

# Conception et Validation d'une Architecture Haptique Indépendante des Modèles

Dominique Henry, Sébastien Druon, Philippe Fraise, André Crosnier

► **To cite this version:**

Dominique Henry, Sébastien Druon, Philippe Fraise, André Crosnier. Conception et Validation d'une Architecture Haptique Indépendante des Modèles. CIFA: Conférence Internationale Francophone d'Automatique, May 2006, Bordeaux, France. lirmm-00102828

**HAL Id: lirmm-00102828**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00102828>**

Submitted on 2 Oct 2006

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

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

# Conception et validation d'une architecture haptique indépendante des modèles

Didier HENRY, Sébastien DRUON, Philippe FRAISSE, André CROSNIER

Laboratoire d'Informatique, Robotique et Microélectronique de Montpellier  
UMR 5506

161 rue Ada, 34392 Montpellier Cedex 5, France

{henry,druon,fraisse,crosnier}@lirmm.fr

*Résumé*— Nous présentons ici une architecture pour la synthèse haptique. Nous nous attachons en particulier à la flexibilité de cette architecture afin de la rendre autant que possible indépendante des modèles utilisés.

Les modèles de déformation, de contact ainsi que le visualisation sont séparés. L'architecture assure la synchronisation temporelle ainsi que la cohérence des différents modèles. Elle a été implémentée et validée expérimentalement pour une simulation simplifiée de découpe virtuelle.

*Mots-clés*— retour haptique, interaction avec un matériau, architecture haptique

## I. INTRODUCTION

Les interfaces haptiques ouvrent depuis plusieurs années de nouvelles perspectives pour l'interaction entre l'homme et la machine. Venant compléter les traditionnels retours d'information textuelle, visuelle ou sonore, elles offrent une information sur les efforts de contact.

Il est par exemple possible d'utiliser une interface haptique pour la téléopération par un système maître-esclave, comme dans [1]. Mais une nouvelle classe d'application est apparue avec l'augmentation de la puissance de calcul des ordinateurs : la simulation d'environnements virtuels.

Une simulation d'interaction haptique s'appuie sur plusieurs éléments : simuler le comportement de l'environnement (déplacement, déformation, rupture, découpe...), donner un retour visuel d'information et enfin offrir un retour haptique. Ces trois composantes s'appuient sur plusieurs éléments logiciels ou bien physiques :

- un modèle de déformation de l'environnement,
- une interface d'affichage (écran),
- un modèle des efforts de contact,
- une interface haptique.

Il existe plusieurs approches pour constituer une simulation complète et cohérente. La première est de s'appuyer sur le modèle de déformation du matériau (l'environnement) afin de calculer les efforts de contact. Cette approche est utilisée par exemple dans [2] ou encore dans [3], [4].

Une seconde approche consiste à séparer les modèles de déformation, c'est-à-dire les lois de comportement du matériau, des modèles de contact. [5] et [6] utilisent une telle approche. La figure 1 fournit un aperçu d'une telle architecture logique. Notons qu'il existe deux classes de dispositifs haptiques, comme expliqué dans [7] : impédance et admittance. Les premiers mesurent des positions et retournent des efforts, les seconds mesurent des efforts et renvoient un

déplacement. Dans cet article, nous travaillons avec une interface haptique à impédance qui mesure une position  $X$  et renvoie une force  $F$ .

L'intérêt de l'approche utilisant deux modèles distincts est de donner plus de flexibilité dans la conception de chaque modèle et d'améliorer les performances du système. Malheureusement, un tel découplage ne se fait que sous certaines contraintes qui assurent la cohérence des modèles.

Dans les travaux cités précédemment, l'architecture n'est utilisée que pour des raisons de performances, ce qui constitue déjà un intérêt suffisant en soi. Toutefois, nous proposons ici d'aller au-delà et de concevoir une architecture permettant aussi de s'affranchir, dans une certaine mesure, des problèmes intrinsèquement liés aux modèles pour la rendre indépendante de ceux-ci.

