiter l'image complète par SVM puisque cette dernière est sensiblement plus lente, pour les mêmes résultats, lorsque les points sont classés avec certitude (à l'extérieur de la région définie par t_a et t_b).

2.3 Solution Proposée

Les travaux qui ont été étudiés dans ce chapitre n'affiche pas vraiment des résultats satisfaisants pour notre système. L'utilisation en paire de visages propres et de SVM,

proposée par Li *et al.* [48, 49], combine fiabilité et vitesse, mais reste malgré tout en dessous de 5 images par seconde. Ce qui reste loin du 30 images par secondes souhaité pour obtenir des résultats en temps réel.

L'utilisation d'un filtre de Kalman apparaît comme le choix le plus intéressant. Le principe récursif du filtre ainsi que ca stabilité face au bruit en font un des filtres les plus puissants dans le genre. La résolution d'un système linéaire pour résoudre le problème de correspondance entre les points de deux images (aux temps t et $t + \Delta t$) est suffisament rapide, mais peut causer des problèmes de divergence. Toutefois, il est possible, sans perte de généralités de supposer que les mouvements du chirurgien seront continus et relativement lents. Le problème n'est donc pas aussi grâve qu'il peut le sembler. L'inconvénient principal de la recherche par points saillants, c'est qu'il faut savoir quoi chercher. Le filtre de Kalman génère un champs d'estimations de mouvement éparse, définissant le mouvement seulement pour un ensemble de points déterminés à l'avance [85]. Ces points doivent idéalement être facilement discernables et non co-planaires, de façon à garantir un suivi rapide et efficace. Il est important de rapeller que notre objectif premier est la simplicité, c'est-à-dire que nous souhaitons minimiser l'intrusion de notre système dans le protocole chirurgical. Le chirurgien pourrait simplement porter un casque aux couleurs voyantes, comme le vert, pour permettre à notre algorithme de le retrouver rapidement. Il serait toutefois impossible de connaître précisément l'orientation de sa tête et donc la direction de son regard. Ce problème peut facilement être contourné en assumant que le chirurgien regardera toujours dans la direction du patient. La projection sera inutile de toute façon dans le cas contraire.

Sans vraiment rajouter à la complexité de notre solution, il pourrait être intéressant d'utiliser un système à lumière infra-rouge. Les caméras actuelles sont très sensibles à l'éclairage infra-rouge de basses fréquences. Nous pourrions utiliser des bandes réfléchissantes, apposées en croix sur le casque du chirurgien, couplées à un projecteur infra-rouge. La lumière ainsi réfléchie pourrait facilement être détectée à

l'intérieur des caméras filmant la scène. Une fois la position et l'orientation de la tête connue, il sera facile d'estimer précisément la position des yeux du chirurgien. Une simple base de données des différents chirurgiens suffiraient pour déterminer la distance exacte entre le sommet de la tête et le centre focal des yeux. En utilisant une lumière invisible à l'oeil, nous nous assurons de ne pas déranger les utilisateurs en les éblouissant ou en créant des ombres pouvant nuire à l'estimation des profondeurs par exemple. Une étude de la distribution des couleurs dans les différentes images recueillies pourraient aider à éliminer les cas ambigus, sans nécessairement ralentir le système. Le suivi effectué dans [91] se faisait déjà en temps réel en 1996.

Une reconstruction du mouvement rigide pourrait également renforcer la recherche. Si une transformation globale est mise à jour à chaque instant, la position finale de la tête pourrait être facilement recalculée. Il suffirait d'apliquer cette transformation 3D au modèle de tête dont on souhaite effectuer le suivi, à partir d'une position initiale donnée. Cette solution risque cependant d'échouer lors d'une opération trop longue. La position étant toujours calculée relativement à la position précédente, les erreurs ont tendances à faire boule de neige et à donner des résultats complètement faux après un certains laps de temps. Il serait cependant possible de réinitialiser la position durant une chirurgie, advenant le cas où les résultats seraient faux au delà d'un certain seuil. En fait, il suffirait de rajouter ces deux systèmes d'estimation de pose au filtre de Kalman et de combiner les trois lectures avec des poids variant en fonction de la vraisemblance de leurs résultats. Les filtres de Kalman sont justement conçus pour facilement concilier plusieurs sources de lectures différentes, avec des modèles de bruits et des taux de vraisemblances différents. Il est possible de se référer au livre de Trucco et Verri [85] pour une introduction sur les filtres de Kalman. Une idée d'un modèle de suivi par filtrage multiple est schématisé dans la figure 2.5.

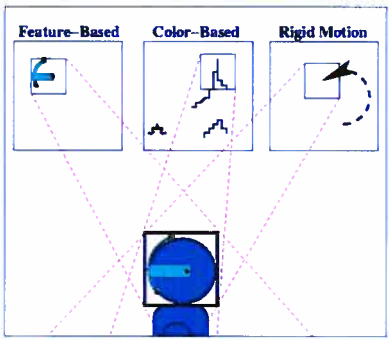

Figure 2.5. À l'aide d'un filtre de Kalman, il est possible de coupler différents types d'analyses d'une scène pour améliorer les résultats lors d'un suivi en 3D.

Chapitre 3

ESTIMATION DE PROFONDEUR

Pour que la projection soit parfaitement ajustée sur le patient, il est essentiel de connaître la forme exacte de son corps durant toute l'opération. Il devient donc nécessaire de construire un système de modélisation 3D efficace permettant d'enregistrer, en temps réel, la forme du corps avec une précision élevée. Un millimètre d'erreur durant une opération critique pourrait être fatal. Nous pouvons, malgré tout, supposer qu'à l'exception de la zone d'intervention, le patient ne sera ni déplacé, ni déformé de façon considérable. Ce dernier étant sous anesthésie, il sera donc plutôt calme et sa respiration sera contrôlée. De plus, bien que nous projèterons des informations reliées à l'opération sur une bonne partie du corps, la région nécessitant une haute précision ne sera pas plus grande que la zone de coupe elle-même (relativement petite). Il serait donc possible d'étudier un système à deux vitesses, analysant précisément une petite région de haute importance tout en gardant un "oeil" sur le reste du corps. Le modèle complet pourrait être actualisé seulement lorsqu'un déplacement supérieur à un certain seuil prédéterminé serait enregistré.

Il existe plusieurs méthodes efficaces pour reconstruire une scène en 3 dimensions. Cependant, certaines ne s'appliquent pas vraiment à notre situation. Par exemple, Dekker *et al.* [23] utilisent un numériseur développé par Hamamatsu Photonics qui permet de créer un modèle complet d'un être humain se tenant debout à l'intérieur d'une sorte de baril. Dans [65], Pintavirooj et Sangworasil utilise une caméra fixe qui étudie un modèle qui tourne de manière contrôlée. Ces solutions sont intéressantes, parce que relativement rapides, mais surtout très précises. Toutefois, il serait difficilement imaginable de les appliquer à une personne alitée et encore moins durant une opération chirurgicale. Les exigences aux niveaux des performances, rapidité et

précision, ainsi que la situation très spécifique de notre étude nous forcent donc à éliminer plusieurs solutions d'emblée.

Ce chapitre sera divisé en deux parties décrivant les principales solutions offertes en vision, avec leurs avantages et inconvénients respectifs. Les algorithmes ont été classés en fonction du nombre de caméras utilisées, mono ou stéréo, pour reconstruire la scène.

3.1 Esimation à Vue Unique

Il est reconnu que nous arrivons à lire la profondeur d'une scène grâce aux quelques centimètres séparant nos deux yeux horizontalement. Toutefois, même en regardant une image plane (2D) ou en fermant un oeil, nous avons toujours une bonne idée de la structure 3D sous-jacente. Il est donc évident que certains indices peuvent être retrouvés dans une image en 2 dimensions pour inférer la profondeur. Plusieurs expériences ont été menées pour retrouver les processus utilisés par le cerveau pour arriver à recréer une scène avec la précision qu'on lui connaît. Les occlusions, les ombres, la perspective linéaire ou la taille relative des objets dans une scène sont autant d'indices qui nous permettent de visualiser la profondeur dans une image. C'est en se basant sur ces constations que certaines solutions ont été avancées pour calculer la profondeur d'une scène à partir d'un point de vue unique. L'image 3.1, tirée de [66] montrent différentes situations où il est possible de tirer de l'information sur la profondeur de la scène. Nous discuterons ici de quelques unes des techniques utilisées pour modéliser des scènes à partir de ce genre d'images. Notre but n'étant pas de faire un résumé de toutes les solutions existantes, mais plutôt de trouver celle qui s'appliquerait le mieux à notre problème, nous nous contenterons ici de faire ressortir les points intéressants et les inconvénients de ces méthodes.

Figure 3.1. Il est possible de trouver des indices sur la géométrie d'une scène à partir de différents éléments d'une image. En haut à gauche, éclairage. En haut à droite, ombre et silhouette. En bas à gauche, texture. En bas à droite, profondeur de champs [66].

3.1.1 Profondeur de champs

La profondeur d'une scène peut être calculée à l'aide de plusieurs images prises d'une même caméra, mais avec des mises au point différentes. Si on connaît les paramètres de la caméra, il est possible de regrouper une série d'images, en comparant les parties floues, pour construire une carte en 3 dimensions de la scène. Un exemple d'algorithme du genre est décrit dans [27]. La partie la plus longue de l'algorithme est d'entraîner le processus pour trouver les opérateurs associés aux différentes profondeurs. Une fois ce travail fait, il devient possible de suivre n'importe quelle scène en temps réel. En effet, si on calcule un certain nombre d'opérateurs $H_s^{\perp}$, associé à des profondeurs s précises, il suffira de trouver celui qui minimise la fonction

$$\Psi(s) = \|H_s^{\perp} I\| \tag{3.1}$$

avec $s \in \mathcal{S}$, l'ensemble des profondeurs pré-calculées, pour chaque point de l'image I. La solution optimale à ce problème représentera grossièrement la carte de profondeurs de la scène. Chaque pixel étant traité séparément, en fonction d'un certain nombre de ses voisins, il est possible de construire un système parallèle très efficace pour le calcul

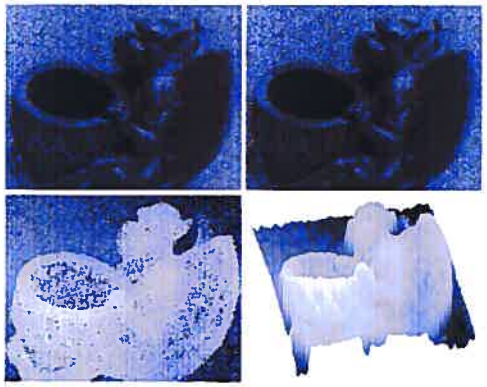

Figure 3.2. Les images du haut sont deux des images d'entrée de l'algorithme avec différents focus. Les images du bas représentent la reconstruction en niveau de gris (gauche) et en 3D (droite) de la scène.

des profondeurs. Toutefois, le fait de calculer un nombre pré-déterminé d'opérateurs discrétise la profondeur et, malgré les solutions offertes par les auteurs de [27] (sur-échantillonage, interpolation linéaire ou bilinéaire, ...), les résultats qu'ils ont obtenus affichent une erreur moyenne (RMS) de presque 4 mm (voire figure 3.2). Ces résultats ne sont donc pas suffisament précis pour être applicables dans notre projet.

3.1.2 Éclairage et Ombres

Jusqu'à récemment, la profondeur calculée à partir de l'ombrage dans une scène offrait des résultats très décevants. En effet, le nombre de suppositions préalables au traitement rendait la mise en application de ces algorithmes pratiquement impossible. La reconstruction par ombrage est un problème mal posé. Il existe en effet une infinité de cartes de normales pouvant recréer la carte d'intensités analysée [58]. En d'autres mots, le nombre de minimums locaux de la fonction d'énergie rend son étude ardue. Dans [63], l'auteur étudie seulement les surfaces lambertiennes, éclairées par une lumière ponctuelle (à l'infinie) et sans auto-réflexion. De plus, pour réussir à obtenir des résultats quelconques, il était nécessaire de connaître la carte

de radiance de la scène ou d'avoir l'albédo précis de la surface étudiée [21]. Ces circonstances permettaient d'obtenir une solution acceptable, mais elles ne se retrouvent pas dans la réalité. Plus récemment, Stewart and Langer [80], ont implanté un système d'illumination global par radiosité semblable à celle d'écrite dans Foley *et al.* [28] :

$$B_i = E_i + \rho_i \sum_{i \leq j \leq n} B_j F_{j-i} \tag{3.2}$$

avec B_i et B_j la radiosité des sections i et j de la scène, E_i l'énergie transmise de i, ρ_i sa réflectivité et F_{j-i} le facteur de forme définissant la relation entre les deux sections. Ce genre de système, efficace pour recréer des images complexes et réalistes, devient rapidement très lourd et ne s'applique absolument pas en temps réel. D. Nandy et J. Ben-Arie [57, 58] ont, quant à eux, implanté un réseau de neuronnes qui apprend à reconnaître des modèles précis, fenêtres 7x7 de l'image [57] ou parties du visage [58], et à les associer à des cartes de profondeurs déjà construites. Ce principe est intéressant, pouvant probablement être adapté pour fonctionner en temps réel, puisqu'une fois l'entraînement exécuté, le travail à faire reste relativement court. Toutefois, le fait que le système doit avoir une idée précise *a-priori* des formes analysées nous fait perdre trop de généralité. La forme du patient ne doit souffrir d'aucune contrainte, puisque ce dernier peut souffrir d'une blessure ouverte modifiant globalement sa forme.

Finalement, pour ce qui est des résultats, la reconstruction par ombrage supporte très mal les discontinuités [80] et la complexité des calculs ne la rend intéressante que pour des images fixes reconstruites durant un post-traitement. Malgré tout, [21] obtient des résultats intéressants en jumelant la stéréo-vision et la reconstruction par ombrage. La figure 3.3 montre des résultats tirés de [57] grâce à leur algorithme par réseau de neuronnes et minimisation par moindres carrés. Toutefois, ces solutions ne pourraient être utilisées ici puisqu'elle sont toutes trop lentes, mais surtout parce que l'éclairage d'une salle d'opération est construit pour ne pas laisser d'ombres pouvant nuire au travail du chirurgien.

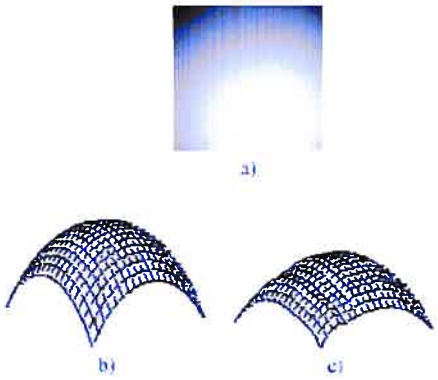

Figure 3.3. a) Image synthétique d'une sphère. b) Forme originale. c) Forme reconstruite par superposition de fenêtres 7x7. La solution n'étant pas très précise, la sphère apparaît légèrement applatie.

3.2 Stéréo-Vision

La stéréo-vision calcule la profondeur à partir de deux images ou plus, prises en même temps, mais à partir de point de vue différents [37]. Ce principe est en fait une imitation du système visuel humain, qui retrouve les profondeurs, entre autres, à l'aide de la différence mesurée entre les images enregistrées par les deux yeux. Le travail effectué par le cerveau pour obtenir les bonnes profondeurs est loin d'être simple et la majorité des algorithmes développés en stéréo-vision souffrent encore de graves lacunes au niveau des performances. Nous traiterons premièrement des solutions utilisant uniquement des caméras, deux ou plus. Nous présenterons ensuite quelques solutions de lumière-structurée, où une des caméras est remplacée par un projecteur.

3.2.1 Caméra-Caméra

Le principal problème de la stéréo-vision est la mise en correspondance entre les différentes images. Trouver les coordonnées de chacun des pixels d'une image I_L à

l'intérieur d'une deuxième image I_R prend rapidement des proportions gigantesques. Pour augmenter encore plus le niveau de difficulté, la présence de bruit, inévitable lorsqu'on travail avec des images réelles, rend le problème de correspondance encore plus laborieux. Heureusement, certaines contraintes peuvent être appliquées pour accélérer la recherche. Ainsi, il est possible de remplacer la recherche 2D dans l'image par une recherche en une dimension à l'aide de la correspondance épipolaire. La figure 3.4 [66] donne une bonne intuition de la géométrie épipolaire. Tous les points d'un plan L passant par le centre des caméras C et C' seront projetés sur les droites l et l'. Il suffit donc de parcourir la ligne l' (en 1D) pour retrouver la projection du point M, peu importe sa profondeur dans la scène. Certaines contraintes peuvent ensuite servir à faciliter la recherche [66] :

Contrainte d'ordre Les points consécutifs d'une ligne épipolaire l seront projetés dans le même ordre sur la ligne l'.

Réciprocité La relation associant le point p au point p' est vrai dans les deux sens. À l'exception des occlusions, le point p' sera nécessairement associé au point p.

Les images de la figure 3.5, prise à l'aide d'un appareil photo Kodak DC-290, montrent la correspondance épipolaire entre les deux photos. On peut voir que chaque ligne passe exactement[1] par les mêmes points dans les deux images. En connaissant les coordonnées d'un point dans les deux images, il est possible d'en obtenir la profondeur par triangularisation. Pour accélérer encore plus la recherche, il est possible de rectifier les images de façon à aligner horizontalement les lignes épipolaires. La recherche de correspondance peut donc se faire directement en 1D. En fait, la majorité des algorithmes présentés plus loin supposent que les images d'entrées ont déjà

[1] Évidement, la présence de bruits ainsi que la simplicité de l'algorithme utilisé nous incitent à tolérer un certain niveau d'erreur. Il est possible d'obtenir de meilleurs résultats en minimisant l'erreur par itérations entre les deux images.

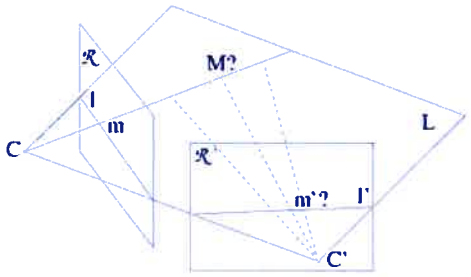

Figure 3.4. Tous les points du plan L sont nécessairement projetés sur les droites l et l' des plans rétiniens respectifs R et R'

Figure 3.5. Chaque droite dans l'image de gauche est associée à son équivalente dans l'image de droite. La géométrie épipolaire nous garantie que tous les points sur une des lignes blanches de l'image de gauche se retrouveront nécessairement sur la droite équivalente de l'image de droite, à l'exception des cas inévitables d'occlusions.

été alignées de la sorte. La figure 3.6 montre le résultat d'une rectification planaire des images de la figure 3.5.

Il existe une panoplie impressionnante de solutions proposées pour résoudre ce problème de correspondance. Ces solutions peuvent se classer en deux catégories, locales et globales. Les méthodes locales cherchent à trouver une solution pour chaque pixel, souvent à l'aide de son entourage immédiat, en choisissant une valeur minimisant une simple somme d'erreurs absolues ou au carré (SSD ou SAD). Tandis que les méthodes dites globales vont plutôt tenter de trouver une solution pour l'ensemble

Figure 3.6. Il est possible de modifier les images étudiées afin d'aligner horizontalement toutes les lignes épipolaires. La recherche en sera ainsi significativement accélérée.

de l'image, en minimisant une fonction d'Énergie :

$$E(f) = E_{smooth}(f) + E_{data}(f) \tag{3.3}$$

où E_{smooth} représente le terme de lissage de l'image et E_{data} définit la correspondance avec l'image d'entrée. La principale différence entre les algorithmes globaux vient du choix de fonctions pour calculer ces deux termes. Szeliski *et al.* ont établi une série de critères d'évaluation pour classer les différentes méthodes existantes [73, 81]. Nous étudierons donc ici celles qui apparaissent les plus intéressantes au niveau précision et vitesse.

Méthodes globales

En règle générale, les algorithmes globaux réussissent à reconstruire les profondeurs de la scène avec beaucoup plus de précision que leurs homologues locaux. Toutefois, cette précision nous vient à un coût non négligeable, le temps de calcul. Par exemple, la résolution par coupe de graphe ("Graph-Cuts") de Zabih *et al.* [16] obtient l'une des meilleurs notes au classement général selon Szeliski [73]. Toutefois, ces résultats prennent facilement plus de 3 minutes pour être calculés. En fait, à part pour quelques cas spéciaux, la méthode de coupures de graphes tente de résoudre un problème NP-complet. Les résultats de cet algorithme semblent très intéressants, affichant

un pourcentage d'erreur d'environ 9% aux points de discontinuité et de seulement 1.5% ailleurs dans les images utilisées dans [73]. Nous pouvons tolérer une certaine flexibilité aux points de discontinuité étant donné que notre algorithme se concentrera spécialement sur les surfaces plus lisses. Malgré tout, bien que les auteurs de [16] soient fiers d'afficher une accélération de leur algorithme, chaque image nécessite entre 150 et 400 secondes de traitement[2].

Les solutions de mise en correspondance globale se basent toutes sur le même principe de base : à partir de deux ensembles de points, représentés à l'aide d'un graphe, nous cherchons à trouver le chemin le moins coûteux qui permet d'associer les points des deux ensembles entre eux. Cox *et al.* [20] effectuent cette recherche linéairement sur les droites épipolaires des caméras. En utilisant la programmation dynamique, avec certaines contraintes au niveau de la disparité maximale, ces derniers arrivent à accélérer la recherche pour obtenir une fonction d'ordre $O(N)$, où N représente le nombre total d'éléments dans une image. Une approche similaire est étudiée par Roy et Cox [70]. Le principe est fort semblable, mais plutôt que de résoudre le problème de minimisation de l'énergie à l'intérieur du graphe par programmation dynamique, les auteurs de [70] utilisent un algorithme de flot maximum. L'algorithme est toutefois beaucoup plus rapide, mais reste cependant impossible d'arriver à du temps-réel avec des temps toujours au dessus d'une minute par image. Un reproche pourrait être fait à tous les algorithmes globaux, la solution étant retrouvée par une succession d'itérations, relativement courtes, il n'est pas vraiment possible de parallèliser le calcul. Du moins, le gain ne serait pas significatif.

La figure 3.7 affiche les résultats des algorithmes de "Graph-Cuts" et de "Max-Flow" appliqués à l'image de l'université de Tsukuba [73].

[2] Les temps donnés ici sont ceux tirés de [16]. Les auteurs de [73] affichent quant à eux un temps de 24 secondes pour l'image de Tsukuba avec le même algorithme. La dimension des images n'est toutefois pas précisée dans [16].

Figure 3.7. (Gauche) Image de reférence de l'université de Tsukuba. (Milieu) Résultats obtenus par la méthode de Graph-Cuts (24 secondes). (Droite) Résultats obtenus par la méthde de Max-Flow.

Fenêtre de corrélation

Les solutions locales sont dites à fenêtre de corrélation parce qu'elles calculent la disparité de chaque pixel indépendamment en fonction d'un nombre restreint de voisins. Le nombre de voisins détermine en fait la taille de la fenêtre. L'hypothèse de lissage est implicite à la fenêtre de corrélation, contrairement aux méthodes globales où cette même hypothèse est décrite par le terme E_{smooth}. Une simple sommation des différences absolues ou au carré (SSD, SAD) des valeurs de la fenêtre suffit pour retrouver une approximation valide. Ce calcul favorise donc le lissage en encourageant toutes les valeurs internes de la fenêtre à avoir la même valeur. Les algorithmes locaux ont d'ailleurs tendance à trop lisser les images, de sorte qu'un flou est souvent ajouté au résultat final. En échange de cette perte de précision, les algorithmes locaux peuvent cependant assurer une convergence, et ce, de façon beaucoup plus rapide que les algorithmes vus précédement.

La solution trouvée localement étant rapide, mais bruitée, ces algorithmes vont souvent être utilisés comme étape de départ aux algorithmes globaux qui nécessitent une bonne approximation de départ pour converger. Les auteurs de [37] réussissent à améliorer grandement leurs résultats en modifiant légèrement l'algorithme de base.

Ainsi, lorsqu'une discontinuité est rencontrée à l'intérieur d'une fenêtre de corrélation, cette dernière sera subdivisée de façon à minimiser les risques d'erreur. Ensuite, si la fonction de corrélation de la fenêtre comporte trop de minimums locaux (plusieurs discontinuités à l'intérieur d'une même fenêtre) et qu'aucun n'est vraiment prédominant, la valeur à ce point sera simplement annulée. Ces cas étant majoritairement causés par le bruit, la perte de quelques "bons" pixels est justifiée par une diminution sensible des erreurs. En effet, une réduction de la moitié du nombre d'erreurs est enregistrée après l'application de ces méthodes. Malgré tout, l'algorithme développé dans [37] n'affiche que 80% de certitude, tel qu'illustré dans la figure 3.8. Mühlmann *et al.* [55] proposent sensiblement les mêmes solutions pour accélérer le traitement, tout en améliorant les performances. Plutôt que d'utiliser l'image en tons de gris, ces derniers travaillent directement avec des images en couleurs. La justification vient du fait que ces images sont censées améliorer de 20% à 25% le ratio signal-bruit. Leurs calculs se font donc sur les trois canaux, rouge, vert et bleu simultanément. Toutefois, les résultats qu'ils affichent sont plutôt décevants, comme on peut le voir dans la figure 3.8, et ce pour à peu près le même temps de calcul que dans [37]. L'image d'une dimension de 384x288 avec 20 niveaux de disparité a été calculée en 218 ms. Il est à noter que la majorité du développement de ce dernier algorithme s'est fait dans le but d'une implémentation matériel, ce qui contrevient à notre objectif d'utiliser du matériel commercialisé et peu dispendieux.

N-Caméra

La mise en correspondance de point peut être grandement facilité si on ajoute des vues supplémentaires. En effet, utiliser plusieurs caméras, plutôt que seulement deux, nous permet d'avoir plus d'informations pour chaque point de l'image. De plus, multiplier les points de vue réduit les zones aveugles où aucune information ne peut être retrouvée. Les calculs supplémentaires ne sont pas aussi encombrants qu'on pourrait s'y attendre, d'autant plus que la majorité du calcul peut être faite en par-

Figure 3.8. (Gauche) Référence de profondeur pour l'image de l'université de Tsukuba. (Centre) Résultat obtenu par Hirschmüller [37] avec fenêtres multiples, filtrage d'erreur et correction des bordures (discontinuités en noir). (Droite) Résultat obtenu à partir d'une image couleur par [55] avec fenêtre 7x7 et filtrage.

allèle. En augmentant le nombre de points de vue, on aggrave cependant le problème d'occlusion. Les points seront plus visibles de façon générale, mais plusieurs ne seront visibles que dans une fraction des images. Plus il y aura de caméras, plus le problème d'occlusion sera dérangeant. Une solution intéressante serait de n'utiliser qu'une partie des vues pour calculer la profondeur à chaque point [42]. En calculant une SSD pour chaque vue et en gardant seulement les meilleurs résultats, on ne garde que les images où le point étudié est visible, réduisant ainsi les chances d'erreurs. D'autres améliorations sont suggérées par Kang *et al.* [42] pour renforcer les résultats, comme d'utiliser des fenêtres de corrélation à taille variable. Ce qui permet d'utiliser plus de points lorsque la région n'est pas texturée et qu'il y a peu d'information de disponible. Le contraire est vrai également, dans une région avec beaucoup de fréquences élevées, il est possible de subdiviser la fenêtre afin de conserver les discontinuités au maximum, un peu comme dans [37]. La vitesse n'étant pas leur objectif premier, les auteurs de [42] renforcent leurs résultats à l'aide d'un algorithme global de coupures de graphe. Cette solution n'est pas suffisamment rapide pour nous, mais les résultats sont très intéressants. Une reconstruction de l'université de Tsukuba est montrée à

Figure 3.9. Calcul par corrélation avec plusieurs caméras, utilisation de la meilleure moitié des fenêtres ("temporal selection") et optmisation par coupures de graphe.

la figure 3.9.

L'approche décrite dans [56] est davantage axée sur la vitesse, permettant à un usager une immersion plus prenante. Leur système parvient à calculer une carte de disparités en approximativement 500 ms[3]. Sans être en temps réel, obtenir 2 ou 3 images par seconde serait probablement satisfaisant dans notre cas. La correspondance est calculée entre deux images par une fonction de "Modified Normalized Cross-Correlation" (MNCC), plus complexe qu'une simple SSD, mais aussi plus précise. On confirme (ou infirme) la disparité calculée par la MNCC à l'aide de la contrainte trifocale [66], en projetant ce point dans une troisième caméra. L'utilisation d'une fonction de corrélation plus complexe permet d'éviter d'avoir recours à une méthode d'optimisation globale dont la solution serait beaucoup plus longue à calculer. Cette méthode affiche cependant un taux d'erreur d'environ 3 mm. Malheureusement, les auteurs de [56] ne démontrent pas l'efficacité de leur système sur les images de l'université de Tsukuba, ce qui rend la comparaison difficile. Toutefois, le taux d'erreur de leur algorithme est également d'environ 3 mm[4].

[3] Évidement, l'utilisation de 5 quad-Pentium III 550 MHz aide quelque peu à accélérer le traitement.

[4] Erreur médiane.

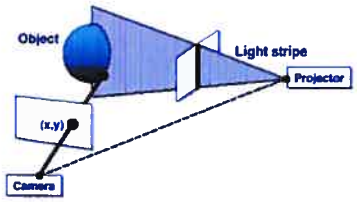

Figure 3.10. Principe de lumière structurée, le plan de lumière projeté intersecte avec l'objet en un point précis, facilement retrouvable à partir d'un capteur (caméra).

3.2.2 Caméra-Projecteur

La complexité du problème de correspondance en stéréo-vision vient en grande partie de l'absence de connaissance *a priori* à propos de la scène analysée. En ne posant aucune contrainte au sujet analysé, on crée un système tout à fait général, toutefois, on ne peut se fier à aucun indice pour accélérer le traitement. En ajoutant un projecteur au système, on construit une projection qui crée des caractéristiques facilement détectables dans la scène [22]. Connaissant certaines informations sur la scène, la correspondance se fait donc de façon plus rapide, sans perdre de généralité. La figure 3.10 montre un exemple de reconstruction à l'aide d'une caméra et d'un laser balayant la surface. Plusieurs techniques ont été étudiées pour calculer la profondeur à l'aide de différents modèles de projection. Les premiers à atteindre des vitesses intéressantes sont Hall-Holt et Rusinkiewicz [34] en 2001. Nous ne traiterons pas des solutions précédentes, généralement trop lentes, mais une bonne revision de la littérature sur le sujet se retrouve dans [95]. L'idée principale est qu'en projettant plusieurs bandes distinctes, plutôt qu'une seule, chaque image apporte plus d'information pour recalculer la profondeur, jusqu'à idéalement reconstruire toute la scène à partir d'une seule image.

Rusinkiewicz *et al.* [34, 72] projettent une série de bandes dont les frontières

varient de façon unique entre chaque image. Les motifs à projeter sont calculés par une recherche du chemin le plus long dans un graphe. Les points du graphe représentent les transitions entre les bandes et les arcs définissent, quant à eux, les conditions garantissant un minimum de frontières fantômes, ou invisibles. En recherchant les transitions entre les bandes, plutôt qu'une bande en particulier, il est possible d'atteindre une précision très élevée. Toutefois, l'algorithme de correspondance entre les bandes étant simpliste, le système ne tolère pas de mouvements brusques et n'est pas efficace lors de changements généraux d'illumination [96]. Leurs résultats sont cependant intéressants, réussissant un calcul des profondeurs à une vitesse de 60 Hz pour des images de 640 par 240 points. Dans [72], la surface est recalculée au rythme de 10 images par seconde, avec une erreur causée par le bruit d'environ 0.1 mm. L'échantillonage se fait en dessous d'un millimètre dans une région de 10 cm^3. Ces dimensions sont faibles, mais s'appliqueraient bien à la zone opératoire.

L'utilisation de bandes en couleurs faciliterait la tâche de mise en correspondance, mais entraîne également un certain nombre de problèmes. Ainsi, pour pouvoir retrouver les correspondances parfaitement, la scène ne devra pas modifier les couleurs entre celles projetées et celles lues par la caméra. Dans le cas contraire, il faudra avoir une idée précise de l'albédo couleur de la scène, c'est-à-dire la "fonction" appliquée par la surface sur les couleurs projetées. Il est à noter que les tests effectués dans [95] et [75] démontrent que la peau humaine ne cause pas de problème à ce niveau. Bien que leurs tests n'aient portés que sur des sujets de couleur blanche, une peau plus foncée ne causerait pas vraiment de problèmes supplémentaires. Yang et Waibel [90] ont d'ailleurs démontré que l'ensemble des couleurs de peau humaine se retrouvent confinées dans une même région du domaine chromatique. Zhang *et al.* [95] effectue son analyse sur chacun des trois canaux simultanément pour trouver les gradients de chaque couleur. Dans [75], la recherche se fait plutôt au niveau de la teinte (H du système de coordonnées HSV), ce qui accélère peut-être la recherche, les auteurs ne donnant pas de détails sur leurs résultats. Le problème de bruit est cependant

plus complexe avec les images en couleurs, les changements d'intensité étant moins absolus. Il faudra également tenir compte d'un problème de diaphonie lors de la projection de motifs couleurs. En pratique, il est difficle d'éviter que les différents canaux de couleurs ne se superposent lors de la lecture par la caméra. Pour réduire ce problème il faut calculer la matrice de diaphonie chromatique (MDC) définissant la relation entre la caméra et le projecteur. À partir de cette matrice, les auteurs de [95] définissent la relation entre la couleur s d'un point dans la caméra et la couleur p de son équivalent dans le projecteur comme :

$$\underbrace{\begin{pmatrix} s^r \\ s^g \\ s^b \end{pmatrix}}_{s} = \underbrace{\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} \rho^r & 0 & 0 \\ 0 & \rho^g & 0 \\ 0 & 0 & \rho^b \end{bmatrix}}_{F} \underbrace{\begin{pmatrix} p^r \\ p^g \\ p^b \end{pmatrix}}_{p} + \underbrace{\begin{pmatrix} o^r \\ o^g \\ o^b \end{pmatrix}}_{o} \tag{3.4}$$

où X est la MDC, F définit l'albédo du point dans la scène et o est l'éclairage ambiant pour ce même point, [95]. Il est à noter que dans cette formule, l'albédo de la scène autant que l'éclairage ambient, sans projection, doivent être connus. Ce calcul ne s'applique donc pas à une scène dynamique, [96]. Le volume de reconstruction est sensiblement plus grand que dans [72] avec une vue de 40 x 25 cm pour des images de résolution 864 x 576, avec une erreur RMS supportable de 0.2 mm. La résolution des images analysées est en fait simplement bornée par la résolution maximale des projecteurs, la résolution des caméras étant, en règle générale, toujours plus élevée.