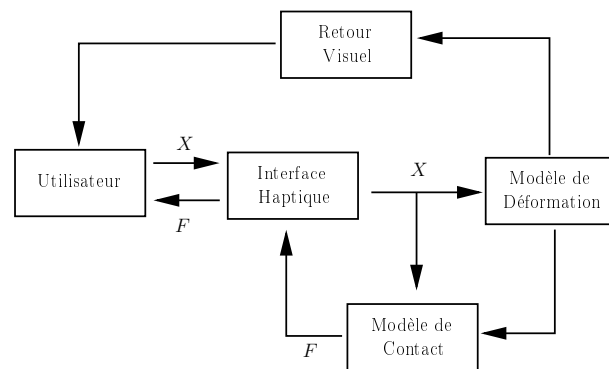


Fig. 1

MODÈLE GÉNÉRAL D'UN SIMULATEUR HAPTIQUE VIRTUEL

Dans un premier temps, nous mettrons en avant les problématiques soulevées par une architecture indépendante des modèles : les questions de fréquence d'échantillonnage et de continuité nous amèneront aux problèmes de synchronisation et de représentation de l'environnement virtuel. Nous proposerons ensuite une architecture s'appuyant sur une représentation géométrique intermédiaire et sur une fenêtre spaciale assurant la continuité locale de la synthèse d'efforts.

L'architecture proposée a été implémentée et nous présenterons un cas simple d'utilisation. Cela nous permettra de valider l'architecture pour les problématiques de re-

présentation géométrique, de synchronisation et de continuité. Nous aborderons aussi la question des performances purement informatiques d'une telle architecture, avant de conclure sur les développements possibles vers une application complète.

## II. PROBLÉMATIQUES

### A. Conditions requises pour les modèles

La modélisation d'un environnement virtuel est un problème en soi. Même pour une classe de matériaux donnée, il y a plusieurs approches possibles. Par exemple, les cas de la modélisation des matériaux mous a été traité par Cotin *et al.* dans [2] mais de nombreuses autres méthodes sont présentées dans [8]. Le cas de la modélisation visco-plastique a été étudié dans [9] et développée plus avant dans [10]. L'aspect haptique propose plusieurs modèles pour un type d'interaction donné.

Dans cet article nous ne discuterons pas des modèles en eux-même mais plutôt des conditions que chaque modèle doit remplir et de la façon de concevoir une architecture cohérente.

#### Conditions visuelles

Afin d'obtenir une simulation réaliste, il faut afficher l'outil virtuel et l'environnement à une fréquence de  $10 \leq F_d \leq 30$  Hz. Le modèle pour le retour visuel doit être un maillage triangulaire (ou toute autre représentation qui peut être convertie facilement en maillage de surface à la fréquence  $F_d$ ).

Ainsi, le modèle de déformation doit avoir une fréquence d'échantillonnage de  $F_d$ , et nous supposons qu'elle est de 30 Hz.

#### Conditions pour l'haptique

De nombreuses études, telles que [8], définissent le taux d'échantillonnage d'une simulation haptique comme  $F_h \approx 1$  kHz. La main humaine est beaucoup plus sensible à la fréquence d'échantillonnage que l'œil humain. Malheureusement, ce qui est gênant pour l'œil, par exemple une animation saccadée, devient dangereux lorsque l'on parle de retour d'effort. C'est pourquoi le taux de rafraîchissement ne doit pas être inférieur à  $F_h$ .

De plus, il faut éviter les discontinuités spaciales, à moins qu'elles ne représentent autre chose que des artefacts de modélisation : une discontinuité ne peut être acceptée que si l'environnement virtuel « affiché » est effectivement discontinu.

### B. Problème de synchronisation

Nous venons de remarquer que les modèles de déformation et de contact ont des besoins différents en terme de fréquence d'échantillonnage. Malheureusement, le modèle de contact dépend du modèle de déformation (voir fig. 1). Ainsi, une des entrées du modèle de contact est sous-échantillonnée. On peut voir cette configuration comme un bloqueur d'ordre zéro.

La solution évidente serait de sur-échantillonner simplement le modèle de déformation. Malheureusement, le problème de performance ne peut pas être ignoré; on peut même le voir comme le problème principal lorsqu'il s'agit

de modéliser et simuler une déformation. C'est pourquoi [2] linéarise les équations discrétisées dans le cas d'une déformation pure (c'est-à-dire sans découpe).

Dès lors, nous devons supposer que les informations sur la topologie locale, calculées à partir du modèle géométrique du matériau, sont sous-échantillonnées à  $F_d = 30$  Hz.

Afin de mieux comprendre les effets de ce bloqueur d'ordre zéro, nous allons supposer que le modèle de déformation peut amener des ruptures dans la structure du matériau simulé (ce qui arrive réellement lors d'une opération de découpe). La figure 2 illustre les efforts  $F(t)$  calculés dans le cas d'un modèle haptique de type élastique, avec un outil modélisé par un point. Ceci reste valide pour des modèles plus complexes.

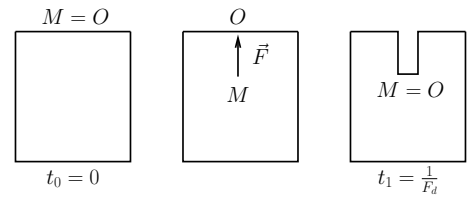


Fig. 2

CALCUL HAPTIQUE ENTRE DEUX PÉRIODES D'ÉCHANTILLONNAGE DU MODÈLE DE DÉFORMATION

À l'instant  $t_0 = 0$ , l'effort retourné est

$$\vec{F}(0) = -K_e \cdot \overrightarrow{OM} = \vec{0} \quad (1)$$

avec  $O$  la projection du point-outil  $M$  sur la surface de contact. Entre  $t_0 = 0$  et  $t_1 = \frac{1}{F_d}$ , l'outil descend dans le matériau avec un mouvement rectiligne et uniforme.  $F(t)$  augmente jusqu'à une valeur maximale  $F_{max} = F(t_1^-)$ . À l'instant  $t_1$ , le modèle de déformation est mis à jour et modifie la topologie du matériau au voisinage de  $M$ , ce qui amène  $OM = 0$ , d'où  $F(t_1^+) = 0$ .

En conséquence, la force  $F(t)$  est discontinue à l'instant  $t_1$  et plus généralement à tout instant  $t = \frac{k}{F_d}$ ,  $k \in \mathbb{N}$ , comme présenté en figure 3 :

$$\forall k \in \mathbb{N}, t_k = \frac{k}{F_d}, F(t_k^-) \neq F(t_k^+) \quad (2)$$

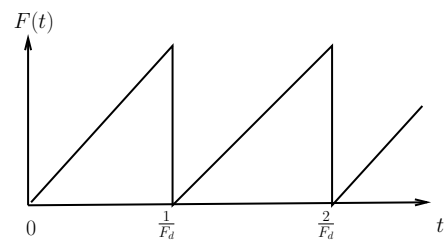


Fig. 3

EFFET DU BLOQUEUR D'ORDRE ZÉRO (SOUS-ÉCHANTILLONNAGE)

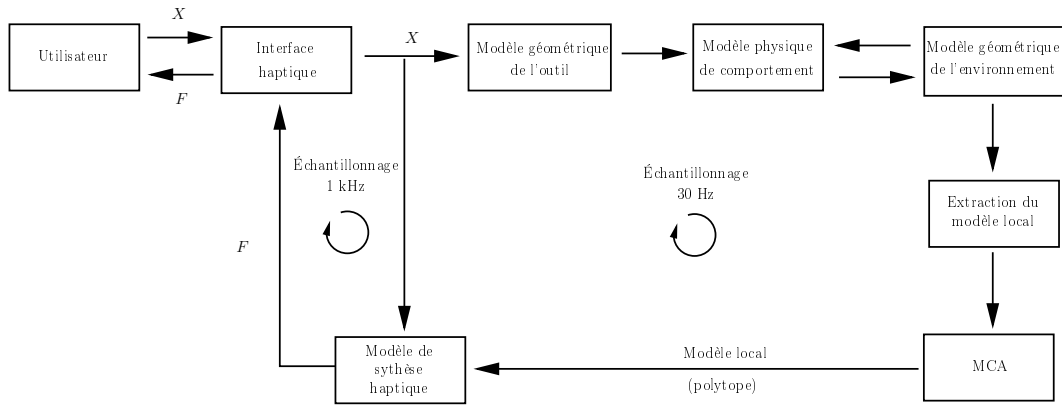


Fig. 4

ARCHITECTURE PROPOSÉE POUR LA SIMULATION DE L'INTERACTION AVEC UN MATÉRIAU VIRTUEL

### C. Représentation des données

Nous venons de remarquer que la fréquence d'échantillonnage de chaque modèle est très différente. Un autre aspect est la représentation de l'environnement. On peut manipuler un modèle de déformation et un modèle haptique qui soient totalement indépendants dans leur conception ; toutefois, la sortie du modèle de déformation doit être cohérente avec l'entrée du modèle de déformation (voir figure 1).