Plutôt que de projeter des bandes parallèles, Scharstein et Szeliski [74] ont tenté d'utiliser un signal sinusoïdal couvrant la totalité de l'image. La couleur de l'image finale au point (x, y) étant définie selon une formule prenant en compte l'albédo couleur de la scène, l'intensité du motif projeté, ainsi que la fréquence et la phase du signal [74]. Toutefois, les auteurs eux-mêmes concluent que ce motif est trop sensible aux déformations optiques à l'intérieur de la caméra et du projecteur ainsi qu'aux inter-réflexions dans la scène. La méthode de [74] montre la précision qu'il est

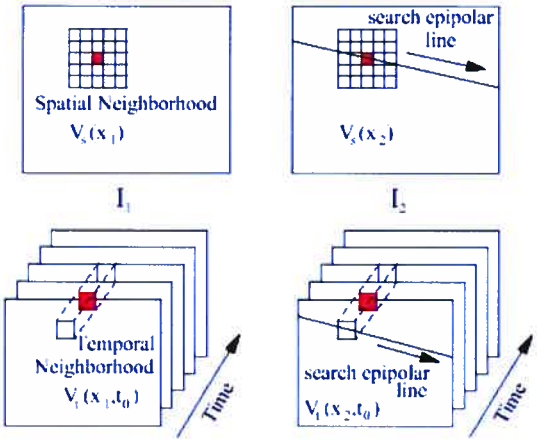

Figure 3.11. Voisinage spatial (gauche), versus voisinage temporel (droite). L'objectif est de trouver l'agencement idéal des deux pour une meilleur reconstruction. La recherche épipolaire se fait spatialement pour un voisinage semblable et temporellement pour une variation temporelle semblable [22].

possible d'obtenir avec un tel système, leurs résultats servent en effet comme base de comparaison pour tester d'autres algorithmes. Ils prennent quelque 80 images pour une reconstruction avec des temps d'exposition variant de 0.1 a 0.5 seconde, ce qui élimine cependant toute chance de traitement en temps réel.

Dans la littérature plus récente, les articles [22, 96] proposent une approche temporelle pour recontruire des scènes animées. Une simple généralisation de la fenêtre de corrélation vers une recherche en 3D, avec le temps comme troisième dimension, permet d'appliquer ce principe à la majorité des algorithmes existants. L'image 3.11 [22] donne une intuition de la recherche effectuée pour un voisinage spatial V_s et temporel V_t. Il est possible de combiner ces recherches pour construire un voisinage V_{st} combinant ces deux approches. Le calcul de corrélation deviendra une somme de SSD (SSSD). Aucun résultat précis n'est donné, mais les exemples montrés par les auteurs sont très prometteurs.

3.3 Résumé

Les solutions n'utilisant qu'une image pour retrouver la profondeur de la scène ne sont pas intéressantes ici, parce que trop approximative, lente et souvent restreignante quant à la scène étudiée. Nous nous tournerons donc vers la stéréo-vision pour numériser le patient. Les solutions de stéréo-vision peuvent être divisées en deux catégories de compromis. Les méthodes rapides (fenêtre de corrélation [37, 55]) réussissent à calculer les profondeurs à une fréquence très élevée, mais elles souffrent toutes d'un manque inacceptable de précision. Tandis que les solutions globales (programmation dynamique [20], "graph-cuts" [16], flot maximum [70]), bien que beaucoup plus précises, sont extrèmement lentes. Les méthodes actives[5] sont définitivement les plus performantes. Il sera donc très avantageux d'utiliser la lumière structurée, en remplacant une des caméras par un projecteur, pour calculer la carte de profondeurs. Les résultats démontrés dans les travaux de Rusinkiewicz [22] et de Zhang [96] ne permettent pas vraiment de conclure. Toutefois, en se basant sur les travaux précédent de ces mêmes auteurs, il est possible de s'attendre à une reconstruction précise et en temps réel. Les travaux portant sur le "bureau du future" [68] ont, quant à eux, démontrés que la projection pourrait être complètement invisible à l'oeil humain. Si on projète rapidement une image en noire et blanc, suivie de son exacte inverse, l'oeil fera la moyenne des deux et ne verra rien de ces projections[6]. La résolution des projecteurs sera une limitation quant à la dimension de la zone de re-

[5] Il est important de voire la différence entre les solutions "actives" en stéréo-vision et celle en suivi 3D. Le terme actif est utilisé dans les deux cas pour référer à des solutions comptant sur un signal extérieur à la scène pour converger plus rapidement. Toutefois, dans le cas de la stéréo-vision, le signal ajouté sera, par exemple, des bandes de lumières, facilement dissimulable pour l'oeil humain. Tandis que les appareils utilisés en suivi actif sont, par définition, incontournable et nuisible au bon déroulement d'une chirurgie.

[6] L'oeil voyant à approximativement 30 images par seconde, si les deux projections se font à l'intérieur d'une intervalle de 1/60 de seconde, la projection sera complètement invisible.

construction. On peut cependant compter sur un espace de reconstruction d'environ $30cm^3$, ce qui sera probablement suffisant dans la plupart des cas, tout en gardant à l'esprit que l'évolution constante des projecteurs nous promet des résultats encore meilleurs pour les prochaines années.

Chapitre 4

MODÉLISATION DE SURFACE

Les algorithmes de stéréo-vision retournent un nuage de points sans véritable structure. Idéalement, chacun des pixels du projecteur aura une profondeur d associée à ses coordonnées (x_p, y_p). Évidement, il est possible que des occlusions viennent rendre la lecture de certains points plus difficile. Pour calculer la projection sur le patient, il faudra intersecter ce nuage de points avec des rayons provenant de la tête du chirurgien. Chaque point à éclairer sur le patient pouvant être vu comme le point d'intersection d'un rayon qui, partant du point de vue du chirurgien, va frapper le patient en un endroit précis et est ensuite projeté dans un des projecteurs. Le point de vue de l'utilisateur n'étant pas le même que celui du projecteur, cette opération ne pourra pas être faite sans un traitement préalable des points. En effet, certains rayons risquent fort de ne rencontrer aucun des points calculés à l'aide de notre algorithme de lumière structurée. Il sera donc nécessaire d'interpoler les profondeurs entre ces différents points afin d'obtenir la garantie qu'aucun rayon ne passera au travers du patient. C'est pourquoi, nous allons tenter de reconstruire une surface représentant le corps du patient, à partir du nuage de points obtenu précédemment. Ce calcul ne doit pas être trop lourd en temps de processeur, ne représentant qu'une fraction du travail à effectuer. Idéalement, il serait intéressant de retrouver la surface modélisée à la même fréquence que la carte de profondeur, pour ne pas créer de décalage trop important. Pour ne pas nuire au travail du chirurgien, un minimum de 5 reconstructions par seconde serait l'idéal. De plus, l'algorithme doit être totalement indépendant de toute intervention humaine. On ne veut pas dépendre d'une quelconque intervention externe ou d'hypothèse *a priori* réduisant la généralité de notre solution.

Gopi et Krishnan [31] ont proposé une classification des différentes solutions exis-

tantes pour le problème de modélisation d'une surface à partir d'un nuage de point. Ces solutions ont majoritairement été conçues pour créer des images réalistes. La qualité du modèle final est donc mise de l'avant par rapport à la vitesse. Notre système étant basé sur plusieurs hypothèses plus ou moins précises (position de la tête du médecin, correspondance de la zone opératoire avec le patient réel, ...) et notre objectif étant simplement d'intersecter le modèle avec un certain nombre de rayons, la précision ainsi que l'esthétisme de ces solutions sont beaucoup moins importantes que leurs performances. Nous ferons donc un bref survol de la littérature sur le sujet, selon la classification de [31] : Subdivision spatiale, Fonction de distance et Reconstruction incrémentale.

4.1 Subdivision Spatiale

La modélisation par subdivision spatiale consiste à représenter l'espace de reconstruction par un ensemble de sous-éléments, appelés voxels[1], de taille relativement petite. En partant d'un volume englobant plus grand que la zone d'intérêt, on peut éliminer les voxels superflus pour obtenir une bonne estimation de la surface. Les algorithmes de subdivision spatiale sont caractérisés par deux approches distinctes. Hoppe *et al.* [39] utilisent une approche de surface. Une surface "idéale" est calculée à l'aide de plans tangents appuyés sur les voisins de chaque point. La surface finale est ensuite interpolée à l'aide d'une minimisation par moindres carrés entre ces plans et les différents points de l'ensemble S de départ. La subdivision spatiale est utilisée pour réduire le nombre de calculs de distance. Ainsi, seul les voxels intersectant la surface "idéale" sont conservés, éliminant tous les points trop éloignés par défaut. L'algorithme développé dans [39] datent de 1992. Il est évident qu'exécuté sur des

[1] Inspiré de pixel (picture elements), voxel désigne donc des éléments de volume ou "volume elements". Généralement les voxels sont des cubes dans l'espace Euclidien, pour simplifier les calculs et garder leur utilisation générale au maximum, mais aucune contrainte n'est formulée quant à leur forme ou leur espace (Euclidien, projectif, affine).

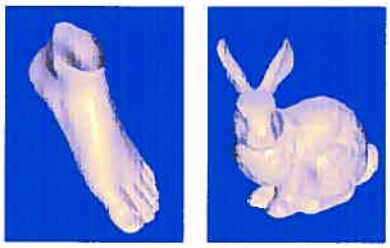

Figure 4.1. La reconstruction du pied pris 15 minutes pour ses quelques 20 000 points. Le lapin de Stanford contenait 36 000 points et fut reconstruit en 23 minutes. Image tirée de [3].

machines plus récentes, les temps de réponse seraient beaucoup plus bas. Leur algorithme prenait malgré tout environ une minute pour un modèle d'à peine 4000 points. Les auteurs de [39] comptent également sur un échantillonage uniforme, ce qu'on ne peut pas garantir dans notre système[2]. L'autre approche de subdivision spatiale est basée sur le volume. Les cellules conservées seront celles qui sont à l'intérieur du nuage de points. La surface finale sera approximée à l'aide de ces voxels. Le résultat final en voxels sera donc un volume, plutôt qu'une simple couche de voxels comme dans l'approche par surface. L'algorithme décrit par Amenta *et al.* [3] utilise les diagrammes de Voronoi et la triangularisation de Delaunay pour calculer la surface, ou croute dans l'article. À partir de cette surface, il est possible de reconstruire le modèle final. Les calculs nécessaires pour la triangularisation de Delaunay sont $O(n^2)$, pour n points. Leurs reconstructions se calculent donc en minutes, mais sont malgré tout impressionnantes, comme on peut le voir dans la figure 4.1. Il est à noter que la plupart des solutions au problème de modélisation, tout comme celle de [3], supportent très mal le bruit.

[2] Notre échantillonage est toutefois très dense, ce qui nous permet de contourner le problème d'uniformité. Il est cependant certain qu'il y aura plusieurs discontinuités dans notre scène, ce qui revient à peu près au même, au point de vue de l'uniformité.

Figure 4.2. Reconstruction à trois différents stages de l'algorithme (320, 1280 et 5120 triangles). Image tirée de [94].

4.2 Fonctions de Distances

La modélisation par fonctions de distances définie la surface de reconstruction comme celle passant par le zéro d'une fonction de distance entre l'ensemble de points S et la surface elle-même. Le terme distance est utilisé ici en tant que distance minimale entre une surface et le point le plus près de l'ensemble S. Yu [94] utilise un réseau de neuronnes pour minimiser la fonction de distance. Chaque point du modèle est représenté par un noeud dans le graphe. Un poids, représentant la distance à la surface, est associé à chacun des arcs. À l'aide des paramètres entrés par l'usager, le système connaît la forme générale de la reconstruction et arrive à retrouver la surface finale par fusions/divisions/inversions des triangles de départ. Évidemment, la solution est intéressante parce qu'imaginative, mais beaucoup trop lente pour nous. La figure 4.2 montre la reconstruction du lapin de Stanford à trois différents stages de l'algorithme.

Le principe des ensembles de niveau ("level sets") est un autre exemple de résolution par fonctions de distances. On peut voir ce principe un peu comme un ballon élastique qui se dégonflerait autour d'un modèle jusqu'à se resserrer complètement autour des points de l'ensemble de départ. Un des facteurs important pour s'assurer de la convergence vers une surface adéquate des "level sets" est d'avoir une bonne initialisation. Zhao *et al.* [99] utilisent un monceau pour trouver la surface englobante rapide-

ment. Chaque point sur la frontière de la surface englobante est ajouté au monceau en fonction de sa distance au nuage de points. Le classement d'un monceau étant $O(log(n))$, pour n points la surface sera calculée au pire en $O(nlog(n))$. En effet, les points les plus éloignés étant traités en premier, il ne sera pas nécessaire de traiter le même point deux fois. En réduisant la résolution de la grille de voxels environ de moitié, les auteurs de [99] réussissent à accélérer par un facteur 10 leurs calculs, et ce, pour une perte acceptable en qualité. Leurs résultats sont d'ailleurs fort intéressants. Bien qu'encore trop lent, le système des "level sets" est très prometteur, beaucoup plus précis que l'approximation que nous utilisons. Il reste malgré tout relativement rapide. Un bon survol du potentiel des "level sets" est présenté par Osher et Fedkiw dans [60].

4.3 Reconstruction Incrémentale

Le principe de reconstruction incrémentale est habituellement relié aux algorithmes gourmands. Une référence de départ est choisie, un point ou un triangle par exemple, et l'algorithme construit la surface en sélectionnant l'élément le plus probable à ajouter à chaque itération. Le "Ball-Pivoting Algorithm" (BPA) [12] est un très bon exemple de reconstruction incrémentale. Une balle se promène sur la surface du nuage de points et choisit à chaque étape un nouveau segment à ajouter à la surface de reconstruction. La sphère se déplace en pivotant autour des axes déjà sélectionnés. Le seul paramètre requis à chaque itération du BPA est donc un triangle de départ. Le rayon ρ de la sphère est déterminé par l'usager au début de l'opération. Il est idéalement sélectionné en fonction de la densité de l'échantillon. Lorsque la sphère arrive à un nouveau point, le triangle formé par ce point et les deux points formant le pivot sera conservé à condition que son orientation pointe dans la même direction générale que la normale de ces trois points. La comparaison se fait simplement en

projetant les deux normales l'une sur l'autre[3] pour savoir si ce triangle fait partie de la surface où si la balle vient d'atteindre une des frontières du modèle. Cette méthodologie à l'avantage de ne pas nécessiter d'information supplémentaire, autre que les points eux-mêmes. Le modèle n'a même pas à être chargé en mémoire au complet puisque le traitement porte uniquement sur l'entourage de la sphère. Il est donc possible de travailler avec un nombre gigantesque de points sans problème. Toutefois, plusieurs distances euclidiennes sont calculées à chaque itération. Le centre de la sphère doit en effet être comparé à tous les points environnant. Le temps de calcul devient donc rapidement non-négligeable. De plus, le rayon de la sphère étant fixe, il est difficile de traiter un échantillonage non uniforme. La sphère peut tomber dans un trou ou encore manquer un élément de la surface si la distance entre les points ou la courbure de la surface est sensiblement plus grande que son rayon. Cohen-Steiner [19] et Petitjean [64] utilisent un algorithme semblable. Le rayon des nouveaux triangles est calculé comme le rayon de la plus grande sphère passant exactement par les trois pointes du triangle, sans toutefois inclure d'autres points de l'ensemble S. Ces triangles forment le Graphe de Gabriel comme on peut le voir dans la figure 4.3. Ce graphe représente un ensemble de triangles qui contient au moins la surface. À partir d'un axe donné, les candidats sont classés par ordre inverse de leur rayon et le triangle sélectionné sera celui qui, tout en étant orienté relativement dans le même direction que ses voisins, affiche le plus petit rayon. Dans [64], la surface recalculée est également fonction de l'axe médial dont tous les points sont équidistants à au moins deux points de la surface[4]. L'objectif est d'avoir un axe médial continu pour la surface reconstruite qui soit le plus semblable possible à l'axe médial discret du nuage de points. Ces méthodes utilisent la triangularisation de Voronoi comme source

[3] Le produit scalaire entre deux vecteurs retourne la longueur de l'ombre créée par le premier vecteur sur le deuxième. Voir [28] pour un résumé des propriétés vectorielles.

[4] L'article de Amenta *et al.* [3] donne une bonne explication sur les diagrammes de Voronoi ainsi que les axes médiaux.

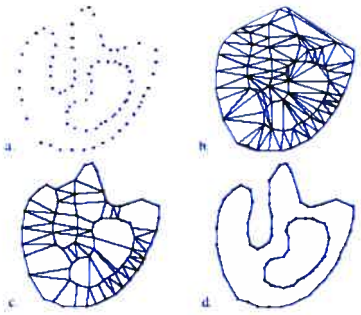

Figure 4.3. Reconstruction polygonale par Petitjean et Da. a) Ensembles de points en 2D. b) Graphe de Delaunay. c) Graphe de Gabriel. d) Interpolant régulier. Réimpression de [64].

d'information de départ. Ils sont donc très lents et supportent mal les échantillons non-uniformes et les surfaces comportant plusieurs discontinuités.

La solution de Gopi et Krishnan [31] est plus intéressante pour nous. Tous les points sont projetés sur un plan en 2D et classés dans un tableau en fonction de la distance à ce plan. L'algorithme démarre avec un point de référence R. À partir de ce point, une liste de nouveaux points potentiels est établie par rapport à une sphère d'influence autour de R. Le bon candidat sera celui qui respecte à la fois les conditions de visibilité et celles d'orientation. Le calcul de visibilité se fait à partir du plan tangent au point de référence. Tous les candidats, ainsi que les segments de frontière déjà calculés sont projetés sur ce plan. Les candidats visibles sont ceux n'intersectant aucun des segments de frontière. La recherche des candidats est d'ordre logarithmique puisqu'elle se fait dans un tableau 2D. Les auteurs obtiennent donc des résultats très intéressants au niveau de la vitesse. En travaillant avec des données provenant d'un laser, il est possible d'éliminer les tests d'orientation et de visibilité lors de la sélection de candidats. En effet, à moins qu'il y ait des discontinuités dans le modèle, un nuage de points provenant d'un laser formera une sorte de tapis de points où tous les points voisins devraient être connectés. Le candidat sélectionné sera donc

simplement le point le plus raproché de R. Leur algorithme arrive à reconstruire des modèles comportant pratiquement un million de points en moins de 7 secondes sur un SGI Onyx2 d'à peine 250 MHz. Toutefois, puisque nous sommes pratiquement assurés d'avoir des discontinuités dans notre modèle, cette solution risque d'être difficilement applicable ici.

4.4 Solution Proposée

Il serait intéressant de suivre de près l'évolution des solutions utilisant les "level sets". Ces dernières ne sont pas encore suffisamment rapides pour être applicables en temps réel, mais les résultats sont prometteurs. Probablement que d'ici peu, l'évolution des CPU et de nouvelles solutions plus rapides permettront une reconstruction presque parfaite en quelques millisecondes. En attendant, il faudra nous contenter de la solution la plus simple possible pour ne pas nuire à l'efficacité de tout le système. Comme nous l'avons mentionné plus haut, la majorité de nos données sont approximées. Reconstruire trop parfaitement le modèle serait donc une perte inutile de temps. Il est possible de prendre tout simplement quatre points qui sont voisins et de former des triangles en trouvant le point milieu. Une idée générale du résultat est affichée dans la figure 4.4. Évidemment, la surface sera approximée très grossièrement, mais si la résolution des projecteurs est élevée, l'erreur ne devrait pas être trop grande. Cette supposition est bonne dans le cas où le nuage de points forme en fait une simple surface, comme avec un laser. Puisque nos profondeurs viendront d'un système de stéréoscopie, cette restriction est sans conséquence. Le traitement de lumière structurée permet d'avoir une surface lisse, du moment que les aberrations sont traitées correctement. La mise en correspondance des différents nuages de points provenant des projecteurs sera le plus gros problème ici. La reconstruction de modèle décrite par Rusinkiewicz *et al.* [72] utilise une version temps-réel de l'algorithme d' "Iterative Closest Points" (ICP) [13] pour enregistrer les différentes surfaces entre elles. Pour

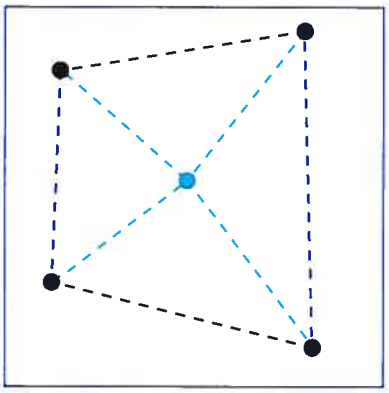

Figure 4.4. La reconstruction la plus simple possible à partir d'un tapis de points. Les points en noir sont des éléments de l'ensemble de départ S. Le point vert est le centre estimé des 4 points (approximation) et les pointillés représentent les frontières des nouveaux triangles.

atteindre une vitesse raisonnable, les auteurs de [72] ont remplacé le calcul de distance en 3D par une simple comparaison par projections. Cette décision est explicable par l'utilisation de carte de distance, qui sont en fait des tableaux 2D de points avec leur profondeur respective. Notre situation étant la même, leur solution s'applique donc également ici. Il est cependant important de noter qu'il y aura beaucoup de discontinuité dans nos cartes de profondeurs. Sans que ce soit un problème très grave, notre objectif n'étant pas de faire des rendus réalistes de la scène, certains algorithmes ont tendance à échouer plus souvent au bord d'une discontinuité.

Chapitre 5

RECALAGE ÉLASTIQUE 3D

Le recalage est un point critique des chirurgies assistées par ordinateurs. En effet, il est fréquent d'utiliser plusieurs modalités d'imageries différentes dans le but de planifier une seule chirurgie. Ces modalités[1] peuvent être morphologiques ou fonctionnelles et affichent toutes des informations différentes et complémemtaires. La figure 5.1 tirée d'un article de Wells *et al.* [88] montre bien l'intérêt qu'il peut y avoir à coupler différentes analyses entre elles pour générer un modèle beaucoup plus parlant. Plusieurs chercheurs ont donc travaillé à développer un système efficace permettant de coupler les différentes images sur un même modèle. Le problème est très vaste et le nombre de solutions développées l'est tout autant. Maintz *et al.* [50, 51] ont défini un ensemble de critères permettant de classifier ces différentes solutions. Leurs critères portent entre autre sur la dimensionalité des modèles, les bases de recalage (naturelles ou artificielles) et sur la nature de la transformation (rigide, projective

[1] Résonnance Magnétique (MR), Tomographie (CT), SPECT, TEP, Échographie, Radiographie, ...

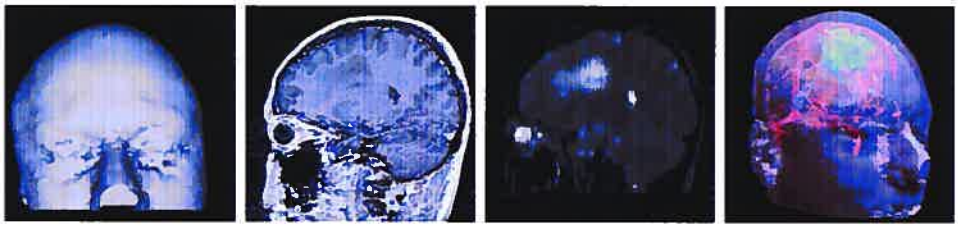

Figure 5.1. Les trois images de gauche ont servi à reconstruire le modèle de droite. On voit très bien à quel point un bon recalage peut faciliter la tâche de planification ainsi que la chirurgie elle-même. Réimpression de [88].

ou élastique). Avant toute chose, il est important de spécifier qu'il n'existe pas actuellement de solution efficace à notre problème. Nous souhaitons en effet effectuer un recalage modèle-vers-patient en 3D, de manière élastique et ce, en temps réel. Ce genre de traitement prend actuellement plusieurs minutes à calculer. Il existe toutefois plusieurs recherches intéressantes, promettant des résultats satisfaisants pour bientôt.

5.1 Complexité du Problème

Des dizaines de chercheurs arrivent à faire du recalage efficacement, mais pas nous, pourquoi? En fait, le problème de recalage n'est pas aussi insurmontable qu'il ne le paraît. Wells *et al.* ont même développé un système de chirurgie assistée par ordinateur (CAO) en temps réel, présentement utilisé au Brigham and Women's Hospital, pour des chirurgies au cerveau [30]. Des solutions applicables étaient déjà développées en 1995, comme dans [41]. Nous devons cependant tenir compte de plusieurs restrictions pour bien implanter notre solution. Dans [41], Maurer *et al.* ont développé un système de CAO du genre de ce que l'on souhaite construire ici, mais ils sont forcés de visser des marqueurs dans la tête du patient pour effectuer leur recalage. Des solutions moins radicales sont apparues depuis. Dans la salle d'opération du futur de Wörn et Hoppe [89], les marqueurs sont simplement collés sur la tête du patient, tandis que Grimson *et al.* comparent des pairs de points sur le patients avec les modèles à recaler[2] [33]. Toutefois, leur problème reste plus simple que ce que l'on tente de faire ici. Toutes les solutions développées pour les chirurgies au cerveau se basent sur une prémise critique, soit que le crâne humain ne se déforme pas durant la chirurgie. Le cerveau aura tendance à se déplacer un peu, mais cette déformation est considérée comme rigide dans la plupart des études actuelles. Le recalage étant rigide, ils ne cherchent donc qu'à trouver 12 paramètres, c'est-à-dire une matrice 3

[2] Évidemment, la supposition que les points à la surface du patient sont les mêmes que ceux de notre modèle implique que les nuages de points des deux surfaces sont très denses.

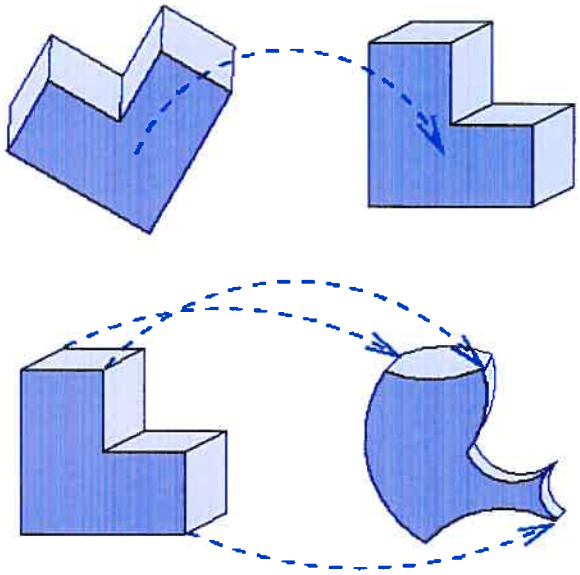

Figure 5.2. La complexité impliquée derrière un recalage élastique est beaucoup plus élevé que lors d'un simple recalage rigide. Chaque point doit être retrouvé indépendamment.

par 3 de rotation R et un vecteur T représentant la translation en 3D de leur modèle. La complexité pour un recalage élastique explose rapidement, comme on peut le voir à la figure 5.2, un déplacement doit être calculé pour chaque élément significatif de l'image. Il faut donc calculer un champ de déplacement D tel que pour deux images I et J, nous puissons avoir $J_f = D(I_f)$ pour chaque point f de l'image I. Les éléments comparés sont habituellement des points, mais peuvent aussi être des vecteurs [71] ou des surfaces [83] par exemple. Tarel et Boujemaa [83] proposent une approche intéressante, combinant un algorithme global à un "Iterative Closest Points" (ICP) pour retrouver la déformation élastique plus rapidement. Une première étape est donc calculée par un algorithme équivalent au K-Moyen [24], utilisé en traitement d'image. Ce dernier permet de trouver une solution grossière, même si la déformation est très grande. Pour augmenter la précision, une deuxième passe est ensuite effectuée avec un ICP, plus lent, mais plus précis. La segmentation initiale est nécessaire pour éviter

que l'ICP ne donne une solution dégénérée dans le cas où le déplacement serait trop grand. Nous ne pourrions pas utiliser une simple minimisation de SSD ici. En effet, cette méthode aura tendance à échouer dans le cas où le bruit dans les images n'est pas indépendant, ce qui est souvent le cas pour les images IMR [17]. De plus, en utilisant différentes modalités, il est fort probable que deux points équivalents n'aient pas nécessairement la même intensité. Maintz *et al.* proposent de comparer les histogrammes des deux images étudiées [51]. Pour une image M et son équivalente N, après une certaine transformation t, ils établissent une table de probabilités $p(n|m)$, pour chaque niveau de gris $m \in M$ et $n \in N$. En considérant les niveaux de gris n et m comme des variables stochastiques, il suffit de trouver la transformation t qui maximisera leur interdépendance. L'image est ensuite subdivisée pour calculer la transformation élastique finale. Chaque sous-image W sera modifiée pour maximiser la probabilité de chacun de ses points $w \in W$ tel que

$$c(t) = \Sigma_{w \epsilon W} p(N(t(w))|M(w)). \tag{5.1}$$

Cette méthode est intéressante puisqu'elle supporte bien les transformations élastiques, mais beaucoup trop longue pour pouvoir être appliquée en temps réel. La solution la plus prometteuse pour notre problème de calcul sera d'utiliser un groupe de machines calculant le recalage en parallèle. L'utilisation de plusieurs ordinateurs personnels nous évitera d'avoir recours à des super-ordinateurs, beaucoup trop dispendieux pour nous, tout en maximisant la puissance de calcul disponible.

5.2 *Solution Hybride Parallélisée*

Plusieurs solutions ont été développées pour arriver à calculer des résultats complexes à partir de processeurs parallèles. Même les ordinateurs personnels supportent maintenant plus d'un processeur. Lorsque les processeurs sont tous dans une même machine, ils peuvent partager la mémoire pour accélérer les échanges d'information. Il devient cependant critique de bien gérer les interruptions pour éviter de traiter des

informations corrompues. Certains ordinateurs utilisent plusieurs dizaines de processeurs en même temps, mais ces machines ne sont pas vraiment accessibles dans le contexte des salles chirurgicales où nous travaillons. Une autre approche est d'utiliser des ressources privées pour chaque processeur. Un réseau de machines ne comportant qu'un ou deux processeurs est assemblé et les différentes machines n'ont pas accès aux ressources des autres sur le réseau. Les échanges sont un peu plus lents que dans une seule machine, mais le coût est sensiblement moins élevé. Il devra toutefois y avoir redondance de l'information, puisque pour accélérer au maximum le système, on ne devra pas avoir à échanger trop d'informations sur le réseau. Chaque ordinateur possèdera l'image au complet en mémoire. Pennec *et al.* ont appliqué l'algorithme des démons de Thirion [84] sur un groupe d'ordinateurs personnels. Le taux d'accélération est presque linéaire par rapport à l'augmentation du nombre de processeurs utilisés [61, 79]. C'est-à-dire que chaque ordinateur supplémentaire utilisé accélère de façon direct le traitement de l'image. Ils réussissent d'ailleurs à obtenir des temps relativement intéressants d'à peine quelques secondes par image. Bien que le recalage pourrait être effectué à partir de n'importe quel filtre simétrique, l'utilisation d'un filtre gaussien comporte certains avantages. En effet, le filtre gaussien est le seul filtre isotropique qui peut être décomposé, pour chacune de ses dimensions, en filtre 1D [17]. Le filtrage peut donc se faire linéairement dans chacune des dimensions de notre espace de reconstruction. Ces lignes étant indépendantes l'une de l'autre, il est possible de faire travailler des ordinateurs différents sur chacune d'elles pour optimiser le temps de calcul.