Il existe de nombreuses méthodes de représentation, parmi lesquelles on trouve les surfaces implicites, les maillages de surface (habituellement triangulaires), les maillages volumique (tétraèdres) ou même les éléments de volume (aussi connus sous le nom « voxel », *Volume Elements*).

D'un point de vue théorique, la représentation choisie doit être continue dans le temps et dans l'espace, et devrait permettre de détecter une collision avec d'autres objets mathématiques, ce qui suppose une mesure d'interpénétration. D'un point de vue pratique, sa mise à jour doit être facile et les données doivent être aussi compactes que possible.

### D. Un modèle unifié

Le modèle simple, présenté en figure 1, amène de nombreuses difficultés telles que la synchronisation et la représentation des données. Cela vient de la séparation entre les aspects liés à la déformation et au contact de la simulation. Toutefois, cette séparation est largement répandue, par exemple par [6].

L'avantage de cette séparation est qu'elle permet d'utiliser des modèles qui sont dédiés et optimisés pour un aspect particulier de la simulation. La simulation de la déformation à un grand taux d'échantillonnage n'est possible que grâce à l'utilisation intensive de pré-calcul, comme dans [2]. Malheureusement, nous avons supposé que l'environnement virtuel simulé subit des modifications topologiques importantes, ce qui nous empêche d'utiliser le pré-calcul. Nous devons trouver un moyen de contourner les problèmes mentionnés précédemment.

## III. ARCHITECTURE PROPOSÉE

### A. Structure améliorée

Afin de gérer les problèmes de synchronisation et de représentation des données, nous proposons une nouvelle architecture présentée en figure 4. Nous verrons que cette nouvelle structure est conforme à la structure présentée en figure 1. Tout d'abord, le retour visuel n'apparaît plus afin de simplifier le schéma. Évidemment, il fait toujours partie du système mais nous n'étudierons pas plus cet aspect de la simulation. Il n'y a aucun changement du point de vue de l'utilisateur : il interagit toujours avec un dispositif haptique à impédance. Le modèle de contact, ou modèle de synthèse haptique, reste le même.

Le modèle de déformation se décompose en trois éléments. Le premier est la représentation géométrique de l'outil. Ce modèle est à la position  $X$  et sert à l'élément suivant pour calculer l'impact de l'outil sur l'environnement virtuel. Le second est le modèle physique de comportement qui représente les lois de déformation (par exemple la mécanique des milieux continus, érosion, loi de Hooke) de l'environnement. Le dernier est le modèle géométrique de l'environnement, c'est-à-dire les informations qui représentent la configuration spatiale du matériau. Comme nous l'avons déjà expliqué, il existe plusieurs manières de le représenter (maillage, voxel, surfaces implicites...). Le modèle de comportement et la représentation géométrique interagissent en boucle, puisque la géométrie est modifiée par le modèle de comportement et que les calculs de ce modèle sont basés, entre autres choses, sur la géométrie. Enfin, le lien entre le modèle de déformation en trois éléments et le modèle haptique doit encore être détaillé.

### B. Représentation unifiée des données

Afin de lier les modèles de déformation et de contact, nous créons une représentation locale (locale au sens de « au voisinage de l'outil ») de la géométrie de l'environnement, comme présenté en figure 5. Nous choisissons une représentation de type voxel. On peut voir le voxel comme une grille spatiale régulière, où chaque cellule cubique contient ou non de la matière. Nous supposons que nous sommes capable de générer un voxel, représentant localement le matériau, à partir de sa représentation au sein