Pour s'assurer d'une solution valide, le plus rapidement possible, il est pratiquement essentiel d'utiliser une approche hybride. En effet, les algorithmes retrouvant une solution élastique directement ont tendance à échouer lorsque la déformation est trop grande [83]. L'image doit donc être traitée une première fois, pour trouver les points saillants. Un recalage rigide est ensuite estimé pour faire correspondre autant que possible les deux modèles. À partir des résultats de ce recalage, d'autres estima-

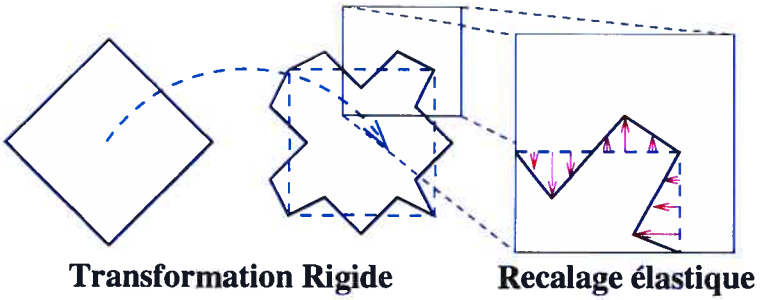

Figure 5.3. Le recalage se fait en deux étapes. On commence par trouver une transformation rigide la plus précise possible. À partir de cette estimation, on peut utiliser un algorithme plus précis pour retrouver les déformations locales.

tions, de plus en plus précises, peuvent être calculées itérativement pour finalement obtenir le recalage élastique final. Ainsi, les meilleures solutions existantes utilisent une approche par niveaux de résolution. Les résultats de chaque itération servent d'hypothèse initiale pour l'itération suivante. Ces étapes sont schématisées dans la figure 5.3. L'algorithme de Rexilius *et al.* prend 5 minutes pour recaler des espaces de 256 x 256 x 124 voxels [69]. Un certain nombre d'éléments de repères, qui peuvent être des points, des polygones ou des voxels par exemple, sont retrouvés dans la scène. Un système de recherche par inter-corrélation est ensuite implanté pour arriver à faire correspondre ces éléments dans les deux images. Une fois ces éléments recoupés, une interpolation élastique est calculée pour retrouver le champ de déformations complet. Ils concluent en soulignant l'importance d'utiliser des contraintes sur l'élasticité des modèles en fonction de leurs propriétés de surface. Ce supplément de travail peut paraître superflu, surtout lorsque l'on cherche à accélérer au maximum les calculs, mais il est important d'éviter que nos solutions dégénèrent complètement.

5.3 Piste de Solution

Les méthodes actuellement utilisées pour planifier une CAO représentent généralement l'information traitée sous forme de volumes. De notre côté, la modélisation du patient se fait à partir d'un nuage de points, qui est ensuite transformé en surface. Avant de pouvoir recaler ces deux modèles entre eux, il sera nécessaire de les tranformer pour qu'ils soient comparables. Les recherches actuelles portent en grande partie sur le recalage de volumes, mais transformer notre surface en volume risque d'être inutilement long. Les modèles pré-opératoires peuvent être modifiés avant la chirurgie sans problème. Il serait donc avantageux d'arriver à travailler directement avec des coordonnées 3D. Une autre alternative serait de travailler avec des surfaces. Les points 3D représentant le patient sont transformés par défaut en surface pour la génération d'images. Si les modèles pré-opatoires étaient représentés par surfaces, le recalage serait plus rapide. Il existe d'autres facteurs, inhérents à notre problème, qui pourraient être utilisés pour trouver une solution suffisamment rapide. Premièrement, si la déformation du patient ne se fait qu'au niveau de la respiration, il serait possible d'interpoler la forme du modèle rapidement et de façon relativement précise. On peut ainsi trouver la position d'une tumeur lorsque les poumons sont remplis d'air et celle lorsque le patient a fini d'expirer et ainsi interpoler la position courante tout au long de l'opération. En effet, étant donné que l'on connait la forme du patient à tout moment durant la chirurgie, il serait possible d'estimer où il se trouve à l'intérieur de son cycle respiratoire. Deuxième point à souligner, puisque l'on souhaite faire le recalage souvent, plusieurs fois par seconde, il est raisonnable d'estimer que les déplacements entre le patient et la projection ne seront pas trop élevés. Ainsi, on pourrait simplement utiliser le résultat du recalage de l'étape précédente comme hypothèse de départ de notre recalage courant. Il serait probablement possible d'éviter ainsi l'étape de recalage rigide nécessaire dans la plupart des solutions actuelles.

5.4 *Solution Proposée*

Le recalage élastique est un problème très complexe à résoudre. En effet, il n'est pas toujours possible d'utiliser une simple fonction linéaire, ou même polynomiale pour la modéliser. En 3D, ce problème dépasse tout à fait les capacités de calcul des ordinateurs personnels actuels. La quantité d'information à gérer est énorme et tous les critères à prendre en compte rendent les solutions impossible à calculer en temps réel. Il faut donc avoir recours à des solutions optimisées reposant sur des algorithmes hybrides utilisant différentes approches en même temps. Les solutions les plus rapides trouvent donc premièrement une fonction globale calculant un recalage rigide pour certains points plus importants seulement. Ensuite, une interpolation est faite entre ces points pour retrouver un champ de transformations plus dense. Les auteurs de [71] ont fait un parallèle entre ce problème et la géostatistique. Ils utilisent l'estimateur de krigeage développé, entre autre, pour reconstruire une surface inconnue à partir d'un ensemble de points donnés [29], pour estimer les déformations locales entre les différents points déjà recalés. Bien que ces améliorations, tout comme le traitement parallèle, améliorent grandement la vitesse et la fiabilité des calculs à effectuer, les résultats sont encore beaucoup trop lents pour pouvoir vraiment avoir une application en temps réel. D'un autre côté, le développement d'un système suffisamment général pour être appliqué à n'importe quel type de chirurgie risque d'être très compliqué. La plupart des solutions étudiées ici portaient sur un type de chirurgie bien précise. La majorité des algorithmes de recalage ont donc été développés pour les chirurgies au cerveau, qui s'évalue relativement bien par un recalage rigide. D'autres études ont été faites sur les chirurgies aux genou ou, comme dans les travaux de Betke *et al.*, aux poumons [14]. Ce qui est intéressant dans [14], c'est qu'ils doivent gérer la respiration du patient. Le recalage ne peut donc pas être estimé comme rigide. De plus, les auteurs de [14] n'utilisent aucun marqueur artificiel. Ils utilisent plutôt certains éléments facilement reconnaissables, comme la colonne vertébrale pour recaler

différentes images d'un même patient. Le problème de généralité est évident ici. Si on souhaite utiliser des marqueurs naturels, comme les vertèbres ou les yeux, il faudra nécessairement redéfinir tous les repères à chaque nouvelle chirurgie.

Chapitre 6

MODÈLE DE PROJECTION ET SIMULATION

À ce stade du programme, nous pouvons supposer que la position du point de vue ainsi que la surface de projection, c'est-à-dire la forme 3D du patient, sont connus. Nous allons maintenant construire une image qui sera projetée dans la scène de façon à ce que le chirurgien voit exactement l'information que l'on souhaite lui envoyer. Cette image est modelisée en deux parties successives. Premièrement, nous construisons une image que le chirurgien devra voir. Cette image n'est pas déformée et serait valide seulement si elle était projetée sur une surface plane. Cette image est nommée image chirurgien, puisque son système d'axes est le même que celui d'une caméra qui serait située à la place de la tête du chirurgien. Ensuite, l'image est déformée de façon à épouser la forme du patient tout en restant valide au niveau des informations vues par le chirurgien. Les points de cette image doivent être déplacés de façon à ce qu'ils représentent correctement, selon le point de vue du chirurgien, l'image que nous avons calculée précédement. La deuxième image est différente pour chacun des projecteurs. Le système de référence de ces images est le même que celui des projecteurs. Nous les nommerons donc images projecteurs. Ce sont ces images que nous allons pouvoir projeter durant l'opération.

6.1 Image Chirurgien

L'information disponible pour la projection, durant la chirurgie, peut se diviser en deux catégories distinctes, c'est-à-dire 2D ou 3D. Les informations 2D se réfèrent à toutes les lectures bio-médicales faites durant l'opération. Ces données peuvent être des images statiques (comme des radiographies), des graphiques (l'ECG du pa-

tient), ou des données alpha-numériques (pression sanguine, température, respiration, ...). La partie 3D sera quant à elle composée d'images perspectives de modèles 3D, généralement calculées avant la chirurgie. Il faut faire attention ici de bien conserver l'effet de profondeur pour assurer la validité de cette projection. La projection 3D comportera par exemple, un modèle de la tumeur à enlever, tel que vue au travers de la peau du patient, ou encore la ligne de coupe à effectuer. Pour pouvoir projeter des informations précises quant à la chirurgie, il sera important d'arriver à intégrer au système les données enregistrées avant l'opération. Les chirurgies d'importance étant habituellement préparées à l'avance, une certaine quantité d'informations est déjà disponible au chirurgien avant l'opération. Ces données sont appellées informations pré-opératoires, elles sont généralement présentes durant la chirurgie via des moniteurs ou des graphiques 2D. Nous allons simplement modifier le médium sur lequel elles vont être propagées. Ces informations sont disponibles sous différentes formes, tel que des images à résonnance magnétique ou des tomographies des différents organes [89]. Une fois que la position des modèles 3D à projeter sera connue, nous pourrons facilement créer une image représentant la vue du chirurgien en projetant sur un plan ces informations. La projection ramenera ces coordonnées 3D vers un système 2D. Nous pourrons donc facilement ajouter à cette image les informations supplémentaires souhaitées. Une fois complète, l'image représentera exactement ce que l'on désire faire voir au chirurgien, nous l'appellerons donc *image chirurgien.* Cette image est en fait le plan image d'une caméra situé dans l'oeil du chirurgien. Nous appellerons cette caméra *caméra chirurgien.* Il est facile de voir les problèmes qui apparaîtraient si la projection était faite à ce moment. Les modèles seraient décalés, annulant tout l'intérêt de ce système, tandis que l'information 2D serait probablement illisible. Nous allons donc devoir reconstruire cette image en fonction des projecteurs et de la surface de projection. L'importance du chapitre 3 ressort ici, puisque l'image projetée sera déformée de façon à s'adapter parfaitement à la surface du patient.

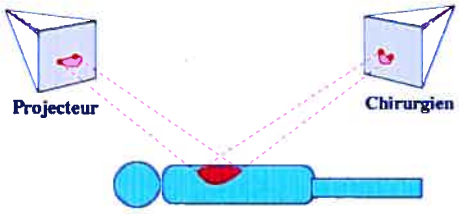

Figure 6.1. Construction des images qui seront projetées sur le patient en fonction du point de vue de l'usager et de la position des projecteurs.

6.2 Image Projecteur

L'image que nous souhaitons projeter peut être vue comme un ensemble de points à illuminer dans la scène. Chaque pixel de l'image chirurgien peut donc être calculé séparément. Si on considère le point de vue, connu, du chirurgien comme une caméra sténopée, on peut lancer un rayon dans la scène pour chacun de ces points. En calculant les intersections de ces rayons avec les objets de la scène, nous obtenons un ensemble de coordonnées en trois dimensions. Chaque point P^c aura donc les coordonnées $[X^c, Y^c, Z^c]^T$, représentant le point dans l'espace selon le système d'axes de la caméra chirurgien. Puisque l'on connaît la position globale dans la scène de cette caméra, il suffit de transformer les points P^c suivant la transformation rigide qui les relie (rotation et translation 3D). Cette opération nous retournera un nouvel ensemble de points $P^s = [X^s, Y^s, Z^s]^T$. Nous pouvons ensuite reprojeter ces points à l'intérieur de chaque projecteur pour trouver exactement quels pixels $P^p = [X^p, Y^p]$ devront être allumés. La reprojection se fera selon la formule suivante :

$$P^p = J M_{int} M_{ext} P^s \tag{6.1}$$

J est la matrice de projection permettant de ramener les points 3D vers des points 2D tandis que M_{int} représente les paramètres internes du projecteur et M_{ext} les paramètres externes, c'est-à-dire la rotation et la translation globale du projecteur.

Ces matrices sont définies comme suit :

$$J = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{6.2}$$

$$M_{int} = A = \begin{pmatrix} \frac{-f}{s_x} & 0 & o_x & 0 \\ 0 & \frac{-f}{s_y} & o_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.3}$$

$$M_{ext} = RT = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{6.4}$$

Les variables de la matrice A sont : f, la longueur focale du projecteur, (s_x, s_y), la taille d'un pixel et (o_x, o_y), le centre du plan de projection. Les r_{ij} et t_a de la matrice M_{ext} représentent la rotation ainsi que la translation globale du projecteur. Ces paramètres n'étant pas indépendants entre eux, des équivalences sont habituellement utilisés (voir Trucco et Verri [85]) pour calculer les différentes matrices. Une bonne calibration pré-opératoire, ainsi que des vérifications ponctuelles seront d'ailleurs nécessaires pour s'assurer que ces valeurs sont toujours exactes. On peut voir, pour un seul projecteur, une schématisation de ces principes dans la figure 6.1. Une fois les coordonnées 3D obtenues, il est possible de comparer les profondeurs d'intersections des rayons avec un seuil représentant grossièrement la surface du patient. Ce seuil permettra d'éliminer les points qui seraient projetés sur des surfaces parasites causant des occlusions. L'occlusion se produisant ici entre le point de vue du chirurgien et la surface de projection, nous éliminons en fait les points que le chirurgien ne voit pas. Une fois que nous connaissons les points qui doivent être affichés, ainsi que leurs coordonnées 3D respectives, nous pouvons les reprojeter dans les projecteurs entourant

la scène. La figure 6.2 schématise cette étape, tout en soulignant l'importance de calculer les occlusions encore une fois. Les points qui sont cachés d'un projecteur seraient projetés un peu partout dans la scène. Cette partie s'avère évidement plus facile à implanter pour la simulation que dans une vraie mise en application, les profondeurs des objets face aux projecteurs n'étant pas réellement connues. Cependant, deux constatations peuvent être prises en considération ici. Premièrement, si la projection se fait sur une surface parasite, les conséquences ne sont pas si grâves. Les points, volant un peu partout, seraient en effet dérangeants, mais il serait surprenant qu'un chirurgien décide de s'opérer la main parce que la projection se fait sur la mauvaise surface. Deuxièmement, si les projecteurs sont correctement positionnés, il sera possible de minimiser grandement ce genre d'occlusions. Pour bien retrouver les couleurs de projection, il est également important de bien connaître le nombre de projecteurs contribuant à chaque point. À l'aide des paramètres de surface aux points de projection (couleur, réflectance, ...), nous pouvons calculer la lumière à projeter nécessaire (intensité, teinte) pour obtenir des résultats satisfaisants. Cette couleur doit ensuite être fondue entre les projecteurs pour s'assurer qu'un point mieux "couvert" n'apparaîsse de manière plus brillante que les autres. La peau se prêtant bien à ce genre de projection [95], les calculs sous-jacents sont simples et peuvent être faits rapidement, sans avoir à tenir compte d'effet de spécularité, changeant selon le point de vue de l'observateur. Il est à noter qu'une fois la chirurgie commencée, la partie modélisée de la projection risque d'être très difficile à conserver. Le sang et les rapides déformations de la surface vont rendre cette projection obsolète rapidement. Garder ces informations à jour deviendrait impossible à ce moment. Toutefois, le reste des informations contenues dans la projection ne devrait pas trop souffrir de ces changements.

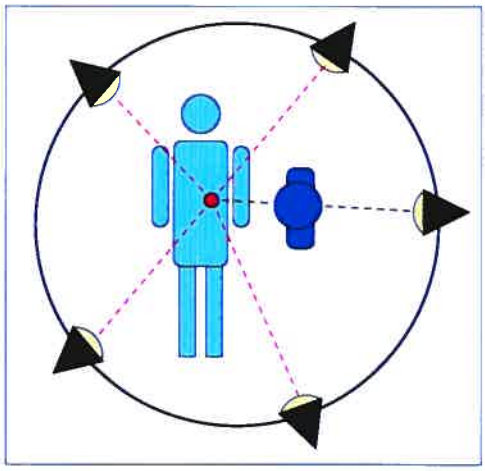

Figure 6.2. Lors de la création des images projecteur, il est important de vérifier les obstruction afin d'éviter de projeter n'importe où dans la scène.