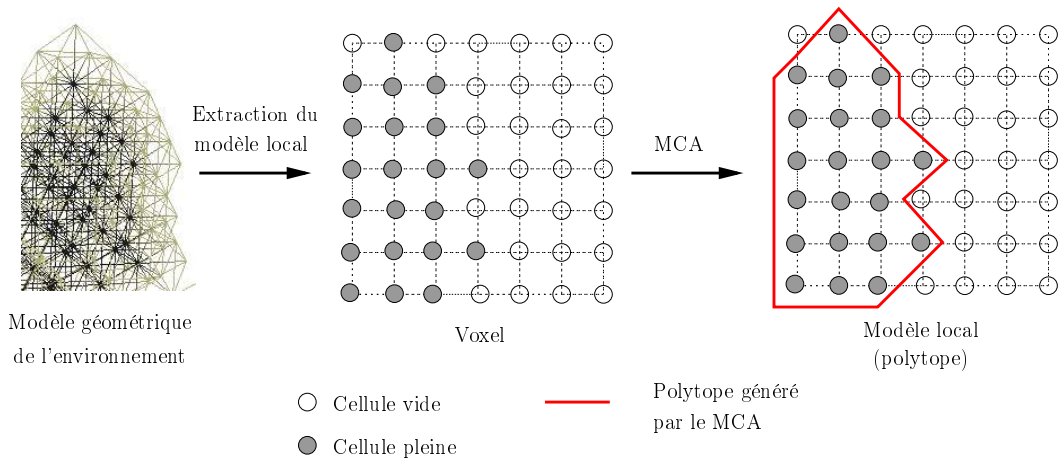


Fig. 5

EXTRACTION ET MCA : DU MODÈLE GÉOMÉTRIQUE À LA REPRÉSENTATION LOCALE

du modèle de déformation. Cela constitue la seule hypothèse que nous faisons sur le modèle de déformation : on peut en extraire une représentation voxel locale.

Le choix de la représentation voxel peut paraître étrange. En effet cette représentation est discrète par nature et sa résolution spatiale doit être suffisamment fine par rapport aux dimensions de l'outil. Mais cela n'est vrai que si la méthode de détection de collision s'appuie directement sur le voxel. Toutefois, on peut obtenir un maillage triangulaire à partir d'un espace voxel en utilisant l'algorithme des *Marching Cubes* (*Marching Cube Algorithm*, MCA) présenté dans [11]. Cet algorithme garantit la continuité de la surface générée à partir de l'espace voxel.

D'un point de vue pratique, le calcul du polytope par le MCA est très efficace grâce à des cas pré-calculés. L'algorithme se base sur la configuration des cellules. Dans le cas 2 D (figure 5) les segments sont générés pour chaque groupe de 4 cellules. Il y a au total  $2^4 = 16$  configurations. Dans le cas 3 D, il y a  $2^8 = 256$  configurations qui peuvent être pré-calculées et appliquées. Grâce à ce pré-calcul, le MCA a de bonnes performances dont nous discuterons par la suite.

Ainsi on peut obtenir une représentation surfacique approchée du matériau au voisinage de l'outil. Il est possible de limiter le nombre total de triangles générés par le MCA par un choix approprié de la résolution spatiale du voxel, ce qui nous permet d'utiliser des algorithmes de détection de collision « naïfs » (c'est-à-dire en testant chaque pair d'objet avec une complexité algorithmique de  $O(n.m)$ ).

Après avoir résolu le problème de détection des collision, le choix du modèle de contact haptique est indépendant du reste du système. Nous discuterons par la suite du modèle utilisé dans cette étude ainsi que les détails de l'algorithme de détection de collision.

### C. Synchronisation et discontinuités

Nous avons conçu une méthode qui fournit au modèle de contact une représentation locale de l'état interne du modèle de déformation, et cela du point de vue de modèle

haptique. Cette représentation est continue dans l'espace par construction, mais nous devons encore étudier sa continuité temporelle. Nous avons expliqué précédemment que le sous-échantillonnage pouvait introduire des discontinuités dans la synthèse haptique (voir fig. 3).

Afin d'éviter un tel comportement qui nuit grandement à la qualité (et à la stabilité) du retour haptique, nous allons limiter la zone d'influence du modèle de déformation. Connaissant le modèle géométrique de l'outil virtuel et sa position  $X(t)$ , on définit une zone « protégée » autour de cet outil. Comme on représente la topologie locale du matériau par un espace voxel, qui est une grille régulière, l'utilisation de boîtes englobantes alignées sur les axes (*Axis Aligned Bounding Box*, AABB) est immédiate.

Soit  $E(t)$  l'espace voxel du modèle haptique à l'instant  $t$  et  $E'(t + \frac{1}{F_d})$  le nouvel espace voxel, c'est-à-dire la sortie du modèle de déformation. L'opération

$$E\left(t + \frac{1}{F_d}\right) = E'\left(t + \frac{1}{F_d}\right) \quad (3)$$

crée des discontinuités dans le polytope au voisinage de l'outil et donc dans les efforts générés.