6.3 Simulation

Nous présentons ici une simulation du traitement expliqué dans ce chapitre. La simulation se fait sur une scène 3D construite à l'aide d'un outil de modélisation (Softimage, XSI [77]). La scène étant synthétique, les suppositions initiales, c'est à dire qu'on connait le point de vue du chirurgien et que le patient a été correctement modélisé en 3D, tiennent sans difficulté. La scène utilisée est grossièrement représentée dans la figure 6.3. On peut remarquer qu'il n'y a pas de caméras dans la scène. En fait, les caméras sont nécessaires pour obtenir les informations que nous avons supposées connues avant de commencer la simulation. Elles ne sont donc plus essentielles à ce stade. La figure 6.4 montre ce que le chirurgien voit en début de simulation, avant la projection. La partie de droite représente la même vue, mais avec un rendu de type fil de fer. On peut donc bien voir la sphère verte, représentant une tumeur quelconque, au travers du patient. La figure 6.5 montre donc le résultat de la projection, entourant la "tumeur", une fois la simulation commencée. Afin d'avoir de bons résultats au niveau de la projection, tout en gardant le système simple, nous

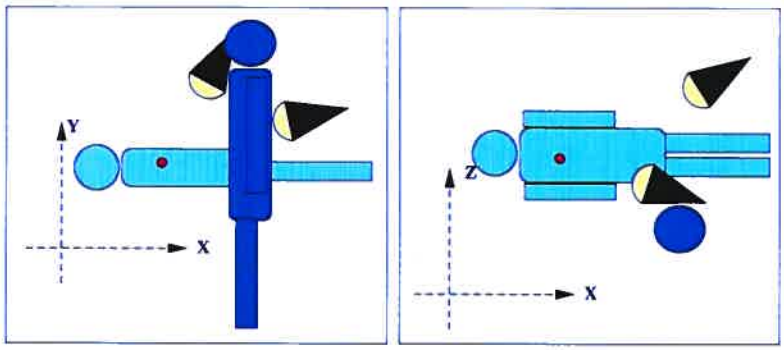

Figure 6.3. Nous avons effectué la simulation à partir d'une scène synthétique tel que vue de côté (à gauche) et de haut (à droite).

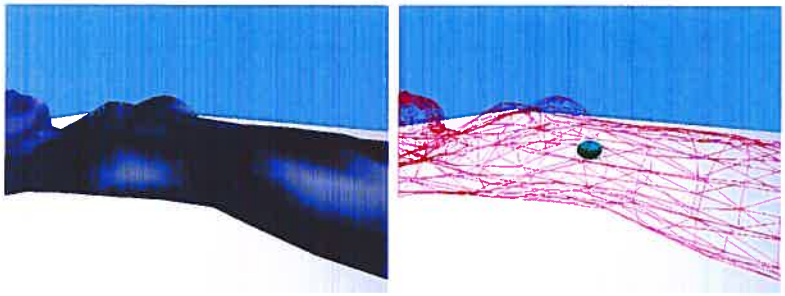

Figure 6.4. La figure de gauche représente la vue du chirurgien avant le début de la simulation, donc avant qu'il n'y ait de projection. La figure de droite représente la même vue, mais avec un rendu en fil de fer, ce qui permet de bien voir la "tumeur" que nous souhaitons mettre en évidence.

Figure 6.5. On peut voir ce que le chirurgien voit, une fois la projection activée.

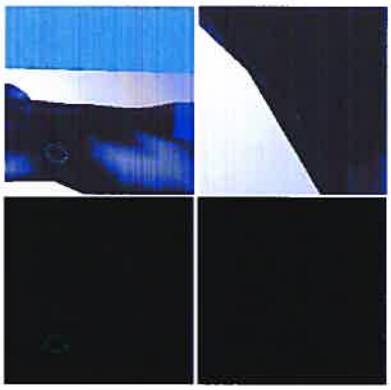

Figure 6.6. L'image du haut montre ce que les projecteurs "voient", comme si une caméra avait été placé à la même place que le projecteur. L'image du bas montre seulement la projection effectuée par chacun des projecteurs.

avons utilisé deux projecteurs passablement rapprochés. On peut voire dans la figure suivante, figure 6.6, le résultat de la projection tel que vue par les projecteurs eux-mêmes. La première rangée d'images représente ce que des caméras verraient si elles étaient situées aux mêmes points que les projecteurs. Tandis que la rangée du bas montre uniquement les points projetés par chacun des projecteurs. On voit très bien dans ces images la déformation effectuée sur le cercle pour qu'il conserve sa forme ronde du point de vu du chirurgien. Le problème d'occlusion est démontré à l'aide de la figure 6.7, où plusieurs points ne sont visibles par aucun des deux projecteurs. La projection est censée encercler la sphère verte complètement. Toutefois on peut voir que le cercle n'est pas complet. Pour que ce système soit réellement efficace, il faudra utiliser plus de projecteurs afin de bien couvrir le patient et ainsi minimiser ce genre d'occlusions. Le problème est d'ailleurs le même pour les caméras. Si une trop grande région du patient devient complètement invisible pendant un certain temps, la projection risque de ne plus être possible.

Les images de la figure 6.5 n'affichent qu'une projection simple de la partie modélisée des données. Puisque tout le traitement a déjà été effectué, il est très

Figure 6.7. La projection est incomplète, plusieurs points sont complètement invisible dans les deux projecteurs de notre simulation.

Figure 6.8. Il est facile d'ajouter de l'information 2D sur le patient une fois que sa forme a été modélisée adéquatement.

facile d'ajouter des informations, dites 2D, à notre projection. En effet, que l'on veuille afficher un graphique ou des caractères alpha-numériques, la façon de déformer l'image reste la même. La différence est uniquement au niveau de l'image chirurgien. Ces nouveaux points n'ont pas à être interceptés dans la scène, puisqu'ils n'existent pas vraiment. Nous allons donc simplement calculer leurs positions 2D dans l'image chirurgien, en s'assurant que tous ces points tombent bien sur le patient. La figure 6.8 montre le résultat de la simulation une fois que l'on a ajouté certaines informa-

tions supplémentaires à la projection. Notre algorithme cherche la meilleure zone de projection à partir du centre des données modélisées. Ainsi, les différentes projections restent regroupées autant que possible dans une même zone, sur le corps du patient. Pour accélérer la recherche, les zones de projection sont trouvées à l'aide de boites englobantes. Ce qui nous donne également la garantie qu'aucune de ces images ne se superposera.

Les calculs que nous utilisons étant simples, nous n'aurons aucune difficulté à générer ces images en temps réel. La projection d'environ 350 points (pour la figure 6.8) ne prend que 25 millisecondes sur un ordinateur de bureau de 1.2 gHz. En voulant implanter un pointeur intelligent, notre objectif n'est pas d'afficher des images trop précises ou détaillées. Il est donc raisonnable de supposer que les projections ne nécessiteront pas plus que quelques milliers de points par image. Il serait toutefois possible d'utiliser encore plus l'accélération matérielle des nouvelles cartes graphiques pour obtenir des résultats encore plus rapides.

Chapitre 7

CONCLUSION ET DISCUSSION

Nous avons défini les lignes directrices pour le développement d'un système de pointeur intelligent par réalité augmentée en situation chirurgicale. Nous avons dû tenir compte de plusieurs contraintes précises pour s'assurer que le système final soit le moins intrusif possible au niveau de la chirurgie. La solution proposée n'utilise donc aucun appareil spécialisé, qui deviendraient rapidement encombrants et dispendieux. Le système pourrait presque fonctionner en temps réel et nous arrivons à déformer les images adéquatement pour donner une impression de profondeur à la projection. Nous avons également fait attention à garder notre proposition générale et automatique pour pouvoir, autant que possible, l'appliquer à n'importe quel type de chirurgie. Notre travail s'est porté premièrement sur le développement d'un modèle de projection efficace pour des surfaces inconnues et déformables. Ce mémoire permet également d'affirmer qu'un tel système pourra effectivement voir le jour d'ici quelques années à peine. Les problèmes qui ne sont pas déjà implantés en temps réel sont en effet le sujet de plusieurs recherches prometteuses.

7.1 Notre Solution

Nous avons dû faire face à cinq problèmes distincts pour définir une application de réalité augmentée complète. Voici un aperçu des voies explorées :

Suivi de la tête : Pour avoir une projection en perspective, il est important de connaître la position de l'observateur. Pour arriver à faire le suivi en temps réel, sans avoir recours à des appareils trop encombrants, nous proposons d'utiliser la segmentation d'image et les filtres de Kalman pour suivre l'observateur sur

plusieurs images. La question reste en fait à savoir ce que l'on va chercher dans la scène. Si le chirurgien portait un casque aux couleurs voyantes[1], il serait facile de trouver la position de son casque et ainsi déduire les coordonnées du point de vue en tant que tel. Pour faciliter encore plus la recherche, nous pourrions utiliser un émetteur à infra-rouge et appliquer des bandes réfléchissantes sur le casque du chirurgien. De cette façon, il serait également possible d'obtenir l'orientation absolue du regard[2], ce qui permettrait de pousser encore plus loin l'expérience de réalité augmentée. Pour garantir la fiabilité des résultats, il serait intéressant de jumeler cette approche avec une recherche de visages par analyse de couleurs ainsi qu'une reconstruction du mouvement rigide de la tête du chirurgien. Ces trois méthodes réunies pourraient minimiser les risques d'obtenir un résultat erroné.

Estimation des profondeurs : Il a été prouvé, dans les travaux sur le bureau du futur [68], qu'il est possible de projeter des bandes de lumière de façon totalement invisible. En utilisant le principe de projection inversée, on arrive à tromper l'oeil, qui mélange les projections entres elles et ne les voit tout simplement pas. Nous pouvons ainsi obtenir, par lumière structurée, un nuage de points denses en trois dimensions, représentant exactement la scène étudiée. De plus, cette estimation se fait en temps réel. La plus grande restriction sera la résolution des caméras et des projecteurs utilisés. Les caméras profitant jusqu'à ce jour d'une meilleure résolution, on peut affirmer sans perte de généralité que la restriction vient en fait seulement des projecteurs. Pour avoir une bonne résolution, les volumes de travail sont actuellement limités à une trentaine de centimètres cubes. Les chirurgies étant de moins en moins invasives, une re-

[1] Vert ou bleu dans un environnement blanc, par exemple, comme c'est souvent le cas en chirurgie.

[2] En retrouvant simplement le casque, l'orientation du regard est considérée comme pointant directement vers la zone de projection.

construction de cette taille sera amplement suffisante dans la majorité des cas. Il serait également possible de garder une surface beaucoup plus grossière pour le reste du corps. De plus, l'évolution continue au niveau des résolutions maximales est tout à fait prometteuse à ce point de vue.

Modélisation de surface : La lumière structurée utilisée à l'étape précédente nous retourne un nuage de points. À partir de ces points, nous allons devoir générer une surface. Les informations que nous avons recueillies jusqu'à maintenant, position de la tête et carte de profondeurs, sont toutes estimées à quelques millimètres près. Il est donc plus important que la modélisation soit effectuée rapidement que précisément. C'est pourquoi nous proposons d'utiliser la solution la plus simple. En partant d'un groupe de quatre points voisins, on peut estimer la position d'un point central et ainsi générer quatre triangles. Une fois que tous les points ont été traités, nous obtenons une surface approximant relativement bien la peau du patient. Il serait toutefois intéressant de suivre de près l'évolution des algorithmes par ensembles de niveau ("Level-Set"). En effet, les surfaces reconstruites par ensembles de niveau sont beaucoup plus précises. La surface est calculée par itération et prend présentement trop de temps, mais plusieurs travaux sont présentement en cours pour accélérer ce processus.

Recalage : Plusieurs solutions existent pour recaler les différentes modalités, recueillies avant l'opération, entre elles. Par recalage, nous entendons donc plutôt l'association des données pré-opératoires avec celles calculées durant l'opération. Ce recalage sera donc de type élastique, en trois dimensions et en temps réel. Ce problème reste le mouton noir de notre système. En effet, le travail nécessaire est encore beaucoup trop lourd pour les machines actuelles. Certaines pistes de solution peuvent malgré tout servir à faciliter le développement d'une solution valide. Ainsi, tous les déplacements dans notre scène devraient se faire de façon relativement lente. Les mouvements seront généralement continus

également. Il est donc plausible de supposer que la solution du recalage au temps t peut servir d'hypothèse de départ au temps $t + \Delta t$. La respiration du patient étant cyclique, on peut également se servir d'interpolations et de tables de référence pour accélérer le traitement. Il reste cependant à souligner le fait qu'il n'existe actuellement aucune solution complète qui arrive aux résultats que nous souhaitons.

Modèle de projection : Les images à projeter sont générées en utilisant une méthode de lancer de rayons. Une image est premièrement calculée en fonction du point de vue du chirurgien, cette image comporte les projections des différents modèles 3D, comme la ligne de coupe ou le pourtour d'une tumeur à enlever, ainsi que certaines informations utiles à la chirurgie. Chaque point de cette image est ensuite considéré indépendamment afin de trouver ses coordonnées à l'intérieur du plan de chacun des projecteurs utilisés. Nous utilisons plusieurs projecteurs pour s'assurer que les projections soient toujours visibles, malgré les multiples risques d'occlusions. Les points de l'image initiale sont donc lancés dans la scène et, à partir des coordonnées 3D ainsi obtenues, nous pouvons calculer précisément quels pixels des projecteurs devront être allumés. Puisque le système suggéré, tout comme notre simulation, utilisera plusieurs projecteurs, nous devrons gérer un problème de recouvrement. Pour éviter que certains points soient plus éclairés que d'autres, l'intensité de chaque point sera divisée par le nombre de projecteurs éclairant ce point.

7.2 Discussion

Comme nous l'avons vu, l'implantation d'un système, tel que nous le souhaiterions, n'est pas encore possible. Il reste encore certains problèmes trop complexes pour être résolus efficacement. Le recalage est définitivement l'élément clé à résoudre pour arriver à développer notre pointeur intelligent de façon complètement automatique

et en temps réel.

Le calibrage des caméras et des projecteurs est un autre problème, qui n'a pas vraiment été traité dans ce mémoire. Il existe des solutions automatiques et valides (voir [68]) pour ce genre de problème. Il reste cependant important de mettre en évidence que l'utilisation de plusieurs caméras et projecteurs implique une phase de calibrage non négligeable.

Nous avons également parlé de parallélisme à quelques reprises. Une solution économique et efficace nécessitera l'utilisation de plusieurs machines. Ici aussi le problème a déjà fait l'objet de plusieurs recherches, mais nous devrons peut-être malgré tout penser à un protocole de communication efficace. Bien que les réseaux actuels soient rapides, nous traitons quand même une quantité d'informations non négligeable.

Finalement, la profondeur de champs des projecteurs pourrait également poser un problème. Les projecteurs actuels peuvent afficher des images sur quelques centimêtres de profondeur sans nécessiter d'ajustement. Toutefois, si la variation de profondeur dans notre zone de projection est trop grande, certains points apparaîtront flous. Il serait donc avantageux d'utiliser différents projecteurs pour chaque région de projection.

7.3 Futur vs Réalité

La photo qui suit montre une scène du film "Final Fantasy : The Spirit Within", développé par SquareSoft inc. en 2001. On y voit une chirugienne opérer un patient à partir d'un modèle lumineux projeté dans les airs. Cette photo permet d'imaginer à quoi pourrait ressembler les chirurgies assistées par ordinateur (CAO) du futur. Évidemment, les technologies actuelles ne permettent pas ce genre d'exploits, mais il est amusant de voir les ressemblances entre les travaux de recherche actuels et les idées que peuvent avoir certains artistes.

Photo tirée du film “Final Fantasy : The Spirit Within”, développé par SquareSoft inc. en 2001.

RÉFÉRENCES

[1] Jeremy Ackerman. UNC ultrasound/medical augmented reality research. *http://www.cs.unc.edu/Research/us/*, Juin 2000.

[2] B. Danette Allen, Gary Bishop, et Greg Welch. Tracking: Beyond 15 minutes of thought. *SIGGRAPH 95, Course 11*, Août 2001.

[3] Nina Amenta, Marshall Bern, et Manolis Kamvysselis. A new Voronoi-based surface reconstruction algorithm. Dans *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 415–421, Orlando, États-Unis, Juillet 1998.

[4] Michel A. Audette, Frank P. Ferrie, et Terry M. Peters. An algorithmic overview of surface registration techniques for medical imaging. *Medical Imaging Analysis*, 4(3):201–217, Septembre 2000.

[5] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environment*, 6(4):355–385, Août 1997.

[6] Ronald T. Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, et Blair MacIntyre. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, Novembre/Décembre 2001.

[7] Ronald T. Azuma et Gary Bishop. Improving static and dynamic registration in an optical see-through HMD. Dans *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 197–204, Orlando, États-Unis, Juillet 1994.

[8] Ronald T. Azuma et Gary Bishop. A frequency-domain analysis of head-motion prediction. Dans *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 401–408, Los Angeles, États-Unis, Septembre 1995.

[9] Michael Bajura, Henry Fuchs, et Ryutarou Ohbuchi. Merging virtual objects with real world: seeing ultrasound imagery within the patient. Dans *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 203–210, Chicago, États-Unis, Juillet 1992.

[10] Michael Bajura et Ulrich Neumann. Dynamic registration correction in video-based augmented reality systems. *IEEE Computer Graphics and Applications*, 15:52–60, 1995.

[11] Sumit Basu, Irfan Essa, et Alex Pentland. Motion regularization for model-based head tracking. Dans *Proceedings of the IEEE International Conference on Pattern Recognition*, pages 611–616, Vienne, Autriche, Août 1996.

[12] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Claudio Silva, et Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.

[13] Paul J. Besl et Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Février 1992.

[14] Margrit Betke, Harrison Hong, et Jane P. Ko. Automatic 3D registration of lung surfaces in computed tomography scans. Dans *Proceedings of the 4th International Conference on International Conference on Medical Image Computing*

and Computer-Assisted Intervention, pages 725–733, Utrecht, The Netherlands, Octobre 2001.

[15] Michael J. Black et Yaser Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. Dans *Proceedings of the 5th IEEE International Conference on Computer Vision 1995*, pages 374–381, Massachussetts Institute of Technology, Cambridge, États-Unis, Juin 1995.

[16] Yuri Boykov, Olga Veksler, et Ramin Zabih. Fast approximate energy minimization via graph cuts. Dans *Proceedings of the 7th IEEE International Conference on Computer Vision 1999*, volume 1, pages 377–384, Corfu, Greece, Septembre 1999.

[17] Pascal Cachier et Xavier Pennec. 3D non-rigid registration by gradient descent on a gaussian-windowed similarity measure using convolutions. Dans *Proceedings of IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 182–189, Hilton Head Island, États-Unis, Juin 2000.

[18] S. Y. Chen et Y. F. Li. Self-recalibration of a colour-encoded light system for automated 3-d measurements. *MeasureSciTech*, 14(1):33–40, Janvier 2003.

[19] David Cohen-Steiner et Frank Da. A greedy delaunay based surface reconstruction algorithm. Rapport Technique ECG-TR-124202-01, Information Society Technologies, 2000.

[20] Ingemar J. Cox, Sunita L. Hingorani, et Satish B. Rao. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, Mai 1996.

[21] James Edwin Cryer, Ping-Sing Tsai, et Mubarak Shah. Combining shape from

shading and stereo using human vision model. Rapport Technique CS-TR-92-25, University of Central Florida, Orlando, États-Unis, 1992.

[22] James Davis, Ravi Ramamoorthi, et Szymon Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. Dans *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 359–366, Madison, États-Unis, Juin 2003.

[23] L. Dekker, I. Douros, B. F. Buxton, et P. Treleaven. Building symbolic information for 3D human body modeling from range data. Dans *Proceedings of the Second International Conference on 3-D Digital Imaging and Modeling*, pages 388–397, Ottawa, Canada, Octobre 1999.

[24] Richard O. Duda, Peter E. Hart, et David G. Stork. *Pattern Classification.* Wiley-Interscience, Toronto, Canada, 2 édition, 2001.

[25] Charles R. Dyer. Volumetric scene reconstruction from multiple views. Dans *Foundations of Image Understanding*, pages 469–489. Kluwer Academic Publishers, Boston, États-Unis, Août 2001.

[26] Douglas Enright, Ronald Fedkiw, Joel Ferziger, et Ian Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83–116, Novembre 2002.

[27] Paolo Favaro et Stefano Soatto. Learning shape from defocus. Dans *Proceedings of 7th European Conference on Computer Vision 2002*, volume 2, pages 735–745, Copenhagen, Denmark, Mai 2002.

[28] James D. Foley, Andries van Dam, Steven K. Feiner, et John F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, Reading, États-Unis, 2 édition, 1990.

[29] André Fortin. *Analyse numérique pour ingénieurs.* Édition de l'École Polytechnique de Montréal, Montréal, Canada, 1995.

[30] David T. Gering, Arya Nabavi, Ron Kikinis, W. Eric L. Grimson, Noby Hata, Peter Everett, Ferenc Jolesz, et William M. Wells. An integrated visualization system for surgical planning and guidance using image fusion and interventional imaging. Dans *Proceedings of the Second International Conference on International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 809–819, Cambridge, England, Septembre 1999.

[31] Meenakshisundaram Gopi et Shankar Krishnan. A fast and efficient projection-based approach for surface reconstruction. *International Journal of High-Performance Computer Graphics*, 1(1):1–12, 2000.

[32] Dmitry O. Gorodnichy, Shahzad Malik, et Gerhard Roth. Affordable 3D face tracking using projective vision. Dans *Proceedings of the 15th International Conference on Vision Interface*, pages 383–391, Calgary, Canada, Mai 2002.

[33] W.E.L. Grimson, G.J. Ettinger, S.J. White, T. Lozano-Pérez, W.M. Wells III, et R. Kikinis. An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization. Dans *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 430–436, Seattle, États-Unis, Juin 1994.

[34] Olaf Hall-Holt et Szymon Rusinkiewicz. Stripe boundary codes for real-time structured-light range scanning of moving objects. Dans *Proceedings of the 8th IEEE International Conference on Computer Vision 2001*, volume 2, pages 359–366, Vancouver, Canada, Juillet 2001.

[35] Chin-Chuan Han, Hong-Yuan Mark Liao, Gwo-Jong Yu, et Liang-Hua Chen.

Fast face detection via morphology-based pre-processing. *Pattern Recognition*, 33(10):1701–1712, 2000.

[36] Antonio Haro, Myron Flickner, et Irfan Essa. Detecting and tracking eyes by using their physiological properties, dynamics, and appearance. Dans *Proceedings of the 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1163–1168, Hilton Head Island, États-Unis, Juin 2000.

[37] Heiko Hirschmüller, Peter R. Innocent, et Jon Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1):229–246, Avril 2002.

[38] Richard Lee Holloway. *Registration Errors in Augmented Reality Systems*. PhD thesis, University of North Carolina at Chapel Hill, 1995.

[39] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, et Werner Stuetzle. Surface reconstruction from unorganized points. Dans *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 71–78, Chicago, États-Unis, Juillet 1992.

[40] Marco C. Jacobs, Mark A. Livingstone, et Andrei State. Managing latency in complex augmented reality systems. Dans *Proceedings of the 1997 Symposium on Interactive 3D Graphics*, pages 49–54, Providence, États-Unis, Avril 1997.

[41] Calvin R. Maurer Jr., J. Michael Fitzpatrick, Robert L. Galloway Jr., Matthew Y. Wang, Robert J. Maciunas, et George S. Allen. The accuracy of image-guided neurosurgery using implantable fiducial markers. Dans *Proceedings of the 9th International Symposium and Exhibition on Computer Assisted Radiology*, pages 1997–1202, Berlin, Germany, Juin 1995.

[42] Sing Bing Kang, Richard Szeliski, et Jinxiang Chai. Handlig occlusions in dense multi-view stereo. Dans *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 103–110, Kauai, États-Unis, Decembre 2001.

[43] Ashish Kapoor et Rosalind W. Picard. Real-time, fully automatic upper facial feature tracking. Dans *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition*, pages 10–15, Washington, États-Unis, Mai 2002.

[44] Shinjiro Kawato et Nobuji Tetsutani. Detection and tracking of eyes for gaze-camera control. Dans *Proceedings of the 15th International Conference on Vision Interface*, pages 348–353, Calgary, Canada, Mai 2002.

[45] Reinhard Koch. 3-d surface reconstruction from stereoscopic image sequences. Dans *Proceedings of the 5th IEEE International Conference on Computer Vision 1995*, pages 109–114, Massachussetts Institute of Technology, Cambridge, États-Unis, Juin 1995.

[46] Kiriakos N. Kutulakos et Steven M. Seitz. A theory of shape by space carving. Dans *Proceedings of the 7th IEEE International Conference on Computer Vision 1999*, pages 307–314, Corfu, Greece, Septembre 1999.

[47] Jan Kybic. *Elastic Image Registration using Parametric Deformation Models*. PhD thesis, École Polytechnique Fédérale de LaÉtats-Unisnne, 2001.

[48] Yongmin Li, Shaogang Gong, Jamie Sherrah, et Heather Liddell. Multi-view face detection using support vector machines and eigenspace modelling. Dans *Proceedings of IEEE International Conference on Knowledge-based Intelligent*

Engineering Systems and Allied Technologies, pages 241–244, Brighton, UK, Août 2000.

[49] Yongmin Li, Shaogang Gong, Jamie Sherrah, et Heather Liddell. Support vector regression and classification based multi-view face detection and recognition. Dans *Proceedings of IEEE International Conference on Face and Gesture Recognition*, pages 300–305, Grenoble, France, Mars 2000.

[50] J. B. Antoine Maintz et Max A. Viergever. An overview of medical image registration methods. *Symposium of the Belgian hospital physicist association*, 12:V:1–22, 1996/1997.

[51] J.B. Antoine Maintz, Eric H.W. Meijering, et Max A. Viergever. General multimodal elastic registration based on mutual information. *IEEE Transaction on Image Processing*, 3338:144–154, 1998.

[52] Shahzad Malik, Gerhard Roth, et Chris McDonald. Robust corner tracking for real-time augmented reality. Dans *Proceedings of the 15th International Conference on Vision Interface*, pages 399–406, Calgary, Canada, Mai 2002.

[53] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 1. Academic Press, 1979.

[54] J.P. Mellor. *Enhanced Reality Visualization in a Surgical Environment.* PhD thesis, Massachusetts Institute of Technology, 1995.

[55] Karsten Mühlmann, Dennis Maier, Jürgen Hesser, et Reinhard Männer. Calculating dense disparity maps from color stereo images, an efficient implementation. *International Journal of Computer Vision*, 47(1):79–88, Avril 2002.

[56] Jane Mulligan, Volkan Isler, et Kostas Daniilidis. Trinocular stereo: A real-time algorithm and its evaluation. *International Journal of Computer Vision*, 47(1):51–61, Avril 2002.

[57] Dibyendu Nandy et Jezekiel Ben-Arie. A neural network approach for reconstructing surface shape from shading. Dans *Proceedings of the 1998 IEEE International Conference on Image Processing*, volume 2, pages 972–976, Chicago, États-Unis, Octobre 1998.

[58] Dibyendu Nandy et Jezekiel Ben-Arie. Shape from recognition and learning: Recovery of 3-d face shapes. Dans *Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2–7, Ft. Collins, États-Unis, Juin 1999.

[59] Office québécoise de la langue française. Le grand dictionnaire terminologique. *http://www.granddictionnaire.com/*, 2003.

[60] Stanley Osher et Ronald Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, New-York, États-Unis, Novembre 2002.

[61] Sébastien Ourselin, Radu Stefanescu, et Xavier Pennec. Robust registration of multi-modal images: towards real-time clinical applications. Dans *Proceedings of the 5th International Conference on International Conference on Medical Image Computing and Computer-Assisted Intervention*, volume 2489, pages 140–147, Tokyo, Japan, Septembre 2002.

[62] Xavier Pennec, Pascal Cachier, et Nicholas Ayache. Understanding the "demon's algorithm": 3D non-rigid registration by gradient descent. Dans *Proceedings of the Second International Conference on International Conference*

on Medical Image Computing and Computer-Assisted Intervention, pages 597–605, Cambridge, England, Septembre 1999.

[63] Alex Pentland. Shape information from shading : A theory about human perception. *Spatial Vision*, 4(2):165–182, 1989.

[64] Sylvain Petitjean et Edmond Boyer. Regular and non-regular point sets: Properties and reconstruction. *Computational Geometry – Theory and Application*, 19(2-3):101–126, 2001.

[65] Chuchart Pintavirooj et Manas Sangworasil. 3D-shape reconstruction based on radon transform with application in volume measurement. Dans *Proceedings of the Winter School of Computer GraphicsInternational Conference*, volume 10, page POS33, Bory, Czech public, Février 2002.

[66] Marc Pollefeys. Tutorial on 3D modeling from images. *In conjunction with European Conference on Computer Vision 2000*, Juin 2000.

[67] Ramesh Raskar, Greg Welch, et Wei-Chao Chen. Table-top spatially-augmented reality: Bringing physical models to life with projected imagery. Dans *Second IEEE and ACM International Workshop on Augmented Reality*, pages 64–74, San Francisco, États-Unis, Octobre 1999.

[68] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, et Henry Fuchs. The office of the future : A unified approach to image-based modeling and spatially immersive displays. Dans *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 179–188, Orlando, États-Unis, Juillet 1998.

[69] Jan Rexilius, Simon K. Warfield, Charles R.G. Guttmann, X. Wei, R. Benson, L. Wolfson, Martha Elizabeth Shenton, Heinz Handels, et Ron Kikinis. A

novel nonrigid registration algorithm and applications. Dans *Proceedings of the 4th International Conference on International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 923–931, Utrecht, The Netherlands, Octobre 2001.

[70] Sébastien Roy et Ingemar J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. Dans *Proceedings of the 6th IEEE International Conference on Computer Vision 1998*, pages 492–502, Bombai, India, Janvier 1998.

[71] Juan Ruiz-Alzola, Carl-Frederik Westin, Simon K. Warfield, Arya Nabavi, et Ron Kikinis. Nonrigid registration of 3D scalar, vector and tensor medical data. Dans *Proceedings of the 5th International Conference on International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 541–550, Pittsburg, États-Unis, Août 2000.

[72] Szymon Rusinkiewicz, Olaf Hall-Holt, et Marc Levoy. Real-time 3D model acquisition. Dans *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 438–446, San Antonio, États-Unis, Juillet 2002.

[73] Daniel Scharstein et Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, Avril 2002.

[74] Daniel Scharstein et Richard Szeliski. High-accuracy stereo depth maps using structured light. Dans *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 195–202, Madison, États-Unis, Juin 2003.

[75] Chanin Sinlapeecheewa et Kiyoshi Takamasu. 3D profile measurement by color pattern projection and system calibration. Dans *Proceedings of the 2002 IEEE International Conference on Industrial Technology*, volume 1, pages 405–410, Bangkok, Thailand, Decembre 2002.

[76] Gregory G. Slabaugh, W. Bruce Culbertson, Thomas Malzbender, et Ronald W. Schafer. A survey of methods for volumetric scene reconstruction from photographs. Dans *International Workshop on Volume Graphics*, pages 81–100, Stony Brook, États-Unis, Juin 2001.

[77] Softimage. Softimage—xsi. *http://www.softimage.com/home/*, 2003.

[78] Sascha Spors et Rudolf Rabenstein. A real-time face tracker for color video. Dans *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–4, Utah, États-Unis, Mai 2001.

[79] Radu Stefanescu, Xavier Pennec, et Nicholas Ayache. Parallel non-rigid registration on a cluster of workstations. Dans Sofie Norager, editeur, *Proceedings of HealthGrid'03*, Lyon, France, Janvier 2003. European Commission, DG Information Society.

[80] A. James Stewart et Michael S. Langer. Towards accurate recovery of shape from shading under diffuse lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):1020–1025, Septembre 1997.

[81] Richard Szeliski et Ramin Zabih. An experimental comparison of stereo algorithms. Dans *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, pages 1–19, Corfu, Greece, Septembre 1999.

[82] San-Lik Tang, Chee-Keong Kwoh, Ming-Yeong Teo, Ng Wan Sing, et Keck-Voon Ling. Augmented reality systems for medical applications. *IEEE Engineering in Medecine and Biology*, pages 49–58, May-June 1998.

[83] Jean Philippe Tarel et Nozha Boujemaa. Une approche floue du recalage 3D : généricité et robustesse. Dans *10ème congrès AFCET, Reconnaissance des Formes et Intelligence Artificielle*, Rennes, France, Janvier 1996.

[84] Jean-Philippe Thirion. Image matching as a diffusion process: an analogy with maxwell's demons. *Medical Image Analysis*, 2(3):243–260, 1998.

[85] Emanuele Trucco et Alessandro Verri. *Introductory Techniques for 3-D Computer Vision.* Prentice-Hall, Inc., Upper Saddle Rier, États-Unis, 1998.

[86] Eric W. Weisstein. Eric W. Weisstein's world of mathematics. *http://mathworld.wolfram.com/*, 2003.

[87] Greg Welch et Eric Foxlin. Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer Graphics and Applications, special issue on Tracking*, 22(6):24–38, November/December 2002.

[88] William M. Wells, Paul Viola, Hideki Atsumi, Shin Nakajima, et Ron Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35–52, 1996.

[89] Heinz Wörn et Harald Hoppe. Augmented reality in the operating theatre of the future. Dans *Proceedings of the 4th International Conference on International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 1195–1196, Utrecht, The Netherlands, Octobre 2001.

[90] Jie Yang et Alex Waibel. Tracking human faces in real-time. Rapport Technique CMU-CS-95-210, Carnegie Mellon University, Pittsburgh, États-Unis, 1995.

[91] Jie Yang et Alex Waibel. A real-time face tracker. Dans *3rd IEEE Workshop on Application of Computer Vision*, pages 142–147, Sarasota, États-Unis, Decembre 1996.

[92] Ming-Hsuan Yang, Norenda Ahuja, et David Kriegman. Face detection using a mixture of factor analyzers. Dans *Proceedings of the 1999 IEEE International Conference on Image Processing*, volume 3, pages 612–616, Kobe, Japon, Octobre 1999.

[93] Ming-Hsuan Yang, David Kriegman, et Narendra Ahuja. Detecting faces in images: A survey. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, pages 34–58, Janvier 2002.

[94] Yizhou Yu. Surface reconstruction from unorganized points using self-organizing neural networks. Dans *Proceedings of IEEE Visualization 1999 Conference*, pages 61–64, San Francisco, États-Unis, Octobre 1999.