Soit AABB la boîte englobante de l'outil. On calcule le nouvel espace voxel pour le modèle de contact comme ceci :

$$\forall v \in E(t) \setminus \text{AABB}, v = v' \quad (4)$$

ce qui signifie que chaque cellule voxel  $v$  utilisée par le modèle de contact est mise à jour par la cellule correspondante  $v' \in E'(t)$  à moins qu'elle n'appartienne à l'AABB de l'outil. La représentation locale du matériau n'est donc mise à jour qu'avec un certain retard, ce qui diminue évidemment la précision mais assure en contrepartie la continuité des efforts ressentis par l'utilisateur. Finalement, la qualité du système est améliorée.

## IV. IMPLÉMENTATION

Afin de valider le principe de découplage, nous avons mis en place une plateforme de test. Le cas d'étude a été plutôt simplifié par rapport à un simulateur de découpe

complet, toutefois chaque élément présenté en figure 4 est implémenté.

### A. Architecture logique et physique

Chaque modèle ou élément a été séparé physiquement pour des raisons de performances qui constituent le principal facteur limitatif dans la conception d'un tel simulateur. Nous ne fournirons pas ici les aspects techniques de l'architecture logicielle et réseau sous-jacente.

L'interface haptique et le calcul haptique (contact) ont été réunis sur le même ordinateur qui est un bi-processeur cadencé à 1,7 GHz fonctionnant sous Linux et équipé de 1 Go de RAM. L'interface haptique est un PHANTOM Premium 6 DOF. Le calcul de la déformation et de l'affichage utilisent un second ordinateur mono-processeur cadencé à 2,6 GHz.

On utilise un protocole UDP sur un réseau local pour transmettre l'espace voxel généré par le modèle de déformation. Ces informations sont envoyées à 30 Hz, c'est-à-dire la fréquence d'échantillonnage  $F_d$ . Le périphérique haptique et le modèle haptique partagent des informations grâce à un accès en mémoire partagée et chaque élément utilise de façon intensive la programmation multi-tâches pour la communication inter-processus et la synchronisation.

### B. Les modèles

#### B.1 Modèle de contact

Le modèle choisi s'inspire du modèle présenté dans [12]. Il s'agit d'un modèle de matériau visqueux. L'outil est représenté par un ensemble de segments. L'équation des efforts est du type :

$$F = - \int_0^{d(t)} K_v \dot{x}(t) dx \quad (5)$$

où  $K_v$  est le coefficient de frottement visqueux,  $d(t)$  la profondeur de pénétration du segment dans le matériau et  $\dot{x}(t)$  la vitesse instantanée d'un point du segment. En faisant d'hypothèse simplificatrice que la vitesse instantanée est la même en tout point du segment, de valeur  $\dot{X}(t)$ , l'équation devient :

$$F = -K_v \dot{X}(t) d(t) \quad (6)$$

L'interface de retour haptique fournit à la fois la position  $X$  en 6 degrés de liberté et la vitesse  $\dot{X}$  en 3 degrés de liberté. Le coefficient  $K_v$  est réglé de manière arbitraire. En ce qui concerne la profondeur de pénétration  $d$ , son calcul se base sur un test d'intersection naïf : l'algorithme des Marching Cubes génère un polytope pour le voxel au sein de l'AABB. L'utilisation de l'algorithme présenté dans [13] nous permet de calculer l'ensemble des points d'intersection entre le polytope qui représente la surface du matériau et le modèle de l'outil (un ensemble de segments).

La figure 6 présente deux cas d'intersection entre un segment et un polytope en deux dimensions ; l'extension au cas 3 D est immédiate. Dans le premier cas (segment  $AB$ ), le nombre d'intersections est pair et la profondeur est  $d = d_1 + d_2$ . Dans le second cas (segment  $A'B'$ ) le nombre d'intersections est impair et nous devons supposer que le point  $A$  est toujours en-dehors du matériau. D'un point de vue physique,  $A$  représente la base de l'outil virtuel et  $B$

son extrémité. On calcule la profondeur de pénétration de la même manière.

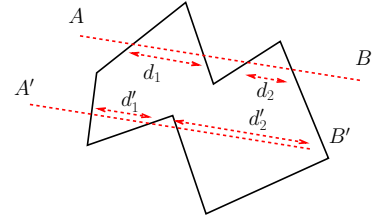


Fig. 6

INTERSECTION ENTRE UN SEGMENT ET UN POLYTOPE — CAS 2 D

#### B.2 Modèle de déformation et visualisation

Afin de valider le principe, nous avons utilisé le plus simple des modèles de déformation/découpe, c'est-à-dire l'érosion. L'outil est représenté par un seul segment et le matériau est modélisé comme un espace voxel. La génération de l'espace voxel au voisinage de l'outil pour le modèle de contact est donc immédiate.

Le modèle de comportement du matériau est simple : toute cellule voxel qui se trouve à une distance inférieure à un seuil donné est effacée.

Le retour visuel du matériau est fourni grâce au MCA, tandis qu'un simple maillage 3 D est utilisé pour l'outil (voir figure 7).

#### C. Performances

Nous avons utilisé le simulateur dans les conditions suivantes. Le matériau virtuel est une boîte rectangulaire de  $6 \times 6 \times 3$  cm et la lame virtuelle mesure 2 cm. La résolution de l'espace voxel généré pour le modèle de contact est de 1 mm. Le MCA fournit une interpolation de ces données, de sorte que cette résolution spatiale n'est qu'un indicateur grossier de la résolution du modèle haptique : le polytope fourni par le MCA est continu et la résolution spatiale haptique n'est alors limitée que par l'interface haptique (environ 0,03 mm).

Expérimentalement, notre implémentation du MCA peut calculer 15 millions de voxel par secondes sur un processeur à 3 GHz (sans affichage). Si l'AABB est de  $6 \times 6 \times 3$  cm (ce qui est bien plus gros que la taille de l'outil) et que la résolution voxel est de 2 mm, et si le calcul est effectué 1000 fois par seconde, nous devons calculer :

$$\frac{60 \times 60 \times 30}{2 \times 2 \times 2} \times 1\,000 = 13,5 \times 10^6 \text{ voxel/s}$$

Puisqu'on modifie la topologie, il faut transférer les données du modèle de déformation au modèle haptique. Si la lame mesure  $50 \times 20 \times 20$  mm et se déplace à 0,1m/s, le volume maximum modifié est (à 10 Hz, 10 mm par pas d'échantillonnage) :

$$50 \times 20 \times (20 + 10) = 30\,000 \text{ mm}^3$$

Avec une résolution voxel de 2 mm, on représente ce volume (par échantillon) par

$$\frac{30\,000}{2 \times 2 \times 2} = 3\,750 \text{ bits}$$

Ceci équivaut à  $3\,700 \times 10\text{ Hz} = 37\text{ kbps}$  (kilo-bit par seconde) ce qui est bien inférieur à 1 Gbps sur un réseau local à haut débit.

## V. CONCLUSION ET PERSPECTIVE

Cette méthode de conception d'un simulateur fournit des résultats satisfaisants dans l'ensemble; les deux modèles fonctionnent ensemble malgré la différence de fréquence d'échantillonnage. Les performances du MCA en terme de charge de calcul et de taux de transfert réseau sont compatibles avec la conception d'une simulation haptique complète. Toutefois, les modèles utilisés sont loin d'être réalistes.

Ainsi, les développements à moyen terme concernent les modèles en eux-mêmes, par exemple pour la simulation de découpe de tissus mous. Pour le modèle haptique, il reste encore à implémenter le modèle présenté dans [12] qui n'est implémenté ici que partiellement : seuls les efforts visqueux sont calculés alors que le modèle comporte aussi des efforts de contact visco-élastiques.

De plus, le modèle de déformation et de découpe demeure un réel problème. Indépendamment l'un de l'autre, ces deux aspects de la simulation ont trouvé de nombreuses réponses comme [14] pour la déformation et [15], [16], [17] pour le problème de découpe de maillage.

Toutefois la méthode que nous avons présentée ici nous permet d'oublier le problème de compatibilité entre les modèles de déformation et de contact, dans une certaine mesure, à la fois du point de vue échantillonnage que représentation.

Enfin, l'infographie nous offre des outils efficaces pour la génération de l'espace voxel en tant que représentation de la topologie locale, particulièrement ceux basés sur les distances signées.