[95] Li Zhang, Brian Curless, et Steven M. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. Dans *Proceedings of the 1st International Symposium on 3D Data Processing Visualization and Transmission*, pages 24–37, Padova, Italy, Juin 2002.

[96] Li Zhang, Brian Curless, et Steven M. Seitz. Spacetime stereo: Shape recovery for dynamic scenes. Dans *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 367–374, Madison, États-Unis, Juin 2003.

[97] Ye Zhang et Chandra Kambhamettu. Robust 3D head tracking under partial occlusion. Dans *Fourth IEEE IEEE International Conference on Automatic Face and Gesture Recognition*, pages 176–182, Grenoble, France, Mars 2000.

[98] Ye Zhang et Chandra Kambhamettu. 3D head tracking under partial occlusion. *Pattern Recognition*, 35(7):1545–1557, Juillet 2002.

[99] Hong-Kai Zhao, Stanley Osher, et Ronald Fedkiw. Fast surface reconstruction and deformation using the level set method. Dans *Proceedings of IEEE Workshop on Variational and Level Set Methods in Computer Vision*, pages 194–202, Vancouver, Canada, Juillet 2001.

[100] Hong-Kai Zhao, Stanley Osher, et Myungjoo Kang. Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Computer Vision and Image Understanding*, 80:295–319, 2000.

[101] Zhiwei Zhu, Kikuo Fujimura, et Qiang Ji. Real-time eye detection and tracking under various light conditions and face orientations. Dans *Proceedings of the symposium on Eye tracking research and applications*, pages 139–144, Mars 2002.

Annexe A

INTELLIGENT POINTER IN COMPUTER ASSISTED SURGERY - DESIGN AND FEASIBILITY

Article publié dans le cadre de la conférence internationale *IEEE Engineering in Medicine and Biology Society* en septembre 2003.

A.1 Intelligent Pointer in Computer Assisted Surgery–Design and Feasibility

N. L. Lewis[1], J. Meunier[1,2]

[2] Départment d'Informatique et de Recherche Opérationnelle, Université de Montréal, Montréal, Canada

[3] Biomedical Engineering Institute, Université de Montréal and École Polytechnique, Montréal, Canada

Abstract–**We present a solution combining the newest technologies in computer vision and preoperatively planned surgical intervention to enhance the efficiency of complex surgery. There have been many solutions proposed to use computer vision in the operating room, but they often depend on annoying and expensive components such as Head Mounted Display (HMD). These systems have proven to be of weak precision and, due to their weight and size, they tend to be really disturbing if worn during a long period of time. The goal of this work is to demonstrate the possibility of a system projecting directly on the patient, in real time, during the surgery. The main advantage is to keep the attention of the surgeon focused directly on his patient at all times. The information that**

can be added to the scene, due to the absence of HMD is evidently restricted to 2D since only one image is projected on the patient (skin, bone, surgery linen etc.) instead of two images for the right and left eyes with HMD, but by using an intelligent pointer to highlight important zones in the line of sight of the surgeon, 3D information can be inferred. This system actually transforms the patient body itself into a visual data source for the surgeon.[3]

Keywords–**Augmented reality, computer assisted surgery (CAS), modeling, non-planar projection**

I. INTRODUCTION

Surgeons can already take advantage of augmented reality (AR) to look directly at the operating field instead of at a monitor to get all kind of visual information. Typical AR systems rely on head-mounted displays (HMD) embedding a small display which projects a virtual image on a semi-transparent glass allowing the simultaneous viewing of the real and virtual scenes. However, because HMD are somewhat cumbersome, their acceptance by surgeons in the operating room is limited. Keeping AR while removing HMD is obviously not easy. One solution offered by Wörn and Hoppe in [A.1] is really promising. They use structured light to create the model of their patient's head and markers to track the head in order to project operation planning information during the surgery. Their system has been created for head surgery, with models that are allowed to move, but not to be deformed. In this paper, we propose a more general solution that could be applied in almost any surgery setup. The main differences are the continuous scanning of the patient and tracking of the surgeon's head for precise perspective projection.

To infer the 3D location of a tumor or any other structure inside the patient's body we also introduce an intelligent pointer to highlight important surgical zones in the

[3] This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds Nature et Technologies du Québec (NATEQ).

line of sight of the surgeon. Our idea follows the work of Raskar et al. "Office of the Future" [A.2] where real-time computer vision techniques are used to dynamically extract depth and reflectance information of surfaces in an office (walls, furniture, etc.) to project images on them without distortion.

This work is a partially implemented proof of concept, so there are still many known issues that will have to be fixed before our system is fully automatic and usable. For now, we have concentrated on projecting a deformed image over a dynamic non-regular surface.

II. METHODOLOGY

Since we want our projection to be readable, with possibly 3D perspective, we have to supervise any movement in the scene. To do so, we have separated this section into four different parts. The first one must be done continually and with accuracy and precision and consist in determining exactly the direction of the surgeon's gaze to create an accurate projection.

In parallel, we must scan precisely the patient's body to keep an up-to-date 3D model surface to project on. Since the patient is not supposed to move neither much, nor fast, this could be done more sparsely in time.

With both these data, the surgeon's gaze direction and the patient 3D model, we can now compute the image we want the surgeon to see. This image could contain 2D data (medical images, alphanumeric or graphic data etc.) and 3D information from an intelligent pointer highlighting important zones in the line of sight of the surgeon.

After the image has been created, the system has to deform it so the information is readable, even if projected over a non-planar projection surface, such that the projected image appears from the surgeon's point of view just as it would be if it was projected on a plane wall.

A. Head Tracking

Even if we do not actually implement this section, we analyze some of the possibilities that have already been suggested. The easiest solution would be to use a

magnetic tracker with sensors located on the surgeon's head. Even if we wish to eliminate any hardware overhead, small sensors could be put on each side of the head. A majority of surgeons already wear lenses that could hold those sensors without being too much of a burden. Yet, this solution is not general enough for us and does in fact require some possibly annoying new materials. Other solutions have been studied. Since we can afford a lot of preprocessing, it could be possible to scan the face of the surgeon before an operation and seek for its signature in images taken from a given camera [A.3,A.4]. Looking for face features, into a full scene, becomes quickly expensive in processing time but a solution is often to limit the search area. Bala [A.5] and Yang [A.6] offer real-time solutions, with reasonable results, that search for color pattern in the scene. This solution could be integrated in our research if the surgeon wears a mask and a cap of a different color than his surgical team-mates.

We underline the fact that what we really are looking for here is only the position of the head. Our projection will be done assuming the surgeon is looking in the good direction, i.e. the surgical zone, but this assumption does not eliminate any generality. In fact, the projected image does not change with movement of the gaze but rather only with the actual position of the head.

B. Modeling

Many solutions have been suggested over the past years to create depth map of models without having to touch them. Because we use a projector in our setup, structured light patterns that have been used successfully in the past by several authors [A.1,A.2,A.7,A.8,A.9] are certainly the best choice. However this approach brings some unavoidable difficulties that must be taken into account such as the already oversaturated light system in the operation room, possible occlusions and synchronization between projector(s) and the camera. These problems have already been addressed in other context [A.2], and could be adapted to the operating room setup. In this paper we use an operating room synthetic model that can be modified dynamically for simulation purposes.

C. Computing the surgeon image

The information relevant to the surgeon and projected on the patient during the intervention could be of two types: 2D and 3D. 2D examples are (1) medical images such as Xrays, (2) graphics such as ECG, (3) alphanumeric data such as blood pressure, temperature, respiration and any other vital signs. 3D examples are tumor location inside the patient or incision position on the patient. In this paper we concentrate on the more difficult 3D type because a solution to this problem gives an immediate solution to the much easier 2D case.

3D information requires a preoperative model of the patient obtained from segmented MRI or CT data for instance [A.1]. Using a registration algorithm, we can associate the preoperative information with the real (intraoperative) patient itself. Then it becomes possible to find the position on the patient skin that we want to get highlighted. Those points are calculated by intersecting a ray between the known surgeon's gaze and the scanned intra-operative patient's model.

Once we know the position of the points to be highlighted, we can compute their projections for the surgeon's point of view and add the remaining 2D data aside. When both the 3D and the 2D information have been computed and gathered, we have the "surgeon image", the plane image we want the surgeon to see. Next, we have to compute the "projector image", so that once projected and deformed over the body's surface, the surgeon can view his (surgeon's) image without any distortion.

D. Computing the projector image

This step computes the projector image that will actually be projected. This image is a distorted image that becomes undistorted when seen form the surgeon's point of view.

For this purpose, a ray-tracing approach is used (fig. A.1). From the previous step, we have a group of points to illuminate that are known from the surgeon's point of view as a 2D image. To obtain the positions of these points in the projector image, we simply "back-project" (after calibration) the points from the surgeon's eyes toward

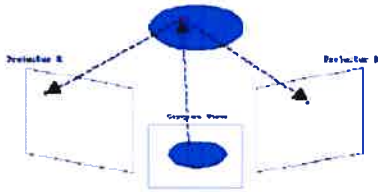

Figure A.1. The construction of the projector images is done with a simple raytracing technique.

the 3D scene (on the patient). These rays intersect with the mesh representing the patient's body in our system and give us the 3D location of each of the illuminated points.

From each of those points, a new ray is thrown to the projector(s). This allows knowing how many projectors "see" each of the points. Having the 2D coordinates in each projector image, we compute the color and intensity of each point, depending on the desired color, the receptive surface attributes (color, reflectance...) and the number of projectors seeing the given 3D point. We used only two projectors in our simulation, but the calculation done can be extended to any number of them without much difficulties.

The points seen from the surgeon's point of view are the result of the blending of the two (or more) projector's images. Using more projectors will reduce the risk of total occlusion during the surgery. The color projected from any given number of projector is simply calculated by dividing the intensity by the number of projector hits for each point. This solves the blending problem rapidly and without effort, but loosing some precision over the final color. For now, as long as it is readable at all time and do not disturb much the user, it is not a priority to obtain a perfect color.

III. RESULTS

In this paper we focus on the fast construction of projector images. These projections can be computed in real-time since it is simply a set of intersections between

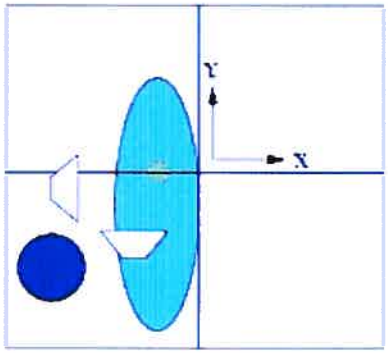

Figure A.2. Top view of the setup used to simulate an operating room. The two trapezoids represent the projectors, the large circle corresponds to the surgeon's head, the ellipsoid is the patient and the smaller circle symbolizes a tumor to be localized inside the patient's body.

some rays and a polygon mesh for the highlighted points. Typically, 200 to 500 points are generated in real-time, more points could be added if necessary without significant increase in time consumption. We run these simulations on a Celeron 1.3GHz personal computer with nVidia GeForce3 video card. This system is intended to work with any offthe-shelf hardware (cameras and projectors), keeping the price of a complete setup as low as possible.

The simulation setup illustrated in fig. A.2 shows approximately the relative position of the projectors, surgeon, patient and tumor used to test our methodology. Fig. A.3 shows the two distorted images of a circular shape used for each projector in order to get a perfect silhouette of the tumor on the patient when seen from the surgeon's point of view. Fig. A.4 presents the same images but this time as they appear when projected on the patient and seen from the projector's point of view. Finally fig. A.5 shows the highlighted silhouette of the tumor seen from the surgeon's point of view when both projections are combined.

The notion of intelligent pointer and 3D information emerges when comparing fig. A.5 to fig. A.6. Fig. A.6 illustrates a mesh (without rendering) of the same scene

as fig. A.5 to reveal the surgical region of interest (tumor) inside the patient. This tumor is pointed out by the intelligent pointer rounded silhouette and is directly in the surgeon's line of sight (when he looks to the patient). This means that if the surgeon moves his (tracked) head a bit aside, the silhouette will adjust accordingly to keep the tumor on his line of sight to help him identify its 3D location. Furthermore, if the patient moves or breath and the tumor remains well registered (in practice, a difficult problem indeed), the pointer will adjust accordingly.

IV. DISCUSSION

We can use two (or more) projectors to ensure that no obstruction happens although only one could be enough, if nothing could possibly block the rays from the projector or if one accepts that in that case, there will be missing data over the patient's body.

Notice that the other parts of a complete system (tracking the surgeon head, modeling the patient) have already their solutions, many of them in real-time as mentioned previously. A complete system could therefore be a parallel system computing each part for a global solution.

The alignment (calibration) and synchronization is always a critical part in any vision algorithm. Raskar et al. [A.2] have developed a way to auto-calibrate the camera and projector together, this would reduce greatly the time and resource needed for manual calibration.

The real-time intra-operative 3D registration of a preoperative model to the patient is the hardest problem to solve. The actual solutions often use cumbersome artificial markers fixed to the patient or require sophisticated algorithms not fast enough [A.1]. We are studying a way to refresh the model only after a threshold displacement has occurred by comparing time-to-time points reading. Slower solutions, without additional markers, could then be used for the registration itself.

V. CONCLUSION

We have presented a solution to use augmented reality (AR) in the operating room

without uncomfortable and expensive components, such as Head Mounted Display (HMD). Moreover, by projecting information directly on the patient the surgeon's focus remains where it is needed, accelerating his work, and reducing the risk resulting from inattention. Although the projected information is 2D, we have shown how, by using an intelligent pointer, 3D information can also be inferred.

ACKNOWLEDGMENT

The authors wish to thank Emric Epstein, Martin Granger-Pichet and Professor Pierre Poulin, from the LIGUM laboratory, for their technical suggestions and support with the intelligent projection system used in our simulations.

REFERENCES

[A.1] H. Wörn and H. Hoppe, Augmented Reality in the Operating Theatre of the Future, in *Proc. 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, Utrecht, The Netherlands, October 2001.

[A.2] R. Raskar, M. Cutts, A. Lake, L. Stesin and H. Fuchs, The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Display, in *Proc. of SIGGRAPH 98*, Orlando

[A.3] A. Kapoor and R. W. Picard, Real-Time, Fully Automatic Upper Facial Feature Tracking, in *Proc. 5th International Conference on Automatic Face and Gesture Recognition*, Washington, D.C., USA, May 2002.

[A.4] D. O. Gorodnichy, S. Malik and G. Roth, Affordable 3D Face Tracking Using Projective Vision, in *Proc. 15th International Conference on Vision Interface*, Calgary, Canada, 2002.

[A.5] L.-P. Bala, K. Talmi and J. Liu, Automatic Detection and Tracking of Faces and Facial Features in Video Sequences, presented at the *Picture Coding Symposium*, Berlin, Germany, 1997.

[A.6] J. Yang and A. Waibel, Tracking Human Faces in Real-Time, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, Rep. CMU-CS-95-210, 1995.

[A.7] L. Zhang, B. Curless and S. M. Seitz, Rapid Shape Acquisition Using Color

Figure A.3. Projector image showing a deformed pattern intended to look like a tumor silhouette from the surgeon's point of view when projected.

Figure A.4. Projected images added to the scene, from the projectors' point of view.

Structured Light and Multi-Pass Dynamic Programming, presented at the *1st International Symposium on 3D Data Processing Visualization and Transmission*, Padova, Italy, June 2002.

[A.8] S. Rusinkiewicz, O. Hall-Holt and M. Levoy, Real-Time 3D Model Acquisition, in *Proc. SIGGRAPH 2002*, San Antonio, Texas, USA, July 2002.

[A.9] O. Hall-Holt and S. Rusinkiewicz, Strip Boundary Codes for Real-Time Structured-Light Range Scanning of Moving Objects, in *Proc. 8th International Conference on Computer Vision*, Vancouver, Canada, July 2001.

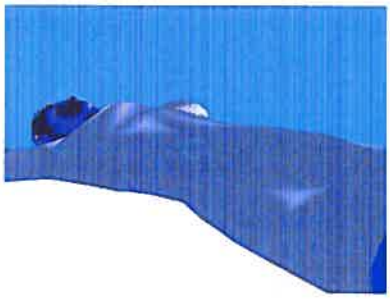

Figure A.5. Surgeon's point of view, showing the added projections on a simulated patient. The (green) silhouette is created from the projections of fig. A.3.

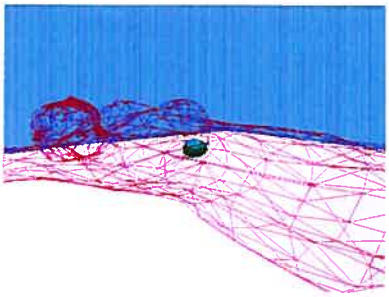

Figure A.6. The surgical area (tumor), highlighted in fig. 5, seen through a wireframed patient.