## RÉFÉRENCES

- [1] V. Jayachandran, T. Rahman, M. Salganicoff, E. Heredia, and W. Harwin. Calibration of closed loop controllers for setting impedances in force-reflecting systems. In *Dynamic Systems and Control Division, Internat. Mech. Eng. Congress and Exposition*, Nov. 1996.
- [2] Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. Real-time elastic deformations of soft tissues for surgery simulation. Rapport de recherche 3511, INRIA, 1998.
- [3] Gilles Debunne. *Animation multirésolution d'objets déformables en temps-réel. Application à la simulation chirurgicale*. PhD thesis, Institut National Polytechnique de Grenoble, Décembre 2000.
- [4] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan H. Barr. Dynamic real-time deformations using space and time adaptive sampling. In *Computer Graphics Proceedings, Annual Conference Series*. ACM Press / ACM SIGGRAPH, Aug 2001. Proceedings of SIGGRAPH'01.
- [5] Y. H. Lu, W. T. Wang, R. H. Liang, and M. Ouhyoung. Virtual sculptor : A feature preserving haptic modeling system. In ACM Press, editor, *Proceedings of ACM International Workshop on Immersive Telepresence*, Juan Les Pins, France, 2002.
- [6] D. Bielser and M. Gross. Interactive simulation of surgical cuts. In IEEE Computer Society Press, editor, *Proceedings of Pacific Graphics*, pages 116–125, 2000.
- [7] R. Adams and B. Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3) :465–474, 1999.
- [8] S. Gibson and B. Mirtich. A survey of deformable modeling in computer graphics. Technical Report TR-97-19, Mitsubishi Electric Research Lab., Cambridge, MA, Nov. 1997.
- [9] Sébastien Druon, André Crosnier, and Loïc Brigandat. Efficient cellular automata for 2d/3d free-form modeling. In *WSCG*, 2003.

- [10] Sébastien Druon. *Modélisation et recalage d'objets et d'environnements en déformation viscoplastique. Aide au geste pour les métiers matière*. PhD thesis, Université de Montpellier 2, 2004.
- [11] William E. Lorensen and Harvey E. Cline. Marching cubes : A high resolution 3D surface construction algorithm. In *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, New York, NY, USA, 1987. ACM Press.
- [12] Didier Henry, Sébastien Druon, Philippe Fraisse, and André Crosnier. Simplified blade-tissue interaction model for haptic feedback. In *Methods and Models in Automation and Robotics*, pages 807 – 812, Aug. 2005.
- [13] Tomas Möller and Ben Trumbore. Fast, minimum storage ray-triangle intersection. *Journal of graphics tools*, 2(1) :21–28, 1997.
- [14] Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. A hybrid elastic model allowing real-time cutting, deformation and force-feedback training and simulation. *Visual Computer*, 16(8) :437–452, 2000.
- [15] Clément Forest, Hervé Delingette, and Nicholas Ayache. Cutting simulation of manifold volumetric meshes. In *Modeling and Simulation for Computer-aided Medicine and Surgery (MS4CMS'02)*, 2002.
- [16] Han-Wen Nienhuys and A. Frank van der Stappen. Combining finite element deformation with cutting for surgery simulations. In A. de Sousa and J.C. Torres, editors, *EuroGraphics Short Presentations*, pages 43–52, 2000.
- [17] Han-Wen Nienhuys and A. Frank van der Stappen. A delaunay approach to interactive cutting in triangulated surfaces. Technical Report UU-CS-2002-044, Institute of information and computing sciences, Utrecht University, 2002.

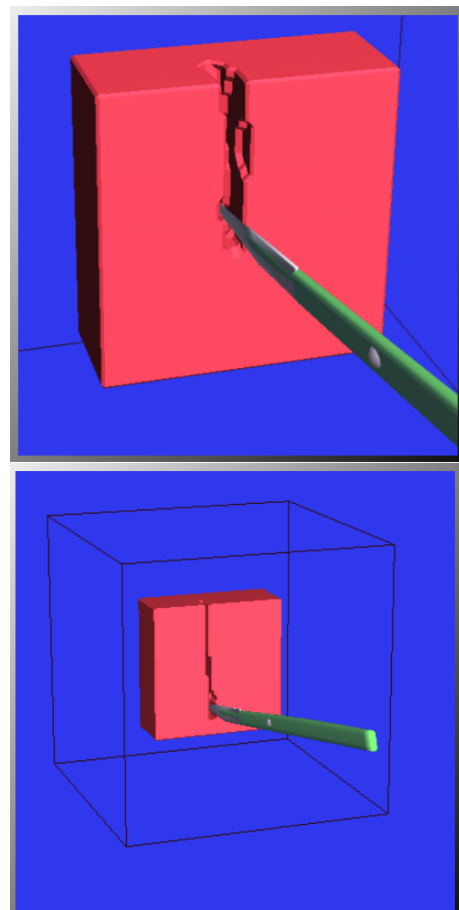


Fig. 7  
RENDU VISUEL DU MATÉRIAU VIRTUEL ET DE L'OUTIL VIRTUEL  
(SCALPEL)